Learning Motion Patterns Using Hierarchical Bayesian Models

by

Xiaogang Wang

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2009

Author .....
Department of Electrical Engineering and Computer Science
May 21, 2009

Certified by .....
W. Eric L. Grimson
Bernard Gordon Professor of Medical Engineering
Thesis Supervisor

Accepted by .....
Terry Orlando
Chairman, Department Committee on Graduate Students

Learning Motion Patterns
Using Hierarchical Bayesian Models

by

Xiaogang Wang

Submitted to the Department of Electrical Engineering and Computer Science
on May 21, 2009, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science

Abstract

In far-field visual surveillance, one of the key tasks is to monitor activities in the scene. Through learning motion patterns of objects, computers can help people understand typical activities, detect abnormal activities, and learn the models of semantically meaningful scene structures, such as paths commonly taken by objects. In medical imaging, some issues similar to learning motion patterns arise. Diffusion Tensor Magnetic Resonance Imaging (DT-MRI) is one of the first methods to visualize and quantify the organization of white matter in the brain in vivo. Using methods of tractography segmentation, one can connect local diffusion measurements to create global fiber trajectories, which can then be clustered into anatomically meaningful bundles. This is similar to clustering trajectories of objects in visual surveillance. In this thesis, we develop several unsupervised frameworks to learn motion patterns from complicated and large scale data sets using hierarchical Bayesian models. We explore their applications to activity analysis in far-field visual surveillance and tractography segmentation in medical imaging.

Many existing activity analysis approaches in visual surveillance are *ad hoc*, relying on predefined rules or simple probabilistic models, which prohibits them from modeling complicated activities. Our hierarchical Bayesian models can structure dependency among a large number of variables to model complicated activities. Various constraints and knowledge can be nicely added into a Bayesian framework as priors. When the number of clusters is not well defined in advance, our nonparametric Bayesian models can learn it driven by data with Dirichlet Processes priors. In this work, several hierarchical Bayesian models are proposed considering different types of scenes and different settings of cameras. If the scenes are crowded, it is difficult to track objects because of frequent occlusions and difficult to separate different types of co-occurring activities. We jointly model simple activities and complicated global behaviors at different hierarchical levels directly from moving pixels without tracking objects. If the scene is sparse and there is only a single camera view, we first track objects and then cluster trajectories into different activity categories. In the meanwhile, we learn the models of paths commonly taken by objects. Under the Bayesian

framework, using the models of activities learned from historical data as priors, the models of activities can be dynamically updated over time. When multiple camera views are used to monitor a large area, by adding a smoothness constraint as a prior, our hierarchical Bayesian model clusters trajectories in multiple camera views without tracking objects across camera views. The topology of multiple camera views is assumed to be unknown and arbitrary. In tractography segmentation, our approach can cluster much larger scale data sets than existing approaches and automatically learn the number of bundles from data. We demonstrate the effectiveness of our approaches on multiple visual surveillance and medical imaging data sets.

Thesis Supervisor: W. Eric L. Grimson
Title: Bernard Gordon Professor of Medical Engineering

Acknowledgments

First of all, I want to thank my parents, especially my mom, for their sacrificial love in the past thirty two years. Whenever I have difficult time, I always can find the strongest support from them.

I want to thank my wife, Man Tang, for her consistent love, support, encouragement and understanding. She sacrificed so much for me and our son. My achievement is impossible without her support. My tough PhD study became enjoyable because of her.

Thanks to ISM brothers and sisters, Pastor Joseph and Lydia Smn. Whenever I have a happy or difficult time, they are always with me. I love this family. I want to express my special thanks to Ce Liu. He cared about my life and gave me a lot of help in many aspects.

Thanks to my advisor Eric Grimson. He gave me a lot of help and guidance on my study, research and career. He read this thesis carefully for multiple times and gave me a lot of feedback in time. Thank my thesis committee members Leslie Kaelbling and Bill Freeman for their helpful suggesions and comments on my thesis work.

I had a great time collaborating with Carl-Fredrik Westin, Kinh Tieu, Xiaoxu Ma, Biswajit Bose, Gerald Dalley and Keng Teck Ma on many projects during my PhD study. I learned a lot from them. Thank Gerald Dalley, Joshua Migdal and Chris Stauffer for their great trackers. Thank Gerald Dalley and Joshua Migdal for their video IO library which makes life much easier. I want to acknowledge valuable discussions with Lauren O'Donnell regarding techniques for clustering fiber tracks. Especially thank Gerald for giving me a lot of technical support in many aspects and allowing me to frequently bug him. Thank Ce Liu, Gerald Galley, Biswajit Bose and Dahua Lin for helping me to prepare my thesis presentation.

Thanks to all the APP members, Chris Stauffer, Tomas Izo, Joshua Migdal, Xiaoxu Ma, Chaowei Niu, Biswajit Bose, Gerald Dalley, Jenny Yuan, Dahua Lin and Deepti Bhatnagar. I miss the time spent with them.

Contents

1 Introduction **17**

1.1 Visual Surveillance 17

1.1.1 Activity Analysis 18

1.1.2 Learning Scene Structures 19

1.2 Tractography Segmentation 20

1.3 Motion Patterns 21

1.4 Contributions and Summary of Approaches 23

1.4.1 Crowded and Complicated Scenes 25

1.4.2 Sparse Scenes with a Single Camera View 29

1.4.3 Sparse Scenes with Multiple Camera Views 33

1.4.4 Tractography Segmentation 36

1.4.5 Summary 37

1.5 Thesis Road Map 38

2 Literature Review **41**

2.1 Activity Analysis Without Tracking Objects 41

2.2 Trajectory Analysis and Scene Modeling with a Single Camera View . 42

2.3 Activity Analysis in Multiple Camera Views 45

2.4 Probabilistic Approaches in Visual Surveillance 46

2.5 Tractography Segmentation 47

2.6 Hierarchical Bayesian Models in Computer Vision Applications 48

2.7 Nonparametric Bayesian Approaches 49

3 Activity Analysis in Crowded and Complicated Scenes 51

3.1 Low-Level Motion Features 52

3.2 Hierarchical Bayesian Models 54

3.2.1 LDA 54

3.2.2 LDA Mixture Model 56

3.2.3 Dirichlet Process 59

3.2.4 HDP 60

3.2.5 HDP Mixture Model 61

3.2.6 Dual-HDP 63

3.2.7 Discussion on the co-clustering framework 65

3.2.8 Example of synthetic data 66

3.3 Visual Surveillance Applications and Experimental Results 67

3.3.1 Discover Atomic Activities 68

3.3.2 Discover Global Behaviors 72

3.3.3 Video Segmentation 74

3.3.4 Activity Detection 75

3.3.5 Abnormality Detection 75

3.3.6 Query Interactions 78

3.3.7 Comparison with Other Methods 80

3.3.8 Discussion 80

4 Trajectory Analysis in A Single Camera View 83

4.1 Modeling Trajectories Using Dual-HDP 83

4.2 Dynamic Dual-HDP 85

4.3 Experimental Results 91

4.3.1 Trajectory Analysis without Dynamic Modeling 91

4.3.2 Trajectory Analysis with Dynamic Modeling 100

4.4 Summary 108

5 Correspondence-Free Activity Analysis in Multiple Camera Views 111

5.1 Feature Space 111

5.2 Trajectory Network . . . 112
5.3 Probabilistic Model . . . 113
5.3.1 Learning and Inference . . . 117
5.3.2 Labeling Trajectories into Activities . . . 118
5.3.3 Detection of Abnormal Trajectories . . . 118
5.3.4 Complexity . . . 119
5.4 Experimental Results . . . 119
5.4.1 Learning Activity Models and Clustering Trajectories . . . 120
5.4.2 Correspondence . . . 132
5.4.3 Abnormality Detection . . . 133
5.4.4 Computational Cost . . . 134
5.4.5 Simulated Data . . . 136
5.5 Discussion . . . 143

6 Tractography Segmentation **145**
6.1 Experimental Results . . . 146

7 Limitations and Future Work **151**
7.1 Low-level features . . . 151
7.2 Design of "Documents" for Activity Analysis in Crowded Scenes . . . 152
7.3 Temporal Logic . . . 152
7.4 Jointly Model Activities and Appearance in Multiple Camera Views . 152
7.5 Guide Tractography Segmentation . . . 153
7.6 Inference . . . 153

8 Conclusion **155**

A Gibbs Sampling for Dual-HDP **159**

List of Figures

1-1 Example of a traffic scene 18

1-2 Tractography in DT-MRI 20

1-3 Tractography segmentation 21

1-4 Compare anatomical structures across subjects 21

1-5 Representations of motion patterns 23

1-6 Examples of crowded and complicated scenes 26

1-7 System diagram for activity analysis in crowded and complicated scenes 27

1-8 Learning motion patterns from temporal co-occurrence of feature values 29

1-9 Examples of paths and trajectories in a parking lot scene 31

1-10 Learning motion patterns from identity co-occurrence 32

1-11 Simulated examples of activities observed in multiple camera views . 34

1-12 Examples of multi-camera settings 35

1-13 An example of multiscale clustering 37

3-1 System diagram for activity analysis in crowded and complicated scenes 53

3-2 Graphical models of LDA and LDA mixture 55

3-3 Graphical models of HDP and HDP mixture 60

3-4 Graphical model of Dual-HDP 64

3-5 Experimental comparison of HDP and Dual-HDP on a toy example . 66

3-6 Distributions of atomic activities learned by Dual-HDP in a traffic scene 69

3-7 Histogram of moving pixels assigned to atomic activities learned by Dual-HDP in a traffic scene 70

3-8 Distributions of atomic activities learned by the LDA mixture model in a traffic scene 70
3-9 Some atomic activities learned by Dual-HDP merger into one atomic activity learned by the LDA mixture model 71
3-10 Distributions of global behaviors over atomic activities learned by Dual-HDP in a traffic scene 73
3-11 Results of video segmentation in a traffic scene 74
3-12 Activity detection in a traffic scene 76
3-13 Vehicle and pedestrian detection based on motions 76
3-14 Results of abnormality detection in a traffic scene 78
3-15 Query result of interaction jay-walking 79

4-1 An example to explain the modeling of semantic regions and activities 84
4-2 Graphical model of dynamic Dual-HDP 86
4-3 Semantic regions at a maritime port learnt from the radar tracks . . . 92
4-4 Clusters of radar tracks from a maritime port 94
4-5 Top 20 abnormal radar tracks from a maritime port 95
4-6 Trajectories collected from a parking lot scene within one week 96
4-7 Some semantic regions learnt from a parking lot scene 97
4-8 Some clusters of trajectories from a parking lot scene 98
4-9 Top 100 abnormal trajectories in the parking lot scene 99
4-10 Simulate trajectories of different activities 100
4-11 Activity classification accuracies of Dual-HDP and two distance-based methods 101
4-12 Dynamic change of semantic region 1 over time learnt from the radar tracks 102
4-13 Dynamic change of semantic region 2 over time learnt from the radar tracks 103
4-14 Dynamic change of semantic region 3 over time learnt from the radar tracks 103

4-15 Dynamic change of semantic region 4 over time learnt from the radar tracks . . . 103
4-16 Dynamic change of semantic region 5 over time learnt from the radar tracks . . . 104
4-17 Abnormal radar tracks detected at different time slices . . . 104
4-18 Dynamic change of a semantic region 1 from parking lot scene over time 106
4-19 Dynamic change of a semantic region 2 from parking lot scene over time 107
4-20 Dynamic change of a semantic region 2 from parking lot scene over time 107
4-21 Abnormal trajectories in the parking lot scene detected at different time slices. . . . 109

5-1 An example of building a network connecting trajectories in multiple camera views . . . 113
5-2 An example to describe the high level picture of our model for trajectory analysis in multiple camera view . . . 114
5-3 Examples of multi-camera settings . . . 120
5-4 Distributions of activity models $(1-4)$ and clusters of trajectories in a parking lot scene . . . 122
5-5 Distributions of activity models $(5-8)$ and clusters of trajectories in a parking lot scene . . . 123
5-6 Distributions of activity models $(9-12)$ and clusters of trajectories in a parking lot scene . . . 124
5-7 Distributions of activity models $(13-14)$ and clusters of trajectories in a parking lot scene . . . 125
5-8 Distributions of activity models $(1-4)$ and clusters of trajectories in a street scene . . . 126
5-9 Distributions of activity models $(5-8)$ and clusters of trajectories in a street scene. . . . 127
5-10 Distributions of activity models $(9-12)$ and clusters of trajectories in a street scene. . . . 128

5-11 Distributions of activity models $(13-16)$ and clusters of trajectories in a street scene. 129
5-12 Activity models learnt in an unsupervised way help to solve the correspondence problem 132
5-13 Some trajectories with low likelihoods from a parking lot scene 134
5-14 Some trajectories with low likelihoods from a street scene 135
5-15 Simulated camera views 137
5-16 Manually drawn paths and simulated trajectories 137
5-17 The accuracies of classifying trajectories into different activities when λ takes different values and T is fixed as 0 138
5-18 Distributions of activity models in a single global views learnt from the simulated data 140
5-19 Distribution of activity models in four camera views learnt from the simulated data. The meaning of colors is the same as Figure 5-4. . . . 141
5-20 Statistics of pairs of trajectories connected on the network 142
5-21 The accuracies of classifying trajectories into different activities when the temporal threshold T change from 0 to 300 seconds 142
5-22 Accuracies of correspondence on the simulated data 143

6-1 Compare the results of two clustering approaches with the ground truth on a data set with $3,152$ fibers 146
6-2 Compare results of our approach and the approach proposed in [107] 147
6-3 Cluster fibers across multiple subjects 149

List of Tables

5.1 Negative log likelihood under our approach and two alternative trajectory networks 131
5.2 Negative log likelihood with models trained on a variable number of cameras 131

Chapter 1

Introduction

1.1 Visual Surveillance

Visual surveillance has been one of the most active research topics in computer vision in recent years. The goal of visual surveillance is to detect, track and recognize objects of interest, understand and describe the behaviors of objects, and efficiently extract useful information for users from a huge amount of video data collected by monitoring cameras. Visual surveillance has a wide variety of applications in both public and private environments, such as homeland security [42, 26, 13, 159, 27, 119], crime prevention [54, 8, 31, 28, 164, 158], traffic control [123, 146, 83, 72, 80], accident prediction and detection [73, 58, 5, 6, 121], and monitoring patients, elderly and children at home [100, 89]. Comprehensive surveys can be found in [56, 102]. These applications require monitoring indoor and outdoor scenes of airports, train stations, highways, parking lots, stores, shopping malls and offices. There is a growing interest in visual surveillance due to the growing availability of cheap sensors and processors, and also a growing need from the public for safety and security.

Currently there may be tens of thousands of cameras in a city collecting a huge amount of data on a daily basis. Researchers are urged to develop intelligent systems to efficiently extract information from a huge amount of data. Because of the large number of cameras, it is also essential for visual surveillance systems to self-adapt to a variety of scenes with minimum human intervention. The focus of visual surveil-

Figure 1-1: Example of a traffic scene. There are many single-agent activities (e.g. cars turn left), multi-agent interactions (e.g. a vehicle comes to stop waiting for pedestrians to cross the street), and global behaviors (e.g. different traffic modes) happening in this scene.

lance research is moving from processing only one video stream with a single camera view from sparse and simple scenes to monitoring crowded and busy scenes with a distributed network of cameras. Intelligence, distribution, adaptability and tolerance are the most important characteristics of modern visual surveillance systems [115].

1.1.1 Activity Analysis

Activity analysis has long been one of the foci of research in visual surveillance. Over the past decade, significant work has been reported on this topic. A literature review can be found in Chapter 2. As an example, Figure 1-1 shows a traffic scene with many different types of activities happening. People expect visual surveillance systems to

- discover typical types of single-agent activities (e.g. a car turns left), multi-agent interactions (e.g. a vehicle stops waiting for pedestrians to cross the street) and global behaviors (e.g. different traffic modes) in these scenes, and provide a summary of them;
- detect/classify activities, interactions and global behaviors;
- detect abnormalities (e.g. pedestrians cross the street outside the crosswalk);
- describe scene evolution in natural language;

- obtain various statistics of activities (e.g. how the frequencies of different types activities vary over time); and

- support query of activities and interactions in a flexible way.

Ideally, a system would learn models of activities to answer such questions in an unsupervised way with as little human labeling effort as possible. we will do activity analysis under different scenarios. It depends on the number of camera views and how crowded the scene is. More concretely, these scenarios include crowded scenes with a single camera views, sparse scenes with a single camera view and sparse scenes with multiple camera views.

1.1.2 Learning Scene Structures

Activities are closely related to the structures of the scenes, such as paths, or entry and exit points, since they regularize the motion of objects. On the one hand, these structures can be identified from the moving patterns of objects related to particular activities [68, 38, 96, 97, 71, 153, 57, 70]. On the other hand, the knowledge of scene structures helps to classify moving patterns into activities, since it provides prior information on activities happening in a scene. In this thesis work the two related problems of activity analysis and scene modeling will be jointly solved. The knowledge of the learnt scene structures is very useful in many surveillance tasks. It supports both high-level activity interpretation and low-level object tracking and classification. It can support activity descriptions with spatial context [119], such as "a car moving off the road" and "a person waiting at a bus stop". It also can improve low-level tracking and classification [11, 49, 75, 55]. For example, if an object disappears, but not at an exit point, then this event is likely to be a tracking failure instead of a true exit. Through learning the lane marker positions on a highway, cast shadows that fall over a lane line can be removed [55]. In classification, people can leverage the fact that vehicles are much more likely than pedestrians to move on the road.

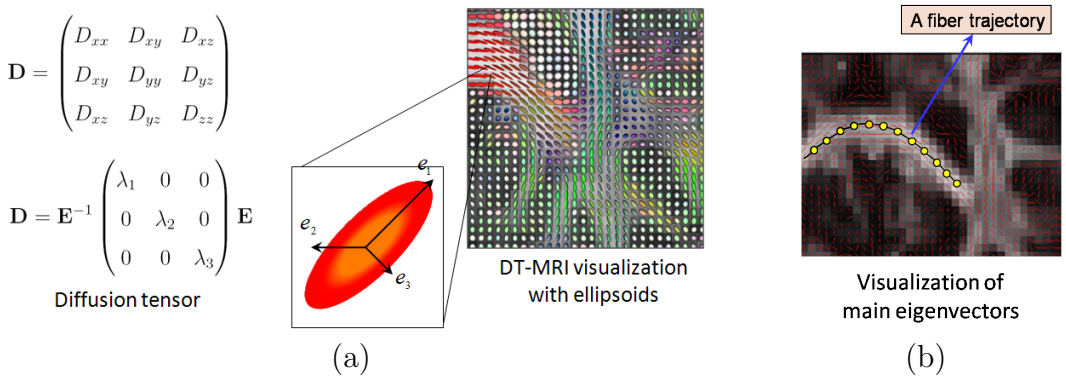
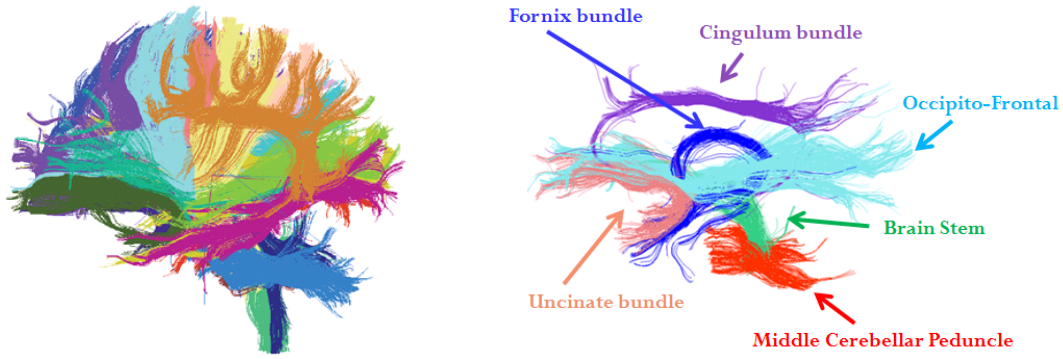


Figure 1-2: Tractography in DT-MRI. (a) In DT-MRI, a local diffusion tensor D, which is a 3×3 symmetric matrix, is used to characterize the motion of water in all directions. The eigenvectors (e_1, e_2 and e_3) of a tensor are computed, and the tensor is visualized with an ellipsoid. (b) Tractography is a technique to extract a fiber trajectory by connecting local diffusion measurements.

1.2 Tractography Segmentation

Some techniques developed in visual surveillance can also be applied to medical imaging where similar issues of of tracking, clustering, abnormality detection and similarity retrieval arise. Diffusion Tensor Magnetic Resonance Imaging (DT-MRI) is an MRI modality that has gained tremendous popularity over the past five years and is one of the first methods that made it possible to visualize and quantify the organization of white matter in the human brain in vivo. As shown in Figure 1-2, DT-MRI measures local diffusivity of a water molecule within the tissue. It provides information about the orientation of white matter fiber tracts. Extracting connectivity information from DT-MRI, termed "tractography", is an especially active area of research, as it promises to model the pathways of white matter tracts in the brain, by connecting local diffusion measurements into global fiber trajectories. As shown in Figure 1-3, in tractography segmentation, fiber trajectories are clustered into bundles which help to identify anatomical structures in the brain. Experts compare the anatomical structures of different subjects to tell whether they are normal. Some examples are shown in Figure 1-4.



(a) (b)

Figure 1-3: Tractography segmentation is a technique to cluster fiber trajectroeis generated from DT-MRI into bundles which correspond to anatomical structures. (a) Full brain tractography segmentation result. Colors represent different bundles. (b) Anatomical labels of some bundles.

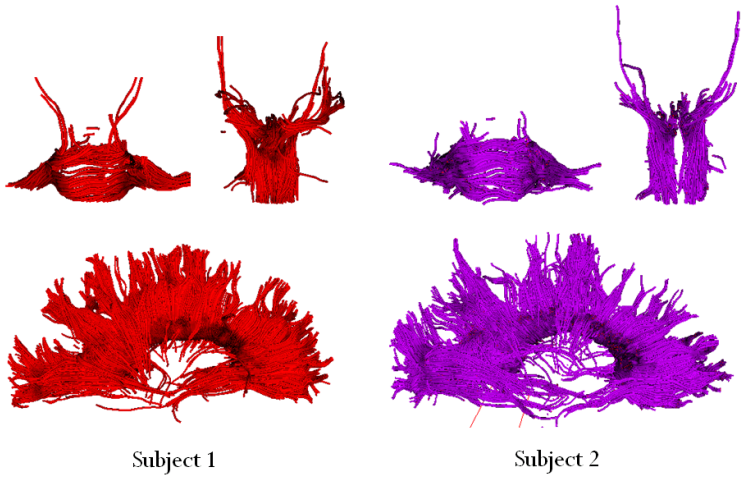


Figure 1-4: Bundles of two different subjects. By comparing anatomical structures across subjects, experts can tell whether they are normal.

1.3 Motion Patterns

In this work a motion pattern means the pathway of an object moving from one location to another. Locations and moving directions are the most important features to describe motion patterns. This thesis focuses on activity analysis in far-field visual surveillance. In far-field settings, objects are small in size and the captured videos are of low resolution and poor quality. It is difficult to compute more complicated features, such as poses, gestures, and appearance of objects. The activities of objects are mainly distinguished by their moving patterns. In comparison, in near-field

surveillance, other features may play a more important role in explaining activities.

A motion pattern has several different representations. As shown in Figure 1-5, it can be represented by a trajectory when an object is tracked in a single camera view. It can be represented by several trajectories when an object is tracked in multiple camera views. Without tracking objects, it can be represented by a set of moving pixels. Through clustering motion patterns, the models of activities and semantically meaningful scene structures can be learnt.

If the scene is sparse and there is only a single camera view, visual surveillance systems typically first detect and track objects, and treat the activity of an object as sequential movements along its trajectory. Through tracking, an activity executed by a single object can be separated from other co-occurring activities, and features related to the activity can be integrated as a track. Trajectories are clustered or classified into different activity categories. The paths commonly taken by objects are learned from the clusters of trajectories.

The view of a single camera is finite and limited by the structures of scenes. In order to monitor activities in a wide area, video streams from multiple cameras have to be used. A typical way of doing multi-camera surveillance is to first infer the topology of camera views, then track objects across camera views, and finally cluster trajectories observed in different camera views. However both inferring the topology of camera views and tracking objects across camera views are notoriously difficult especially when the number of cameras is large and the topology of the camera views is arbitrary.

Many scenes, such as airports, train stations, street intersections and shopping malls, are very crowded. It is difficult to track objects in these scenes because of frequent occlusions. Alternatively, moving pixels instead of trajectories can be used as features to model activities. Existing approaches typically treat a video clip as an integral entity and compute a motion feature vector from the moving pixels detected in the whole video clip. The whole video clip is labeled as one of the activity categories, and flagged as normal or abnormal. However, when there are many different types of activities happening in a busy scene simultaneously, it is difficult to separate a

(a) (b) (c)

Figure 1-5: Motion patterns have different representations. A motion pattern can be represented by (a) a trajectory when an object is tracked in a single camera view, (b) several trajectories when an object is tracked in multiple camera views independently or (c) a set of moving pixels without tracking objects.

particular activity from other co-occurring activities without relying on detection and tracking. Activity analysis in crowded and complicated scenes is still a very challenging problem not well solved yet.

Extracting fiber trajectories by tractography from DT-MRI has some similarity with tracking objects in visual surveillance. Tractography segmentation aims to cluster fiber trajectories into anatomically meaningful bundles. This is like clustering trajectories of objects. Thus similar trajectory analysis approaches can be applied in both fields [153, 107]. Through clustering trajectories, anatomical structures can be identified in the brain from DT-MRI and semantically meaningful scene structures can be found in far-field visual surveillance. In this work, we study the problem of learning motion patterns with applications to both visual surveillance and tractography segmentation.

1.4 Contributions and Summary of Approaches

This thesis work focuses on learning motion patterns with its applications to activity analysis in far-field visual surveillance and tractography segmentation in medical imaging. In visual surveillance, we will do activity analysis under different scenarios. It depends on the number of camera views and how crowded the scene is. The contributions of this work are summarized as four-fold. First, we apply nonparametric hierarchical Bayesian models to learn motion patterns in visual surveillance and tractography segmentation. Second, we proposed an unsupervised framework to

jointly model simple activities, complicated interactions and global behaviors and to separate co-occurring activities in crowded scenes without tracking objects. Third, when existing approaches analyze activities in multiple camera views, they require tracking objects across camera views and knowing the topology of camera views. We propose a novel approach for activity analysis from multiple camera views without tracking objects across camera views and without knowing the topology of camera views. Fourth, we proposed a novel dynamic nonparametric method to update models of activities over time. Our approaches can also be used to solve clustering problems on other data sets.

There are many advantages of using nonparametric hierarchical Bayesian models for activity analysis and tractography segmentation.

(1) Many existing activity analysis approaches in visual surveillance are *ad hoc*, relying on predefined rules or simple probabilistic models. They have difficulty modeling complicated activities. Under a Bayesian framework, visual surveillance tasks are formulated in a principled way. In this framework, abnormality has a probabilistic explanation. Hierarchical Bayesian models can structure dependency among a large number of variables to model complicated activities. Various constraints and knowledge can be nicely added into a Bayesian model as priors. For example, using the models of activities learned from historical data as priors, the models of activities can be dynamically updated over time. By adding a smoothness constraint on the distributions of trajectories over activities as a prior, our hierarchical Bayesian model clusters trajectories in multiple camera views without tracking objects across camera views.

(2) In our dynamic hierarchical Bayesian approach, the models of activities are updated over time and thus can better explain activities at different times. For example, some activities which are abnormal at a particular time may become normal at a different time.

(3) When analyzing activities in multiple camera views, our approach assumes that the topology of camera views is arbitrary. The camera views can have large overlap, small overlap, or even no overlap.

(4) Many existing approaches have difficulty deciding the numbers of clusters of moving pixels, video clips and trajectories. Nonparametric Bayesian models can learn the numbers of clusters driven by data. This provides a solution when the number of clusters is not well defined in advance.

(5) While many existing approaches need expensive human effort to adjust parameters or label data, our approaches are unsupervised with little human labeling effort.

(6) Our approaches have lower space complexity and can cluster larger scale data sets than many existing approaches.

Our approaches developed in four applications are summarized in the following subsections.

1.4.1 Crowded and Complicated Scenes

Many scenes, such as street intersections, train stations, airports, and shopping malls (see Figure 1-6), of great interest for security purpose are very crowded. It is difficult to detect and track objects because of frequent occlusions. Most of the tracking based activity analysis approaches are expected to fail in these scenes. Although many approaches were proposed for activity analysis directly using motion feature vectors without tracking objects, they assumed that there was only one type of activity happening in each short video clip. The whole video clip was labeled as one of the activity categories, and flagged as normal or abnormal. They held this assumption at least at the training stage. In these crowded and complicated scenes, many different types of activities happen simultaneously. It is difficult for existing approaches to separate co-occurring activities without supervision. A detailed literature review can be found in Section 2.1.

We proposed an unsupervised framework using a nonparametric hierarchical Bayesian model, Dual Hierarchical Dirichlet Processes (Dual-HDP), to jointly model simple activities, such as cars turning right and pedestrians crossing the street, which are called atomic activities, and global behaviors, such as different traffic modes, in the scene directly from moving pixels without tracking objects. Global behavior is defined as

Figure 1-6: Examples of crowded and complicated scenes, such as traffic scenes, train stations and shopping malls. In these scenes, it is difficult to track objects because of frequent occlusions among objects and it is difficult to separate different types of activities which happen at the same time.

a combination of different types of co-occurring atomic activities in the scene. After the models of atomic activities are learned, they can be used as units to detect interactions between objects such as jay-walking (when a pedestrian is crossing the street, a vehicle is simultaneously approaching).

Our system diagram is shown in Figure 1-7. We compute local motions (locations and moving directions of moving pixels) as our low-level visual features. This avoids the difficult tracking problem in crowded scenes. We do not use global motion features ([164, 161]) because, in these complicated scenes, multiple different types of activities often occur simultaneously and we want to separate them. Each moving pixel has a feature value which includes location and direction of motion. An observed long video sequence is uniformly divided into many short video clips. Global behavior is a combination of atomic activities occurring in the same video clip. Thus, there exist two hierarchical structures in both the data set (long video sequence $\rightarrow$ short video clips $\rightarrow$ moving pixels) and visual surveillance tasks (global behaviors $\rightarrow$ atomic activities). So, it is natural to employ a hierarchical Bayesian approach to

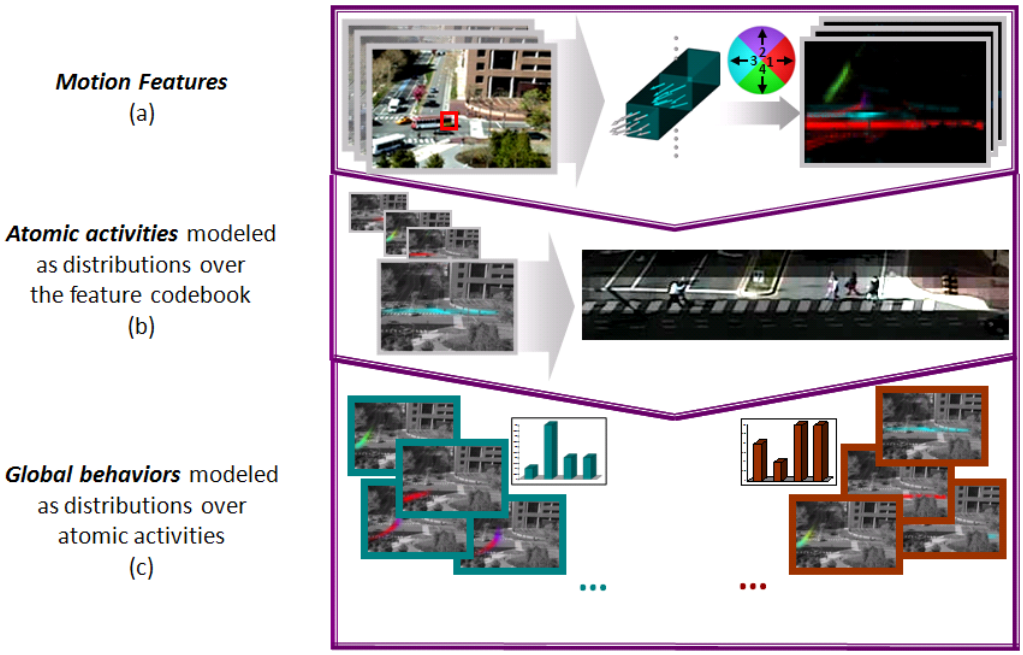


Figure 1-7: Diagram of the system for activity analysis in crowded and complicated scenes. Our framework connects low-level motion features, middle-level atomic activities and high-level global behaviors. (a) The observed long video sequence is uniformly divided into short clips as documents. In each video clip, moving pixels are detected and quantized into visual words based on their locations and motion directions. The four quantized directions are represented by four colors. Each video clip has a distribution over visual words. (b) Atomic activities (e.g. pedestrians cross the road) are discovered and modeled as distributions over visual words. Moving pixels are clustered into atomic activities. (c) Video clips are clustered into global behaviors, which are modeled as distributions over atomic activities.

connect three elements in visual surveillance: low-level motion features, middle-level atomic activities, and high-level global behaviors. Atomic activities are modeled as distributions over low-level visual features and global behaviors are modeled as distributions over atomic activities. Moving pixels are clustered into atomic activities and video clips are clustered into global behaviors. Abnormal video clips and moving pixels are detected as those with low data likelihood under the learned hierarchical Bayesian model. As explained in [44], a hierarchical Bayesian model learned from a data set with hierarchical structure has the advantage of using enough parameters to fit the data well while avoiding overfitting problems since it is able to use a population distribution to structure some dependence into the parameters. In our case, the same types of atomic activities repeatedly occur in different video clips. By sharing a common set of atomic activity models across different video clips, the models of atomic activities can be well learned from enough data. On the other hand, atomic activities are used as components to further model more complicated global behaviors, which are clusters of video clips. This is a much more compact representation than directly clustering high dimensional motion feature vectors computed from video clips. Under hierarchical Bayesian models, surveillance tasks such as motion segmentation, video segmentation, activity detection, and abnormality detection are formulated in a transparent, clean, and probabilistic way compared with the *ad hoc* nature of many existing approaches.

Dual-HDP advances the existing language processing model, Hierarchical Dirichlet Processes (HDP) [140]. HDP is a nonparametric Bayesian model. It clusters words often co-occurring in the same documents into one topic and learns the number of topics from data. Dual-HDP co-clusters both words and documents. Documents containing similar sets of topics of words are clustered. It automatically decides the numbers of both word topics and document clusters. Directly using HDP to solve this problem, HDP can only cluster moving pixels into atomic activities. Since video clips have different combinations of atomic activities, Dual-HDP has an extra layer of hierarchical Dirichlet processes to model the clusters of video clips on the top of atomic activities. Dual-HDP is similar to the nonparametric model, called Nested

(a) (b)

Figure 1-8: Learning motion patterns from temporal co-occurrence of feature values. An atomic activity generates continuous motions in time. Two features have strong temporal correlation if they are related to the same atomic activity. The models of atomic activities can be learned from the temporal co-occurrence of feature values. (a) Moving pixels in a short video clip. They are colored into different atomic activity category. (b) The spatial distribution of an atomic activity. The two locations marked by yellow boxes are on the pathway of vehicles. When a car is passing by, we observed moving pixels at these two locations around the same time. The atomic activity model has large distribution on both locations with temporal correlation.

Dirichlet Process, proposed by Rodriguez et al. [117] independently applied to health care quality analysis. It is also closely related to the Transformed Dirichlet Process proposed by Sudderth et al. [137] applied to object recognition.

Under our framework, video clips are treated as documents and moving pixels are treated as words. An atomic activity generates temporally continuous motions. Thus local motions caused by the same atomic activity often co-occur in the same short video clips. This is called temporal co-occurrence. As an example shown in Figure 1-8, the models of atomic activities are learned from the temporal co-occurrence of feature values. Video clips containing similar sets of atomic activities are clustered into the same global behavior. Our model holds the "bag-of-words" assumption. It does not capture complicated temporal logic in activity, such as two people meet each other, walk together, and then separate. This part of thesis work was previously published in [151, 152].

1.4.2 Sparse Scenes with a Single Camera View

If there is only a single camera view and the scene is sparse, we first detect and track objects, and then cluster trajectories of objects into different activity categories based

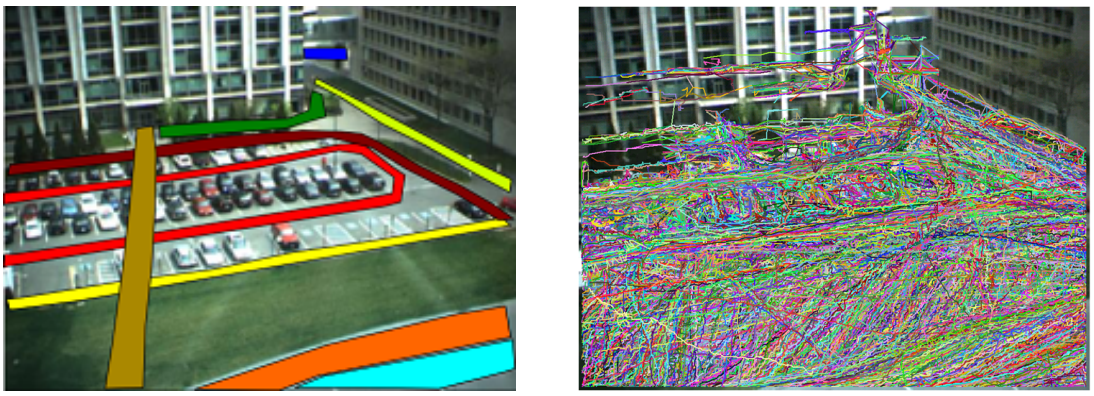
on their spatial distributions and moving directions. In the meanwhile the models of paths commonly taken by objects are learned. Abnormal trajectories are detected as those with low data likelihood under the learned hierarchical Bayesian models.

We now briefly explain several basic concepts and assumptions held in our proposed framework under this setting. There are paths in the physical world. Paths may have spatial overlap. The intersections of paths are called semantic regions. A path is composed of several semantic regions. Objects on the same paths have similar moving patterns, which is called an activity. Some examples of paths and trajectories can be found in Figure 1-9. Although some paths, such as roads of vehicles can be recognized by their physical features, some paths cannot be. For example, pedestrians may take a short cut on a grass field. A trajectory, which only records the positions of an object, is a history of the movement of an object in a camera view. The points on trajectories are called observations.

The scene of a camera view is quantized into small cells. When an object moves, it connects two cells far apart in a camera view by its trajectory. This is called identity co-occurrence as explained in Figure 1-10. Our probabilistic model is based on some simple, general assumptions on the spatial and temporal features related to activities. (1) Cells located in the same semantic region are likely to be connected by trajectories. (2) trajectories passing through the same path (a set of semantic regions) belong to the same activity.

We propose an unsupervised framework using Dual-HDP for trajectory analysis. Under our framework, trajectories are treated as documents and the observations (positions and moving directions of objects) on the trajectories are treated as words. Topics are semantic regions. Observations are clustered into semantic regions based on identity co-occurrence and trajectories passing through the same set of semantic regions are clustered into the same activity (path). This part of the thesis work was previously published in [150].

Dual-HDP is a static model. We further extend Dual-HDP to a dynamic Dual-HDP model which can online update the models of activities over time. A dynamically updated model can better explain activities at the current time. For example, some



(a) (b)

Figure 1-9: Examples of paths and trajectories in a parking lot scene. We can track objects in sparse scenes. (a) Examples of paths in the parking lot scene. (b) Trajectories collected from the parking lot scene within one week. Random colors are used to distinguish individual trajectories.

activities which are abnormal at one particular time may become normal at a different time. We can analyze how the models of activities and paths change over time. Under dynamic Dual-HDP, the data is divided into subsets in temporal order. Dynamic Dual-HDP has a much lower complexity than Dual-HDP, since it clusters subsets of trajectories incrementally and does not keep any historical data in memory. This property is very useful if people need to cluster a huge data set collected from several months or years. Dynamic Dual-HDP is related to the dynamic topic model proposed by Blei et al. [16], which was a parametric model assuming that the number of topics are fixed.

Most of the existing trajectory analysis approaches cluster trajectories and detect abnormal trajectories by defining the pairwise distances/similarities between trajectories. A detailed review can be found in Section 2.2. This framework has several drawbacks. First, there is no global probabilistic framework to model activities happening in the scene. They have an *ad hoc* nature especially on the definition of distance measures. Abnormal trajectories are usually detected as those with a larger distance to other trajectories. Their abnormality detection lacks a probabilistic explanation. Second, they usually do not provide a solution to the number of clusters and require that the cluster number is known in advance. Third, some approaches required that

(a) (b)

Figure 1-10: Learning motion patterns from identity co-occurrence. Identity co-occurrence means that two feature values are on the same trajectories and are related to the same object. (a) Two locations marked by yellow boxes are on the same trajectory. (b) Two locations marked by yellow boxes are in the same semantic region and thus they co-exist on many trajectories. Based on the identity co-occurrence information, a semantic region model with large distribution on both two locations can be learned.

two trajectories were temporally aligned in order to compute their distance. Thus they are sensitive to misdetection and tracking errors. Fourth, calculating the distances/similarities between all pairs of samples is computationally inefficient, with a complexity of $O(N^2)$ in both time and space, where N is the number of trajectories.

Our framework differs from previous trajectory analysis and scene modeling approaches in the following aspects.

- Different from existing distance-based clustering approaches, it clusters trajectories using a generative model. There is a natural probabilistic explanation for the detection of abnormal trajectories.

- Using Dirichlet Processes, the number of activity categories and semantic regions are automatically learnt from data instead of being manually specified.

- It does not require that trajectories are temporally aligned while many existing approaches do. So it is more robust to tracking errors.

- The space complexity of Dual-HDP is $O(N)$ instead of $O(N^2)$ in the number of trajectories. We use collapsed Gibbs sampling is to do inference. The time complexity of each collapsed Gibbs sampling iteration is $O(N)$. However, there is no theoretical justification on the convergence of the collapsed Gibbs sampling.

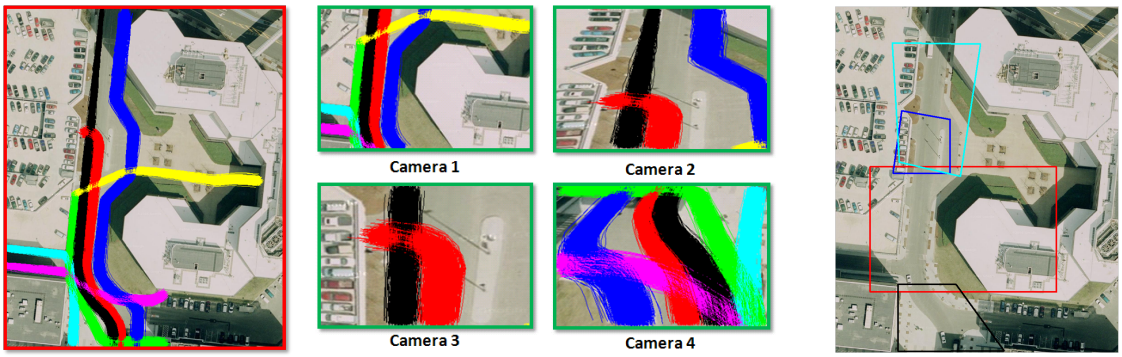
- Dynamic Dual-HDP online updates the models of activities, while existing approaches are static and run in a batch mode. Dynamic Dual-HDP clusters trajectories incrementally. It has much lower space and time complexities than Dual-HDP and existing approaches.

- Our approach clusters trajectories based on identity co-occurrence, different from distance based methods which cluster trajectories close in space. Considering the case when vehicles move on two side-by-side lanes in the same direction, distance-based methods will group trajectories of these vehicles into one cluster while our approach will learn the two lanes as different semantic regions and separate trajectories into two clusters since there are few vehicles crossing lanes to connect locations in different lanes.

1.4.3 Sparse Scenes with Multiple Camera Views

In order to monitor activities in a wide area, video streams from multiple cameras have to be used. We first track objects in each camera view independently, and then group trajectories, which belong to the same activity category but are observed in different camera views, into one cluster. The distributions of a path in multiple camera views are jointly modeled. This is more challenging than activity analysis in a single camera view since it is difficult to track objects across camera views. Examples of activities observed in multiple camera views can be found in Figure 1-11.

Many systems [61, 23, 78, 85, 25, 79, 74, 64, 136, 98, 113, 126, 65, 125, 143, 48, 145, 128, 39] using multiple camera views for visual surveillance were proposed in past years. They were based on various assumptions on the number of cameras, the topology and geometry of camera views, and the calibration of cameras. A detailed review of related work can be found in Section 2.3. Most of these approaches focused on tracking objects across camera views. In general, this is a very difficult problem. Because of the structures of the scenes, the distribution and configuration of cameras could be quite arbitrary and unknown. The camera views may have any combination of large, small, or even no overlap. The objects in camera views may move on one



(a) (b) (c)

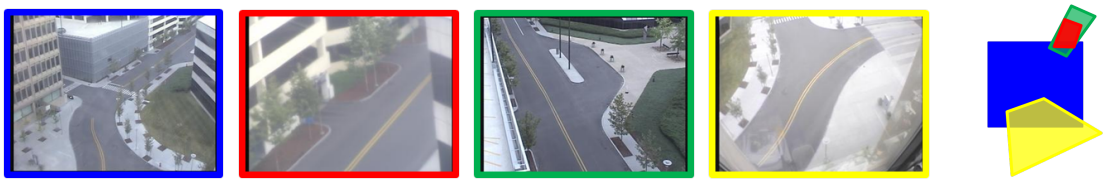
Figure 1-11: Simulated examples of activities observed in multiple camera views. Trajectories of different activity categories are marked by different colors. Trajectories are obtained from simulation. The purpose of our activity analysis in multiple camera view is to group trajectories which belong to the same activity category but are observed in different camera views, into one cluster. (a) Activities observed in a single giant camera view, which is usually unavailable in real life. (b) The same activities as in (a) are observed in four different camera views, which are similar to the camera views available in real life. (c) The fields covered by four camera views. They are marked by polygons in colors: red (camera 1), blue (camera 2), cyan (camera 3) and black (camera 4).

or multiple ground planes. Analyzing activities over such a multi-camera network is quite challenging. A natural way of doing multi-camera surveillance is to first infer the topology of camera views [98, 113], solve the correspondence problem [61, 85, 25, 136, 79, 64, 113, 125, 65, 126, 48, 128], stitching trajectories of the same object observed in different camera views into a complete trajectory, and then analyze the stitched trajectories using the same approaches developed for a single camera view. However both inferring the topology of camera views and solving the correspondence problem are notoriously difficult especially when the number of cameras is large and the topology of camera views is arbitrary.

The ultimate goal of some surveillance systems is activity analysis instead of solving correspondence. In this thesis, we propose an unsupervised hierarchical Bayesian model for activity analysis in multiple camera views without doing inference on the topology of camera views and without solving the correspondence problem. Further more, even though correspondence is not a prerequisite, after the models of activities have been learnt, they can help to solve the correspondence problem, since if two

Camera 1 Camera 2 Camera 3 Camera 4

(a) Parking lot scene



Camera 1 Camera 2 Camera 3 Camera 4

(b) Street scene

Figure 1-12: Examples of multiple camera views and their topology in two data sets, a parking lot scene and a street scene. When the topology of camera views is plotted, the fields of camera views are represented by different colors: blue (camera 1), red (camera 2), green (camera 3), yellow (camera 4). However, our approach does not require knowledge of the topology of the cameras in advance.

trajectories observed in different camera views belong to the same activity, they are likely to correspond to the same object.

We assume that the topology of camera views is unknown and quite arbitrary. The camera setting in our approach is as follows.

- The cameras are static and synchronized but not calibrated.
- The fields covered by these camera views may have no overlap or any amount of overlap. However we assume that when an object exits a camera view, it is already in or will enter one of the other camera views within a predefined time threshold T unless it moves out of range of the entire set.
- Objects may move on different ground planes.

Examples of multi-camera settings are shown in Figure 1-12.

This framework is an extension of the framework of clustering trajectories in a single camera view as described in Section 1.4.2. Using only identity co-occurrence, trajectories within a single camera view can be clustered, but we cannot cluster

trajectories across camera views since we do not track objects across camera views. In this framework, we assume that the trajectories of the same object observed in different camera views have temporal correlation. If two trajectories are observed in two camera views around the same time, they are more likely to be the same object and thus belong to be same activity category. A smoothness constraint is added into the Bayesian model as prior to cluster trajectories across camera views based on both identity co-occurrence and temporal co-occurrence information.

A network is first built by connecting with an edge trajectories that are observed in different camera views and whose temporal extents are close. Then a probabilistic model, in which a path has joint distribution in all the camera views, is built. If two trajectories are connected by an edge in the network, the prior of the smoothness constraint requires that they have similar distributions over paths. This part of the thesis work was previously published in [154, 155].

1.4.4 Tractography Segmentation

We use a technology similar to clustering trajectories of objects in a single camera views to cluster fiber trajectories into anatomically meaningful bundles. Existing tractography segmentation approaches had difficulty clustering very large scale data sets. Full brain tractography typically generates 10 thousand to 100 thousand fibers for each subject. Sometimes fibers from multiple subjects need to be clustered together. Existing approaches only cluster fibers from a subregion, or sample a small subset, such as five thousand fibers, from the large data set and learn the models of bundles from this subset. Thus important information from the full data set is lost. Many of these methods have difficulty deciding the number of clusters. It was shown that clustering performance of these approaches changed dramatically when different numbers of clusters were chosen [101]. A detailed review of existing approaches can be found in 2.5.

We use Dual-HDP to cluster fibers and learn the models of bundles from a training set without supervision. The 3D space of the brain is quantized into voxels. If two voxels are connected by many fibers, both of the voxels have large weights in the

(a) (b) (c)

Figure 1-13: An example of multiscale clustering. Multiscale clustering makes it easy for experts to identify anatomical structures across different scales. The spatial range of the whole brain is $200 \times 200 \times 200$. (a): The clustering result when the space is quantized into voxels of size $11 \times 11 \times 11$. The bundles correspond to structures at a large scale. (b): One bundle from (a). (c): The space is quantized into voxels of size $3 \times 3 \times 3$ and the bundle in (b) is further clustered into smaller bundles corresponding to structures at a finer scale.

model of the same bundle, which means that they are on the same pathway of white matter tracts. Our approach can cluster a large data set without sampling it and can learn the number of clustering driven by data. If we need to analyze a new subject, we use Dynamic Dual-HDP to cluster fibers from new subjects. The models of bundles learnt from training data are used as priors, and models are adapted to new data. Instead of fixing the number of clusters, our approach can create new clusters for structures which are observed in the new data but not in the training data.

Our framework can be extended to multiscale clustering. First we cluster fibers using a large size of voxels and bundles correspond to structures at a large scale. Then each bundle can be further clustered using a smaller size of voxels, leading to structures at a finer scale. An example is shown in Figure 1-13. Multiscale clustering makes it easier for experts to identify white matter structures across different scales.

1.4.5 Summary

All these proposed frameworks learn motion patterns from the co-occurrence of feature values (locations and moving directions) using hierarchical Bayesian models. Temporal co-occurrence is used to cluster moving pixels and learn the models of atomic activities in crowded scenes. Identity co-occurrence is used to cluster tra-

jectories and learn the models of paths in a single camera views. Both these two kinds of co-occurrence are used to cluster trajectories across camera views. These approaches are applicable to different types of scenes and have different limitations. If all types of activities most of the time happen together in the scene, which means that all the video clips have similar combinations of atomic activities, then temporal co-occurrence is not enough to separate co-occurring atomic activities and to learn their models. In that case, the approach described in Section 1.4.1 will not work well. This problem becomes more serious when the field covered by the camera view is larger and thus activities in the scene have more temporal overlap. The approach proposed in Section 1.4.2 only works well in sparse scenes where objects are trackable and the identity co-occurrence information can be reliably obtained.

Hierarchical Bayesian models allow us to jointly model simple and complicated motion patterns at different levels. Various knowledge and constraints can be nicely added into Bayesian frameworks as priors. Thus we can better solve problems which are difficult for nonBayesian approaches, such as modeling activities in multiple views, dynamically updating the models of activities, and clustering data of new subjects in tractography segmentation.

One commonality of these hierarchical Bayesian models proposed in this work is that they extract and quantize low-level features, and explore the co-occurrence of features at different hierarchical levels. These models can be applied to other fields such as language processing, object recognition and scene categorization where co-clustering problems arise. For example, our models can co-cluster words and documents in language processing. They can jointly cluster images into scene categories and cluster image patches into object classes at two different levels. These applications are interesting research directions in the future work.

1.5 Thesis Road Map

The remaining part of this thesis is organized as follows. Chapter 2 reviews related work on activity analysis in visual surveillance, tractography segmentation, hierarchi-

cal Bayesian models and nonparametric Bayesian approaches. Details of hierarchical Bayesian models developed in each of the four applications and experimental results are presented in Chapter 3 - 6. Chapter 3 first reviews two existing language processing models LDA and HDP. Then LDA mixture model, HDP mixture model, and Dual-HDP model are proposed. They advance LDA and HDP. Dual-HDP is applied to activity analysis in crowded scenes without tracking objects. Experimental results on a traffic scene are presented. In Chapter 4, Dual-HDP is applied to trajectory analysis in a single camera view and a dynamic Dual-HDP model is proposed to update the models of activities over time. Experimental results on radar tracks collected from a sea port and trajectories collected from a parking lot scene are presented. In Chapter 5, a trajectory network based hierarchical Bayesian model is proposed to cluster trajectories in multiple camera views. This approach is evaluated on a street scene and a parking lot scene. In Chapter 6, Dual-HDP and dynamic Dual-HDP are used for tractography segmentation and evaluated on multiple DT-MRI data sets. In Chapter 7, we discuss the limitations of this work and point out future directions we are interested in investigating. Finally contributions of this work are reiterated in Chapter 8.

Chapter 2

Literature Review

2.1 Activity Analysis Without Tracking Objects

Many approaches [164, 161, 132, 29, 158, 156] directly used motion feature vectors to describe video clips without tracking objects. For example, Zelnik-Manor and Irani [161] modeled and clustered video clips using multi-resolution histograms. Zhong et al. [164] also computed global motion histograms and did word-document analysis on videos. Their words were frames instead of moving pixels. They clustered video clips through the partition of a bipartite graph. Without object detection and tracking, a particular activity cannot be separated from other activities simultaneously occurring in the same clip, as is common in crowded scenes. These approaches treated a video clip as an integral entity and flagged the whole clip as normal or abnormal. They were often applied to simple data sets where there was only one kind of activity in a video clip. It is difficult for these approaches to model both single-agent activities and more complicated global behaviors. Although there were actions/events modeling approaches [114, 15, 105, 127, 84], which allowed one to detect and separate co-occurring activities, they were usually supervised. For example, Ke [76] proposed a supervised approach to detect human action in crowded scenes. At the training stage, they required manually isolating activities or a training video clip only contained one kind of activity. It is difficult for these approaches to separate co-occurring activities without supervision.

Ali and Shah [2] proposed an approach for motion segmentation at every frame in crowded videos based the spatial and temporal smoothness of motion flows. However, they did not have activity models shared by the entire video sequence and did not model global behaviors in the scene.

Our approach uses hierarchical Bayesian models to jointly model single-agent activities and more complicated global behaviors at different levels. It can separate co-occurring activities without supervision. Activity models are shared by the entire video sequence. Following our thesis work described in Section 1.4.1 and 3 and [151, 152], Li et al. [86, 63] used topic models for activity analysis scene segmentation based on moving pixels.

2.2 Trajectory Analysis and Scene Modeling with a Single Camera View

Many of existing trajectory analysis approaches cluster trajectories and detect abnormal trajectories by defining the pairwise similarities/distances between trajectories. The proposed trajectory similarities/distances include Euclidean distance [41], Hausdorff distance and its variations [71, 153], hidden Markov model [112], and Dynamic Time Warping (DTW) [77]. Some approaches required that two trajectories are temporally aligned when computing their distance. An alignment method, long common subsequence (LCSS) analysis was proposed [147, 22]. Instead of matching all points on a trajectory, it disregarded some outlier points. Similarly, Piciarelli and Foresti [110] defined a distance measure only matching part of the trajectory. A comparison of different similarity/distance measures can be found in [163]. Based on the computed similarity matrix, some standard clustering algorithms such as spectral clustering [104], graph-cuts [129], agglomerative and divisive hierarchical clustering [12, 87], and fuzzy c-means [57, 59] were used to group trajectories into different activity categories. These similarity/distance-based approaches have several drawbacks. First, there is no global probabilistic framework to model activities happening in the

scene. They have an *ad hoc* nature especially on the definitions of distance measures. Abnormal trajectories are usually detected as those with a larger distance to other trajectories. Their abnormality detection lacks a probabilistic explanation. Second, they usually do not provide a solution to the number of clusters. They often require that the cluster number is known in advance. Third, some approaches required temporal alignment of trajectories, which is sensitive to misdetection and tracking errors. Fourth, calculating the similarities between all pairs of samples is computationally inefficient, with a complexity of $O(M^2)$ in both time and space, where M is the number of trajectories. Some clustering algorithms such as spectral clustering need to compute the eigenvectors and eigenvalues of the similarity matrix and their computational cost is even higher. The complexity of labeling a new trajectory as one of the activity categories or as an abnormality is $O(M)$, since similarity-based approaches require computing the similarity between the new trajectory and each of the trajectories in the training set. Visual surveillance systems often require processing data collected over weeks or even months. These approaches have difficulty clustering very large data sets. If we monitor a parking lot for a month, there may be half million trajectories collected. It is even impossible to load such a huge similarity matrix into memory. Our approach uses a nonparametric Bayesian model for trajectory analysis. Under this framework, abnormality detection has a probabilistic explanation. The number of clusters of trajectories is learned driven by data with Dirichlet processes as priors. It does not require the computation of the similarity matrix. The space complexity of clustering trajectories is $O(M)$ instead of $O(M^2)$. The complexity of labeling a new trajectory as one of the activity categories or as abnormality is $O(K)$ where K is the number of clusters.

Another kind of approaches used features vectors from trajectories for clustering instead of computing pairwise distances. Because trajectories have variable length, it is difficult to directly use them as features for clustering. Subsampling may be required to let all the trajectory have the same length [96, 11, 88, 59]. Alternatively, some approaches projected trajectories into features spaces where feature vectors of fixed size were computed for clustering purpose. For example, the coefficients of least

squares polynomials [99, 160], Chebyshev polynomials [99], Haar transform [90], and discrete Fourier transform (DFT) [103] were used as features. Johnson and Hogg [68] first quantized the flow vectors of observations and then treated the pdf of a trajectory on the quantized flow vectors as a feature vector. Porikli [111] used HMM to characterize trajectories. Subspace methods such as PCA [10] and ICA [12] were used to construct a new space of lower dimensionality to improve clustering. A trajectory was summarized by the hidden state parameters and the transition matrix. Saleemi et al. [120] represented a trajectory with spatio-temporal variables (start location, destrination location and transition times). Basharat et al. [9] used the transitions of the state (destination location and size) of an object on a trajectory as features. The feature vectors of trajectories were clustered using algorithms such as k-means, self-organization map [68, 139], and fuzzy self-organizing neural network [60]. Kernel density estimation [120] and Gaussian mixture model [9] were used to estimated the probability distribution of trajectories in the feature space. Many of these approaches were sensitive to tracking errors. For example, the coefficients of DFT may change significantly if a trajectory is broken in the middle. Our approach does not compute the feature vectors of trajectories and it is more robust to tracking errors.

One way to decide the number of clusters is to minimize or maximize some objective criteria. Clustering is performed a number of times by varying the number of clusters. Hu et al. [57, 59] proposed a tightness and separation criterion, which measured how close trajectories in clusters were compared to the distance between clusters. Similar criteria were used in [111, 7] for spectral clustering. The Bayesian information criterion (BIC) was used in [67].

Trajectory clustering is also related to the problem of modeling scene structures. It takes a lot of effort to manually input these structures, and they cannot be reliably detected based on the appearance of the scene. In some cases, e.g. detecting shipping fairways on the sea, there is no appearance cue available at all. It is of interest to detect these structures by trajectory analysis. Usually paths were detected by modeling the spatial extents of trajectory clusters [38, 97, 71, 153]. Semantic regions were detected as intersections of paths [97, 110]. Entry and exit points were detected

at the ends of paths [134, 153].

2.3 Activity Analysis in Multiple Camera Views

Considerable work has been done to solve the challenging correspondence problem of trajectories observed in multiple camera views. One way is to manually label salient points in the scene and record their coordinates in the 3D world. After mapping 2D image planes to the 3D world [144, 52], objects can be tracked in multiple camera views. When the camera views overlap, static features can be selected to compute an assumed homography between two camera views [20] and camera views are calibrated to a single global ground plane. Trajectories in different camera views can be stitched based on their spatial proximity on the common ground plane. In general, automatically finding correspondence of static features between different views is difficult.

Lee et al. [85], Sheikh and Shah [128], and Stauffer and Tieu [136] calibrated multiple camera views using tracking data from moving objects. They also assumed that camera views had significant overlap and that objects moved on the same ground plane. Lee et al. [85] and Sheikh and Sheikh [128] assumed that the topological arrangement of camera views was known. Stauffer and Tieu [136] could automatically infer it, but with high complexity ($O(N^2)$ where N is the number of camera views).

When the camera views are disjointed or their overlap is small, automatic calibration is difficult and the appearance of objects is often used as a cue to correspondence [64, 65, 125, 48, 148]. This is a very challenging problem and not well solved yet. The appearance of objects may significantly change because of different cameras settings and different poses of objects. Many objects, such as cars and pedestrians, have similar appearance, confusing correspondence. In far-field settings, objects may only cover a few pixels, making matching difficult. Other approaches [98, 143] inferred the topology of disjoint camera views using the transition time between cameras.

Even given similarities between trajectories observed in different camera views, solving the correspondence problem is still difficult because of the large search space,

especially when there are many trajectories and cameras. It requires searching in the solution space of N-partite graphs, where N is the number of cameras [128]. In general, if there are more than two cameras, the problem is NP hard in the number of trajectories [43]. It has solution in polynomial time only with some particular topologies of camera views and the topology has to be known [64].

In summary, all these trajectory correspondence approaches had various assumptions on the topology and geometry of camera views, and they faced difficulties of camera calibration, appearance matching, inference on the topology of camera views, and high computational cost to search for the optimal solution. Given a general setting of a camera network, solving the correspondence problem is difficult. One of the contributions of this thesis is that we directly cluster trajectories into activities and model distributions of paths over a multi-camera network without solving the correspondence problem. So our method has less restriction on the topology of camera views, the structures of the scene and the number of cameras.

2.4 Probabilistic Approaches in Visual Surveillance

Probabilistic approaches were widely applied to object detection, tracking and event detection in visual surveillance [14, 108, 40, 54, 45, 106, 109]. Nillius et al. [106] used a Bayesian network to associate the identities of isolated tracks. Oliver et al. [108] used Coupled HMM to model the interaction between two objects. However, there are few studies on using graphical models to cluster tracks of objects into motion patterns. Pang et al. [109] proposed a Bayesian filtering framework to group targets which are moving together in similar directions and are close in space. It was evaluated on a very small data set only including four trajectories. This approach did not cluster whole trajectories since the group identities of targets might change dynamically. It only grouped targets moving at the same time, which means that trajectories were temporally aligned. In order to group targets observed at different time using this approach, trajectories have to be first aligned, which is one of the major difficulties in clustering trajectories since targets might be misdetected during some time windows

and trajectories might be broken or associated incorrectly because of tracking errors. Our models of clustering trajectories do not require the alignment of trajectories.

2.5 Tractography Segmentation

Automatically clustering fibers from DT-MRI has drawn a lot of attention in recent years. A typical framework is to first define a pairwise similarity/distance between fibers and then to input the similarity matrix to standard clustering algorithms. Various distances between fibers were proposed. Brun et al. [21] proposed a 9-D fiber shape descriptor, and computed the Euclidean distances between descriptors. Jonasson et al. [69] measured the similarity between two fibers by counting the number of points sharing the same voxel. Gerig et al. [46] proposed three measures related to Hausdorff distance: closest point distance, mean of closest distances and Hausdorff distance. Various clustering algorithms, such as hierarchical clustering (single-link and complete-link) [46, 157], fuzzy c-means [130], k-nearest neighbors [34], normalized cuts [21] and spectral clustering [69, 107] were used. Mean of closest distances and spectral clustering were popular among possible choices [107, 69].

These clustering algorithms required manually specifying the number of clusters or a threshold for deciding when to stop merging/splitting clusters, both of which are difficult to know especially when the data sets are complicated and noisy. Moberts et al. [101] showed that the performance of clustering varied dramatically when different numbers of clusters were chosen. To avoid this difficulty, O'Donnell and Westin [107] first chose a large cluster number, such as 200, for spectral clustering and then manually merged clusters to obtain models for white matter structures.

Another drawback of this framework is the high space and time complexities of computing pairwise distances between fibers when the data set is large. Whole brain tractography produces between $10,000$ and $100,000$ fibers per subject. It is difficult to compute a $100,000 \times 100,000$ similarity matrix or even to store it in memory. Some clustering algorithms, such as normalized cuts and spectral clustering, need to compute the eigenvectors and eigenvalues of this huge similarity matrix. This

problem becomes more serious when clustering fibers of multiple subjects. The current solutions are to cluster only a small portion (such as $5,000$ fibers) of the whole data set after subsampling or to do some numerical approximation based on the sampled subset [107]. However, important information from the full data set may be lost after subsampling.

Maddah et al. [94, 93, 95] proposed a probabilistic approach to cluster fibers without computing pairwise distances. They used a Dirichlet distribution as a prior to incorporate anatomical information. It was a parametric model, assuming that the number of clusters was known and required manual initialization of cluster centers. [95] required establishing point correspondence which was difficult, while our approach does not.

2.6 Hierarchical Bayesian Models in Computer Vision Applications

In computer vision, hierarchical Bayesian models have been applied to scene categorization [36], object recognition [131, 118, 138, 137, 149], and human action recognition [105]. References [36, 138, 137, 105] were supervised learning frameworks in the sense that they needed to manually label the documents. The video clip in [105] usually contained a single activity and [105] did not model interactions among multiple objects. References [131] and [118], which directly applied language processing models, probabilistic latent semantic analysis [53] and latent Dirichlet allocation (LDA) [18], were unsupervised frameworks assuming a document contains only one major topic. When we apply Dual-HDP to model trajectories and video clips, we assume that a document has multiple major topics. In our previous work [149], a novel hierarchical Bayesian model, Spatial Latent Dirichlet Allocation (SLDA), was proposed to learn models of object classes from image collections without requiring labeled data. SLDA improved LDA by modeling spatial and temporal structures among visual words, which are essential for solving many computer vision problems.

2.7 Nonparametric Bayesian Approaches

There has been a lot work [35, 50, 24, 165] on time dependent Dirichlet Process (DP) models published in recent years. Griffin et al. [50] proposed a framework, called Order-Based Dependent Dirichlet Processes (DDP), to model time series data. Caron et al. introduced a class of time-varying DP mixture models using a generated polya urn scheme. These works modeled time dependency of DP mixtures without more complicated hierarchical structures (such as hierarchical Dirichlet processes). Our dynamic Dual-HDP model is most relevant to [116] and [133]. Ren et al. [116] proposed a Dynamic Hierarchical Dirichlet Process model which was applied to music segmentation and analysis and gene expression data. In [116] the data of different time interval all shared the same set of topics, which did not change over time. It only modeled the dynamic change of the mixture weights of topics. Srebro et al. [133] integrated Ordered-Based DDP [50] into hierarchical topic models. They also assumed that topics were fixed over time. However, in our problem it is important for our dynamic Dual-HDP to model the dynamic change of topics, which reveals the change of the spatial distribution of semantic regions over time. Furthermore, dynamic Dual-HDP has a more complicated hierarchical structure with two layers of hierarchical Dirichlet processes.

In visual surveillance related tasks, Fox and Willsky et al. [40] used Dirichlet process to solve the problem of data association for multi-target tracking in the presence of an unknown number of targets.

Bayesian models involving Dirichlet process mixtures (DPM) are at the heart of the modern nonparametric Bayesian movement. DPM was applied to medical image analysis in recent years because of its capability to learn the number of clusters and its flexibility to adapt to a wide variety of data. Adelino [1] used a DPM model for brain MRI tissue classification. In [81, 142] DPM models were used to model spatial brain activation patterns in functional magnetic resonance imaging. In [66], Jbabdi et al. modeled the connectivity profiles of a brain region as an infinite mixture of multivariate Gaussian distributions with a Dirichlet Process prior. To the best of our

knowledge, our work is the first to use hierarchical Dirichlet process mixture models for tractography segmentation to automatically learn the number of clusters from data.

Chapter 3

Activity Analysis in Crowded and Complicated Scenes

This chapter presents hierarchical Bayesian models and experimental results on activity analysis in crowded and complicated scenes without tracking objects. Section 1.4.1 briefly summarizes our approach and Section 2.1 reviews related work.

There are some hierarchical Bayesian models for language processing, such as LDA [18] and HDP [140], from which we can borrow. Under LDA and HDP models, words, such as "professor" and "university", often co-existing in the same documents are clustered into the same topic, such as "education". HDP is a nonparametric model and automatically learns the number of topics from data while LDA requires knowing that in advance. We perform word-document analysis on video sequences. Moving pixels are quantized into visual words and short video clips are treated as documents. Directly applying LDA and HDP to our problem, atomic activities (corresponding to topics) can be discovered and modeled, however modeling global behaviors in the scene is not straightforward, since these models cannot cluster documents. Although LDA and HDP allow inclusion of more hierarchical levels corresponding to groups of documents, they require first manually labeling documents into groups. For example, [140] modeled multiple corpora but required knowing to which corpus each document belonged; [36] used LDA for scene categorization, but had to label each image in the training set into different categories. These were supervised frameworks. We propose

three novel hierarchical Bayesian models, LDA mixture model, HDP mixture model and Dual-HDP model. They co-cluster words and documents in an unsupervised way. In the case of visual surveillance, this means we can learn atomic activities as well as global behaviors without supervision. In fact, the problems of clustering moving pixels into atomic activities and of clustering video clips into global behaviors are closely related. The global behavior category of a video clip provides a prior for possible activities happening in that clip. It helps to cluster moving pixels into atomic activities. On the other hand, first clustering moving pixels into atomic activities provides an efficient representation for modeling interactions since it dramatically reduces the data dimensionality. Thus video clips can be clustered into global behaviors efficiently and effectively. We jointly solve these two problems together under a co-clustering framework. LDA mixture model assumes that the number of different types of atomic activities and global behaviors happening in the scene is known. HDP mixture model learns the number of categories of atomic activities driven by data. Dual-HDP learns the numbers of categories of both atomic activities and global behaviors.

The following sections explain the computation of motion features (Section 3.1), hierarchical Bayesian models (Section 3.2), and show the results in the application area of visual surveillance (Section 3.3).

3.1 Low-Level Motion Features

Our data is a video sequence from a far-field scene (a traffic scene as shown in Figure 1-6 (a) is used for our experiments) recorded by a fixed camera. There are myriads of activities and interactions in the video. It also involves many challenging problems, such as lighting changes, occlusions, a variety of object types, object view changes and environmental effects.

We compute local motions as our low-level features. Moving pixels are detected in each frame as follows. We compute the intensity difference between two successive frames, on a pixel basis. If the difference at a pixel is above a threshold, that pixel is detected as a moving pixel. The motion direction at each moving pixel is obtained by

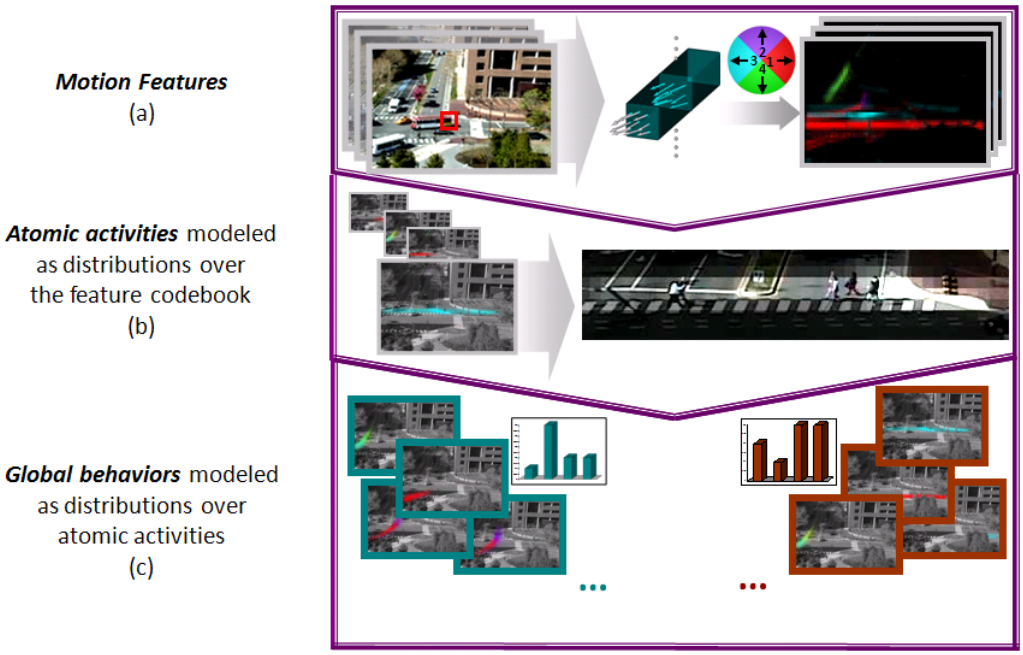


Figure 3-1: Diagram of the system for activity analysis in crowded and complicated scenes. Our framework connects low-level motion features, middle-level atomic activities and high-level global behaviors. (a) The observed long video sequence is uniformly divided into short clips as documents. In each video clip, moving pixels are detected and quantized into visual words based on their locations and motion directions. The four quantized directions are represented by four colors. Each video clip has a distribution over visual words. (b) Atomic activities (e.g. pedestrians cross the road) are discovered and modeled as distributions over visual words. Moving pixels are clustered into atomic activities. (c) Video clips are clustered into global behaviors, which are modeled as distributions over atomic activities.

computing optical flow [91]. The moving pixels are quantized according to a feature codebook, as follows. Each moving pixel has two features: position and direction of motion. To quantize position, the scene (in the size of 480×720 in our experiments) is uniformly divided into cells of size 10 by 10. The motion of a moving pixel is quantized into four directions as shown in Figure 3-1(a). Hence the size of the feature codebook is $48 \times 72 \times 4$, and thus each detected moving pixel is assigned a word from the codebook based on rough position and motion direction. Deciding the size of the codebook is a balance between the descriptive capability and complexity of the model. The whole video sequence is uniformly divided into non-overlapping short clips, each 10 seconds in length. In our framework, video clips are treated as documents and moving pixels are treated as words for word-document analysis as described in Section 3.2.

3.2 Hierarchical Bayesian Models

LDA [18] and HDP [140] were originally proposed as hierarchical Bayesian models for language processing. In these models, words that often co-exist in the same documents are clustered into the same topic. We extend these models by enabling clustering of both documents and words, thus finding co-occurring words (topics) and co-occurring topics (global behaviors). For far-field surveillance videos, words are quantized local motions of pixels; moving pixels that tend to co-occur in clips (or documents) are modeled as topics. Our goal is to infer the set of atomic activities (or topics) from video by learning the distributions of features that co-occur, and to learn distributions of atomic activities that co-occur, thus finding global behaviors. Three new hierarchical Bayesian models are proposed in this section: LDA mixture model, HDP mixture model, and Dual-HDP model.

3.2.1 LDA

Figure 3-2(a) shows the LDA model of [18]. Suppose the corpus has M documents. Each document is modeled as a mixture of K topics, where K is assumed known.

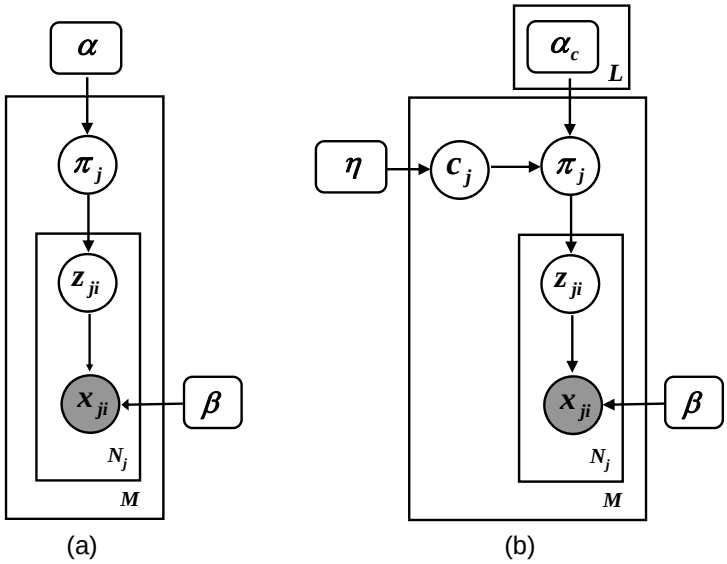


Figure 3-2: (a) LDA model proposed in [18]; (b) our LDA mixture model.

Each topic k is modeled as a multinomial distribution $\beta_k = [\beta_{k1}, \ldots, \beta_{kW}]$ over a codebook of size W. $\beta = \{\beta_k\}$. $\alpha = [\alpha_1, \ldots, \alpha_K]$ is a Dirichlet prior on the corpus. For each document j, a parameter $\pi_j = [\pi_{j1}, \ldots, \pi_{jK}]$ of a multinomial distribution over the K topics is drawn from Dirichlet distribution $Dir(\pi_j|\alpha)$,

$$p(\pi_j|\alpha) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \pi_{j1}^{\alpha_1 - 1} \cdots \pi_{jK}^{\alpha_K - 1},$$

where $\Gamma(\cdot)$ is a gamma function. For each word i in document j, a topic label $z_{ji} = k$ is drawn with probability π_{jk}, and word x_{ji} is drawn from a discrete distribution given by $\beta_{z_{ji}}$. π_j and z_{ji} are hidden variables. α and β are hyperparameters to be optimized. Given α and β, the joint distribution of topic mixture π_j, topics $\mathbf{z}_j = \{z_{ji}\}$, and words $\mathbf{x}_j = \{x_{ji}\}$ is:

$$\begin{aligned} p(\mathbf{x}_j, \mathbf{z}_j, \pi_j|\alpha, \beta) =& p(\pi_j|\alpha) \prod_{i=1}^{N_j} p(z_{ji}|\pi_j) p(x_{ji}|z_{ji}, \beta) \\ =& \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \pi_{j1}^{\alpha_1 - 1} \cdots \pi_{jK}^{\alpha_K - 1} \prod_{i=1}^{N_j} \pi_{j z_{ji}} \beta_{z_{ji} x_{ji}} \end{aligned} \tag{3.1}$$

where N_j is the number of words in document j. Unfortunately, the marginal likelihood $p(\mathbf{x}_j|\alpha,\beta)$ and thus the posterior distribution $p(\pi_j,\mathbf{z}_j|\alpha,\beta)$ are intractable for exact inference. Thus in [18], a Variational Bayes (VB) inference algorithm used a family of variational distributions:

$$q(\pi_j,\mathbf{z}_j|\gamma_j,\phi_j) = q(\pi_j|\gamma_j)\prod_{i=1}^{N_j} q(z_{ji}|\phi_{ji}) \tag{3.2}$$

to approximate $p(\pi_j,\mathbf{z}_j|\alpha,\beta)$, where the Dirichlet parameter γ_j and multinomial parameters $\{\phi_{ji}\}$ are free variational parameters. The optimal (γ_j,ϕ_j) is computed by finding a tight lower bound on $\log p(\mathbf{x}_j|\alpha,\beta)$.

3.2.2 LDA Mixture Model

This LDA model in [18] does not cluster documents. All the documents share the same Dirichlet prior α. In activity analysis, we assume that video clips (documents) of the same type of global behavior would include a similar set of atomic activities (topics), so they could be grouped into the same cluster and share the same prior over topics. Our LDA mixture model is shown in Figure 3-2(b). The M documents in the corpus will be grouped into L clusters. Each cluster c has its own Dirichlet prior α_c. For a document j, the cluster label c_j is first drawn from discrete distribution η, and π_j is drawn from $Dir(\pi_j|\alpha_{c_j})$. Given $\{\alpha_c\}$, β, and η, the joint distribution of hidden variables c_j, π_j, $\mathbf{z}_j$ and observed words $\mathbf{x}_j$ is

$$p(\mathbf{x}_j,\mathbf{z}_j,\pi_j,c_j|\{\alpha_c\},\beta,\eta) = p(c_j|\eta)p(\pi_j|\alpha_{c_j})\prod_{i=1}^{N} p(z_{ji}|\pi_j)p(x_{ji}|z_{ji},\beta) \tag{3.3}$$

The marginal log likelihood of document j is:

$$\log p(\mathbf{x}_j|\{\alpha_c\},\beta,\eta) = \log\sum_{c_j=1}^{L} p(c_j|\eta)p(\mathbf{x}_j|\alpha_{c_j},\beta) \tag{3.4}$$

Our LDA mixture model is relevant to the model proposed in [36]. However, the hidden variable c_j in our model was observed in [36]. So [36] required manually labeling documents in the training set, while our framework is totally unsupervised. This causes a different inference algorithm to be proposed for our model. Using VB [18], $\log p(\mathbf{x}_j|\alpha_{c_j}, \beta)$ can be approximated by a tight lower bound $L_1(\gamma_{jc_j}, \phi_{jc_j}; \alpha_{c_j}, \beta)$,

$$\begin{aligned}
\log p(\mathbf{x}_j|\alpha_{c_j}, \beta) &= \log \int_{\pi_j} \sum_{\mathbf{z}_j} p(\pi_j, \mathbf{z}_j, \mathbf{x}_j|\alpha_{c_j}, \beta) d\pi_j \\
&= \log \int_{\pi_j} \sum_{\mathbf{z}_j} \frac{p(\pi_j, \mathbf{z}_j, \mathbf{x}_j|\alpha_{c_j}, \beta) q(\mathbf{z}_j, \pi_j|\gamma_{jc_j}, \phi_{jc_j})}{q(\mathbf{z}_j, \pi_j|\gamma_{jc_j}, \phi_{jc_j})} d\pi_j \\
&\geq \int_{\pi_j} \sum_{\mathbf{z}_j} q(\mathbf{z}_j, \pi_j|\gamma_{jc_j}, \phi_{jc_j}) \log p(\mathbf{x}_j, \mathbf{z}_j, \pi_j|\alpha_{c_j}, \beta) d\pi_j \\
&\quad - \int_{\pi_j} \sum_{\mathbf{z}_j} q(\mathbf{z}_j, \pi_j|\gamma_{jc_j}, \phi_{jc_j}) \log q(\mathbf{z}_j, \pi_j|\gamma_{jc_j}, \phi_{jc_j}) d\pi_j \\
&= L_1(\gamma_{jc_j}, \phi_{jc_j}; \alpha_{c_j}, \beta). \qquad (3.5)
\end{aligned}$$

However because of the marginalization over c_j, hyperparameters are still coupled even using VB. So we use both Expectation-Maximization (EM) [30] and VB to estimate hyperparameters. After using VB to compute the lower bound of $\log p(\mathbf{x}_j|\alpha_{c_j}, \beta)$, an averaging distribution $q(c_j|\gamma_{jc_j}, \phi_{jc_j})$ can provide a further lower bound on the log likelihood,

$$\begin{aligned}
\log p(\mathbf{x}_j|\{\alpha_c\}, \beta, \eta) &\geq \log \sum_{c_j=1}^{L} p(c_j|\eta) e^{L_1(\gamma_{jc_j}, \phi_{jc_j}; \alpha_{c_j}, \beta)} \\
&= \log \sum_{c_j=1}^{L} q(c_j|\gamma_{jc_j}, \phi_{jc_j}) \frac{p(c_j|\eta) e^{L_1(\gamma_{jc_j}, \phi_{jc_j}; \alpha_{c_j}, \beta)}}{q(c_j|\gamma_{jc_j}, \alpha_{jc_j})} \\
&\geq \sum_{c_j=1}^{L} q(c_j|\gamma_{jc_j}, \phi_{jc_j}) \left[\log p(c_j|\eta) + L_1(\gamma_{jc_j}, \phi_{jc_j}; \alpha_{c_j}, \beta)\right] \\
&\quad - \sum_{c_j=1}^{L} q(c_j|\gamma_{jc_j}, \phi_{jc_j}) \log q(c_j|\gamma_{jc_j}, \phi_{jc_j}) \\
&= L_2(q(c_j|\gamma_{jc_j}, \phi_{jc_j}), \{\alpha_c\}, \beta, \eta) \qquad (3.6)
\end{aligned}$$

L_2 is maximized when choosing

$$q(c_j|\gamma_{jc_j}, \phi_{jc_j}) = \frac{p(c_j|\eta)e^{L_1(\gamma_{jc_j},\phi_{jc_j};\alpha_{c_j},\beta)}}{\sum_{c_j} p(c_j|\eta)e^{L_1(\gamma_{jc_j},\phi_{jc_j};\alpha_{c_j},\beta)}}. \tag{3.7}$$

Our EM algorithm for hyperparameters estimation is:

1. For each document j and cluster c_j, find the optimal values of the variational parameters $\{\gamma^*_{j,c_j}, \phi^*_{j,c_j} : j = 1, \ldots, M; c_j = 1, \ldots, L\}$ to maximize L_1 (using VB [18]).

2. Compute $q(c_j|\gamma^*_{jc_j}, \phi^*_{jc_j})$ using (3.7) to maximize L_2.

3. Maximize L_2 with respect to $\{\alpha_c\}$, β, and η. β and η are optimized by setting the first derivative to zero,

$$\eta_c \propto \sum_{j=1}^{M} q(c_j = c|\gamma^*_{jc}, \phi^*_{jc}) \tag{3.8}$$

$$\beta_{kw} \propto \sum_{j=1}^{M}\sum_{c_j=1}^{L} q(c_j|\gamma^*_{jc_j}, \phi^*_{jc_j}) \left[\sum_{i=1}^{N} \phi^*_{jc_j ik} x^w_{ji}\right] \tag{3.9}$$

where $x^w_{ji} = 1$ if $x_{ji} = w$, otherwise it is 0. The $\{\alpha_c\}$ are optimized using a Newton-Raphson algorithm. The first and second derivatives are:

$$\frac{\partial L_2}{\partial \alpha_{ck}} = \sum_{j=1}^{M} q(c_j = c|\gamma_{jc}, \phi_{jc})[\Psi(\sum_{k=1}^{K}\alpha_{ck}) - \Psi(\alpha_{ck}) + \Psi(\gamma_{jck}) - \Psi(\sum_{j=1}^{k}\gamma_{jck})] \tag{3.10}$$

$$\frac{\partial^2 L_2}{\partial \alpha_{ck_1}\alpha_{ck_2}} = \sum_{j=1}^{M} q(c_j = c|\gamma_{jc}, \phi_{jc})[\Psi'(\sum_{k=1}^{K}\alpha_{ck}) - \delta(k_1, k_2)\Psi'(\alpha_{ck_1})] \tag{3.11}$$

where Ψ is the first derivative of log Gamma function.

L_2 monotonously increases after each iteration.

3.2.3 Dirichlet Process

Both LDA and LDA mixture are parametric Bayesian models. They require that the numbers of topics and clusters of documents are manually specified. Using Dirichlet Process (DP) [37] as prior, they can be extended to nonparametric Bayesian models which learn the number of clusters driven by data.

DP is used as a prior to sample probability measures. It is defined by a concentration parameter γ, which is a positive scalar, and a base probability measure H. A probability measure G randomly drawn from $DP(\gamma, H)$ is always a discrete distribution,

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}, \tag{3.12}$$

which can be obtained from a stick-breaking construction [124]. In Eq (3.12), ϕ_k is a parameter vector sampled from H, δ_{ϕ_k} is a Dirac delta function centered at ϕ_k, and π_k ($\sum_{k=1}^{\infty} \pi_k = 1$) is a non-negative scalar constructed by

$$\pi_k = \pi'_k \prod_{l=1}^{k-1} (1 - \pi'_l),$$

$$\pi'_k \sim Beta(1, \gamma).$$

G can be used as a prior of infinite mixture models. Let $\{x_i\}$ be a set of observed data points. x_i is sampled from a density function $p(\cdot|\theta_i)$ parameterized by θ_i, and θ_i (which is one of the ϕ_ks in Eq (3.12)) is sampled from G. Data points sharing the same parameter vector ϕ_k are clustered together under this mixture model. Given parameter vectors $\theta_1, \ldots, \theta_N$ of N data points, the parameter vector θ_{N+1} of data point x_{N+1} can be sampled from a prior by integrating out G,

$$\theta_{N+1}|\theta_1, \ldots, \theta_N, \gamma, H \sim \sum_{k=1}^{K} \frac{n_k}{N + \gamma} \delta_{\theta^*_k} + \frac{\gamma}{N + \gamma} H. \tag{3.13}$$

There are K distinct parameter vectors $\{\theta^*_k\}_{k=1}^{K}$ (identifying K components) among $\theta_1, \ldots, \theta_N$. n_k is the number of points with parameter vector θ^*_k. θ_{N+1} can be assigned

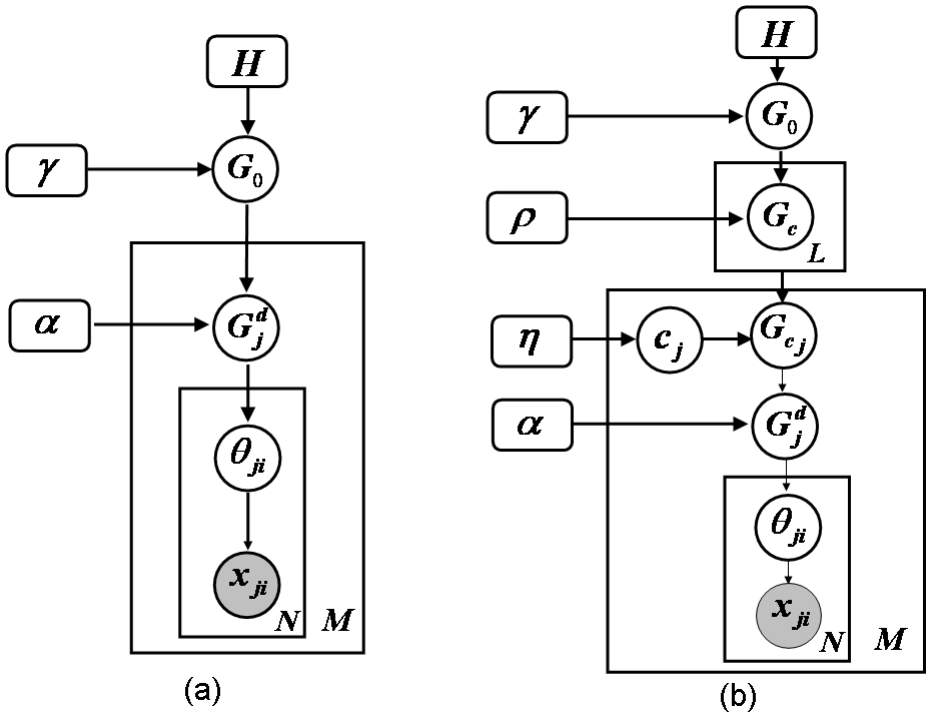


Figure 3-3: (a) *HDP* model proposed in [140]; (b) our *HDP* mixture model.

as one of the existing components (x_{N+1} is assigned to one of the existing clusters) or can sample a new component from H (a new cluster is created for x_{N+1}). The posterior of θ_{N+1} is

$$p(\theta_{N+1}|x_{N+1}, \theta_1, \ldots, \theta_N, \gamma, H) \propto p(x_{N+1}|\theta_{N+1}) p(\theta_{N+1}|\theta_1, \ldots, \theta_N, \gamma, H). \tag{3.14}$$

It is likely for the infinite mixture model with DP prior to create a new component if existing components cannot well explain the data. There is no limit to the number of components. These properties make DP ideal for modeling data clustering problems when the number of clusters is not well-defined in advance.

3.2.4 HDP

HDP proposed by Teh et al. [140] is a nonparametric hierarchical Bayesian model and its corresponding parametric model is LDA. HDP automatically learns the number of topics driven by data. The graphical model of HDP is shown in 3-3. In HDP, a prior distribution G_0 over the whole corpus is sampled from a Dirichlet process,

$G_0 \sim DP(\gamma, H)$. $G_0 = \sum_{k=1}^{\infty} \pi_{0k}\delta_{\phi_k}$. ϕ_k is the parameter of a topic, which is modeled as a multinomial distribution over the codebook. ϕ_k is sampled from Dirichlet prior H. All the words in the corpus are sampled from some topics $\{\phi_k\}$. $\{\pi_{0k}\}$ are mixture weights over topics. For each document j, a random measure G_j^d is drawn from a Dirichlet process with concentration parameter α and base probability measure G_0: $G_j^d|\alpha, G_0 \sim DP(\alpha, G_0)$. Each G_j^d has support at the same locations $\{\phi_k\}_{k=1}^{\infty}$ as G_0, i.e. all the documents share the same set of topics, and can be written as $G_j^d = \sum_{k=1}^{\infty} \pi_{jk}\delta_{\phi_k}$. G_j^d is a prior distribution of all the words in document j. For each word i in document j, a parameter vector θ_{ji} of a topic is drawn from G_j^d (θ_{ji} is sampled as one of the ϕ_k's). Word x_{ji} is drawn from discrete distribution $Discrete(\theta_{ji})$,

$$p(x_{ji}|\theta_{ji}) = \theta_{jix_{ji}},$$

where $\theta_{ji} = (\theta_{ji1}, \ldots, \theta_{jiW})$. In [140], Gibbs sampling schemes were used to do inference under an HDP model.

3.2.5 HDP Mixture Model

In our HDP mixture model, as shown in Figure 3-3 (b), clusters of documents are modeled and each cluster c has a random probability measure G_c. G_c is drawn from Dirichlet process $DP(\rho, G_0)$. For each document j, a cluster label c_j is first drawn from discrete distribution $p(c_j|\eta)$. Document j chooses G_{c_j} as the base probability measure and draws its own G_j^d from Dirichlet process $G_j^d \sim DP(\alpha, G_{c_j})$. We also use Gibbs sampling for inference. In our HDP mixture model, there are two kinds of hidden variables to be sampled: (1) variables $\mathbf{z} = \{z_{ij}\}$ assigning words to topics, base distributions G_0 and $\{G_c\}$; and (2) cluster label c_j. The key issue to be solved in this thesis is how to sample c_j. Given c_j is fixed, the first kind of variables can be sampled using the same scheme described in [140]. We will not repeat the details here. We focus on the step of sampling c_j, which is the new part of our model compared with HDP in [140].

At some sampling iteration, suppose that there have been K topics, $\{\phi_k\}_{k=1}^{K}$,

generated and assigned to the words in the corpus (K is variable during the sampling procedure). G_0, G_c, and G_j^d can be expressed as,

$$G_0 = \sum_{k=1}^{K} \pi_{0k}\delta_{\phi_k} + \pi_{0u}G_{0u},$$

$$G_c = \sum_{k=1}^{K} \pi_{ck}\delta_{\phi_k} + \pi_{cu}G_{cu},$$

$$G_j^d = \sum_{k=1}^{K} \omega_{jk}\delta_{\phi_k} + \omega_{ju}G_{ju}^d,$$

where G_{0u}, G_{cu}, and G_{ju}^d are distributed as Dirichlet process $DP(\gamma, H)$. Note that the prior over the corpus (G_0), a cluster of documents (G_c) and a document G_j^d share the same set of topics $\{\phi_k\}$. However, they have different mixtures over topics.

Using the sampling schemes in [140], topic mixtures $\pi_0 = \{\pi_{01}, \ldots, \pi_{0K}, \pi_{0u}\}$, $\pi_c = \{\pi_{c1}, \ldots, \pi_{cK}, \pi_{cu}\}$ are sampled, while $\{\phi_k\}$, G_{0u}, G_{cu}, G_{ju}^d, and $\omega_j^d = \{\omega_{j1}, \ldots, \omega_{jK}, \omega_{ju}\}$ can be integrated out without sampling. In order to sample the cluster label c_j of document j, the posterior $p(c_j = c|(m_{j1}, \ldots, m_{jK}), \pi_0, \{\pi_c\})$ has to be computed where m_{jk} is the number of words assigned to topic k in document j and is computable from $\mathbf{z}$:

$$\begin{aligned} & p(c_j = c|(m_{j1}, \ldots, m_{jK}), \pi_0, \{\pi_c\}) \\ \propto & p(m_{j1}, \ldots, m_{jK}|\pi_c)p(c_j = c) \\ = & \eta_c \int p(m_{j1}, \ldots, m_{jK}|\omega_j^d)p(\omega_j^d|\pi_c)d\omega_j^d. \end{aligned}$$

$p(m_{j1}, \ldots, m_{jK}|\omega_j^d)$ is a multinomial distribution. Since G_j^d is drawn from $DP(\alpha, G_c)$,

$p(\omega_j^d|\pi_c)$ is a Dirichlet distribution $Dir(\omega_j^d|\alpha \cdot \pi_c)$. Thus we have

$$
\begin{aligned}
& p(c_j = c|(m_{j1}, \ldots, m_{jK}), \pi_0, \{\pi_c\}) \\
\propto & \eta_c \int \frac{\Gamma(\alpha\pi_{cu} + \alpha \sum_{k=1}^K \pi_{ck})}{\Gamma(\alpha\pi_{cu}) \prod_{k=1}^K \Gamma(\alpha\pi_{ck})} \omega_{ju}^{\alpha\pi_{cu}-1} \prod_{k=1}^K \omega_{jk}^{\alpha\pi_{ck}+m_{jk}-1} d\omega_j^d \\
\propto & \frac{\Gamma(\alpha\pi_{cu} + \alpha \sum_{k=1}^K \pi_{ck})}{\Gamma(\alpha\pi_{cu}) \prod_{k=1}^K \Gamma(\alpha\pi_{ck})} \frac{\Gamma(\alpha\pi_{cu}) \prod_{k=1}^K \Gamma(\alpha\pi_{ck} + m_{jk})}{\Gamma(\alpha\pi_{cu} + \sum_{k=1}^K (\alpha\pi_{ck} + m_{jk}))} \\
= & \eta_c \frac{\Gamma(\alpha)}{\Gamma(\alpha + N_j)} \frac{\prod_{k=1}^K \Gamma(\alpha \cdot \pi_{ck} + m_{jk})}{\prod_{k=1}^K \Gamma(\alpha \cdot \pi_{ck})} \propto \eta_c \frac{\prod_{k=1}^K \Gamma(\alpha \cdot \pi_{ck} + m_{jk})}{\prod_{k=1}^K \Gamma(\alpha \cdot \pi_{ck})}.
\end{aligned} \tag{3.15}
$$

where Γ is the Gamma function.

So the Gibbs sampling procedure repeats the following two steps alternatively at every iteration:

1. given $\{c_j\}$, sample $\mathbf{z}$, π_0, and $\{\pi_c\}$ using the schemes in [140];
2. given $\mathbf{z}, \pi_0$, and $\{\pi_c\}$, sample cluster labels $\{c_j\}$ using posterior Eq 3.15.

In this section, we assume that the concentration parameters γ, ρ, and α are fixed. In actual implementation, we give them a vague gamma prior and sample them using the scheme proposed in [140]. Thus these concentration parameters are sampled from a broad distribution instead of being fixed at a particular point.

3.2.6 Dual-HDP

In this section, we propose a Dual-HDP model which automatically learns both the number of word topics and the number of document clusters. In addition to the hierarchical Dirichlet processes which model the word topics, there is another layer of hierarchical Dirichlet processes modeling the clusters of documents. Hence we call this a Dual-HDP model. The graphical model of Dual-HDP is shown in Figure 3-4. In the HDP mixture model, each document j has a prior G_{c_j} drawn from a finite mixture $\{G_c\}_{c=1}^L$. In Dual-HDP, G_{c_j} is drawn from an infinite mixture,

$$
Q = \sum_{c=1}^{\infty} \epsilon_c \delta_{G_c} \tag{3.16}
$$

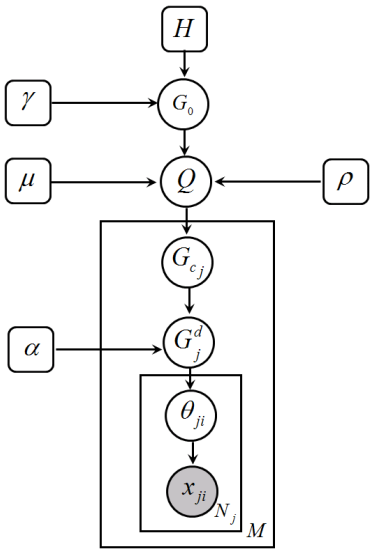


Figure 3-4: The graphical model of Dual-HDP.

Notice that G_c itself is a random distribution with infinite parameters. When a Dirichlet process was first developed by Ferguson [37], the location parameters (such as ϕ_k in Eq. 3.12) could only be scalars or vectors. MacEachern [92] made an important generalization and proposed the Dependent Dirichlet Processes (DDP). DDP replaces the locations in the stick-breaking representation with stochastic processes and introduces dependence in a collection of distributions. The parameters $\{(\pi_{ck}, \phi_{ck})\}_{k=1}^{\infty}$ of G_c can be treated as a stochastic process with index k. Q can be treated as a set of dependent distributions, $Q = \{Q_k = \sum_{c=1}^{\infty} \epsilon_c \delta_{(\pi_{ck}, \phi_{ck})}\}_{k=1}^{\infty}$. So we can generate Q through DDP.

As shown in Figure 3-4, Q is sampled from $DDP(\mu, \rho, G_0)$. In Eq (3.16), $\epsilon_c = \epsilon'_c \prod_{l=1}^{c-1}(1-\epsilon'_l)$, $\epsilon'_c \sim Beta(1, \mu)$, and $G_c \sim DP(\rho, G_0)$. Similar to the HDP mixture model in Figure 3-3 (b), $G_0 \sim DP(\lambda, H)$ is the prior over the whole corpus and generates topics shared by all of the words. $\{G_c\}_{c=1}^{\infty}$ all have the same topics in G_0, i.e. $\phi_{ck} = \phi_k$. However they have different mixtures $\{\pi_{ck}\}_{k=1}^{\infty}$ over these topics.

Each document j samples a probability measure G_{c_j} from Q as its prior. Different documents may choose the same prior G_c, thus they form one cluster. So in Dual-HDP, the two infinite mixtures Q and G_0 model the clusters of documents and words respectively. The following generative procedure is the same as HDP mixture model. Document j generates its own probability measure G_j^d from $G_j^d \sim DP(\alpha, G_{c_j})$.

Word i in document j samples topic ϕ_k from G_j^d and samples its word value from $Discrete(\phi_k)$.

Gibbs sampling was also used for inference and learning on Dual-HDP. The Gibbs sampling procedure can be divided into two steps:

1. given the cluster assignment $\{c_j\}$ of documents is fixed, sample the word topic assignment $\mathbf{z}$, masses π_0 and π_c on topics using the schemes in [140];

2. given $\mathbf{z}$, masses π_0 and π_c, sample the cluster assignment $\{c_j\}$ of documents. c_j can be assigned to one of the existing clusters or to a new cluster. We use the Chinese restaurant franchise for sampling. See details in the Appendix A.

3.2.7 Discussion on the co-clustering framework

We propose three words-documents co-clustering models. Readers may ask why we need a co-clustering framework? Can we first cluster words into topics and then cluster documents based on their distributions over topics, or solve the two problems separately? In visual surveillance applications, the issue is about simultaneously modeling atomic activities and global behaviors. In the language processing literature, there has been considerable work dealing with word clustering [53, 18, 140] and document clustering [122, 33, 162] separately. Dhillon [32] showed the duality of words and documents clustering: "word clustering induces document clustering while document clustering induces words clustering". Information on the category of documents helps to solve the ambiguity of word meaning and vice versus. Thus a co-clustering framework can solve the two closely related problems in a better way. Dhillon [32] co-clustered words and documents by partitioning a bipartite spectral graph with words and documents as vertices. However, one cluster of documents only corresponded to one cluster of words. [53, 18] showed that one document may contain several topics. In a visual surveillance data set, one video clip may contain several atomic activities. Our co-clustering algorithms based on hierarchical Bayesian models can better solve these problems.

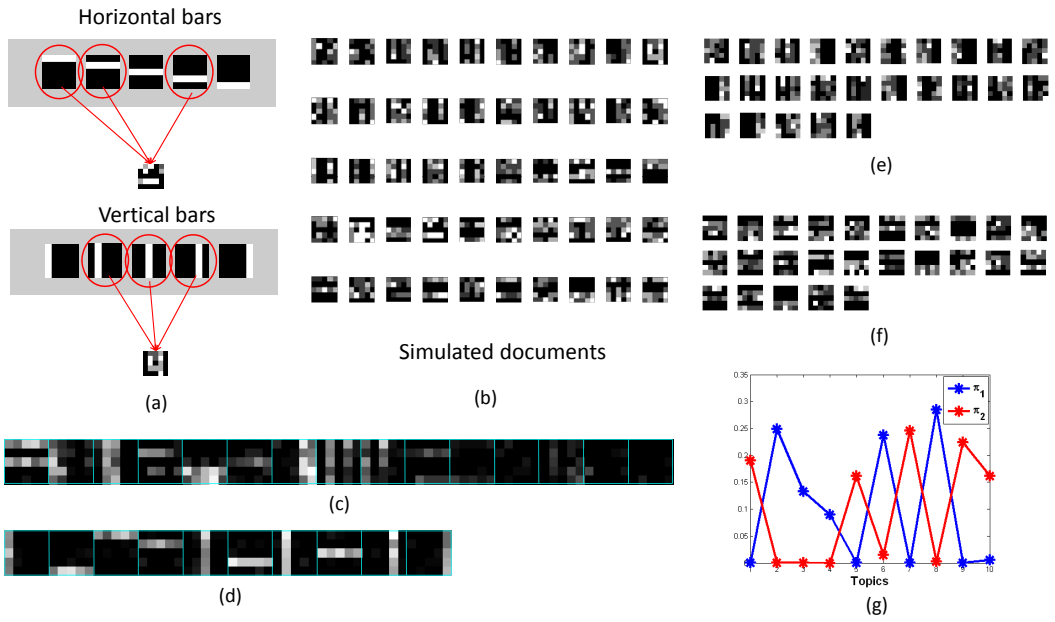


Figure 3-5: Experimental comparison of HDP and Dual-HDP on a toy example. (a) There are ten topics with distributions along horizontal bars and vertical bars. A synthetic document can be generated in one of the two ways. It randomly combines several vertical bar topics and sample words from them or randomly combines several horizontal bar topics. (b) The simulated documents. (c) Topic distributions learnt by HDP. (d) Topics distributions learnt by Dual-HDP. Documents are grouped into two clusters shown in (e) and (f). (g) Topic mixtures of two clusters π_1 and π_2.

3.2.8 Example of synthetic data

We use an example of synthetic data to demonstrate the strength of our hierarchical Bayesian models (see Figure 3-5). The toy data is similar as that used in [51]. The word vocabulary is a set of 5×5 cells. There are 10 topics with distributions over horizontal bars and vertical bars (Figure 3-5 (a)), i.e., words tend to co-occur along the same row or column, but not arbitrarily. The document is represented by a image with 25 pixels in a 5×5 grid. Each pixel is a word and the intensity of a pixel is the frequency of the word. If we generate documents by randomly choosing several topics from the ten, adding noise to the bar distributions, and sampling words from these bars, there are only two levels of structures (topics and words) in the data and the HDP model in [140] can perfectly discover the 10 topics. However, in our experiments in Figure 3-5, we add one more level, clusters of documents, to the data. Documents are from two clusters: a vertical-bars cluster and a horizontal-bars cluster.

If a document is from the vertical-bars cluster, it randomly combines several vertical bar topics and sample words from them, otherwise, it randomly combines horizontal bar topics. As seen in Figure 3-5 (c), HDP has much worse performance on this data. There are two kinds of correlation among words: if words are on the same bar, they often co-exist in the same documents; if words are all on horizontal bars or vertical bars, they are also likely to be in the same documents. It is improper to use a two-level HDP to model data with a three-level structure. 15 topics are discovered and many of the topics include more than one bar as shown in Figure 3-5 (c). Using our HDP mixture model and Dual-HDP model to co-cluster words and documents, the 10 topics are discovered nearly perfectly as shown in Figure 3-5(d). In the meanwhile, the documents are grouped into two clusters as shown in Figure 3-5 (e) and (f). The topic mixtures π_1 and π_2 of these two clusters are shown in Figure 3-5 (g). π_1 only has large weights on horizontal bar topics while π_2 only has large weights on vertical bar topics. Thus our approach recovers common topics (i.e. words that co-occur) and common documents (i.e. topics that co-occur). For Dual-HDP, as initialization all the words are in one topic and all the documents are in one cluster.

3.3 Visual Surveillance Applications and Experimental Results

After computing the low-level visual features as described in Section 3.1, we divide our video sequence into 10 seconds long clips, each treated as a document, and feed these documents to the hierarchical Bayesian models described in Section 3.2. In this section, we explain how to use the results from hierarchical Bayesian models for activity analysis. We will mainly show results from Dual-HDP, since it automatically decides the number of word topics and the number of document clusters, while LDA mixture model and HDP mixture model need to know those in advance. However, if the number of word topics and the number of document clusters are properly set in LDA mixture model and HDP mixture model, they provide very

similar results. Experimental results are from a traffic scene. The video sequence lasts 90 minutes. Some video examples of our results can be found from our website (http://groups.csail.mit.edu/vision/app/research/HBM.html).

3.3.1 Discover Atomic Activities

In visual surveillance, people often ask "what are the typical activities and global behaviors in this scene?" The parameters estimated by our hierarchical Bayesian models provide a good answer to this question.

As explained in Section 1.4.1, an atomic activity usually causes temporally continuous motions and does not stop in the middle. So the motion features caused by the same kind of atomic activity often co-occur in the same video clips. Since the moving pixels are treated as words in our hierarchical Bayesian models, the topics of words are actually a summary of typical atomic activities in the scene. Each topic has a multinomial distribution over the motion feature codebook, specified by β in LDA mixture model and $\{\phi_k\}$ in Dual-HDP. (ϕ_k can be easily estimated given the words assigned to topic k after sampling).

Our Dual-HDP model automatically discovers 29 atomic activities in the traffic scene. In Figure 3-6, we show the distributions of these atomic activities over locations and moving directions. The atomic activities are sorted by size (the number of moving pixels assigned to the atomic activity) from large to small. The numbers of moving pixels assigned to atomic activities are shown in Figure 3-7. Atomic activity 2 is "vehicles make a right turn". Atomic activities 5, 14, and 20 are "vehicles make left turns". Atomic activities 6 and 9 are "vehicles cross road d, but along different lanes". Atomic activities 1 and 4 are "vehicles pass road d from left to right". This activity is broken into two atomic activities because when vehicles from g make a right turn (see atomic activity 2) or vehicles from road e make a left turn (see atomic activity 14), they also share the motion pattern in atomic activity 4. Atomic activities 10 and 19 are "vehicles come to stop behind the stop lines". Atomic activities 13, 17 and 21 are "pedestrians walk on crosswalks". When people pass the crosswalk a, they often stop at the divider between roads e and f waiting for vehicles to pass by. So this

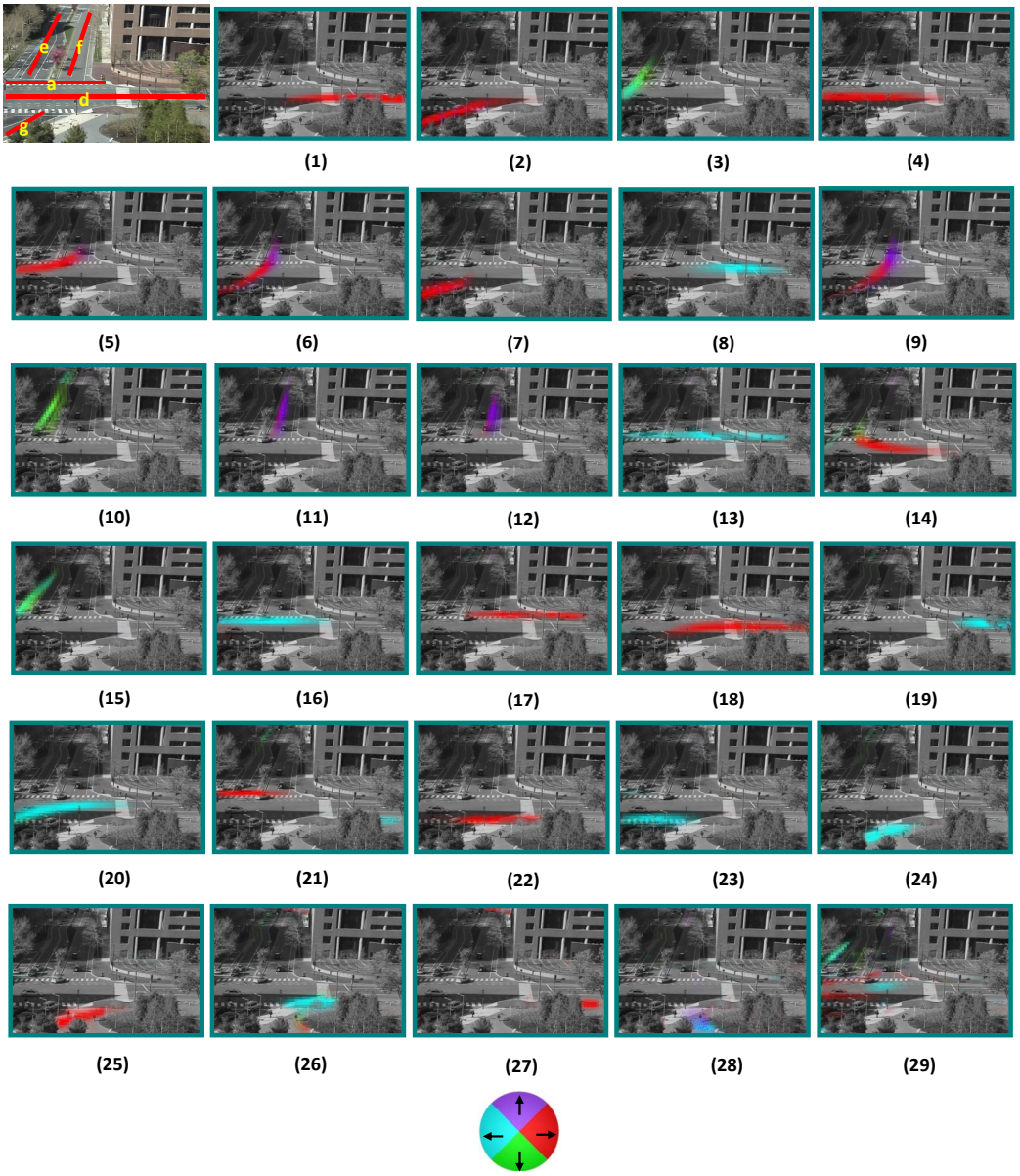


Figure 3-6: Distributions of 29 atomic activities (topics) discovered by our Dual-HDP models. Colors represent four quantized moving directions: red ($\rightarrow$), magenta ($\uparrow$), cyan ($\leftarrow$), and green ($\downarrow$). The intensity represents the density of distributions. The atomic activities are sorted according to how many moving pixels in the data set are assigned to them (from large to small). For convenience, we label roads and crosswalks as $a, b, \ldots$ in the first image. From these distributions, we can guess some activities happening in this scene, such as vehicles turning left (atomic activity 5 and 20), pedestrians crossing streets (atomic activity 13, 17, 21, 22 and 23), and vehicles crossing the intersection (atomic activity 6 and 9).

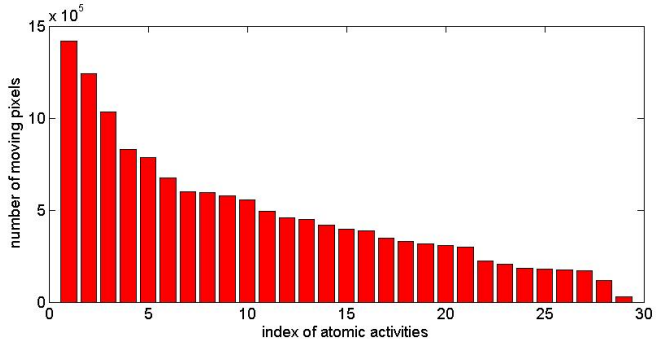


Figure 3-7: Histogram of moving pixels assigned to 29 atomic activities in Figure 3-6.

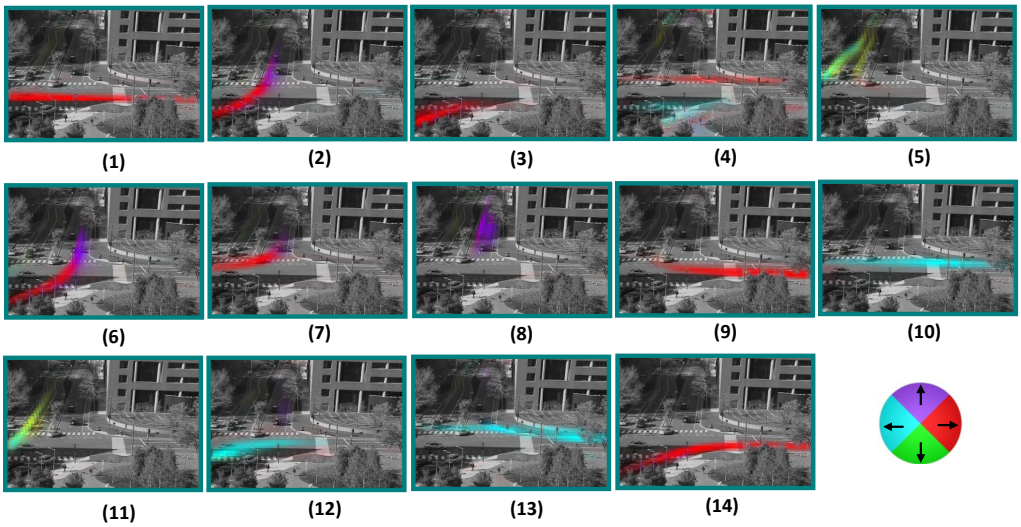
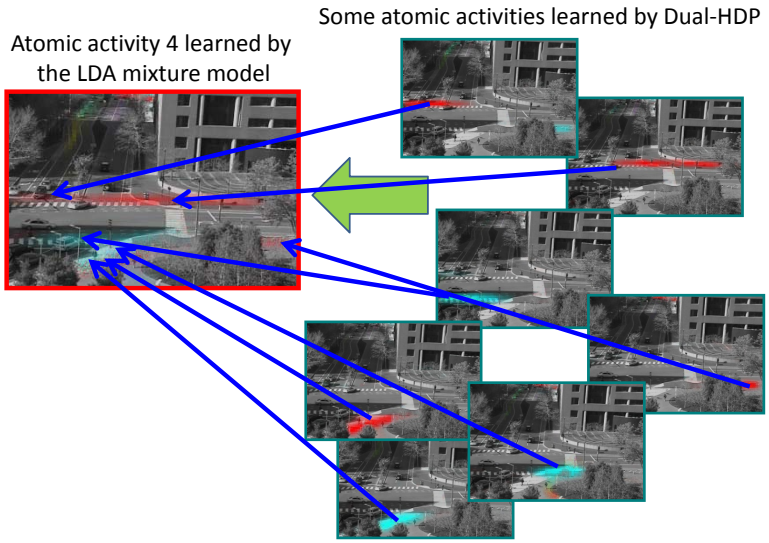


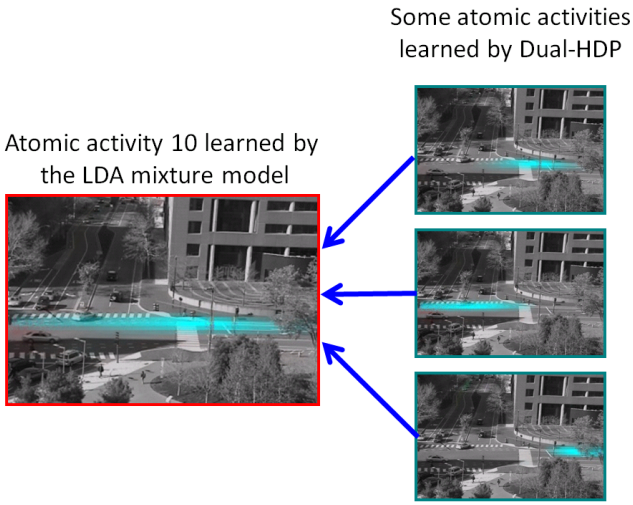
Figure 3-8: Distributions of atomic activities learned by the LDA mixture model when the number of atomic activities is fixed as 14.

activity breaks into two atomic activities 17 and 21.

When the number of atomic activities is set as 29, the LDA mixture model provides similar result as Dual-HDP. In Figure 3-8, we show the results from the LDA mixture model when choosing 14 instead of 29 as the number of atomic activities. Several atomic activities discovered by Dual-HDP merge into one atomic activity in the LDA mixture model. For example, as shown in Figure 3-9 (a), atomic activities 17, 21, 23, 24, 25, 26, and 27 related to pedestrian walking learned by Dual-HDP as shown in Figure 3-6 merge into a single atomic activity 4 learned by the LDA mixture model shown in Figure 3-8. As shown in Figure 3-9 (b), atomic activities 8, 16 and 19 in Figure 3-6 merge into atomic activitiy 10 in Figure 3-8.



(a)



(b)

Figure 3-9: Some atomic activities learned by Dual-HDP merger into one atomic activity learned by the LDA mixture model. (a) When the number of atomic activities is set as 14 in the LDA mixture model, Atomic activities 17, 21, 23, 24, 25, 26, and 27 related to pedestrian walking learned by Dual-HDP merge into one atomic activity learned by the LDA mixture model. (b) When the number of atomic activities is set as 14 in the LDA mixture model, Atomic activities 8, 16 and 19 related to pedestrian walking learned by Dual-HDP merge into one atomic activity learned by the LDA mixture model.

3.3.2 Discover Global Behaviors

Global behavior in the scene can be explained as a combination of co-occurring atomic activities, or equivalently, topics, under our framework. In our hierarchical Bayesian models, the video clips are automatically clustered into different global behaviors. The topics mixtures ($\{\alpha_c\}$ in the LDA mixture model and $\{\pi_c\}$ in Dual-HDP) as priors of clusters of video clips provide a good summary of global behaviors. Figure 3-10 plots the mixture weights π_c of five global behaviors on atomic activities learned by Dual-HDP. They are different traffic modes. Global behavior 1 explains traffic moving in a vertical direction. Vehicles from road e and g move vertically, crossing road d and crosswalk a. 3, 6, 7, 9 and 11 are major atomic activities in this traffic mode, while the weights on other atomic activities related to horizontal traffic (1, 4, 5, 8, 16 and 20), and pedestrians walking on crosswalk a and b (13, 17, 21 and 23), are very low. Global behavior 2 explains "vehicles from road g make a right turn to road a while there is not much other traffic". In this traffic mode, vertical traffic is forbidden because of the red light while there are no vehicles traveling horizontally on road d, so these vehicles from g can make a right turn. Global behavior 3 is "pedestrians walk on the crosswalks while there is not much traffic". In this traffic mode, several atomic activities (21, 13, 17) related to pedestrian walking have much higher weights than their average distribution on the whole video sequence. Atomic activities 10 and 15 also have large weights in this global behavior and they explain that vehicles on road e come to stop behind the stop line when pedestrians walk on the crosswalk. Global behavior 4 is "vehicles on road d make a left turn to road f". Atomic activities 5, 11, and 12 related to vehicles turning left have large weights. Atomic activities 1 and 4 also have large weights since horizontal traffic from left to right is allowed at this time. However atomic activities 8, 16 and 20 have very low weights, because traffic from right to left conflicts with this left turn activity. Global behavior 5 is horizontal traffic. In this traffic mode, atomic activities 13, 17 and 21 have relatively high weights, since pedestrians are allowed to walk on a. In the second row of Figure 3-10, we show an example video clip for each type of global behavior.

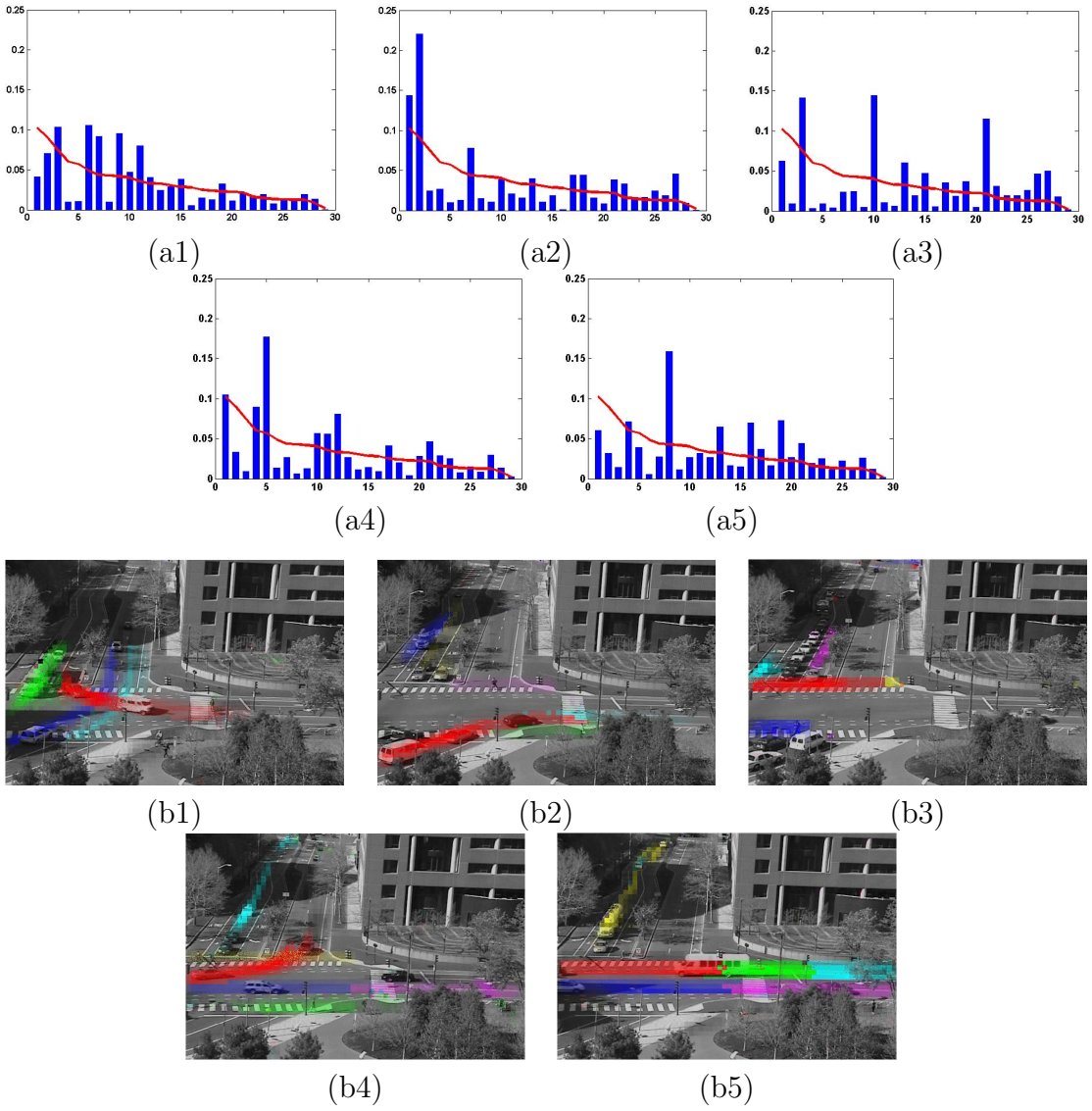


Figure 3-10: Distributions of global behaviors over atomic activities learned by Dual-HDP in a traffic scene. Video clips are clustered into five global behaviors. In (a1)-(a5), we plot the mixture weights $\{\pi_c\}$ over 29 atomic activities as prior of each global behavior represented by blue bars. For comparison, the red curve in each plot is the average mixture weights over atomic activities on the whole data set. The x-axis is the index of atomic activities. The y-axis is the mixture weight over atomic activities. In (b1)-(b5), we show a video clip as an example for each type of global behavior and mark the motion distributions of the five largest atomic activities in that vide clip. Notice that colors distinguish different atomic activities in the same video (the same color may correspond to different atomic activities in different video clips) instead of representing motion directions as in Figure 3-6.

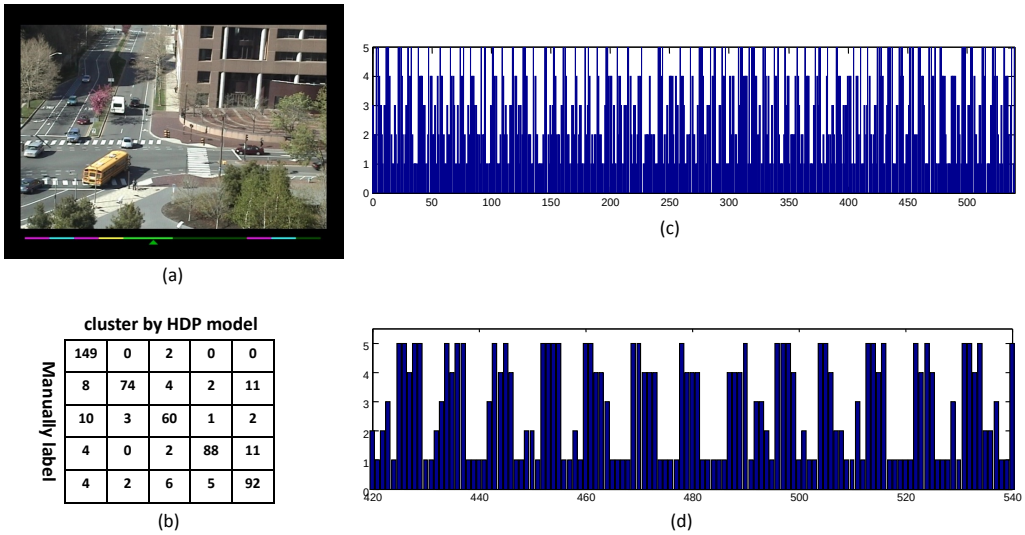


Figure 3-11: Results of segmenting a long video sequence into different global bahaviors. (a) The snapshot of our video segmentation result; (b) the confusion matrix; (c) the segmentation result of the 90 minutes long video sequence; (d) zoom in of the segmentation result of the last 20 minutes of video. In (c) and (d), the x-axis is the index of video clips in temporal order, and the y-axis is the label of the five global behaviors shown in Figure 3-10.

In each video clip, we choose the five largest atomic activities and plot their motion distributions by different colors.

3.3.3 Video Segmentation

Given a long video sequence, we can segment it based on different types of global behaviors. Our models provide a natural way to complete this task in an unsupervised manner since video clips are automatically separated into clusters (global behaviors) in our model. To evaluate the clustering performance, we create a ground truth by manually labeling the 540 video clips into five typical interactions in this scene as described in Section 3.3.2. The confusion matrix between our clustering result and the ground truth is shown in Figure 3-11 (b). The average accuracy of video segmentation is 85.74%. Figure 3-11 shows the labels of video clips in the entire one and half hours of video and in the last 20 minutes. Note the periodicity of the labels assigned. We can observe that each traffic cycle lasts around 85 seconds.

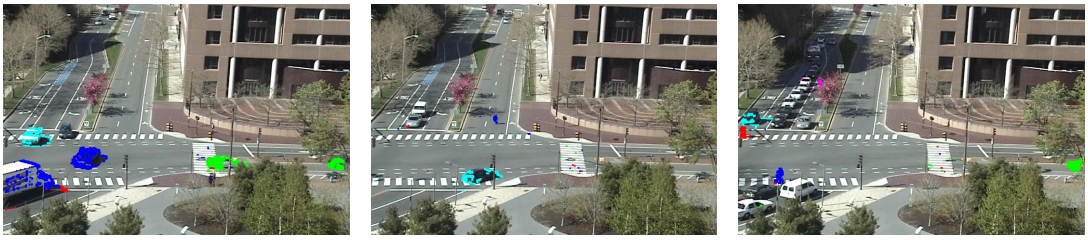
3.3.4 Activity Detection

We also want to localize atomic activities happening in the video. Since in our hierarchical Bayesian models, each moving pixel is labeled as one of the atomic activities, activity detection becomes straightforward. In Figure 3-12, we choose five ten seconds long video clips as examples of the five different global behaviors, and show the activity detection results on them. Since our motion detection method is very simple, the detected moving pixels are quite noisy. They are not smooth in space and time. However, they are well labeled into different activity categories, because the activity models are shared by all the video clips and they are well learned from a huge amount of moving pixels.

As an extension of activity detection, we can detect vehicles and pedestrians based on motions. It is observed that the vehicle motions and pedestrian motions are well separated among atomic activities. However, the user first needs to label each of the discovered atomic activities as being related to vehicles or pedestrians. Then we can classify the moving pixels into vehicles and pedestrians based on their atomic activity labels. Figure 3-13 shows some detection results. This approach cannot detect static vehicles and pedestrians. Most existing object detectors are based on the appearance of objects. However, our approach is based on motions and activity models. It is complementary to appearance based vehicle and pedestrian detectors. Further more, it does not require labeling training examples.

3.3.5 Abnormality Detection

In visual surveillance, detecting abnormal video clips and localizing abnormal activities in the video clip are of great interest. Under the Bayesian models, abnormality detection has a nice probabilistic explanation by the data likelihood of every video clip and every moving pixel rather than by comparing similarity between samples. Computing the likelihood of documents and words under the LDA mixture model has been described in Section 3.2.2 (see Eq 3.5). Computing the data likelihood under the HDP mixture model and Dual-HDP model is not straightforward. We need to



(1) (2) (3)

(4) (5)

Figure 3-12: Activity detection. Five video clips are chosen as examples of the five global behaviors shown in Figure 3-10. We show one key frame of each video clip. The motions are clustered into different activities marked by different colors. However since there are so many atomic activities, we cannot use a uniform color scheme to represent all of them. In this Figure, the same color in different video clips may indicate different activities. Clip 1 has atomic activities 1 (green), 3 (cyan) and 6 (blue) (see these atomic activities in Figure 3-6). Clip 2 has atomic activities 2 (cyan) and 13 (blue). Clip 3 has atomic activities 15 (cyan), 7 (blue) and 21 (red). Clip 4 has atomic activities 1 (red), 5 (blue), 7(green), 12 (cyan) and 15 (yellow). Clip 5 has atomic activities 8 (red), 16 (cyan), 17 (magenta) and 20 (green).

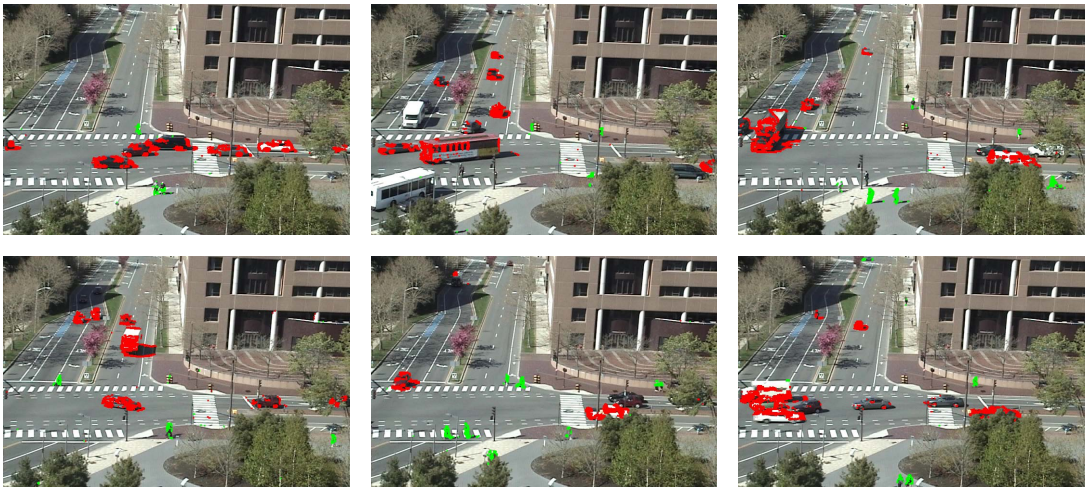


Figure 3-13: Vehicle and pedestrian detection based on motions. Vehicle motions are marked by red color and pedestrian motions are marked by green color.

compute the likelihood of document j given other documents, $p(\mathbf{x}_j|\mathbf{x}^{-j})$, where $\mathbf{x}^{-j}$ represents the whole corpus excluding document j. For example, in the HDP mixture model, since we have already drawn M samples $\{\mathbf{z}^{-j(m)}, \{\pi_c^{(m)}\}, \pi_0^{(m)}\}_{m=1}^{M}$ from $p(\mathbf{z}^{-j}, \{\pi_c\}, \pi_0|\mathbf{x})$ which is very close to $p(\mathbf{z}^{-j}, \{\pi_c\}, \pi_0|\mathbf{x}^{-j})$, we approximate $p(\mathbf{x}_j|\mathbf{x}^{-j})$ as

$$p(\mathbf{x}_j|\mathbf{x}^{-j}) = \frac{1}{M}\sum_m \sum_{c_j} \int_{\omega_j} \sum_{\mathbf{z}_j} \sum_i p(x_{ji}|z_{ji}, \mathbf{z}^{-j(m)}, \mathbf{x}^{-j}) p(\mathbf{z}_j|\omega_j) p(\omega_j|\pi_{c_j}^{(m)}) \eta_{cj} d\omega_j \tag{3.17}$$

$p(\omega_j|\pi_{c_j}^{(m)})$ is a Dirichlet distribution. If $(u_1, \ldots, u_T)$ is the Dirichlet prior on ϕ_k,

$$p(x_{ji}|z_{ji}, \mathbf{z}^{-j(m)}, \mathbf{x}^{-j}) = (u_{x_{ji}} + n_{x_{ji}})/(\sum_{t=1}^{T}(u_t + n_t))$$

is a multinomial distribution, where n_t is the number of words in $\mathbf{x}^{-j}$ with value t assigned to topic z_{ji} (see [140]). The computation of

$$\int_{\omega_j} \sum_{\mathbf{z}_j} p(x_{ji}|z_{ji}, \mathbf{z}^{-j(m)}, \mathbf{x}^{-j}) p(\mathbf{z}_j|\omega_j) p(\omega_j|\pi_{c_j}^{(m)})$$

is intractable, but can be approximated with a variational inference algorithm as in [18]. The likelihood computation in Dual-HDP model is very similar to that in the HDP mixture model. The only difference is to replace η_{c_j} with $\epsilon_{c_j}^{(m)}$ in Eq 3.17.

Figure 3-14 shows the top five detected abnormal video clips. Red color highlights the regions with abnormal motions in the video clips. There are two abnormal activities in the first video. A vehicle is making a right-turn from road *d* to road *f*. This is uncommon in this scene because of the layout of the city. Actually there is no topic explaining this kind of activity in our data (topics are summaries of typical activities). A person is simultaneously approaching road *f*, causing abnormal motions. In the successive video clip, we find that the person is actually crossing road *f* outside the crosswalk region. This video clip ranked fourth in abnormality. In the second and third videos, bicycles are crossing the road abnormally. The fifth video is another example of a pedestrian crossing the road outside the crosswalk.

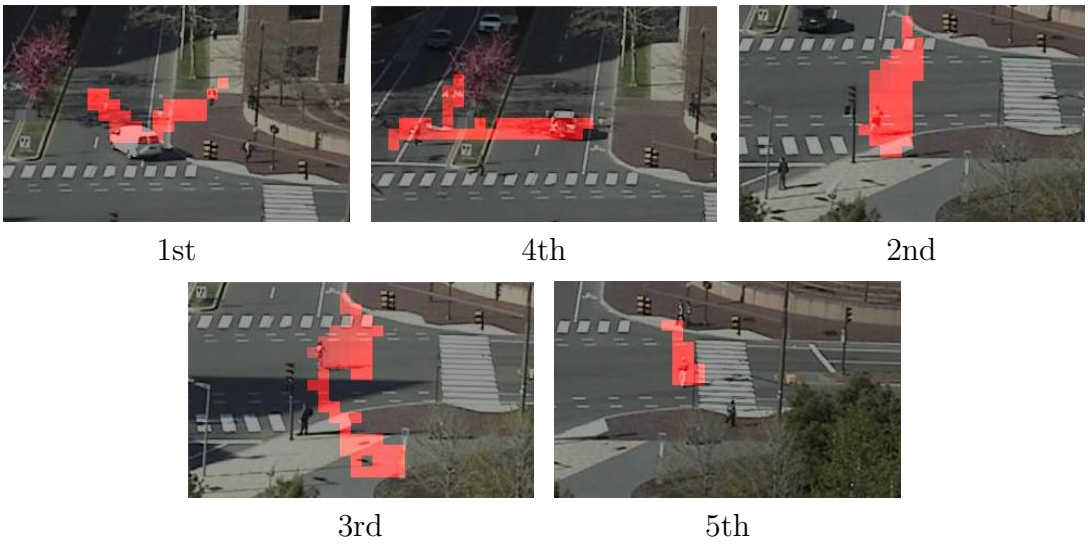


Figure 3-14: Results of abnormality detection. We show the top five video clips with the highest abnormality (lowest likelihood). In each video clip, we highlight the regions with moving pixels with high abnormality.

3.3.6 Query Interactions

Under our framework, it is convenient to use atomic activities as units to query interactions of interest. For example, suppose we want to detect the interaction of jay-walking. We simply pick two atomic activities involved in this interaction, i.e. "pedestrians walk on crosswalk a from right to left (atomic activity 13) while vehicles are approaching in vertical direction (atomic activity 6)", and specify a query distribution q ($q(6) = q(13) = 0.5$ and the weights on other atomic activities are zeros). Each video clip j has a distribution p_j over atomic activities. We match $\{p_j\}$ with the query distribution using relative entropy between q and p_j,

$$D(q||p_j) = \sum_{k=1}^{K} q(k) log \frac{q(k)}{p_j(k)} \tag{3.18}$$

Figure 3-15 (d) shows the result of querying examples of "pedestrians walk on crosswalk a from right to left while vehicles are approaching in vertical direction". All the video clips are sorted by matching similarity. There are 18 jay-walking instances in this data set, and they are all found among the top 37 examples out of the 540 clips in the whole video sequence. The top 12 retrieval results are all correct.

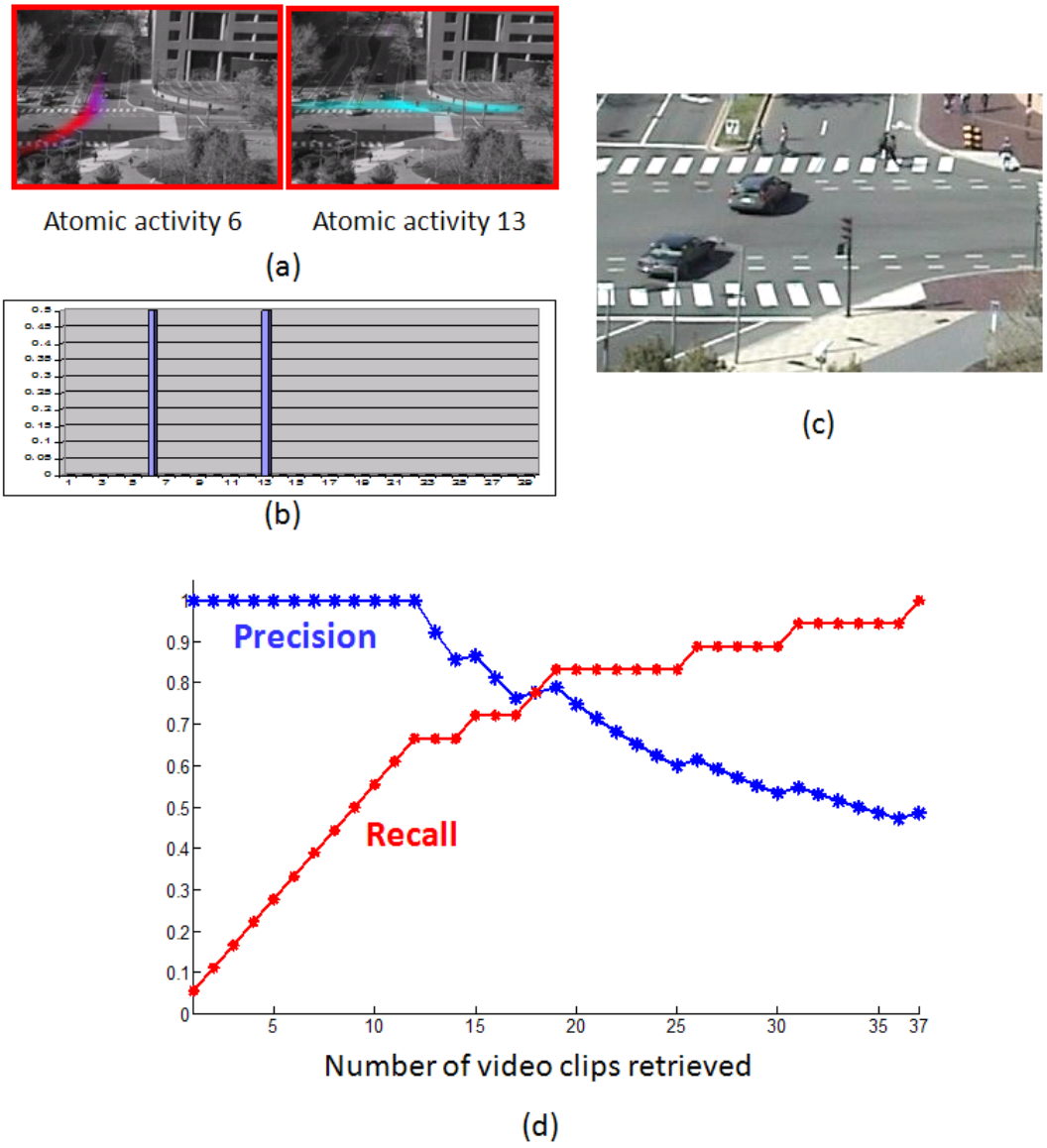


Figure 3-15: Query result of interaction jay-walking. (a) Two atomic activities 6 and 13 involved in the interaction jay-working. (b) A query distribution is drawn with large weights on atomic activities 6 and 13 and zeros weights on other topics. (c) An example of jay-walk retrieval. (d) Precision and recall with the number of video clips retrieved. There are totally 18 jay-walking instances according to labeling. They are all found among the top 37 video clips out of the total 540 clips in the data set.

3.3.7 Comparison with Other Methods

Another option to model interactions is to first use the original LDA (a) or HDP (b) as a feature reduction step. A distribution p_j over topics or a posterior Dirichlet parameter (γ_j in Eq 3.2) is associated with each document. Then one can cluster documents based on $\{p_j\}$ or $\{\gamma_j\}$ as feature vectors. [18] used this strategy for classification. K-means on $\{p_j\}$ only has 55.6% accuracy of video segmentation on this data set (KL divergence is the distance measure), while the accuracy of our Dual-HDP model is 85.74%. It is hard to define a proper distance for Dirichlet parameters. We cannot get meaningful clusters using $\{\gamma_j\}$.

We also evaluate the algorithm proposed in [164], which used global motion to describe each frame, on this data set. [164] also adopted word-document analysis and used spectral graph partitioning. However, it did not model local atomic activities and the interactions or activities were directly modeled as a distribution over global motion instead of atomic activities. Although their method worked well on simple data sets in [164], where usually there was only one kind of activity in each video clip, it failed on our complicated scene with many activities co-occurring. We did not find meaningful interactions from the discovered clusters using their approach on our data. The formation of clusters is dominated by the amount of traffic flow instead of the types of traffic. The detected abnormal examples are videos with relatively small amounts of motion and do not really include interesting activities.

3.3.8 Discussion

The space complexities of the three proposed models are all $O(KW) + O(KL) + O(KM) + O(N)$, where K is the number of topics, W is the size of the codebook, L is the number of document clusters, M is the number of documents and N is the total number of words. Using EM and VB, the time complexity of the learning and inference of the LDA mixture model is $O(ML) + O(NK) + O(LK^2)$. Running on a computer with 3GHz CPU, it takes less than one hour to process an 1.5 hours video sequence. The Gibbs sampling inference of HDP mixture model and Dual-

HDP model is much slower. The time complexity of each Gibbs sampling iteration is $O(NK) + O(ML)$. It is difficult to provide theoretical analysis on the convergence of Gibbs sampling. It takes around 12 hours to process an 1.5 hours video sequence. In recent years, variational inference was proposed for HDP [141] and it is faster than Gibbs sampling. A possible extension of this work is to explore variational inference algorithms under HDP mixture model and Dual-HDP model. Currently our algorithm is running in a batch mode. However, once the model has been learnt from a training video sequence and fixed, it can be used to do motion/video segmentation and abnormality detection on new video stream in an online mode.

Chapter 4

Trajectory Analysis in A Single Camera View

Having explained activity analysis in crowded scenes in Chapter 3, we will consider the scenario when the scenes are sparse and we can track objects in this chapter. We will presents the results of using Dual-HDP and dynamic Dual-HDP for trajectory analysis in a single camera view. Objects are first tracked in a single camera view. Trajectories of objects are clustered into different activity categories and the models of paths commonly taken by objects are learned. Dual-HDP assumes the models of activities do not change over time. Dynamic Dual-HDP online dynamically updates models of activities.

4.1 Modeling Trajectories Using Dual-HDP

We treat a trajectory as a document and the observations on the trajectory as words. The positions and moving directions of observations on a trajectory are computed as features which are quantized according to a codebook. The codebook uniformly quantizes the space of the scene into small cells and the velocity of objects into several directions. A trajectory is modeled as a bag of quantized observations without temporal order.

In the physical world, objects move along some paths. We refer to the intersections

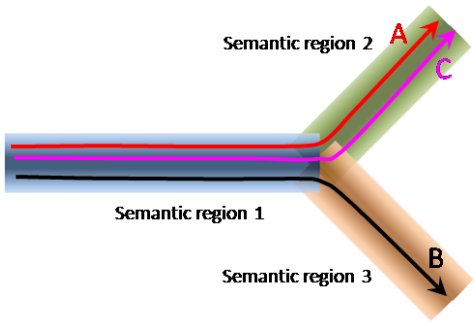


Figure 4-1: An example to explain the modeling of semantic regions and activities. Semantic regions are the overlap regions of paths. There are three semantic regions (indicated by different colors) which form two paths. Both trajectories A and C pass through regions 1 and 2, so they are clustered into the same activity. Trajectory B passes through regions 1 and 3, so it is clustered into a different activity.

of paths as semantic regions, i.e. two paths may share one semantic region as shown in Figure 4-1. When Dual-HDP is used to model trajectories, topics reveal semantic regions shared by trajectories, i.e. many trajectories pass through one semantic region with common directions of motion. A semantic region is modeled as a multinomial distribution over the space of the scene and moving directions. If two trajectories pass through the same set of semantic regions, they are on the same path and thus they are clustered into the same activity. In our Dual-HDP model, each cluster of documents (trajectories) has a prior distribution over topics (semantic regions). It is learnt in an unsupervised way. All the trajectories clustered into the same activity share the same prior distribution. Using Dirichlet Processes, Dual-HDP can learn the number of semantic regions and the number of activity categories from data.

In Figure 4-1, an example is shown to explain the modeling. There are three semantic regions (indicated by different colors) which form two paths. Both trajectories A and C pass through regions 1 and 2, so they are clustered into the same activity. Trajectory B passes through regions 1 and 3, so it is clustered into a different activity.

With the "bag-of-words" assumption, our approach does model the first order temporal information among observations since the codebook encodes the moving directions. It can distinguish some activities related to temporal features. For example, if objects visit several regions in opposite temporal order, they must pass through the same region in opposite directions. In our model, that region splits into two topics

because of the velocity difference. So these two activities can be distinguished by our model, since they have different topics.

4.2 Dynamic Dual-HDP

Under Dual-HDP, when the models of activities and semantic regions are learnt and fixed, classifying unseen trajectories into existing activity categories and detecting abnormal trajectories can be done in an online mode. However, there are still some reasons to extend the Dual-HDP model to a dynamic Dual-HDP model. First, people have interest in the dynamic change of models of activities over time. For example, exploring when a new mode of activity appears, when an old mode of activity disappears, and when a particular kind of activity becomes more dominant than other activities in the scene is of interest in surveillance applications. Abnormality detection may also change over time. An activity may be detected as an abnormality when it first appears in the scene. However, when more and more instances occur, it becomes typical. Similarly, a typical activity at an earlier time may becomes abnormal when it rarely happens later. Second, when a surveillance system monitors an area over months or even years, it is difficult to load all the huge amount of data once into memory and process it. Dynamic Dual-HDP learns the models of activities incrementally over time and does not have to keep old data.

In order to learn models of activities dynamically, one option is to divide the entire data set into subsets according to the temporal order and learn the activity models of each subset independently using Dual-HDP. This has two problems. First, the activity models learnt in different subsets are not aligned. Without manually permuting the activity models properly, people cannot observe how these models change over time. Second, since different subsets do not share information, if there is not enough data in a subset, the activity models cannot be well learnt from it. Blei et al. [16] proposed a model which allowed the topics to be dynamically updated. However, it assumed that the number of topics was fixed. Allowing the addition of new emerging activity

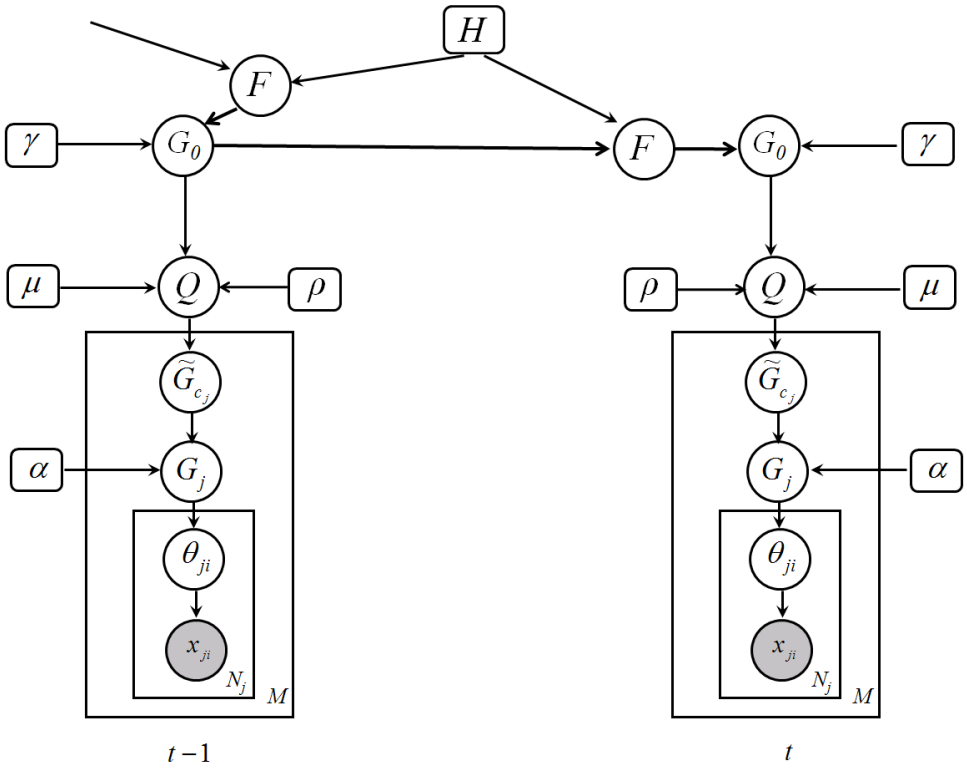


Figure 4-2: Graphical model of dynamic Dual-HDP

models over time is of considerable interest in surveillance applications.

The graphical model of the dynamic Dual-HDP is shown in Figure 4-2. The data is divided into subsets based on temporal intervals (e.g. a subset includes trajectories happening within one hour). The key difference is that G_0^{t-1}, which is the infinite mixture of topics of words learnt at time interval $t-1$, is used as prior to predict G_0^t, which is the mixture of topics learnt at the next time interval t. Assume that K^{t-1} topics have been generated from the data up to time $t-1$. Then G_0^{t-1} can be represented as

$$G_0^{t-1} = \sum_{k=1}^{K^{t-1}} \pi_{0k}^{t-1} \delta_{\phi_k^{t-1}} + \pi_{0u}^{t-1} G_{0u}^{t-1} \tag{4.1}$$

where the first K^{t-1} topics have been assigned to the data up to time $t-1$, and $\pi_{0u}^{t-1} G_{0u}^{t-1} = \sum_{k=K^{t-1}+1}^{\infty} \pi_{0k}^{t-1} \phi_k^{t-1}$ is the remaining part of the infinite mixture of topics, none of which is assigned to any data up to time $t-1$ [140]. Both $\{\pi_{0k}^{t-1}\}_{k=1}^{K^{t-1}}$ and $\{\phi_k^{t-1}\}_{k=1}^{K^{t-1}}$ can be sampled from the data up to $t-1$. We assume that they are learnt

and fixed before predicting G_0^t.

In order to use the mixture of topics learnt up to $t-1$ as the prior of G_0^t, we first normalize $\{\pi_{0k}^{t-1}\}_{k=1}^{K^{t-1}}$ to $\{\hat{\pi}_{0k}^{t-1}\}_{k=1}^{K^{t-1}}$

$$\hat{\pi}_{0k}^{t-1} = \frac{\pi_{0k}^{t-1}}{\sum_{k'=1}^{K^{t-1}} \pi_{0k'}^{t-1}}.$$

Then a base probability measure F^t is constructed for G_0^t,

$$F^t = \omega^t \sum_{k=1}^{K^{t-1}} \hat{\pi}_{0k}^{t-1} \delta_{\phi_k^t} + (1-\omega^t)H, \tag{4.2}$$

where

$$\phi_k^t \sim Dir(\xi_k^t \cdot \phi_k^{t-1} + H), \tag{4.3}$$

H is the same Dirichlet distribution as in Section 3.2.6 without changing over time, ω^t is a scalar between 0 and 1, and ξ_k^t is a positive scalar. G_0^t is sampled from a Dirichlet process,

$$G_0^t \sim DP(\gamma^t, F^t), \tag{4.4}$$

where γ^t is a positive scalar. We will give more details about ω^t, ξ_k^t, and γ^t later. From Eq 4.2 and 4.4, we observe that the random measure G_0^t at time t includes the K^{t-1} topics generated before t and new topics never seen before. The weights π_0^t over topics change over time. In Eq 4.3, when a topic ϕ_k^t at t was observed before and thus has a corresponding topic model ϕ_k^{t-1} at $t-1$, ϕ_k^t is sampled from a Dirichlet distribution including ϕ_k^{t-1} as prior knowledge. Thus dynamic Dual-HDP also models the dynamic change of models of topics $\{\phi_k^t\}$ instead of assuming that they are fixed over time as in [116] and [133]. In the following, we explain the inference by Gibbs sampling.

Suppose that at a sampling step there are K^t topics assigned to the data up to t (K^t changes during Gibbs sampling on the subset of t). Then an explicit construction

Algorithm 1 Inference under the dynamic Dual-HDP

Input trajectories (documents) collected from T time slices, $\{w_{ji}^t\}, t = 1, \ldots, T$.
Output models of activities and semantic regions and cluster labels of trajectories at different times.
Initialization $K^0 = 0$, $n_k^0 = 0$, $s^0 = 0$.
for $t = 1$ to T **do**
  repeat
    given other variables, sample the topic assignment $\{z_{ji}\}$ of the words $\{w_{ji}^t\}$ observed at time t, and sample the topic mixtures $\{\tilde{\pi}_{ck}\}$ of trajectory clusters using the Chinese restaurant franchise sampling scheme proposed in [140].
    given other variables, sample the cluster labels c_j of trajectories (documents) observed at time t, and sample the mixtures $\{\epsilon_c\}$ of trajectory clusters in 3.16 using the Chinese restaurant franchise sampling proposed in Appendix A.
    given other variables, sample topic models $\{\phi_k^t\}$ and mixtures $\{\pi_{0k}^t\}$ from the models $\{\phi_k^{t-1}\}$ and $\{\pi_{0k}^{t-1}\}$ learnt at time $t-1$ and the data observed at time t using Eq 4.19 and 4.20.
  until converge
  update n_k^t and s^t using Eq 4.17 and 4.18.
end for

for G_0^t is given as,

$$G_0^t = \sum_{k=1}^{K^{t-1}} \pi_{0k}^t \delta_{\phi_k^t} + \sum_{k=K^{t-1}+1}^{K^t} \pi_{0k}^t \delta_{\phi_k^t} + \pi_{0u}^t G_{0u}^t. \tag{4.5}$$

$\{\phi_k^t\}_{k=1}^{K^{t-1}}$ are the topics existing before t. They will be updated using the data observed at t. They are the same variables as in Eq 4.2. $\{\phi_k\}_{k=K^{t-1}+1}^{K^t}$ are the new topics assigned to the data observed at t. $\pi_{0u}^t G_{0u}^t = \sum_{k=K^t+1}^{\infty} \pi_{0k}^t \phi_k^t$ is the remaining part of the infinite mixture of topics, none of which is signed to any data up to time t. From Eq 4.2 and 4.4, $G_{0u}^t \sim DP(\gamma^t(1-\omega^t), H)$. $\boldsymbol{\pi}_0^t = (\pi_{01}^t, \ldots, \pi_{0K^t}^t, \pi_{0u}^t)$ and $\{\phi_k^t\}_{k=1}^{K^t}$ are the variables to be sampled. Given $\boldsymbol{\pi}_0^t$ and $\{\phi_k^t\}$, the sampling of other variables is the same as Dual-HDP. We focus on sampling $\boldsymbol{\pi}_0^t$ and $\{\phi_k^t\}$ given other variables. Suppose the topic assignments to words at t are given. In the Chinese Restaurant Franchise sampling used by HDP and Dual-HDP, let n_{kw} be the number of words with value w assigned to topic k, n_k be the total number of words assigned to topic k, s_j be the number of big tables serving dish (topic) k, and s be the total

number of big tables [1]. n_{kw}, n_k, s_k, s are all statistics from the data subset of time t. Since G_0^{t-1} provides prior of G_0^t as shown in Eq 4.2, 4.3 and 4.4,

$$p(\boldsymbol{\pi}_0^t|\{\hat{\pi}_{0k}^{t-1}\}_{k=1}^{K^{t-1}}) = Dir(\gamma^t\omega^t\hat{\pi}_{01}^{t-1},\ldots,\gamma^t\omega^t\hat{\pi}_{0K^{t-1}}^{t-1},0,\ldots,0,\gamma^t(1-\omega^t)). \tag{4.6}$$

When $1 \leq k \leq K^{t-1}$,

$$p(\phi_k^t|\phi_k^{t-1}) = Dir(\xi_k^t \cdot \phi_k^{t-1} + H), \tag{4.7}$$

and when $K^{t-1} < k \leq K^t$,

$$p(\phi_k^t) = Dir(H). \tag{4.8}$$

The data likelihoods are

$$p(n_{k1},\ldots,n_{kW}|n_k,\phi_k^t) = Multinomial(n_k,\phi_k^t), \tag{4.9}$$

where W is the size of the codebook, and

$$p(s_1,\ldots,s_{K^t}|s,\boldsymbol{\pi}_0^t) = Multinomial(s,\boldsymbol{\pi}_0^t). \tag{4.10}$$

So $\boldsymbol{\pi}_0^t$ and ϕ_k^t can be sampled from posteriors,

$$\begin{aligned}&\boldsymbol{\pi}_0^t|\{s_k\}_{k=1}^{K^t},\{\hat{\pi}_{0k}^{t-1}\}_{k=1}^{K^{t-1}}\\ \sim&Dir(s_1+\gamma^t\omega^t\hat{\pi}_{01}^{t-1},\ldots,s_{K^{t-1}}+\gamma^t\omega^t\hat{\pi}_{0K^{t-1}}^{t-1},s_{K^{t-1}+1},\ldots,s_{K^t},\gamma^t(1-\omega^t)),\end{aligned} \tag{4.11}$$

when $1 \leq k \leq K^{t-1}$,

$$\phi_k^t|\{n_{kw}\}_{w=1}^W,\phi_k^{t-1} \sim Dir(n_{k1}+\xi_k^t\cdot\phi_{k1}^{t-1}+u_1,\ldots,n_{kW}+\xi_k^t\cdot\phi_{kW}^{t-1}+u_W), \tag{4.12}$$

where $H = (u_1,\ldots,u_W)$. When $K^{t-1} < k \leq K^t$,

$$\phi_k^t \sim Dir(n_{k1}+u_1,\ldots,n_{kW}+u_W). \tag{4.13}$$

[1]The meanings of big tables and dishes in Chinese Restaurant Franchise are defined in [140] and Appendix A.

Properly choosing ω^t, γ^t and ξ_k^t, we can control how much the old data up to $t-1$ influences the inference of models of the current time t. In this work, we choose

$$\omega^t = \frac{r \cdot s^{t-1}}{r \cdot s^{t-1} + \gamma}, \tag{4.14}$$

$$\gamma^t = r \cdot s^{t-1}, \tag{4.15}$$

$$\xi_k^t = r \cdot n_k^{t-1}. \tag{4.16}$$

r is a scalar between 0 and 1 controlling how fast the influence of old data decrease. n_k^t and s^t are the accumulated effective numbers of words assigned to topic k and big tables. They are updated over time,

$$n_k^t = r \cdot n_k^{t-1} + n_k, \tag{4.17}$$

$$s^t = r \cdot s^{t-1} + s. \tag{4.18}$$

Remind that n_k and s are the statistics obtained from the subset of t. At initialization $n_k^0 = 0$ and $s^0 = 0$. Then Eq 4.11 and 4.12 become

$$\begin{aligned} &\boldsymbol{\pi}_0^t | \{s_k\}_{k=1}^{K^t}, \{\hat{\pi}_{0k}^{t-1}\}_{k=1}^{K^{t-1}} \\ \sim &Dir(s_1 + r s^{t-1} \hat{\pi}_{01}^{t-1}, \ldots, s_{K^{t-1}} + r s^{t-1} \hat{\pi}_{0K^{t-1}}^{t-1}, s_{K^{t-1}+1}, \ldots, s_{K^t}, \gamma), \end{aligned} \tag{4.19}$$

$$\phi_k^t | \{n_{kw}\}_{w=1}^W, \phi_k^{t-1} \sim Dir(n_{k1} + r n_k^{t-1} \phi_{k1}^{t-1} + u_1, \ldots, n_{kW} + r n_k^{t-1} \phi_{kW}^{t-1} + u_W). \tag{4.20}$$

When the data becomes older, its influence on the current models is weaker. The decreasing rate is r. The inference under dynamic Dual-HDP is summarized in Algorithm 1.

In our problem, dynamic Dual-HDP is applied to online learning of activity models and online abnormality detection, where we assume that data in the future is unknown. Thus in Eq 4.11 and 4.12, π_0^t and ϕ_k^t are sampled from the posteriors given $\hat{\pi}_0^{t-1}$ and ϕ_k^{t-1} without knowing $\hat{\pi}_0^{t+1}$ and ϕ_k^{t+1}. If we assume that data both in the past and in the future is known, the posteriors are more complicated than Eq 4.11

and 4.12, and the Gibbs sampling inference may require keeping all data collected from the whole period in the memory. In the current sampling algorithm, we only need to keep the data observed at the current time slice for inference. All the old data can be removed, as its information has been included in the activity models $\{\pi_{0k}^{t-1}\}$ and ϕ_k^{t-1} and sufficient statistics n_k^{t-1} and s^{t-1}.

4.3 Experimental Results

4.3.1 Trajectory Analysis without Dynamic Modeling

Our nonparametric hierarchical Bayesian models are evaluated on radar tracks collected from a maritime port, visual tracks collected from a parking lot, and simulated data. The results of Dual-HDP without dynamic modeling will be presented first. The results of Dynamic Dual-HDP will be reported in 4.3.2.

Results on Radar Tracks

In this section, experiments are done on a relatively small data set which has 577 radar tracks collected from a maritime port data set. They were acquired by multiple collaborating radars along the shore and recorded the locations of ships on the sea. Many existing approaches were evaluated on data sets with similar sizes as this one. According to the feedback of an expert who is familiar with the sea port, the semantic regions and clusters learned by our approaches make intuitive sense.

23 semantic regions are discovered by our model. In Figure 4-3, we display the distributions of semantic regions (sorted by the number of observations assigned to semantic regions) over space and moving directions. As shown in Figure 4-3, the 1st, 4th, 6th, 8th and 15th semantic regions are five side by side shipping fairways, where ships move in two opposite directions. For comparison, we segment the five fairways using a threshold on the density, and overlay them in Figure 4-3 (c) in different colors, green (1st), red (4th), black (6th), yellow (8th), and blue (15th). Since they are so close in space, they cannot be separated using spatial distance based trajectory

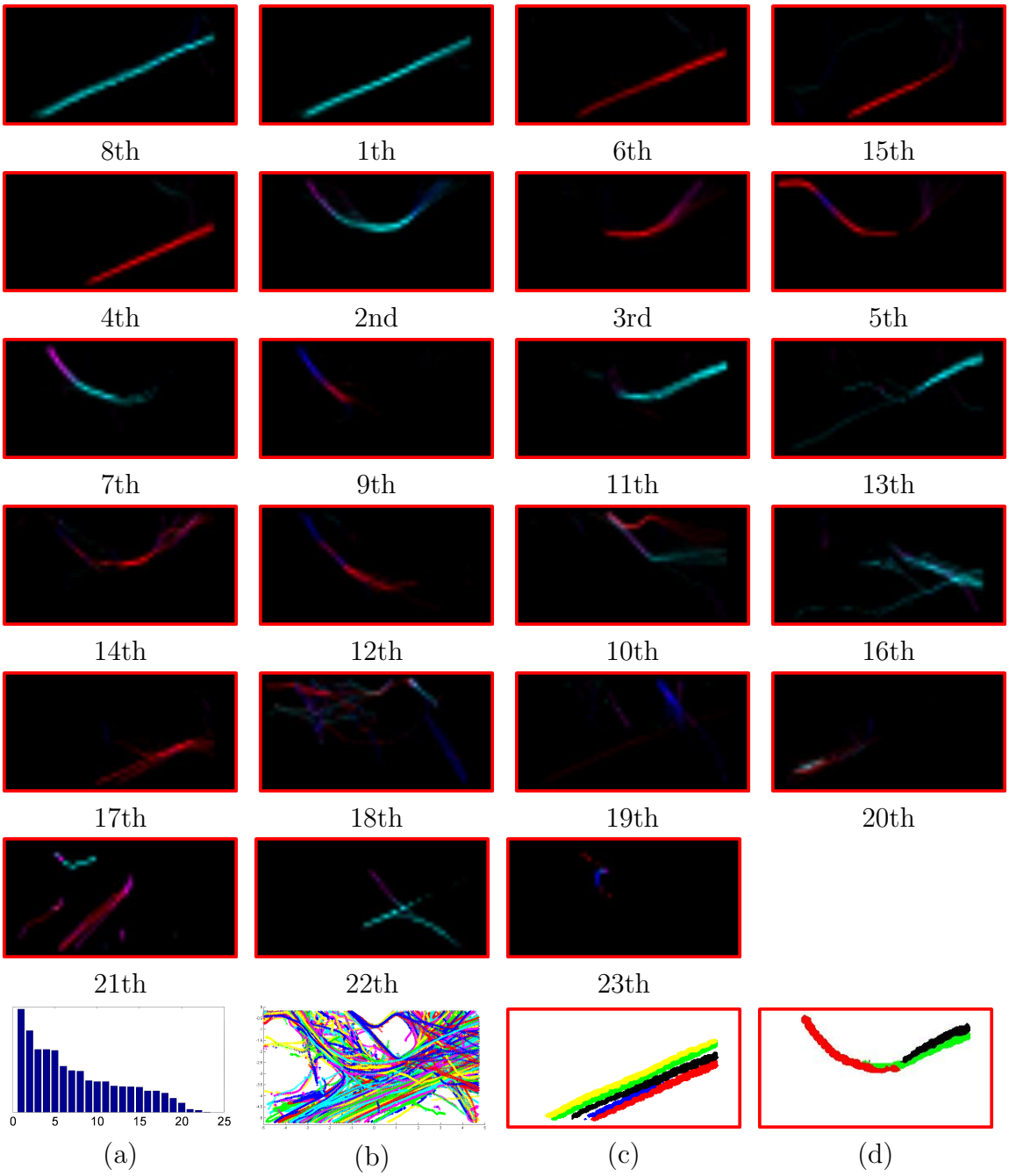
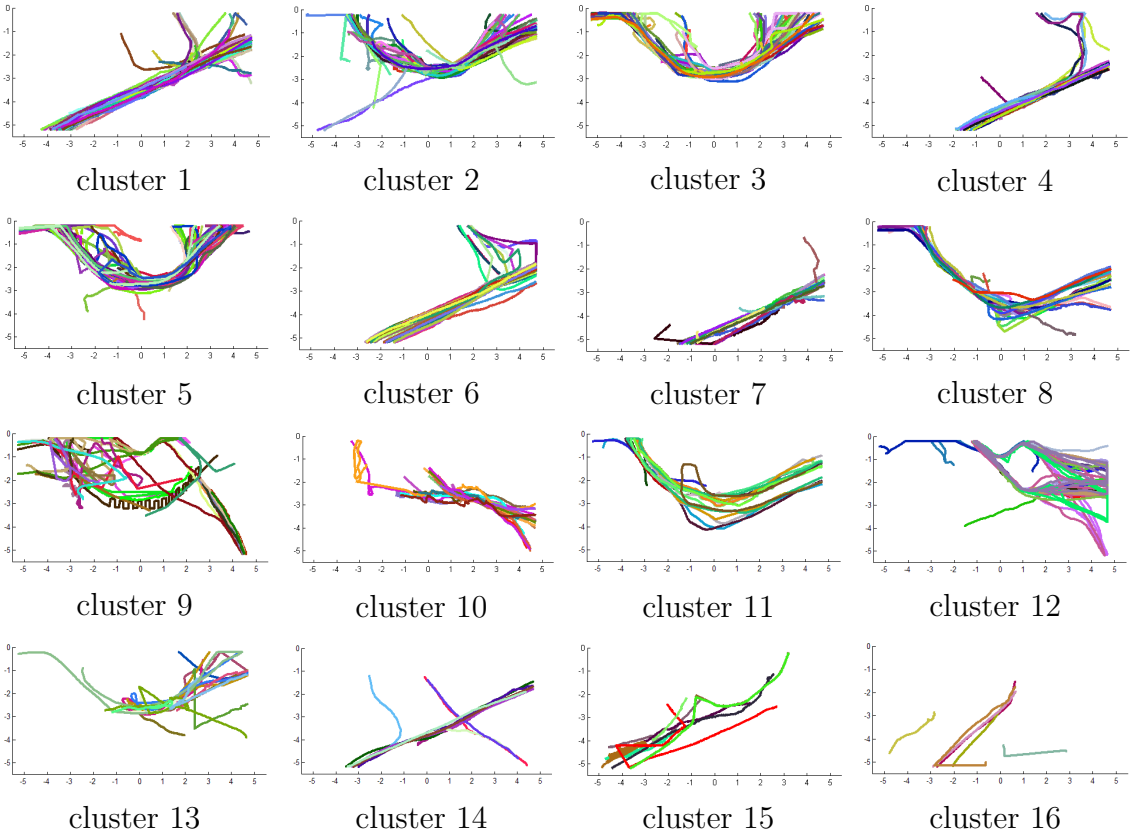


Figure 4-3: Semantic regions at a maritime port learnt from the radar tracks. Distributions of semantic regions over space and moving directions are shown (for easier comparison, they are not shown in order). Colors represent different moving directions: $\rightarrow$ (red), $\leftarrow$ (cyan), $\uparrow$ (magenta), and $\downarrow$ (blue). (a) Histogram of observations assigned to different semantic regions. (b) All of the radar tracks. (c) Compare the 1st, 4th, 6th, 8th, and 15th semantic regions. They are five by five shipping lanes. (d) Compare the 7th, 11th, and 13th semantic regions. Ships first move along the 7th semantic region and then diverge along the 11th and 13th semantic regions.

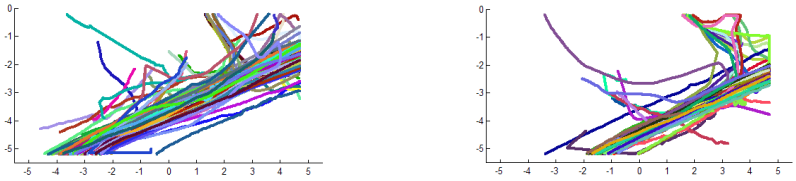
clustering approaches. In Figure 4-3 (d), we compare the 7th, 11th, and 13th semantic regions also by overlaying the segmented regions in red, green, and black colors. This explains the fact that ships first move along the 7th semantic region and then diverge along the 11th and 13th semantic regions.

Our approach groups trajectories into 16 clusters. In Figure 4-4, we plot the eight largest clusters and some smaller clusters. Clusters 1, 4, 6 and 7 are close in space but occupy different regions. Clusters 3 and 5 occupy the same region, but ships in the two clusters moves in opposite directions. Clusters 2 and 5 partially overlap in space. As shown in Figure 4-3(d), ships first move along the same way and then diverge in different directions. Clusters 2 and 5 share the same semantic region. Only modeling semantic regions using HDP cannot separate these two clusters. For comparison, in the last two sub-figures of Figure 4-4 we also show two clusters of the result using Euclidean distance and spectral clustering [41] and setting the number of clusters as 16. In this approach a similarity matrix is computed by comparing the distance between each pair of trajectories. Then spectral clustering is used to compute an embedded space. Trajectories are projected to the embedded space and clustered by k-means. Some fine structures of shipping fairways cannot be separated using a spatial distance based clustering method. One of the advantages of our approach is that it learns the number of clusters from data. When spatial distance based clustering methods are evaluated on this data set, choosing an improper cluster number, say 8 or 25, causes the clustering performance to significantly deteriorate.

In Figure 4-5, we display the top 20 abnormal trajectories based on their normalized log-likelihoods $log(p(\mathbf{w}^j|\mathbf{w}^{-j}))/N_j$. There are two possible reasons for the abnormality. (1) The trajectory does not fit any major semantic regions. Many examples can be found in Figure 4-5. (2) The trajectory fits more than one semantic region, but the combination of the semantic regions is uncommon. The red trajectory in Figure 4-5 (a), and the red and green trajectories in Figure 4-5 (b) are such examples.

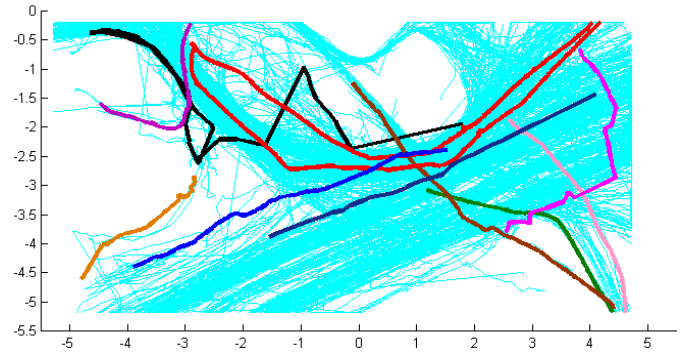


(a)

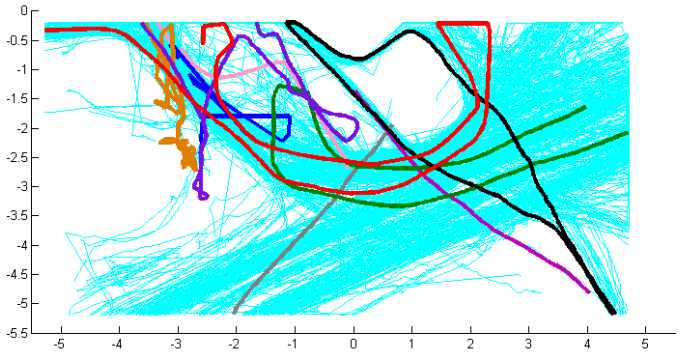


(b)

Figure 4-4: Clusters of radar tracks from a maritime port. Random colors are used to distinguish individual trajectories. For comparison the last two sub-figures show some trajectory clusters of the result using Euclidean distance and spectral clustering [41]. Some clusters in (a) merge into one cluster in (b).



(a) Top $1 - 10$



(b) Top $11 - 20$

Figure 4-5: Top 20 abnormal radar tracks are plotted in different colors. Other trajectories are plotted in cyan color. These abnormal trajectories do not fit any major semantic region or fit more than semantic regions whose combinations are uncommon.

Results on tracks from a parking lot

There are $N = 40,453$ trajectories in the parking lot data set collected over one week. They are plotted in Figure 4-6. Because of the large number of samples, similarity based clustering methods require both large amounts of space (6GB) to store the $40,453 \times 40,453$ similarity matrix and high computational cost to compute the similarities of around $800,000,000$ pairs of trajectories. If spectral clustering is used, it is quite challenging to compute the eigenvectors of such a huge matrix. It is difficult for many existing approaches to work on this large data set. The space complexity of our nonparametric Bayesian approach is $O(N)$ instead of $O(N^2)$. The time complexity of each Gibbs sampling iteration is $O(N)$. It is difficult to provide theoretical analysis on the convergence of Gibbs sampling. However, we can gather

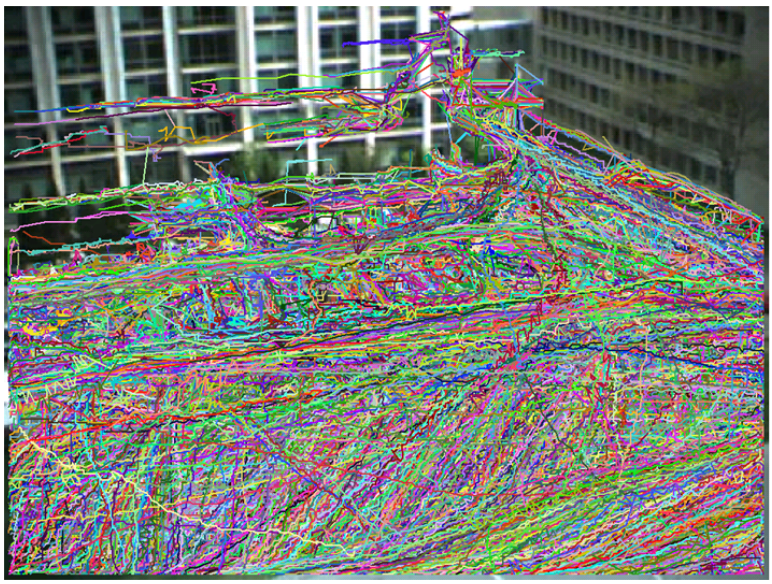


Figure 4-6: Trajectories collected from a parking lot scene within one week. Random colors are used to distinguish individual trajectories.

empirical observations by plotting the likelihoods of data sets over Gibbs sampling iterations. On the smaller radar data set, the likelihood curve converges after $1,000$ iterations. This takes around 1.5 minutes running on a computer with 3GHz CPU. On the parking lot data set, which is 70 times large than the radar data set in the number of trajectories, the likelihood curve converges after $6,000$ iterations. It takes around 6 hours. In our experiments, the time complexity of our approach is much smaller that $O(N^2)$

30 semantic regions and 22 clusters of trajectories are learnt from this data set. Some of them are shown in Figures 4-7 and 4-8. The first and third semantic regions explain vehicles entering and exiting the parking lot. Most other semantic regions are related to pedestrian activities. Because of opposite moving directions, some regions split into two semantic regions, such as semantic regions 2 and 7, 9 and 12, 5 and 14. Similarly objects on trajectories (see Figure 4-8) in clusters 2 and 3, 5 and 11 are moving in opposite directions. Many outlier trajectories are in small clusters, such as clusters 20, 21 and 22. The top 100 abnormal trajectories are shown in Figure 4-9. Most of these trajectories detected as abnormal are pedestrians walking on the grass field and and pedestrians crossing the parking lot over empty parking spaces.

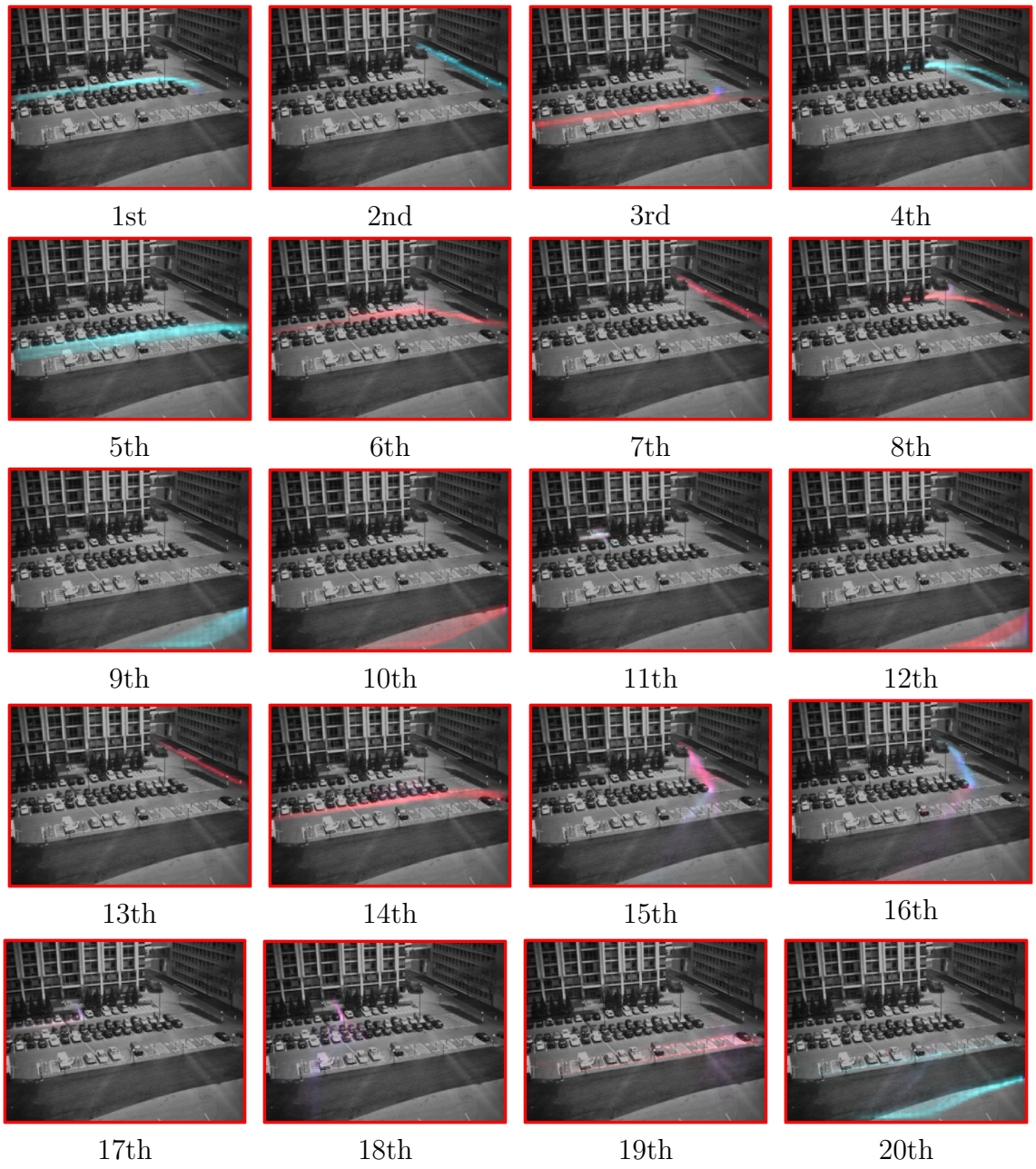


Figure 4-7: Some semantic regions learnt from a parking lot scene. The meaning of colors is the same as Figure 4-3.

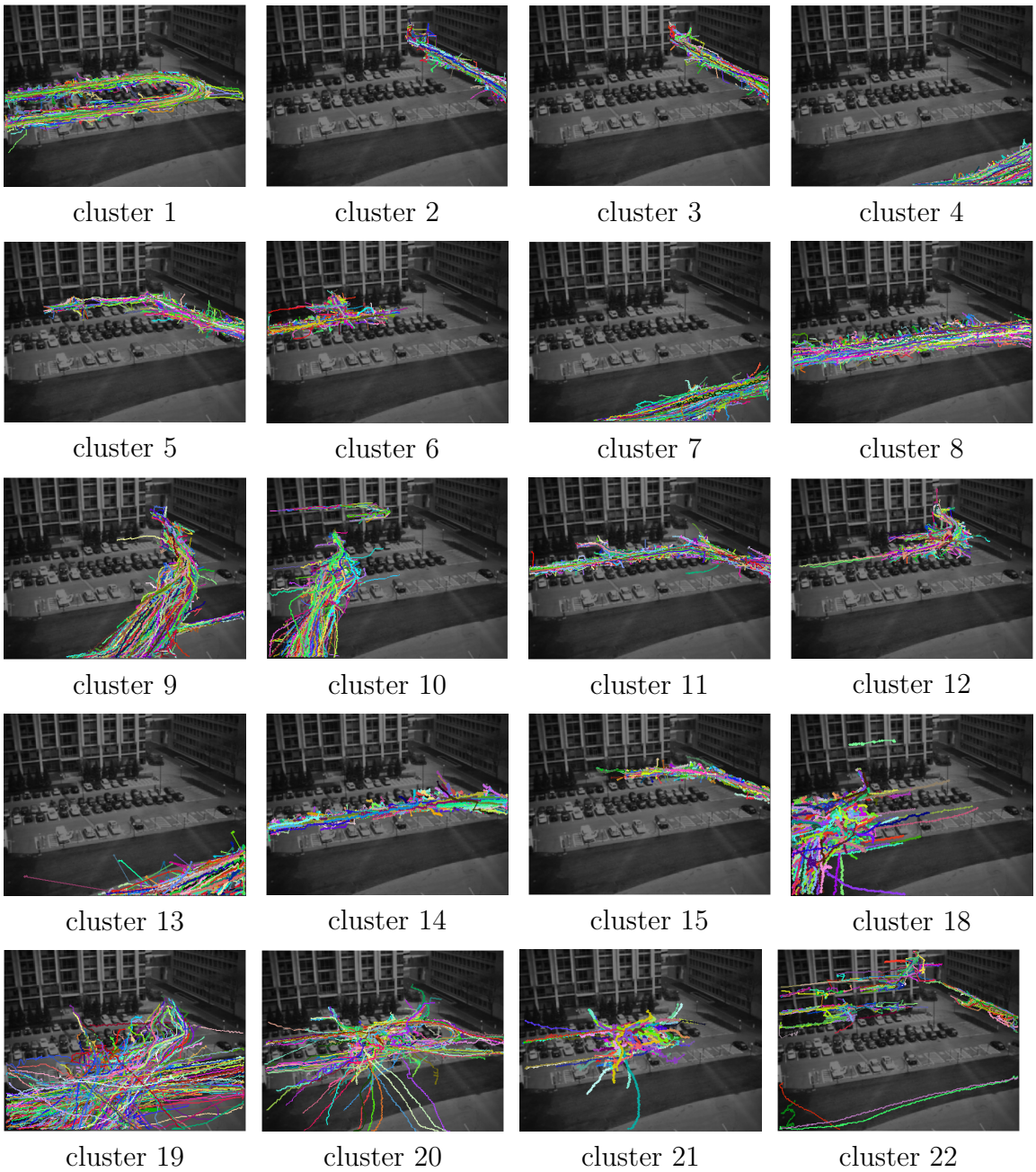


Figure 4-8: Some clusters of trajectories from a parking lot scene. In cluster 1, objects enter the parking lot and make U-turn. Cluster 2 and 3 occupy the same region but their trajectories move in opposite directions. In cluster 6, objects come out from a build and leave the parking lot. In cluster 9 and 10, objects cross the grass field.

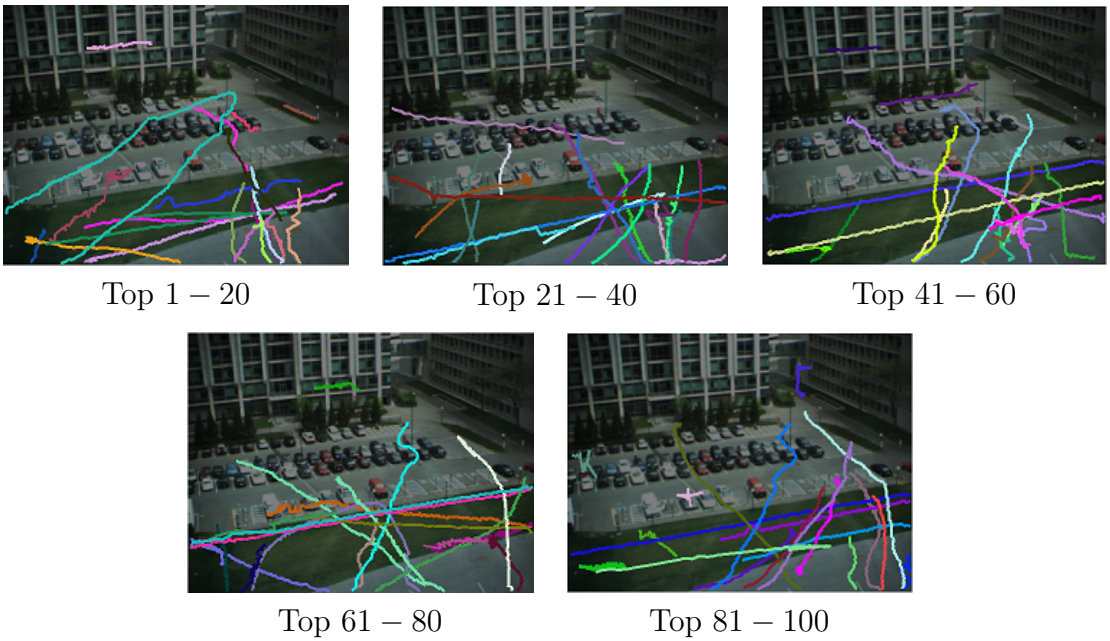
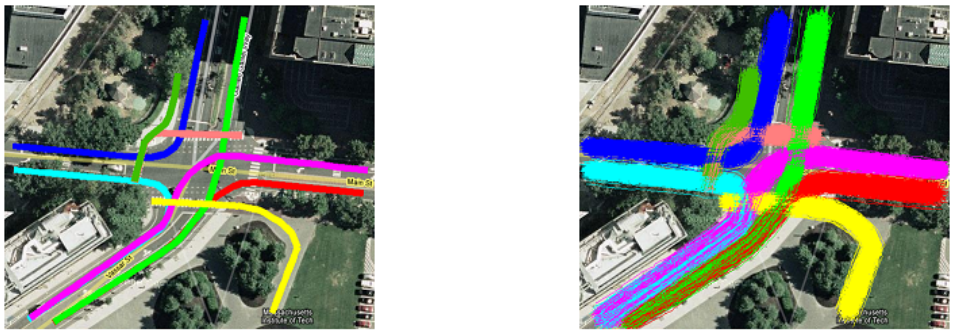


Figure 4-9: Top 100 abnormal trajectories in the parking lot scene. Many of them are pedestrians walking on the grass field.

Evaluation on Simulated Data

In this section, we simulate trajectories to evaluate how robust our model is to tracking errors. As shown in Figure 4-10 (a), eight paths are manually drawn on a scene. Some paths share the same semantic regions. A trajectory is randomly assigned to one of the eight predefined activities. A trajectory samples the location of its starting point from a Gaussian distribution centered at the starting point of its path with variance $\sigma_1 = 5$. It samples the remaining points sequentially following the direction specified by the path, with additive Gaussian noise of variance $\sigma_2 = 2$. The simulated trajectories are shown in Figure 4-10 (b). In reality, some trajectories are broken because of occlusions and scene clutter during tracking. In our simulation, we decide whether a trajectory is broken in a random way with probability r ($0 \leq r \leq 1$). If a trajectory is broken, the breaking point is uniformly sampled along the trajectory. A larger r simulates the case when there are more tracking errors. There are other types of tracking errors, such as wrong associations, not simulated in this experiment. However, breaking is one of the most common tracking errors, since some other tracking errors can be transferred to breaking errors by simply stopping tracking when the tracker is confused or there is



(a) (b)

Figure 4-10: Simulate trajectories of different activities. (a) The central lines of eight paths manually drawn in the scene. They are distinguished by different colors. (b) Trajectories simulated from the eight paths. They are also displayed in the eight colors.

not enough evidence to support the hypothesis. After the trajectories are clustered by our algorithm, we manually specify each of the eight clusters as an activity category, so each trajectory is assigned an activity label by our algorithm. By comparing with the ground truth, the accuracy of activity classification is computed. Figure 4-11 plots the activity classification accuracies with different r. It is observed that the performance does not significant drop when tracking errors increase. This shows that our algorithm is robust to tracking errors to some extent. We also compare our algorithm with two distance-based methods which use Euclidean distance [41] and modified Hausdorff distance [153] to compute distance between two trajectories. The modified Hausdorff distance compares both spatial distance and velocity difference of observations on the trajectories. The method using Euclidean distance requires that trajectories are temporally aligned. The performance of these two distance-based methods, especially when using Euclidean distance, drops significantly when the trajectory data set has tracking errors.

4.3.2 Trajectory Analysis with Dynamic Modeling

In this section, the results of the dynamic Dual-HDP model will be presented.

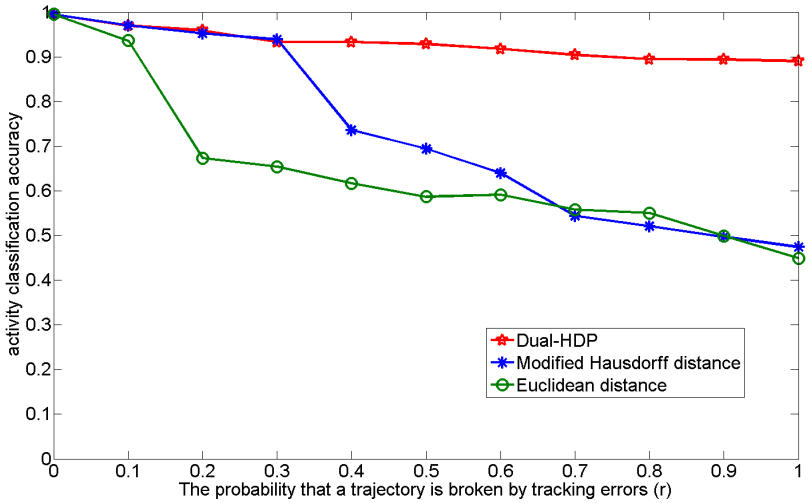


Figure 4-11: Activity classification accuracies of Dual-HDP and two distance-based methods (Euclidean distance [41] and modified Hausdorff distance [153]) when the simulated trajectories are broken with different probabilities from 0 to 1. The modified Hausdorff distance compares both spatial distance and velocity difference of observations on the trajectories.

Results on Radar Tracks

In this section we conduct experiments on a much larger data set than that used in Section 4.3.1. It includes 8,478 radar tracks collected from 304 hours. The trajectories are divided into $T = 304$ slices by hours. In Figure 4-12, 4-13, 4-14, 4-15, and 4-16, we show the semantic regions learnt at different time slices. The model of a semantic region learned at a previous time slice is used as a prior to update the model of the semantic region at the next time slice. Thus we can observe how the models of semantic regions change over time. The first subfigure shows when this semantic region first appears as a new mode under Dual-HDP. Figure 4-12 shows the dynamic change of semantic region 1. Semantic region 1 first appears at the 35th hour and its shape changes over time. As shown in Figure 4-13, the topic related to semantic region 2 first appears at the 47th hour. However, it appears noisy in the first few time slices. Its shape forms after the 112nd hour. This mode gradually disappears after 244 hours. Semantic region 3 and 4 are first coupled in the same topic at the early stage (see the first two subfigures in Figure 4-15). They are well separated when more data is observed later on.

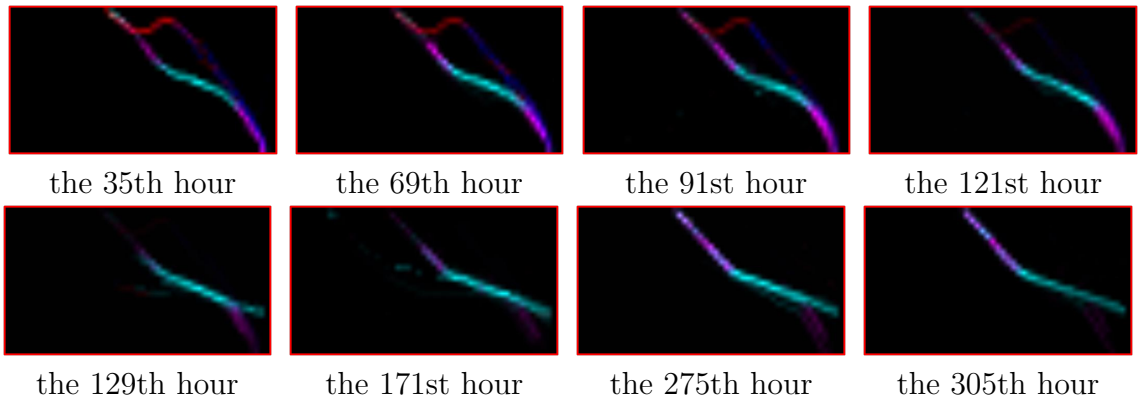


Figure 4-12: Dynamic change of semantic region 1 over time learnt from the radar tracks. In the first subfigure, we show when the semantic region first appears, i.e. semantic region 1 first appears as a new mode learnt by the Dual-HDP model at the 35th hour. Figure 4-13, 4-14, 4-15, and 4-16 follow the same convention.

Figure 4-17 shows the abnormal radar tracks detected at different time. Since the activity models and semantic regions change over time, the detected abnormal trajectories are also different depending on the time context. Trajectories detected as abnormal at some time slices may become normal when they appear at other time slices. For example, as shown in Figure 4-17, some abnormal trajectories detected at the 7th hour and the 9th hour actually pass through semantic regions 2, 3, 4, and 5. However, these modes are learnt later. In the first few hours, only a few trajectories passing through these semantic regions are observed. So they are detected as abnormal. When more trajectories of the same activities are observed and the activity models are well learnt later, they will not be detected as abnormal any more.

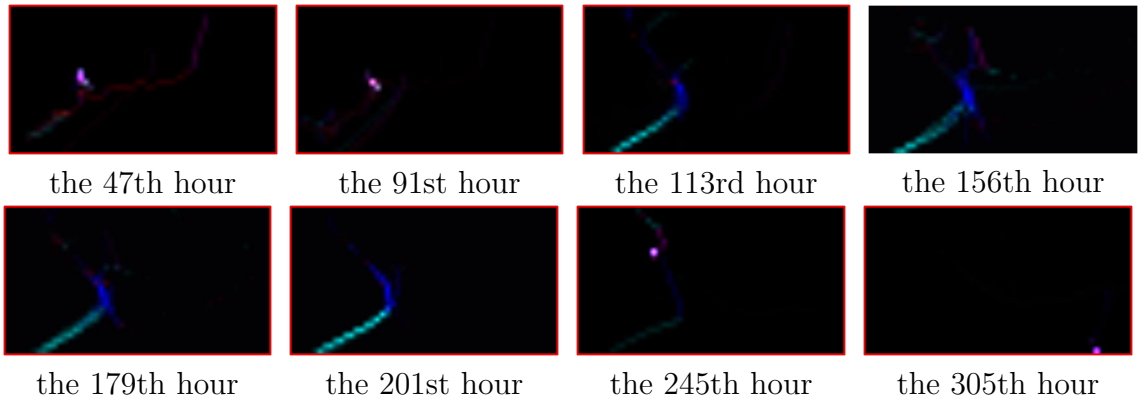


Figure 4-13: Dynamic change of semantic region 2 over time learnt from the radar tracks. This semantic region first appears at the 47th hour. However, it appears noisy in the first few time slices. Its shape forms after the 112nd hour. This mode gradually disappears after 244 hours.

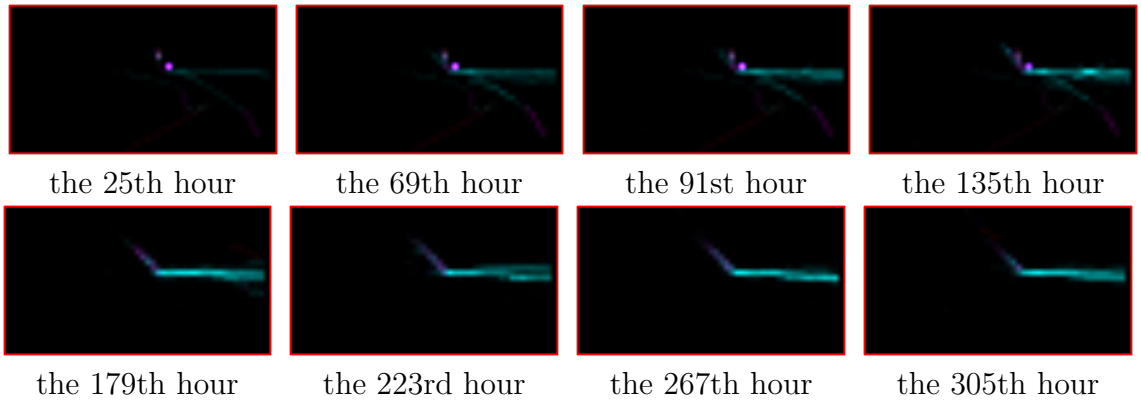


Figure 4-14: Dynamic change of semantic region 3 over time learnt from the radar tracks.

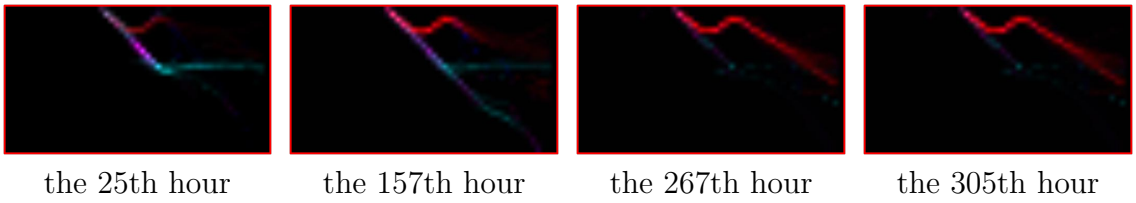
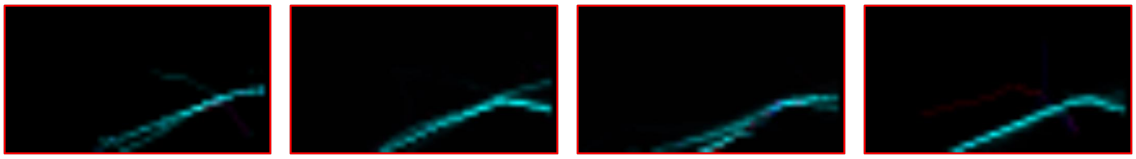
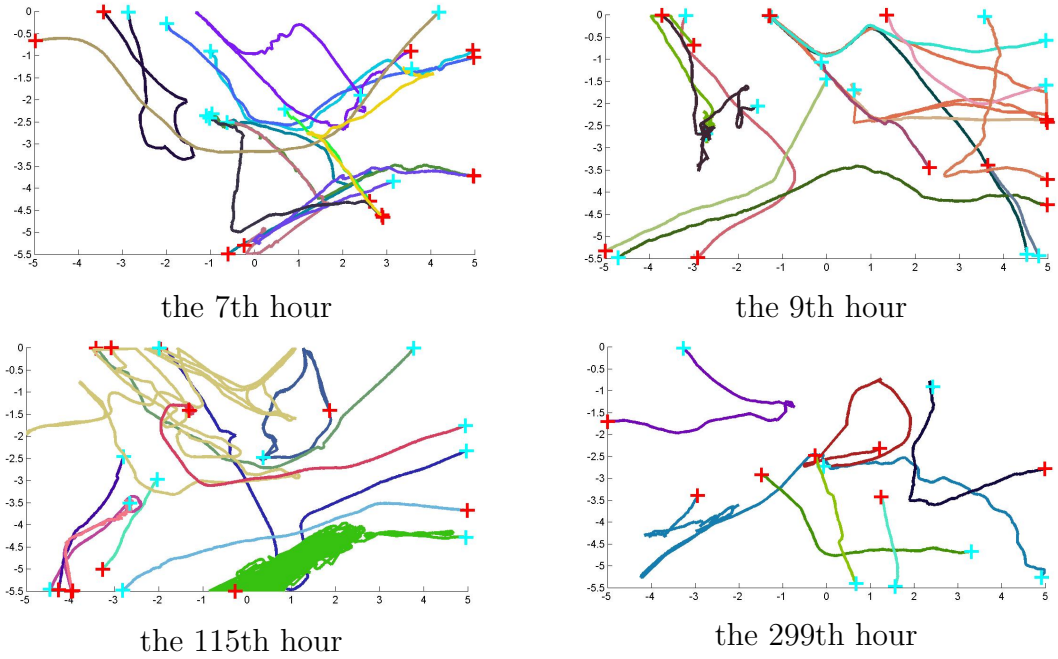


Figure 4-15: Dynamic change of semantic region 4 over time learnt from the radar tracks. Semantic region 3 and 4 are first coupled in the same topic at the early stage (see the first two subfigures). They are well separated when more data is observed later on.



the 15th hour | the 93rd hour | the 221st hour | the 305th hour

Figure 4-16: Dynamic change of semantic region 5 over time learnt from the radar tracks.



the 7th hour | the 9th hour | the 115th hour | the 299th hour

Figure 4-17: Abnormal radar tracks detected at different time slices. The same threshold of data likelihood is used for all the time slices. Some abnormal trajectories detected at the 7th hour and the 9th hour actually pass through semantic regions 2, 3, 4, and 5. However, these modes are learnt later. In the first few hours, only a few trajectories passing through these semantic regions are observed. So they are detected as abnormal. When more trajectories of the same activities are observed and the activity models are well learnt later, they will not be detected as abnormal any more.

Results on Tracks from a Park Lot

The $40,453$ trajectories of the parking lot data set are collected from one week. We divide them into time slices by hours. Figure 4-19, 4-18, and 4-20 show the dynamic change of semantic regions over time. We can observe some cyclic change of the distributions of semantic regions. There are fewer activities happening around midnight and early in the morning. The distributions of semantic regions are sparser compared with those in the afternoon and in the evening. Also, because there are few cars parking at night and early in the mornign, pedestrians can cross the parking lot over empty parking spaces. In Figure 4-18, the shape of the semantic region at time slice between 13 o'clock and 14 o'clock on May 16 changes, because there are more people from the top entering the parking lot and exiting from left at the particular time interval. In Figure 4-20, the shape of semantic region 3 also changes over time. People may exit the parking lot from the left of the scene or from a gate in the middle area of the scene (somewhere between two rows of trees).

Figure 4-18: The dynamic change of semantic region 1 over time learnt from the trajectories collected from a parking lot. The same background image is used for all time slices. The background image is not representative of each time slice (i.e. there are not this many cars parked there late at night). There are fewer activities in the parking lot around midnight, so the distributions are sparser. Also there are few cars parking there and pedestrians can cross the parking lot over empty parking spaces. There are more activities in the afternoon and in the evening. So the distributions are denser at that time. The shape of the semantic region can change.

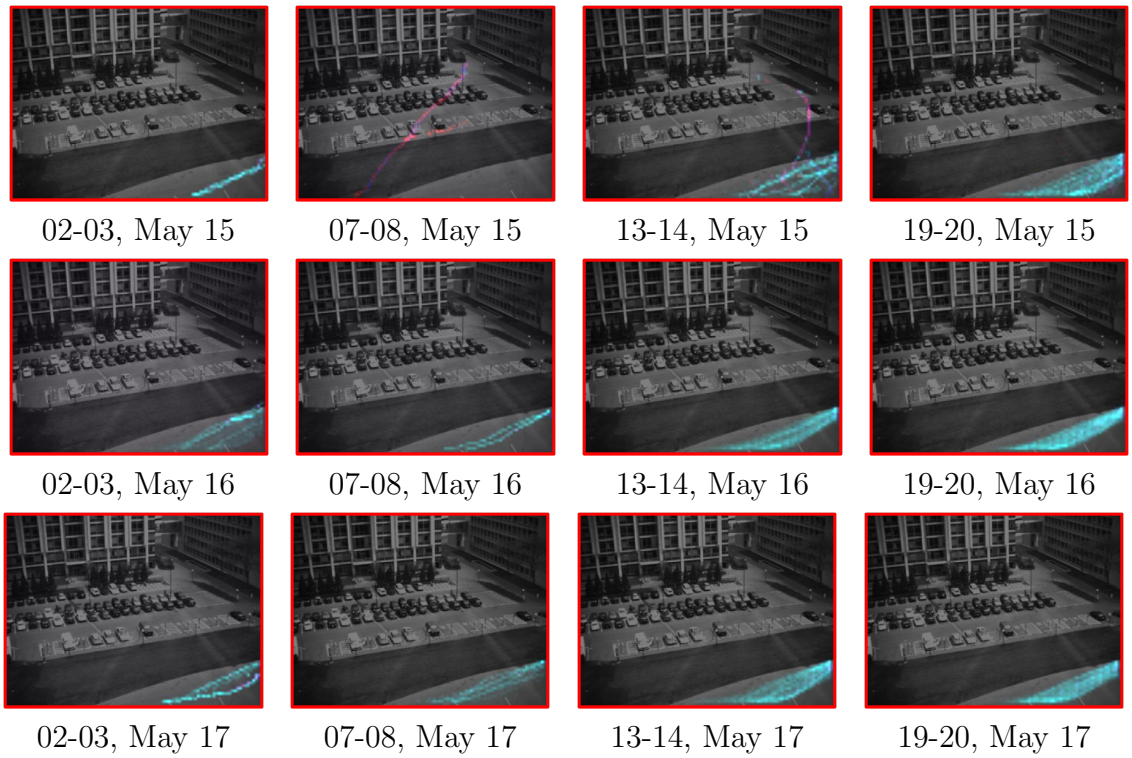


Figure 4-19: The dynamic change of semantic region 2 over time learnt from the trajectories collected from a parking lot.

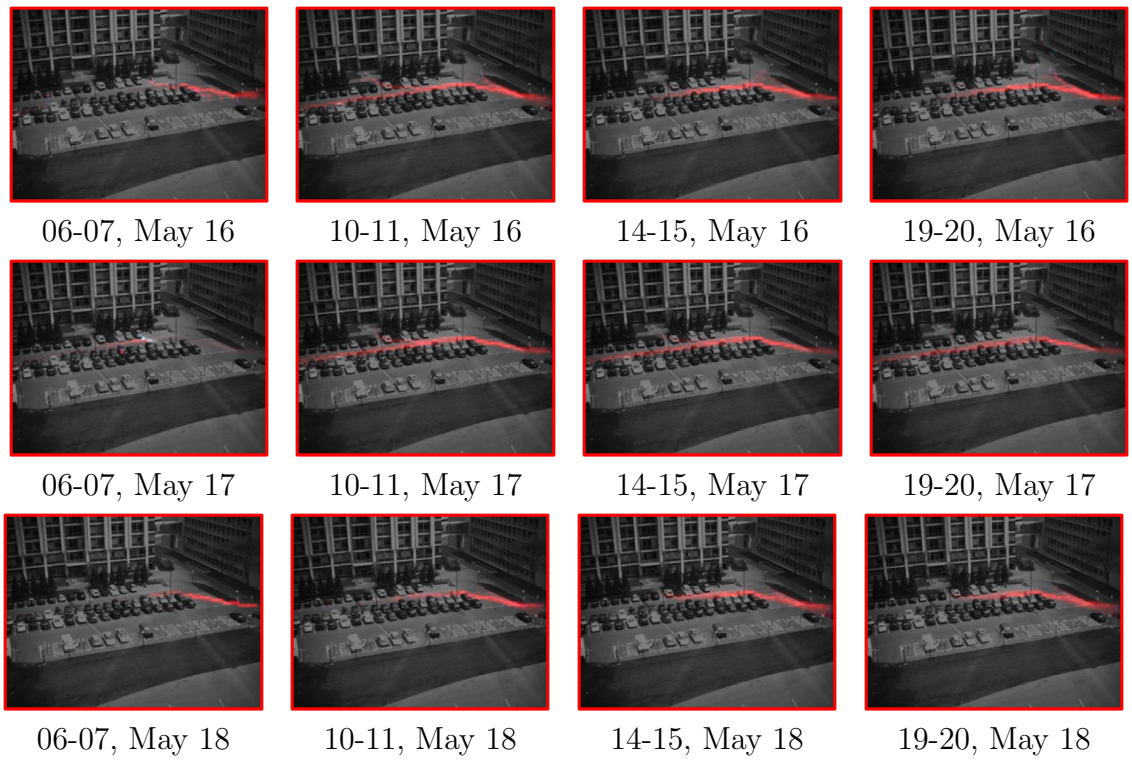


Figure 4-20: The dynamic change of semantic region 3 over time learnt from the trajectories collected from a parking lot.

Figure 4-21 shows the abnormal trajectories detected at different time. Between 7 o'clock and 8 o'clock in the morning everyday, many trajectories passing through the bottom right corner of the scene are detected as abnormal. As shown in Figure 4-19, there are not many trajectories of this activity happening in the morning. When they start to appear in a large number between 8 o'clock and 9 o'clock, they are detected as abnormal. In contrast, these kinds of trajectories are not detected as abnormal between 14 o'clock and 15 o'clock in the afternoon. Many trajectories detected as abnormal are those passing through the grass field. An interesting example occurred between 13 o'clock and 15 o'clock on May 16th. A worker was mowing the grass around this time. Many trajectories moving back and forth horizontally on the grass field are detected as an abnormality between 13 o'clock and 14 o'clock. However, after the model of this activity has been well learnt at this time slice, similar trajectories are not detected as abnormal in the next hour.

4.4 Summary

In this chapter, Dual-HDP and dynamic Dual-HDP are used to learn models of semantic regions and cluster trajectories. Dual-HDP assumes that the models of semantic regions and activities are static, while dynamic Dual-HDP updates the models of semantic regions and activities over time. Dynamic Dual-HDP cluster trajectoris incrementally and thus it can process a huge set of trajectories collected from a very long period with low computational complexities. Both these two approaches are evaluated on radar tracks and trajectories collected from a parking lot. Their results make intuitive sense. Dual-HDP is quantitively compared with other clustering methods on a simulated data set and it achieves much better performance especially when there are significant tracking errors. However, a quatitive evaluation on real data sets is difficult because it is not easy to find the ground truth. One possible way is to compare with user study. However, different people have different judgement on clustering and abnormality detection. Further more, the subjects need to observe a large set of trajectories in order to well understand the scene happening in the scene.

Figure 4-21: Abnormal trajectories in the parking lot scene detected at different time slices. The same threshold of data likelihood is used for all the slices. Between 7 o'clock and 8 o'clock in the morning everyday, many trajectories passing through the bottom right corner of the scene are detected as abnormal. There are not many trajectories of this activity happening in the morning. When they start to appear in a large number between 8 o'clock and 9 o'clock, they are detected as abnormal. Between 13 o'clock and 15 o'clock on May 16th, a worker was mowing the grass around this time. Many trajectories moving back and forth horizontally on the grass field are detected as an abnormality between 13 o'clock and 14 o'clock. However, after the model of this activity has been well learnt at this time slice, similar trajectories are not detected as abnormal in the next hour.

This makes user study time comsuming. Quantitive evaluation of trajectory analysis on real data sets is an important research direction as future work.

Chapter 5

Correspondence-Free Activity Analysis in Multiple Camera Views

Chapter 4 assumes a single camera view. If we need to monitor activities in a large area, video streams from multiple camera views have to be used. This chapter presents a hierarchical Bayesian model of clustering trajectories in multiple camera views. We group trajectories, which belong to the same activity category but are observed in different camera views, into one cluster. The distributions of a path in multiple camera views are jointly modeled. It is more challenging than activity analysis in a single camera view since we do not track objects across camera views.

5.1 Feature Space

Objects are tracked in each of the camera views independently using the Stauffer-Grimson tracker [135]. Similar to Section 4, a trajectory is treated as a document. The locations and moving directions of observations of an object are computed as features and quantized to visual words according to a codebook of its camera view. However, the codebook is built from multiple camera views. In each camera view, the space of the view is uniformly quantized into small cells and the velocity of objects is quantized into several directions. A global codebook concatenates the codebooks of all the camera views. Thus the word value of an observation i is indexed by (c_i, x_i, y_i, d_i)

in the global codebook. c_i is the camera view in which i is observed. (x_i, y_i) and d_i are the quantized coordinates and moving direction of observation i in camera c_i.

5.2 Trajectory Network

A network is built connecting trajectories observed in multiple camera views based on their temporal extents. Each trajectory is a node on the network. Let t_{si} and t_{ei} be the starting and ending time of trajectory i. T is a positive temporal threshold. It is roughly the maximum transition time of objects moving between adjacent camera views. If trajectories a and b are observed in different camera views and their temporal extents are close,

$$(t_{sa} \leq t_{sb} \leq t_{ea} + T) \vee (t_{sb} \leq t_{sa} \leq t_{eb} + T), \tag{5.1}$$

then a and b will be connected by an edge on the network. This means that a and b may be the same object since they are observed by cameras around the same time. There is no edge between two trajectories observed in the same camera view. An example can be found in Figure 5-1. As shown in (a), the views of cameras 1 and 2 overlap and are disjoint with the view of camera 3. Trajectories 1 and 2 observed by cameras 1 and 2 correspond to the same object moving across camera views. Their temporal extents overlap as shown in (b), so they are connected by an edge in the network as shown in (d). Trajectories 3 and 4 observed by cameras 1 and 3 correspond to an object crossing disjoint views. Their temporal extents have no overlap but the gap is smaller than T as shown in (c), so they are also connected. Trajectories 3 and 6, 5 and 7 do not correspond to the same objects, but their temporal extents are close, so they are also connected in the network. A single trajectory 3 can be connected to two trajectories (4 and 6) in other camera views. An edge in the network indicates a possible correspondence candidate only based on the temporal information of trajectories. But we do not really solve the correspondence problem when building the trajectory network, since many edges are actually false correspondences. The

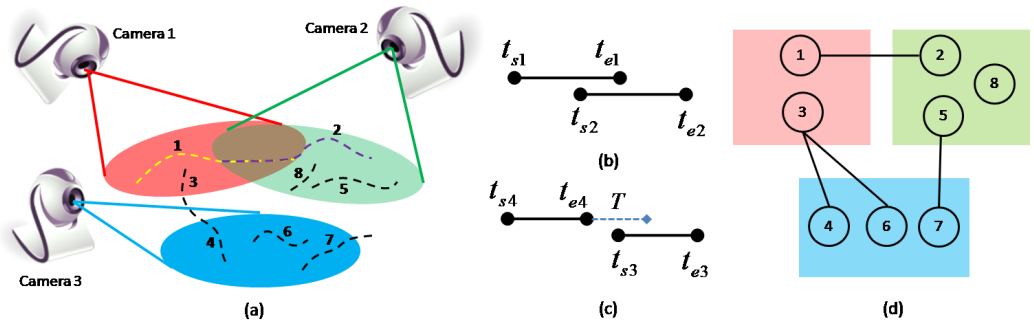


Figure 5-1: An example of building a network connecting trajectories in multiple camera views. (a) Trajectories in three camera views. (b) The temporal extents of trajectories 1 and 2. (c) The temporal extents of trajectories 3 and 4. (d) The network connecting trajectories. Trajectories 1 and 2 observed by cameras 1 and 2 correspond to the same object moving across camera views. Their temporal extents overlap, so they are connected by an edge in the network. Trajectories 3 and 4 observed by cameras 1 and 3 correspond to an object crossing disjoint views. Their temporal extents have no overlap but the gap is smaller than T, so they are also connected. Trajectories 3 and 6, 5 and 7 do not correspond to the same objects, but their temporal extents are close, so they are also connected in the network. A single trajectory 3 can be connected to two trajectories (4 and 6) in other camera views.

network simply keeps all of the possible candidates.

5.3 Probabilistic Model

In this section, we describe our probabilistic model which clusters trajectories in different camera views into activities and models paths across camera views. Documents are trajectories, words are observations, and topics are activities (paths). Each activity has a distribution over locations and moving directions in different camera views, and corresponds to a path. In previous topic models, documents are generated independently. However, we assume that if two trajectories in different camera views are connected by an edge in the network, which means that they may correspond to the same object since they are observed by cameras around the same time, they tend to have similar distributions over activities. Thus the distributions of an activity (a path of objects) in different camera views can be jointly modeled. In Figure 5-2, we use an example to describe the high level picture of our model. Trajectories a and b

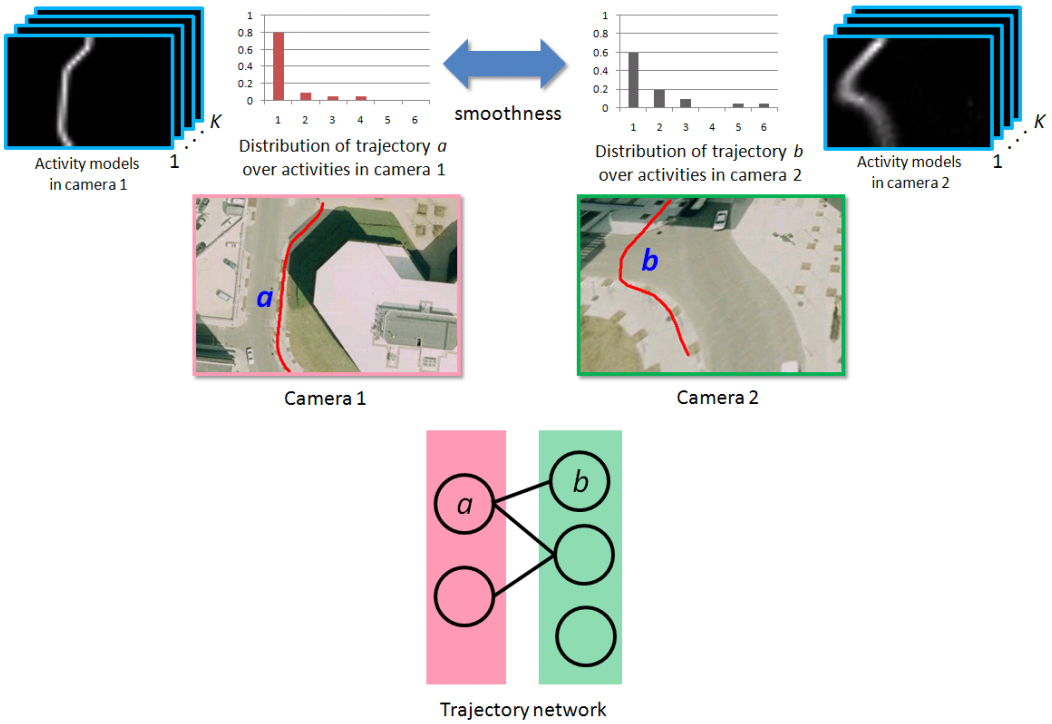


Figure 5-2: An example to describe the high level picture of our model. Trajectories a and b are observed in different camera views and are connected by an edge in the trajectory network. Points on trajectories are assigned to activity categories by fitting activity models. Thus both a and b have distributions over activities. The smoothness constraint requires that their distributions over activities are similar in order to have small penalty. In this example, both trajectory a and b have a larger distribution on activity 1, so the models of activity 1 in two different camera views can be related to the same activity.

are observed in different camera views and are connected by an edge in the trajectory network. Points on trajectories are assigned to activity categories by fitting activity models. Thus both a and b have distributions over activities. The smoothness constraint requires that their distributions over activities are similar in order to have small penalty. In this example, both trajectory a and b have a larger distribution on activity 1, so the models of activity 1 in two different camera views can be related to the same activity.

Let M be the number of trajectories. Each trajectory j has N_j observations. Each observation i on trajectory j has a visual word value x_{ji} which is an index of the global codebook. Observations will be clustered to one of the K activity categories.

Let z_{ji} be the activity label of observation i on trajectory j. Each activity k has a multinomial distribution ϕ_k over the global codebook. So an activity is modeled as a distribution over space and moving directions in all the camera views. ϕ_k is sampled from a Dirichlet prior

$$p(\phi_k|\beta) = Dir(\phi_k;\beta), \tag{5.2}$$

where $Dir(\cdot;\cdot)$ is Dirichlet distribution and β is a flat hyperparameter. If a visual word x_{ji} has activity label z_{ji}, its data likelihood is

$$p(x_{ji}|z_{ji},\{\phi_k\}) = \phi_{z_{ji}x_{ji}}. \tag{5.3}$$

Eq 5.2 and 5.3 are the same as modeled in LDA.

Each trajectory has a random variable θ_j which is the parameter of a multinomial distribution over K activities. Activity labels $\{z_{ji}\}$ of observations are sampled from θ_j. If two trajectories j_1 and j_2 are connected by an edge on the network, they are neighbors and the smoothness constraint requires that θ_{j_1} and θ_{j_2} are similar and the distributions of $\{z_{j_1i}\}$ and $\{z_{j_2i}\}$ are similar. The joint distribution of $\{\theta_j\}$ and $\{z_{ji}\}$ are modeled as,

$$\begin{aligned}
&p(\{\theta_j\},\{z_{ji}\}|\alpha,\gamma)\\
\propto&\prod_{j=1}^{M}\prod_{k=1}^{K}(\theta_{jk})^{\alpha-1}\prod_{\{j_1,j_2\}\in E}\prod_{k=1}^{K}(\theta_{j_1k})^{\gamma\cdot n_{j_2k}}(\theta_{j_2k})^{\gamma\cdot n_{j_1k}}\prod_{j=1}^{M}\prod_{i=1}^{N_j}\theta_{jz_{ji}}\\
=&\prod_{j=1}^{M}\prod_{k=1}^{K}\theta_{jk}^{\alpha-1+\gamma\sum_{j'\in\Omega_j}n_{j'k}}\prod_{j=1}^{M}\prod_{i=1}^{N_j}\theta_{jz_{ji}}\\
=&\prod_{j=1}^{M}\left[\frac{\prod_{k=1}^{K}\Gamma(\alpha+\gamma\sum_{j'\in\Omega_j}n_{j'k})}{\Gamma(K\cdot\alpha+\gamma\sum_{j'\in\Omega_j}\sum_{k=1}^{K}n_{j'k})}\mathrm{Dir}(\theta_j;\alpha+\gamma\sum_{j'\in\Omega_j}n_{j'1},\ldots,\alpha+\gamma\sum_{j'\in\Omega_j}n_{j'K})\prod_{i=1}^{N_j}\theta_{jz_{ji}}\right]
\end{aligned} \tag{5.4}$$

$\Gamma(\cdot)$ is the Gamma function. n_{jk} is the number of observations assigned to activity k on trajectory j. E is the set of pairs of neighboring trajectories which are connected. Ω_j is the set of trajectories connected with j. α is a flat Dirichlet prior as a hyperpa-

rameter. $(\sum_{j'\in\Omega_j} n_{j'1}, \ldots, \sum_{j'\in\Omega_j} n_{j'K})$ is the histogram of observations assigned to K activity categories on the neighboring trajectories of j. It is used as the Dirichlet parameter for θ_j, after being weighted by a positive scalar γ and added to a flat prior α. Let $\rho_k = \alpha + \gamma \cdot \sum_{j'\in\Omega_j} n_{j'k}$. According to the properties of the Dirichlet distribution, if $\theta_j \sim \mathrm{Dir}(\rho_1, \ldots, \rho_K)$, the expectation of θ_j is $(\rho_1/\sum \rho_k, \ldots, \rho_K/\sum \rho_k)$ and its variation is small if $\sum \rho_k$ is large. Notice that z_{ji} is sampled from θ_j and θ_j has a constraint added by $z_{j'i'}$ on its neighboring trajectories. So trajectory j tends to have a similar distribution over activities as its neighboring trajectories, which means that they are smooth. A larger γ puts a stronger smoothness constraint. If $\gamma = 0$, Eq 5.4 is the same as in LDA where $\{\theta_j\}$ are sampled from a Dirichlet prior $Dir(\cdot;\alpha)$ independently.Given Eq 5.2, 5.3 and 5.4, finally the joint distribution of $\{\phi_k\}$, $\{\theta_j\}$, $\{z_{ji}\}$ and $\{x_{ji}\}$ is

$$
\begin{aligned}
& p(\{\phi_k\}, \{\theta_j\}, \{z_{ji}\}, \{x_{ji}\} | \alpha, \beta, \gamma) \\
=& p(\{\theta_j\}, \{z_{ji}\} | \alpha, \gamma) \prod_{k=1}^{K} p(\{\phi_k\}|\beta) \prod_{j=1}^{M} \prod_{i=1}^{N_j} p(\{x_{ji}\}|\{z_{ji}\}, \{\phi_k\}) \\
=& \prod_{j=1}^{M} \left[\frac{\prod_{k=1}^{K} \Gamma(\alpha + \gamma \sum_{j'\in\Omega_j} n_{j'k})}{\Gamma(K\cdot\alpha + \gamma \sum_{j'\in\Omega_j} \sum_{k=1}^{K} n_{j'k})} \mathrm{Dir}(\theta_j; \alpha + \gamma \sum_{j'\in\Omega_j} n_{j'1}, \ldots, \alpha + \gamma \sum_{j'\in\Omega_j} n_{j'K}) \right] \\
& \prod_{k=1}^{K} \mathrm{Dir}(\phi_k; \beta) \prod_{j=1}^{M} \prod_{i=1}^{N_j} \left(\theta_{j z_{ji}} \cdot \phi_{z_{ji} x_{ji}} \right). \qquad (5.5)
\end{aligned}
$$

5.3.1 Learning and Inference

We do inference by Gibbs sampling. It turns out that $\{\theta_j\}$ and $\{\phi_k\}$ can be integrated out during the Gibbs sampling procedure.

$$
\begin{aligned}
& p(\{z_{ji}\},\{x_{ji}\}|\alpha,\beta,\gamma) \\
=& \int_{\{\phi_k\}}\int_{\{\theta_j\}} p(\{\theta_j\},\{\phi_k\},\{z_{ji}\},\{x_{ji}\}|\alpha,\beta,\gamma)d\{\theta_j\}d\{\phi_k\} \\
=& \int_{\{\phi_k\}} \prod_{k=1}^{K} p(\phi_k|\beta) \prod_{j=1}^{M}\prod_{i=1}^{N_j} p(\{x_{ji}\}|\{z_{ji}\},\{\phi_k\})d\{\phi_k\} \int_{\{\theta_j\}} p(\{\theta_j\},\{z_{ji}\}|\alpha,\gamma)d\{\theta_j\} \\
\propto& \int_{\{\phi_k\}} \prod_{k=1}^{K}\prod_{w=1}^{W} \phi_{kw}^{\beta-1} \prod_{j=1}^{M}\prod_{i=1}^{N_j} \phi_{z_{ji}x_{ji}} d\{\phi_k\} \int_{\{\theta_j\}} \prod_{j=1}^{M}\prod_{k=1}^{K} \theta_{jk}^{\alpha-1+\gamma\sum_{j'\in\Omega_j} n_{j'k}} \prod_{j=1}^{M}\prod_{i=1}^{N_j} \theta_{jz_{ji}} d\{\theta_j\} \\
=& \int_{\{\phi_k\}} \prod_{k=1}^{K}\prod_{w=1}^{W} (\phi_{kw})^{\beta+m_{kw}-1} d\{\phi_k\} \int_{\{\theta_j\}} \prod_{j}\prod_{k} (\theta_{jk})^{\alpha+n_{jk}+\gamma\cdot\sum_{j'\in\Omega_j} n_{j'k}-1} d\{\theta_j\} \\
=& \prod_k \frac{\prod_w \Gamma(\beta+m_{kw})}{\Gamma(W\cdot\beta+m_{k\cdot})} \prod_j \frac{\prod_k \Gamma\left(\alpha+n_{jk}+\gamma\cdot\sum_{j'\in\Omega_j} n_{j'k}\right)}{\Gamma\left(K\cdot\alpha+n_{j\cdot}+\gamma\cdot\sum_{j'\in\Omega_j} n_{j'\cdot}\right)},
\end{aligned}
\tag{5.6}
$$

where W is the size of the global codebook, m_{kw} is the number of observations assigned to activity k with value w, $m_{k\cdot}$ is the total number of observations assigned to activity k, n_{jk} is the number of observations assigned to activity k on trajectory j, and $n_{j\cdot}$ is the total number of observations on trajectory j. Then the conditional distribution of z_{ji} given all the other activity labels $\mathbf{z}^{-ji}$ is

$$
p(z_{ji}=k|\mathbf{z}^{-ji},\{x_{ji}\},\alpha,\beta,\gamma) \propto \frac{\beta+m_{k,x_{ji}}^{-ji}}{W\cdot\beta+m_{k,\cdot}^{-ji}} \cdot \frac{\alpha+n_{jk}^{-ji}+\gamma\sum_{j'\in\Omega_j} n_{j'k}}{K\cdot\alpha+n_{j\cdot}^{-ji}+\gamma\sum_{j'\in\Omega_j} n_{j'\cdot}}, \tag{5.7}
$$

where $m_{kx_{ji}}^{-ji}$, $m_{k\cdot}^{-ji}$, n_{jk}^{-ji}, and $n_{j\cdot}^{-ji}$ are the same statistics as $m_{kx_{ji}}$, $m_{k\cdot}$, n_{jk}, and $n_{j\cdot}$ except that they have excluded observation i on trajectory j. To have a large posterior in Eq 5.7, the first term requires that the value of observation i should fit the model of activity k, and the second term requires that its activity label is consistent with those of observations on the same trajectory and neighboring trajectories, with γ controlling the weight of neighboring trajectories. The models of activities are not

explicitly learnt during the Gibbs sampling procedure, but they can be estimated from any single sample of $\{z_{ji}\}$,

$$\widehat{\phi}_{kw} = \frac{\beta + m_{kw}}{W \cdot \beta + m_{k\cdot}} \tag{5.8}$$

5.3.2 Labeling Trajectories into Activities

A trajectory is labeled as activity k, if most of its observations are assigned to k. The activity label of an observation can be obtained during the Gibbs sampling procedure based on Eq 5.7. However, there may be an over smoothing effect, since in some cases most of the trajectories being the neighbors of trajectory j do not correspond to the same object as j. In this work, we adopt an alternative labeling approach which actually achieves better performance in experiments. As shown by the experimental results in Section 5.4, the activity models learnt from Gibbs sampling are distinctive enough to label trajectories. After the activity models have been learnt and fixed at the end of Gibbs sampling, which uses Eq 5.7 and 5.8, we ignore the smoothness constraint among trajectories and label the observation as

$$z_{ji} = \arg\max_{k} \hat{\phi}_{kx_{ji}} \tag{5.9}$$

This is also used to label an unseen new trajectory.

5.3.3 Detection of Abnormal Trajectories

When detecting abnormal trajectories, we also ignore the smoothness constraint and fix the learnt activity models $\{\hat{\phi}_k\}$. A trajectory is detected as an abnormality if it does not fit any activity model well. Then abnormality detection is reduced to the Latent Dirichlet Allocation model proposed in [18]. The likelihood of a trajectory j under the learnt activity models $\{\hat{\phi}_k\}$ is

$$p(\mathbf{w}_j = \{x_{ji}\}|\alpha, \{\hat{\phi}_k\}) = \int p(\theta_j|\alpha) \left(\prod_{i=1}^{N_j} \sum_{z_{ji}} p(z_{ji}|\theta_j) p(x_{ji}|\hat{\phi}_{z_{ji}}) \right), \tag{5.10}$$

where $p(\theta_j|\alpha)$ is a Dirichlet distribution, and both $p(z_{ji}|\theta_j)$ and $p(x_{ji}|\hat{\phi}_{z_{ji}})$ are discrete distributions. Since the computation of Eq 5.10 is intractable, in [18] a variational approach was used to compute a lower bound of Eq 5.10. A trajectory is flagged as abnormal if its lower bound is small.

5.3.4 Complexity

In order to simplify the notation, we assume that all the trajectories have the same number of observations, which is a fixed constant. The spatial complexity of our approach is $O(WK) + O(MK)$, while that of similarity based approaches is at least $O(M^2)$. The storage of similarity based approaches is unmanageable when M is huge. W is the size of the codebook, K is the number of activity categories, and M is the number of trajectories. In our approach, the time complexity of each Gibbs sampling iteration is $O(M)$, however it is difficult to provide theoretical analysis on the convergence of Gibbs sampling. Similarity based approaches have to compute the similarity of $O(M^2)$ pairs of trajectories and if spectral clustering is used, it is quite challenging to compute the eigenvectors of a huge $M \times M$ similarity matrix when M is large. The time complexity of our approach to label a new trajectory into one of the activity categories or detect a new trajectory as abnormal is $O(K)$[1], while the time complexity of similarity based approaches is at least $O(M)$. So our approach is much more efficient when the number of trajectories is huge.

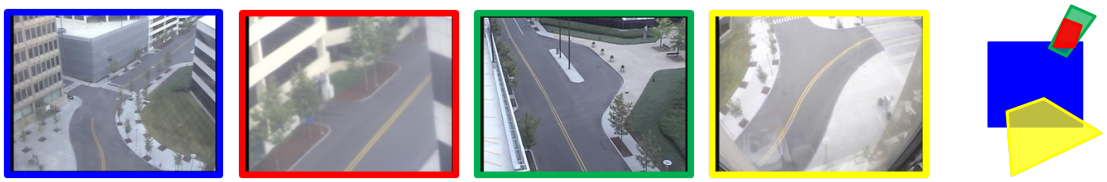
5.4 Experimental Results

We evaluate our approach on two data sets, a parking lot scene and a street scene. Each has four camera views. Each camera view is of size 320×240. To build the codebook, each camera view is quantized into 64×48 cells. Each cell is of size 5×5. The moving directions of moving pixels are quantized into four directions. There are tracking errors in both of the two data sets. For example, a track may break into

[1]In abnormality detection, a variational approach [18] is used to compute a lower bound of the data likelihood (Eq 5.10) in an iterative process. We assume the number of iterations is small.

Camera 1 Camera 2 Camera 3 Camera 4

(a) Parking lot scene



Camera 1 Camera 2 Camera 3 Camera 4

(b) Street scene

Figure 5-3: Camera views and their topology in two data sets, a parking lot scene and a street scene. When the topology of camera views is plotted, the fields of camera views are represented by different colors: blue (camera 1), red (camera 2), green (camera 3), yellow (camera 4). However, our approach does not require knowledge of the topology of the cameras in advance.

fragments because of interactions among objects. In order to obtain more quantitative evaluation, we simulate some trajectories whose activity categories are known as the ground truth, and evaluate our approach on the simulated data.

5.4.1 Learning Activity Models and Clustering Trajectories

Parking Lot Scene

The parking lot data set has 22,951 trajectories, collected from 10 hours during the day time over 3 days. Inspection shows that it is a fairly busy scene. The topology of its four camera views is shown in Figure 5-3 (a). The view of camera 1 has no overlap with other camera views. However, the gap between the views of cameras 1 and 2 is small. The views of cameras 2 and 3 have small overlap. The views of cameras 3 and 4 have large overlap. Our approach does not require the knowledge of the topology of cameras. Fourteen different activities are learnt from this data set. They are shown in Figure 5-4 - 5-7. For each activity, we plot its distribution over space and moving directions in the four camera views and show the trajectories clustered into

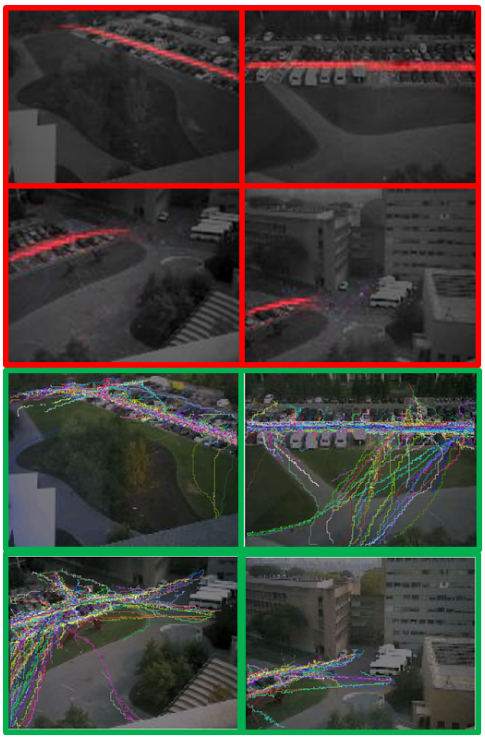
this activity. When visualizing activity models, moving directions are represented by different colors, and the density of distributions over space and moving directions is proportional to the brightness of colors (high brightness means high density). When plotting trajectories, random colors are used to distinguish individual trajectories.

In Figure 5-4, activity 1 captures vehicles and pedestrians entering the parking lot. It has a large extent in space and is observed by all four cameras. Activity 4 captures vehicles and pedestrians leaving the parking lot. In activities 5 (Figure 5-5) and 7 (Figure 5-5), pedestrians are walking in the same direction but on different paths. From the distributions of their models, it is observed that the two paths are side by side but well separated in space. The path of activity 6 occupies almost the same region as that of activity 5. However, pedestrians are moving in opposite directions in these two activities, so the distributions of their models are plotted in different colors. In activity 8 (Figure 5-5), pedestrians appear from behind the trees and a building as observed by cameras 3 and 4 and disappear from a gate of the parking lot in the view of camera 2.

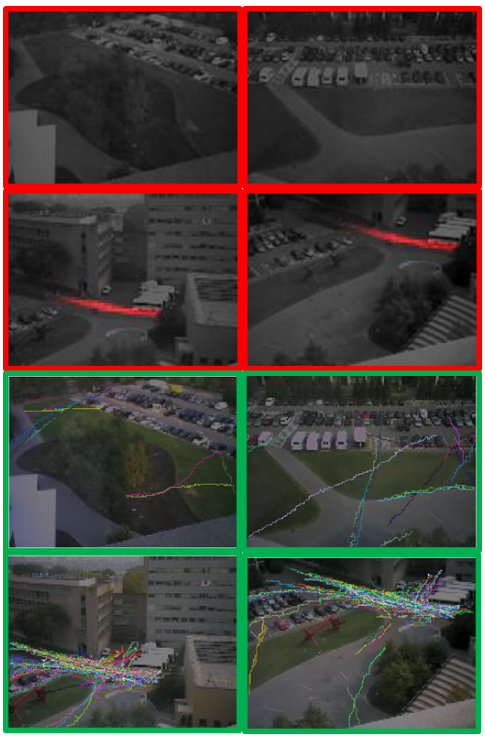
Street Scene

The topology of the four cameras of the street scene is shown in Figure 5-3 (b). Camera 1 has a distant view of the street. Camera 2 zooms in on the top-right part in the view of camera 1. The view of camera 3 has overlap with the views of cameras 1 and 2. It extends the top-right part of the view in camera 1 along the street. The view of camera 4 partially overlaps with the bottom region of the view in camera 1. There are $14,985$ trajectories in this data set, collected from 30 hours during day time over four days. Sixteen activities are learnt in this scene. They are shown in Figure 5-8 - 5-11.

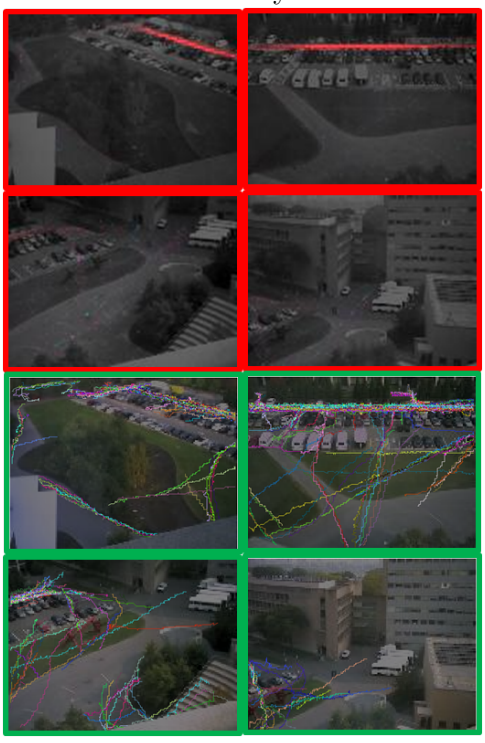
Activity 1 (Figure 5-8) captures vehicles moving on the road. It is observed by all of the four cameras. Vehicles first move from the top-right corner to the bottom-left corner in the view of camera 4. Then they enter the bottom region in the view of camera 1 and move upward. Some vehicles disappear at the exit points observed in the views of cameras 2 and 3, and some move further beyond the view of camera 3.



Activity 1



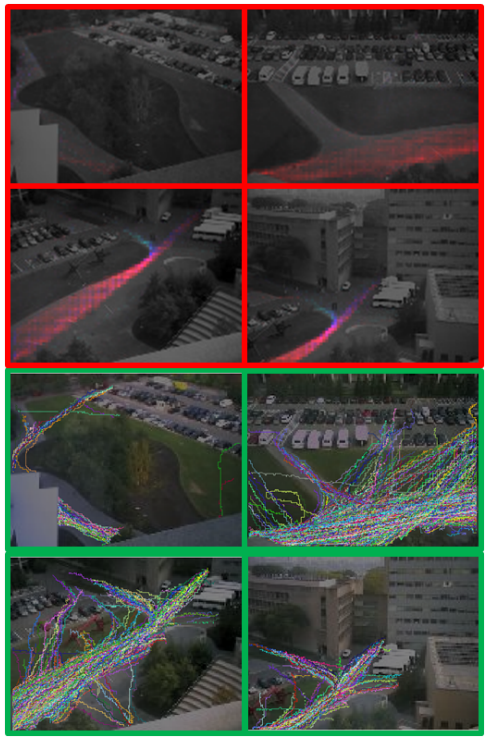
Activity 2



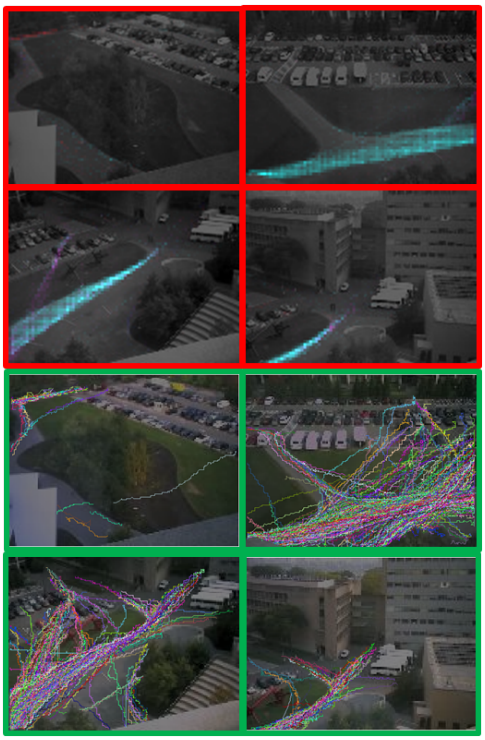
Activity 3

Activity 4

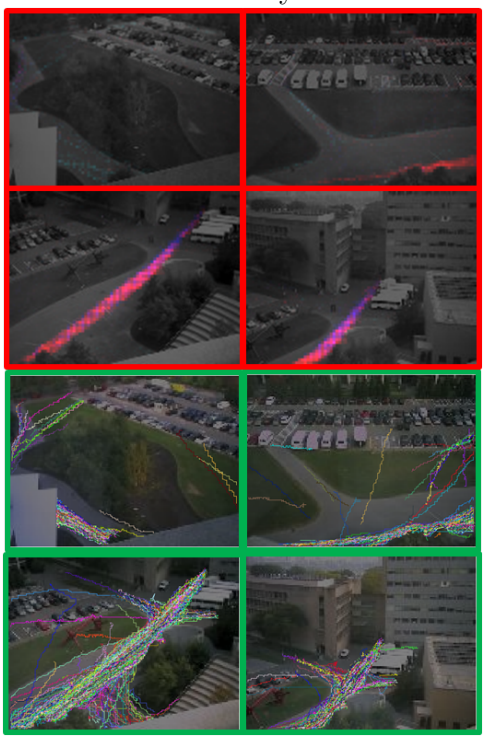
Figure 5-4: Distributions of activity models $(1-4)$ and clusters of trajectories in a parking lot scene. When plotting the distributions of activity models (in the four red windows on the top), different colors are used to represent different moving directions: $\rightarrow$ (red), $\leftarrow$ (cyan), $\uparrow$ (blue), $\downarrow$ (magenta). When plotting trajectories clustered into different activities (in the four green windows at the bottom), random colors are used to distinguish individual trajectories.



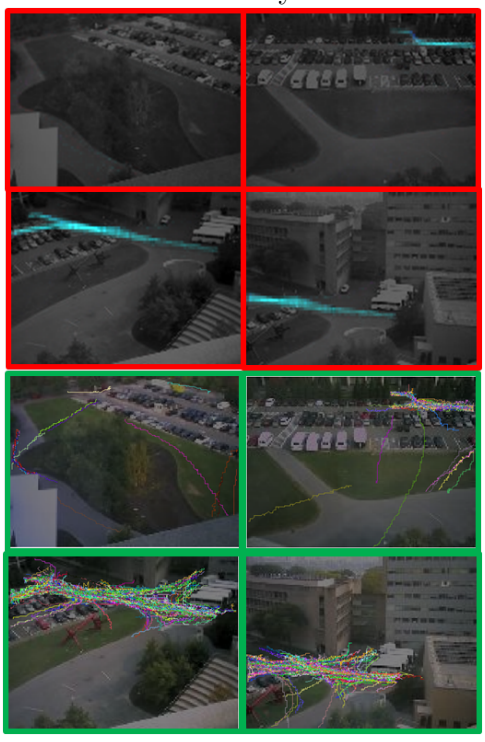
Activity 5



Activity 6

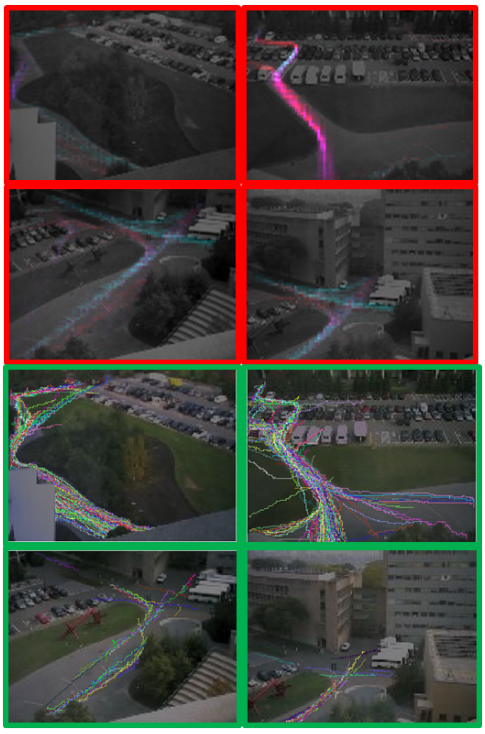


Activity 7



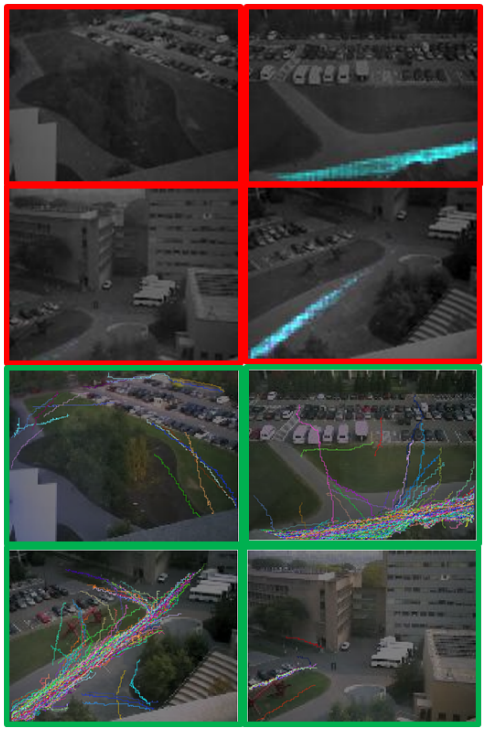
Activity 8

Figure 5-5: Distributions of activity models $(5-8)$ and clusters of trajectories in a parking lot scene.

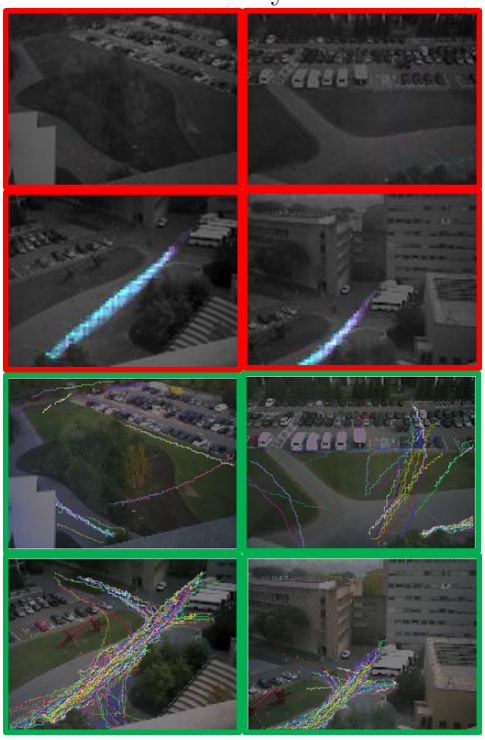


Activity 9

Activity 10

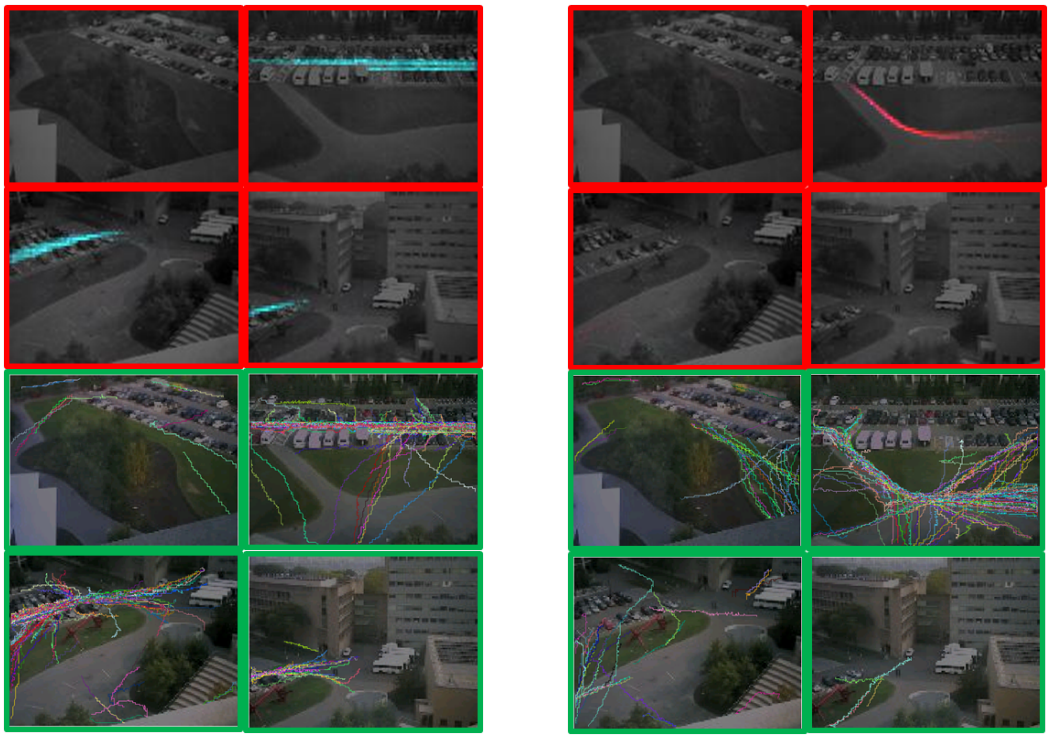


Activity 11



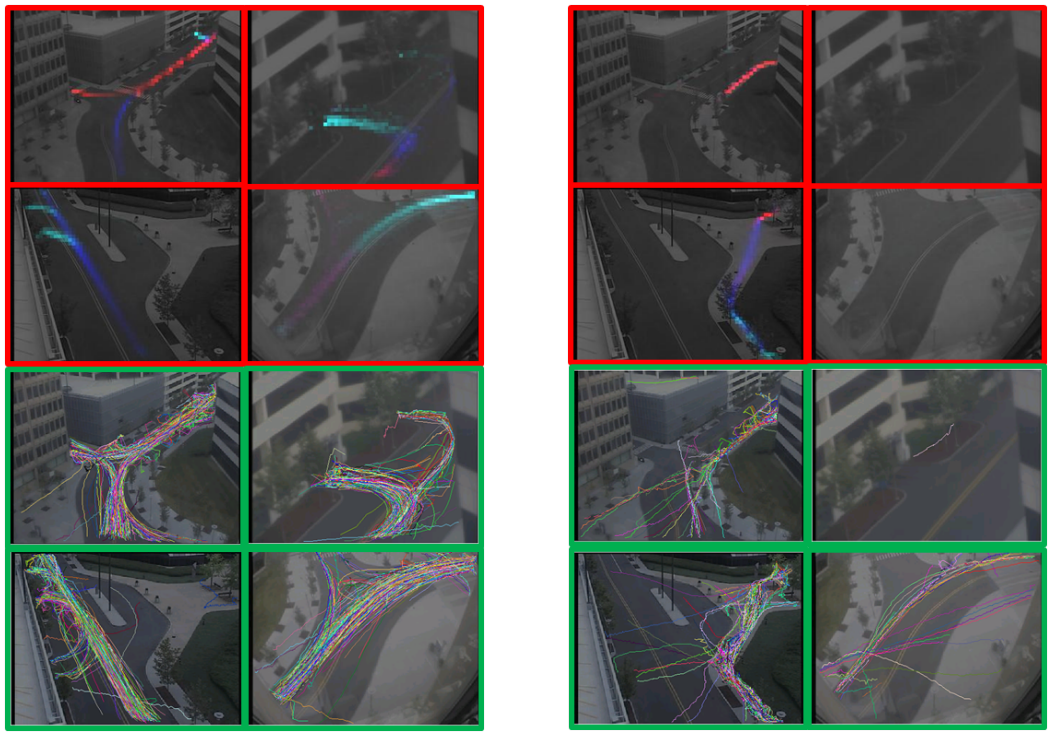
Activity 12

Figure 5-6: Distributions of activity models $(9-12)$ and clusters of trajectories in a parking lot scene.

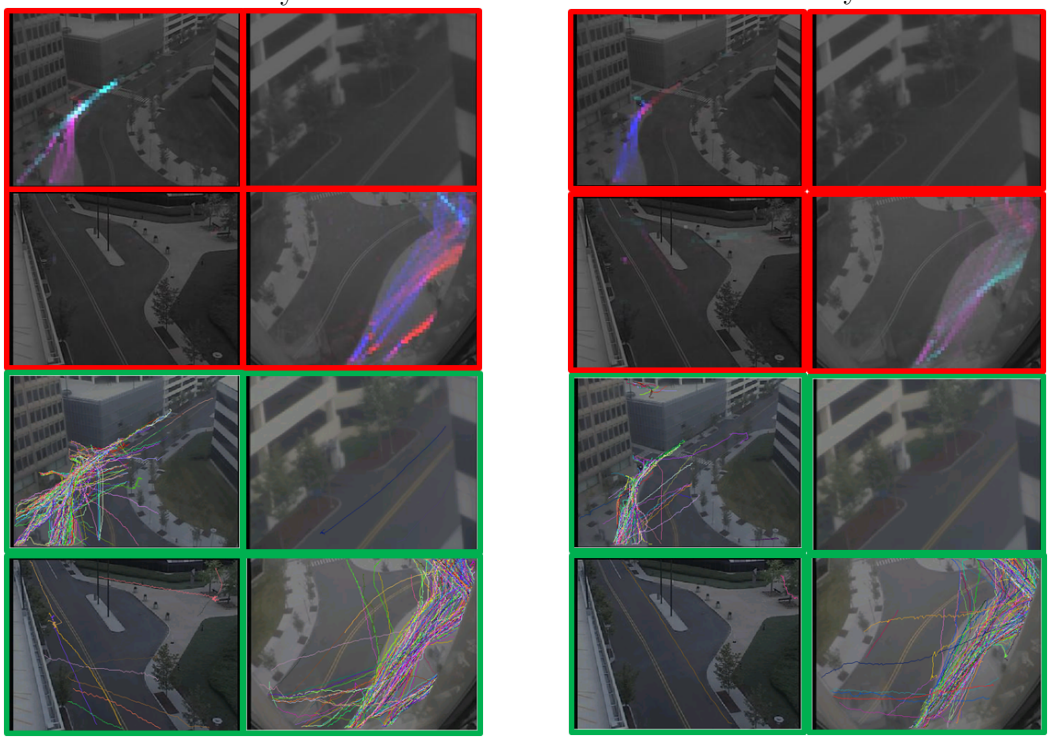


Activity 13 Activity 14

Figure 5-7: Distributions of activity models $(13 - 14)$ and clusters of trajectories in a parking lot scene.

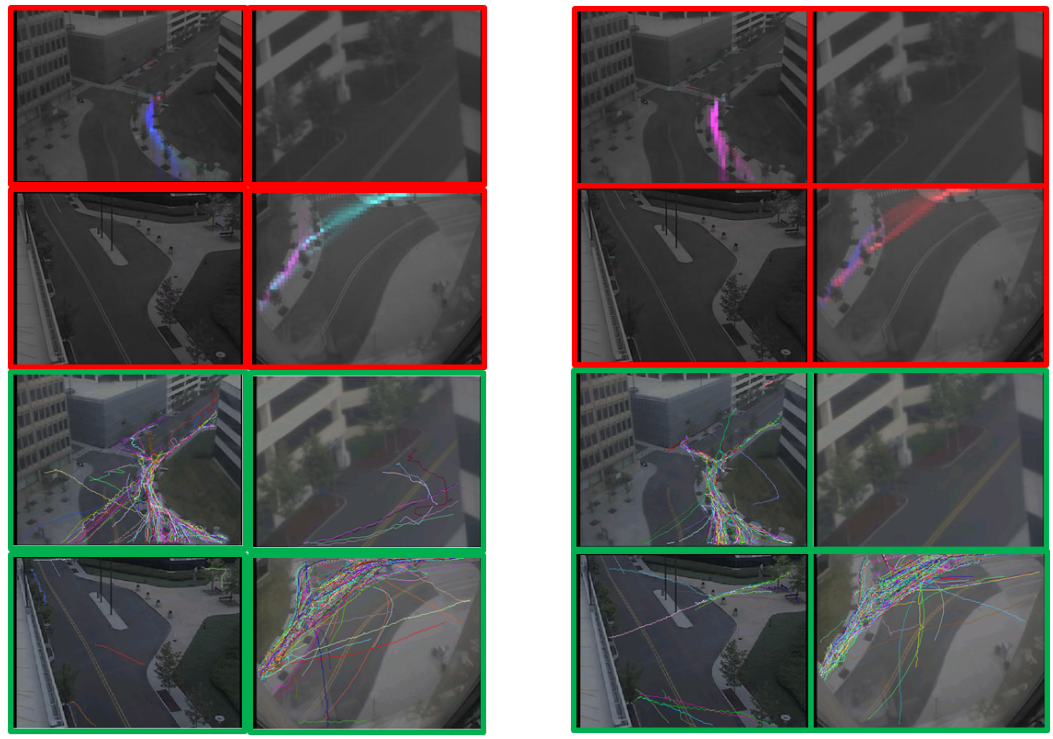


Activity 1 Activity 2



Activity 3 Activity 4

Figure 5-8: Distributions of activity models $(1-4)$ and clusters of trajectories of the street scene. The meaning of colors is the same as Figure 5-4.



Activity 5 Activity 6

Activity 7 Activity 8

Figure 5-9: Distributions of activity models $(5-8)$ and clusters of trajectories of the street scene.

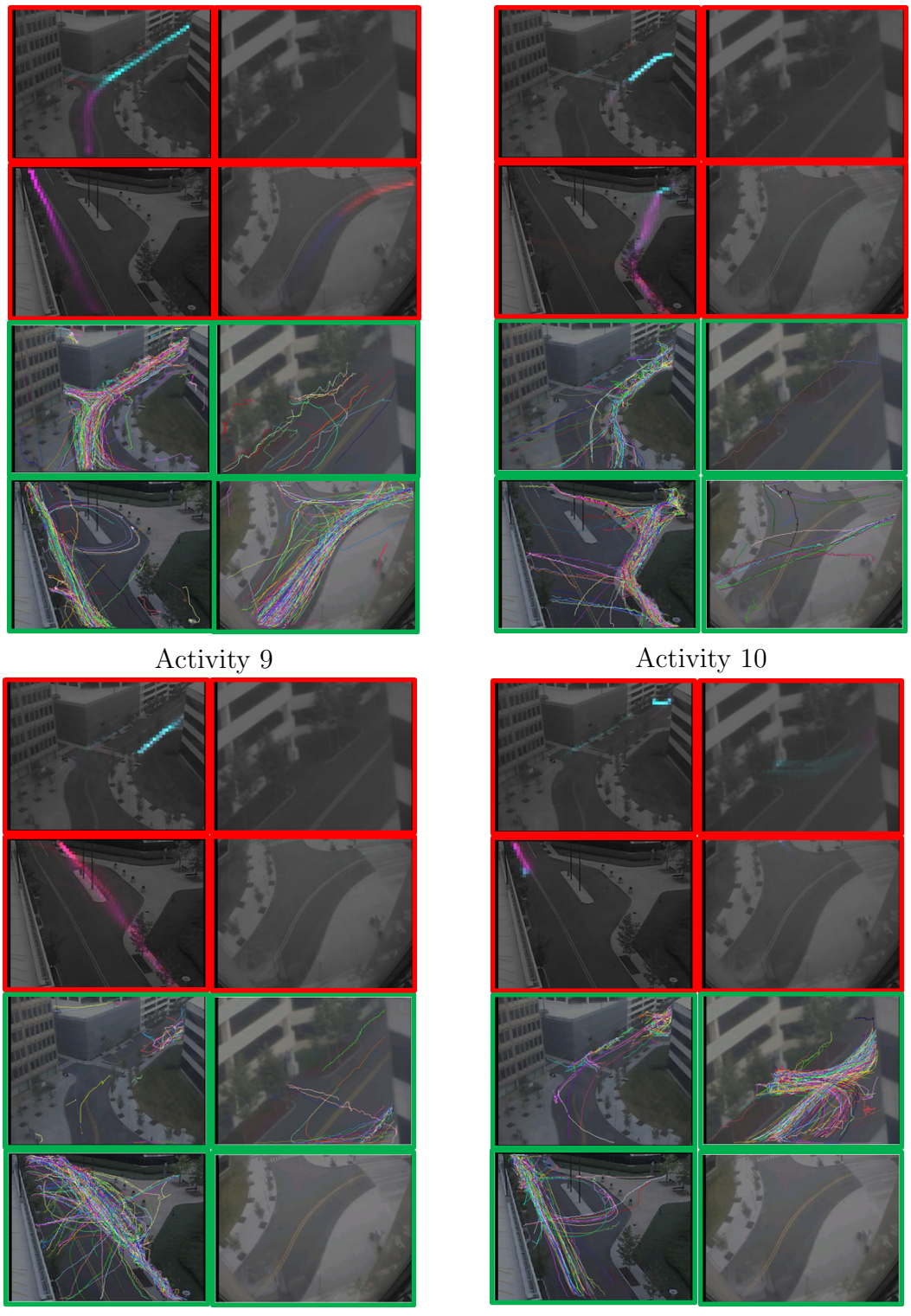
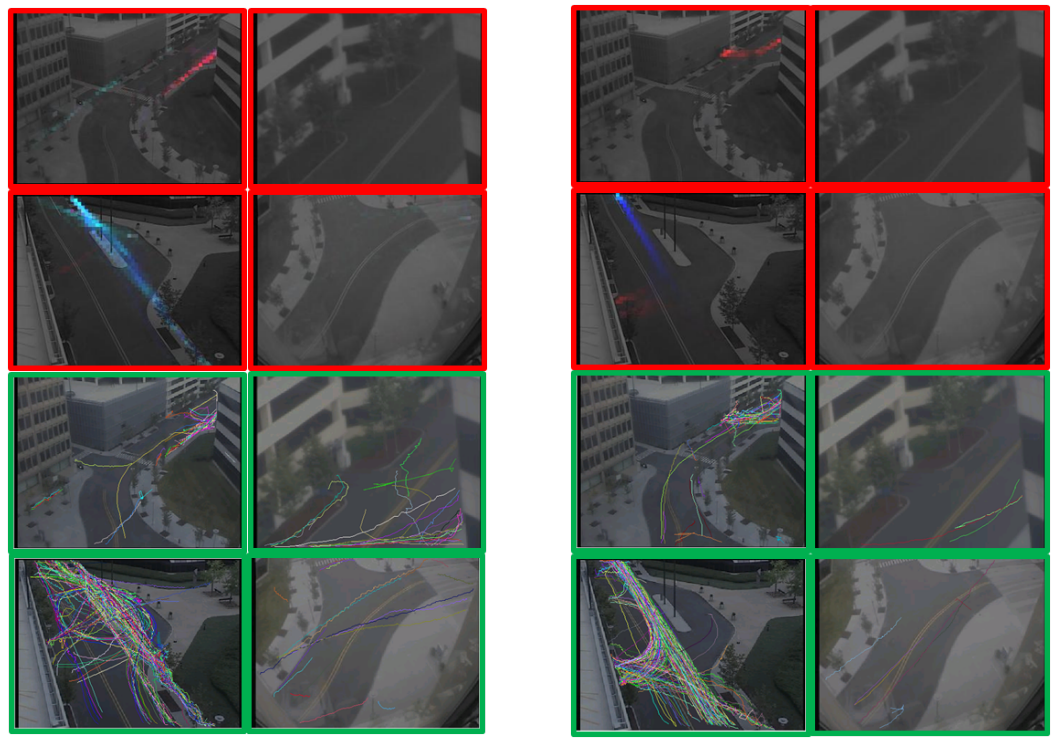
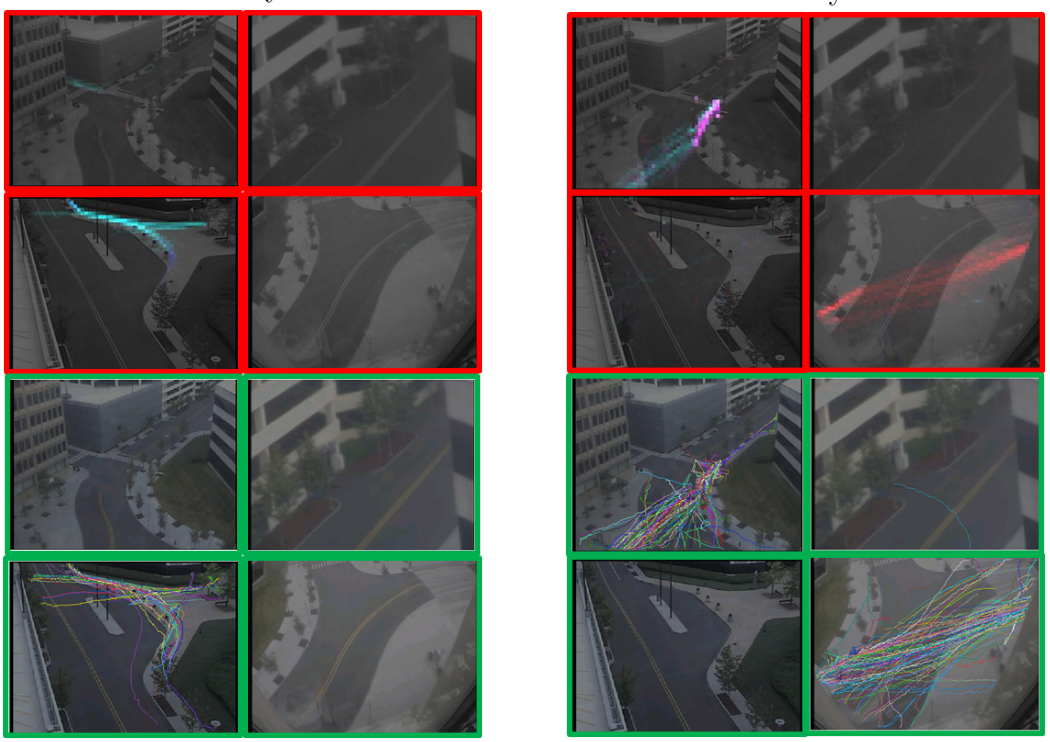


Figure 5-10: Distributions of activity models $(5-8)$ and clusters of trajectories of the street scene.



Activity 13 Activity 14



Activity 15 Activity 16

Figure 5-11: Distributions of activity models $(13 - 16)$ and clusters of trajectories of the street scene.

In activities 4 (Figure 5-8), 6 (Figure 5-9), and 7 (Figure 5-9), pedestrians first walk along the sidewalk in the view of camera 1, and then cross the street as observed by camera 4. The paths of activities 6 and 7 occupy similar regions in the view of camera 1, but their paths diverge in the view of camera 4. The paths of activities 3 and 4, 5 and 6 occupy the same regions but pedestrians are moving in opposite directions on them.

As shown in Figure 5-4 - 5-11, the models of activities reveal some structures, such as paths commonly taken by objects, and entrance and exit points in the scene. Some paths are less related to the appearance of the scene. For example, some paths cross the street outside the crosswalk in the street scene. Usually paths have spatial extents in multiple camera views. These regions can be detected by simply thresholding the density of the distributions of activities (ϕ_k in Eq 5.5). As observed, in these two very large data sets there are many outlier trajectories, which do not fit any activity model well, such as those crossing the grass fields in the parking lot scene. They are finally assigned some activity at random or because part of the trajectory fits a particular activity.

Negative log likelihood on testing data

Since clustering trajectories into activities is unsupervised learning, we compute the negative log likelihood on testing data to evaluate its performance. This is the log of perplexity in proportion to the number of bits required to encode the testing data. It measures how unseen testing data fits the model learnt from training data. Two hundred trajectories randomly sampled from each camera serve as the test set; the remaining trajectories are used for training. To compare models with different trajectory networks, the activity models $\{\phi_k\}$ are learnt with smoothness constraint added by the trajectory network. Once $\{\phi_k\}$ are learnt and fixed, the negative log likelihood is computed on the test data ignoring the smoothness constraint.

First, we compare our approach with two alternatives: (1) unconnected network; (2) network with random correspondences[2]. The former completely abandons the

[2]First find correspondence candidates using Eq 5.1. Instead of fully connecting these candidates

Table 5.1: Negative log likelihood under our approach and two alternative trajectory networks.

	Our approach	Unconnected	Random
Parking Lot	130.3	200.3	176.8
Street	85.7	228.8	135.2

Table 5.2: Negative log likelihood with models trained on a variable number of cameras. The test data is 200 trajectories from a single camera. The activity models in that camera are jointly learnt with different number of cameras (from 1 to 4). The last column is a baseline model trained on data whose cluster labels of trajectories are randomly assigned.

	1	2	3	4	Random
Parking Lot	120.9	121.3	122.8	123.3	425
Street	40.0	41.5	44.9	42.2	168

smoothing constraint, so it cannot jointly model the distributions of a single activity in multiple camera views. The latter simulates the case when correspondence is poor. Both alternatives result in higher negative log likelihood as shown in Table 5.1.

We also compare against models learned with trajectories from a single to all of the cameras. Models learned from a subset of the cameras will necessarily have lower negative log likelihood for trajectories within those cameras; however, they are limited to modeling joint activities only in a subset of the camera views. Our model captures joint activities in all cameras simultaneously, and only exhibits a small increase in the negative log likelihood as shown in Table 5.2.

Temporal Threshold

The temporal threshold T in Eq 5.1 determines the connectivity on the trajectory network. If a camera view A is disjoint from other views and it takes objects more than T seconds to cross the smallest gap between A and other views, then there is no way to extend a path in A to other views. If T is large and the scene is busy, the network will have too many "noisy" edges which connect two trajectories actually

as in our model, a trajectory is randomly connected with only one of the candidates in a different camera view.

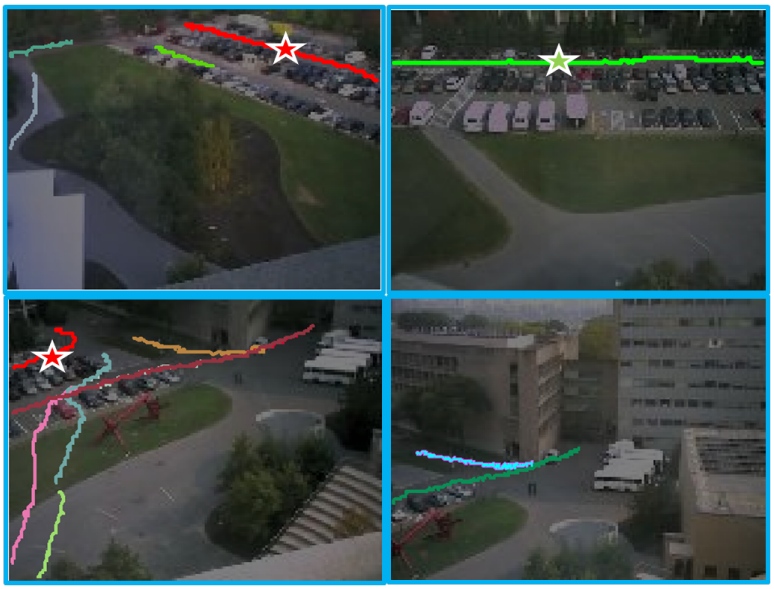


Figure 5-12: Activity models learnt in an unsupervised way help to solve the correspondence problem. We pick a query trajectory from one of the camera views and mark it using green color and a star. All the trajectories in other camera views satisfying Eq 5.1 are plotted in random colors. The red color and red stars mark the trajectories with the same activity category as the query trajectory. They are likely to correspond to the same object. So the information on activity category can dramatically reduce the search space when solving the correspondence problem.

corresponding to different objects. Under-smoothing could lead to the same activity separated into different clusters, while over-smoothing could lead to different activities joining into the same cluster. Empirically, we achieved similar results with a wide range of values for T: for the street scene data set, good results are achieved when T varies between 0 and 30 seconds; for the parking lot data set, the range of good values of T is roughly from 3 to 15 seconds because the parking lot scene is busier and the view of camera 1 is disjoint from other camera views. There is quantitative evaluation of T on a simulated data set in Section 5.4.5.

5.4.2 Correspondence

Although our activity analysis approach does not require correspondence among trajectories in different camera views, after the models of activities have been learnt in an unsupervised way, they can help to solve the correspondence problem, since if two

trajectories belong to the same activity and are connected by an edge, they are likely to correspond to the same object. For example, see Figure 5-12. We pick a query trajectory from one of the camera views and mark it using green color and a star. All the trajectories in other camera views satisfying Eq 5.1 are plotted in random colors. The red color and red stars mark the trajectories with the same activity category as the query trajectory. They are likely to correspond to the same object. So the information on activity category can dramatically reduce the search space when solving the correspondence problem.

When there are more than two cameras views, the correspondence problem is NP hard in the number of trajectories. Finding an approximate solution to this NP hard problem is not the focus of this thesis. So we demonstrate the capability of our activity models by doing correspondence among trajectories in two camera views. Given the distances between trajectories, correspondence of trajectories in two cameras views can be solved by the Hungarian algorithm [82] in polynomial time. The distance $D(a, b)$ between two trajectories a and b which are in different views is defined as follows. Each point on a trajectory is assigned to one of the activities according to Eq 5.9. Thus each trajectory j has a distribution p_j over activities. If trajectories a and b satisfy the temporal constraint Eq 5.1, then the distance between them is

$$D(a,b) = \sum_{k=1}^{K} p_a(k) log\left(\frac{p_a(k)}{p_b(k)}\right) + \sum_{k=1}^{K} p_b(k) log\left(\frac{p_b(k)}{p_a(k)}\right), \tag{5.11}$$

which is Jensen-Shannon divergence; otherwise $D(a, b) = \infty$.

We manually label 200 trajectories observed in camera view 1 and 4 in the street scene as ground truth. Our approaches achieves 93.2% correspondence accuracy on this data set. As comparison, an appearance based correspondence approach proposed in [148] only has an accuracy of 79.8% on this data set.

5.4.3 Abnormality Detection

In Figure 5-13 and 5-14 we plot some trajectories with low data likelihoods, which have been normalized by the length of trajectories, and are detected as abnormality

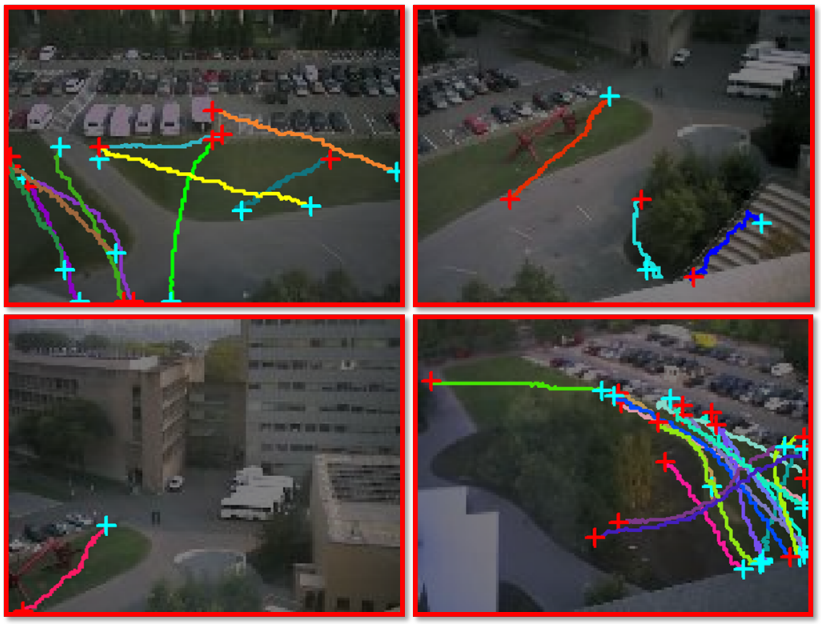


Figure 5-13: Some trajectories with low likelihoods from a parking lot scene. Random colors are used to distinguish individual trajectories. In order to indicate the moving direction of a trajectory, the starting and ending points of a trajectory is marked by + in red and cyan colors. Many of them are pedestrians walking on the grass field.

from the parking lot scene and the street scene. All of the trajectories are sorted by abnormality and the top 30 are shown. Some very short trajectories most likely caused by tracking errors are not shown here. In the parking lot scene, most of the detected abnormal trajectories are pedestrians walking on the grass field. In the street scene, abnormal activities include pedestrians walking on the grass fields, pedestrians crossing the street, pedestrians walking in the middle of the street, and vehicles moving along a wrong lane.

5.4.4 Computational Cost

Running on a computer with 2GHz CPU, it takes about two hours to learn the activity models from $22,951$ trajectories of the parking lot data set and 40 minutes to learn the activity models from $14,985$ trajectories from the street scene. When the activity models are learnt and fixed, it takes less than 0.03 second to compute the likelihood of a trajectory in order to detect abnormality, and it is much faster to label a new trajectory as some activity category.

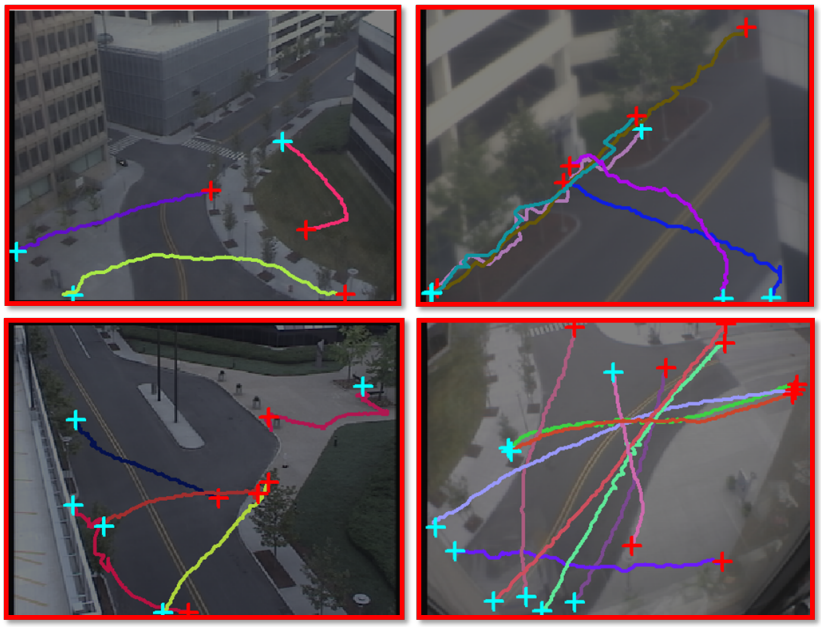


Figure 5-14: Some trajectories with low likelihoods from a street scene. Random colors are used to distinguish individual trajectories. In order to indicate the moving direction of a trajectory, the starting and ending points of a trajectory is marked by + in red and cyan colors. Abnormal activities include pedestrians walking on the grass fields, pedestrians crossing the street, pedestrians walking in the middle of the street, and vehicles moving along a wrong lane.

5.4.5 Simulated Data

In order to quantitatively evaluate our algorithm, we simulate data used as the ground truth. As shown in Figure 5-15 (a), we choose a scene which covers almost the same area of the street scene we used in Section 5.4.1. On a satellite image, we manually draw the fields covered by four camera views. The fields are convex four-sided polygons. These fields are converted to a standard camera view in size of 240×360 through projective transformation. The views observed by four cameras are shown in Figure 5-15 (b). We manually draw the central lines of eight paths on the satellite image (Figure 5-16 (a)) and simulate 8000 trajectories. We assume trajectories have almost the same speed, since speed does not play an important role in our algorithm. The starting points of trajectories are generated sequentially as follows.

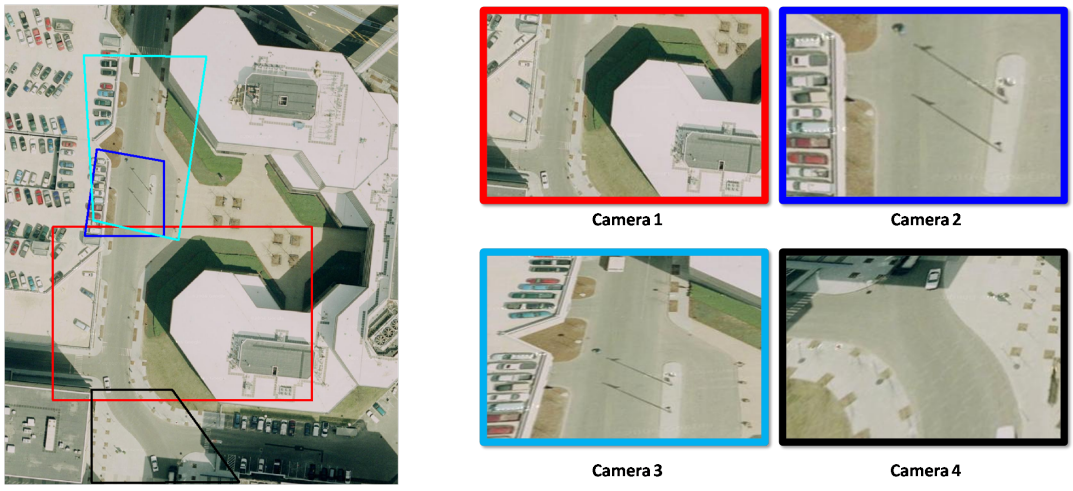
$$t_{s(i+1)} = t_{si} + \Delta t_{i+1}, \tag{5.12}$$

$$\Delta t_{i+1} \sim Exponential(\lambda). \tag{5.13}$$

t_{si} is the starting time of the ith trajectory. The temporal difference $\Delta t_{i+1} = t_{s(i+1)} - t_{si}$ between two successive trajectories is sampled from a exponential distribution with mean λ. A trajectory i is randomly assigned to one of the eight predefined activities, k. Trajectory i samples the location of its starting point from a Gaussian distribution centered at the starting point of path k with variance σ_1 ($\sigma_1 = 5$ in this simulation). Then i samples the remaining points sequentially with the velocity specified by path k and being added to Gaussian noise with variance σ_2 ($\sigma = 2$ in this simulation). The simulated trajectories in the global views and each of the four camera views are shown in Figure 5-16 (b) and (c).

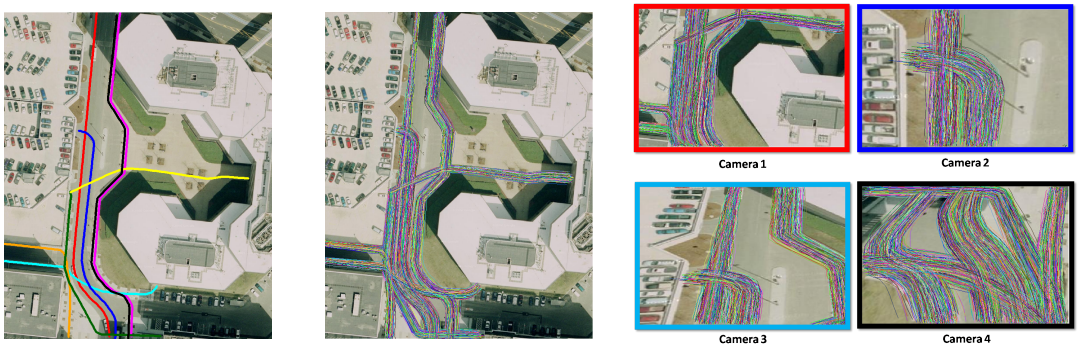
Learning activity models and clustering trajectories

λ is the parameter controlling how busy the scene is. When λ is smaller, more objects co-exist in the scene at the same time, which means that there are more edges on the trajectory network and it is harder for our algorithm to jointly learn the models of activities in different camera views. In our experiments, we change the value of



(a) (b)

Figure 5-15: (a) The global view of the scene where the data is simulated and the fields covered by four camera views. The fields are marked by polygons. Colors are used to distinguish cameras. (b) The views observed by cameras after projective transformation.



(a) (b) (c)

Figure 5-16: (a) The central lines of eight paths manually drawn in the scene. They are distinguished by colors: 1 (red), 2 (blue), 3 (dark green), 4 (magenta), 5 (black), 6 (cyan), 7 (yellow), and 8 (orange). (b) Trajectories generated from the eight paths. (c) Trajectories observed in four cameras.

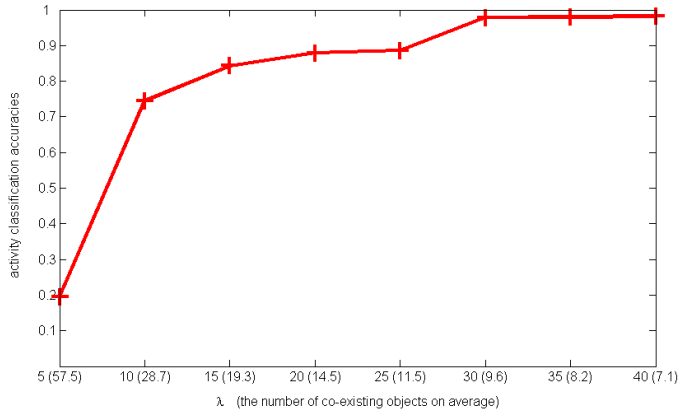


Figure 5-17: The accuracies of classifying trajectories into different activities when λ takes different values and T is fixed as 0. Our approach achieves good performance when the scene is fairly busy.

λ from 5 seconds to 40 seconds. Based on the speed set for this experiment, the time an object spent to pass through a path varies from 170 seconds to 410 seconds. It depends on the length of the path. When λ takes values from 5 seconds to 40 seconds, the averaged number of objects co-existing in the scene varies from 57.5 to 7.1 (see Figure 5-17). After the trajectories are clustered by our algorithm, we manually specify each of the eight clusters as an activity category, so each trajectory is assigned an activity label by our algorithm. By comparing with the ground truth, the accuracy of activity classification is computed. The accuracies when choosing different λ values are shown Figure 5-17. The accuracy is high ($> 97.8\%$) when $\lambda \geq 30$ seconds. The models of activities in a single global view and four camera views learnt from the simulated data when $\lambda = 30$ seconds are shown in Figure 5-18 and 5-19. Notice that when $\lambda = 30$ seconds, if we randomly sample a time point, there are around 9.6 objects co-existing in the scene on average. Each trajectory is connected to 12.4 trajectories by edges on the network on average. When λ decreases, some trajectories of different activities merge into one cluster. When $\lambda = 5$ seconds, the scene is very busy (there are 57.5 objects co-existing in the scene on average), all of the trajectories are merged into one cluster and our algorithm cannot learn any useful activity models from this data set. Each trajectory is connected to 73.0 trajectories by edges on the network on average.

We further look into the structure of the trajectory network constructed according to the temporal extents of trajectories in the data set when $\lambda = 30$ seconds. Figure 5-20 shows the number of edges which are related to different combinations of activities and camera views according to the ground truth. The entry of (k_1, k_2) on the table of camera views i_1 and i_2 are the number of edges connecting two trajectories, one of which is in camera view i_1 and belongs to activity k_1, and the other of which is in camera view i_2 and belongs to activity k_2. There are six 8×8 tables. As we mentioned earlier, the edges on the trajectory network indicate possible correspondence candidates based on the temporal extents of trajectories. If the correspondence can be solved just using temporal information, all of the nonzero numbers in the table will be on the diagonal. Actually many off diagonal entries have nonzero numbers, which indicate false correspondences, whose ambiguity cannot be solved by only using temporal information. The ratio between the numbers of edges on diagonal and off diagonal is 0.2732. This ratio can be understood as a signal-to-noise ratio in some sense. There are many more false correspondences than true correspondences. However, these false correspondences almost uniformly distribute among different combinations of activities and work as background noise. So if a trajectory of activity k_1 is connected with another trajectory of activity k_2, k_2 is more likely to be the same as k_1 than any one of the other activties. When the scene is busier, the signal-to-noise ratio is lower. When $\lambda = 5$, the ratio is 0.1692 and our algorithm fails. Notice that the signal-to-noise ratio is $1/7 = 0.1429$, if trajectories are randomly connected without using any temporal information.

Figure 5-21 plots the classification accuracies when λ is fixed as 40 seconds and the temporal threshold T in Eq 5.1 changes from 0 seconds to 300 seconds. The results stay at a high accuracy when T varies in a large range between 0 second and 40 seconds. There is some interesting correlation between Figure 5-17 and Figure 5-21. The performance of our algorithm drops if there are too many edges on the trajectory network, which means that the "signal-to-noise" ratio is low. The number of edges increases if λ decreases, which means that the scene is busier and there are more objects co-existing in the scene, or T increases. From Figure 5-17, when T is

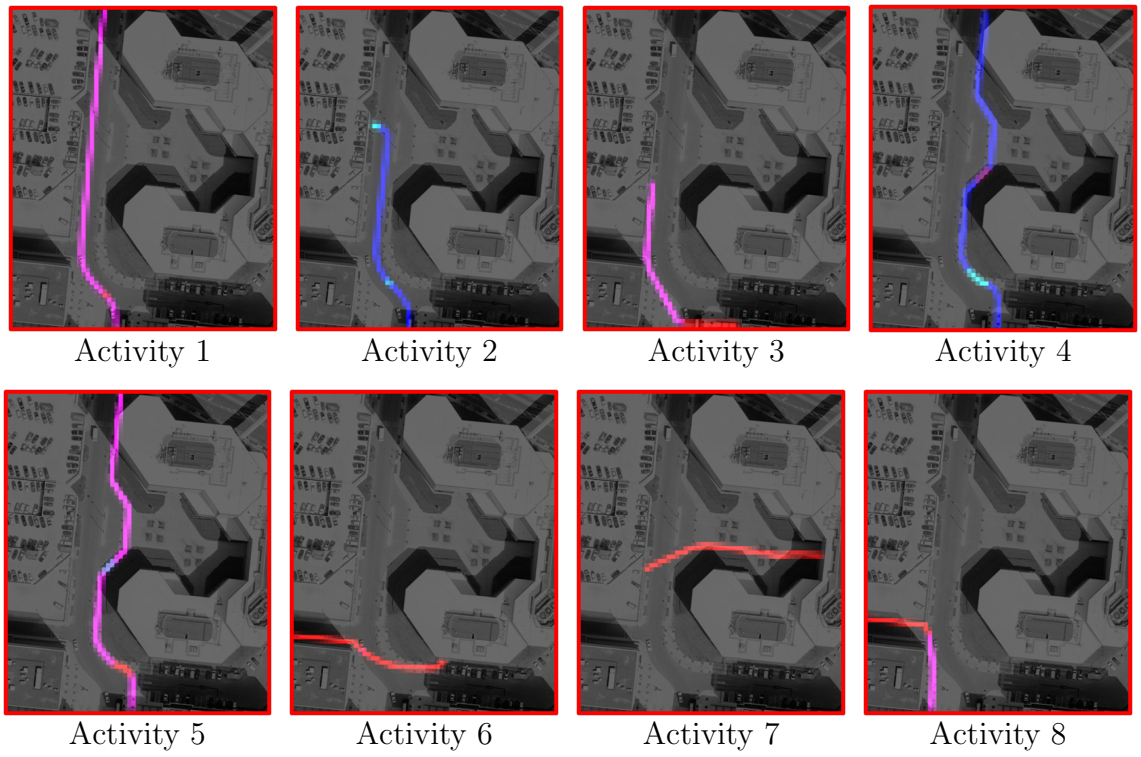


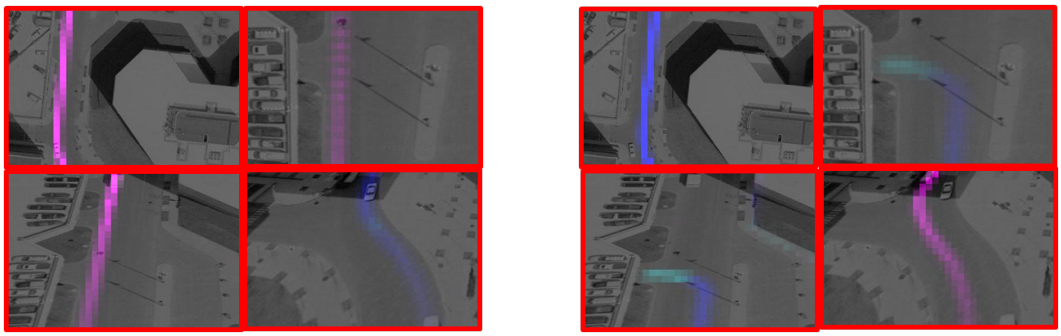
Figure 5-18: Distributions of activity models in a single global views learnt from the simulated data. The meaning of colors is the same as Figure 5-4.

fixed at 0 second, $\lambda = 30$ seconds seems to be a turning point on the accuracy curve. On average, there are around two more objects co-occurring when $\lambda = 30$ seconds compared with $\lambda = 40$ seconds. From Figure 5-21, when λ is fixed at 40 seconds, $T = 40$ seconds seems to be a turning point on the accuracy curve. Compared with $T = 0$ second, the temporal window in Eq 5.1 extends for $2 \times T = 80$ seconds. In 80 seconds, there are around two more objects appearing on average when $\lambda = 40$ seconds. So there are approximately the same number of edges under two settings. For $(\lambda = 30, T = 0)$, on average each trajectory is connected to 12.4 trajectories by edges on the network, and for $(\lambda = 40, T = 40)$ this number is 13.0.

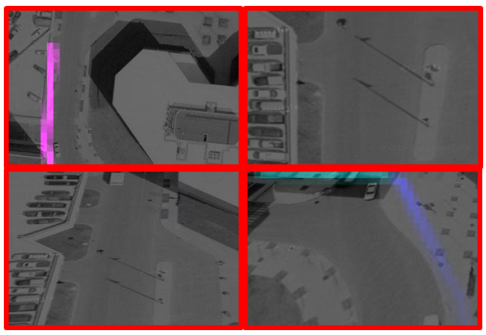
Using activity models to solve the correspondence problem

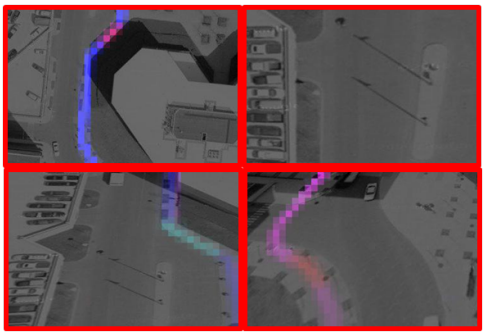
As mentioned in Section 5.4.2, the learnt activity models can help to solve the correspondence problem. We evaluate the performance on simulated data.

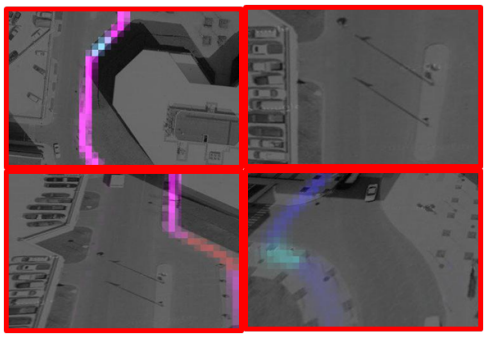
We choose camera views 1 and 4 which are shown in Figure 5-15. 1000 trajecto-

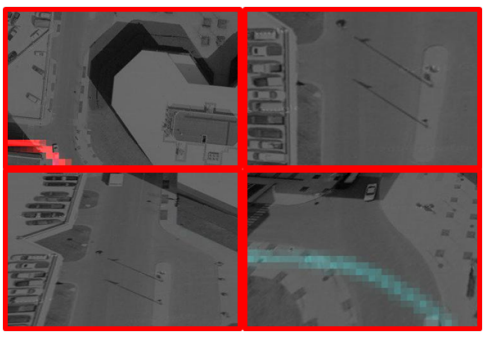

Activity 1

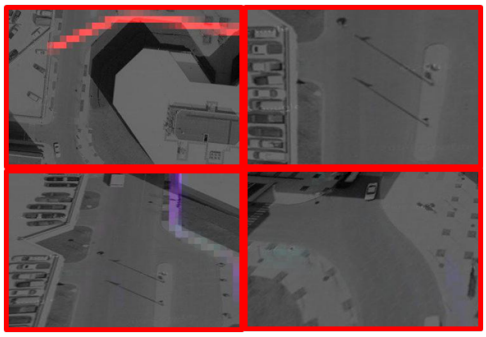
Activity 2


Activity 3


Activity 4


Activity 5


Activity 6


Activity 7

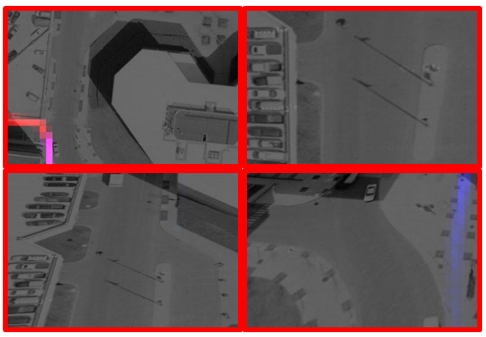

Activity 8

Figure 5-19: Distribution of activity models in four camera views learnt from the simulated data. The meaning of colors is the same as Figure 5-4.

Camera 1 / Camera 2

	1	2	3	4	5	6	7	8
1	1747	666	0	0	0	0	3	0
2	698	1657	0	0	0	0	3	0
3	599	615	0	0	0	0	3	0
4	811	719	0	0	0	0	1	0
5	811	753	0	0	0	0	2	0
6	485	477	0	0	0	0	0	0
7	898	924	0	0	0	0	11	0
8	529	483	0	0	0	0	0	0

Camera 1 / Camera 3

	1	2	3	4	5	6	7	8
1	1107	638	0	1036	1011	0	502	0
2	984	1176	0	973	1114	0	477	0
3	780	598	0	814	860	0	298	0
4	1124	692	0	1361	1040	0	539	0
5	1124	733	0	1055	1374	0	581	0
6	687	452	0	639	776	0	120	0
7	1199	905	0	1105	1230	0	1439	0
8	724	465	0	678	799	0	160	0

Camera 1 / Camera 4

	1	2	3	4	5	6	7	8
1	1831	948	927	881	914	870	0	821
2	945	1779	950	858	850	770	0	746
3	769	594	1796	714	633	656	0	586
4	865	888	1028	1875	975	966	0	897
5	948	939	1014	970	1900	811	0	810
6	550	459	763	654	509	1522	0	475
7	1015	961	1184	1074	951	938	0	884
8	609	484	737	597	579	503	0	1457

Camera 2 / Camera 3

	1	2	3	4	5	6	7	8
1	1703	447	0	662	833	0	260	0
2	744	1292	0	753	701	0	201	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	3	0	0	3	0	0	6	0
8	0	0	0	0	0	0	0	0

Camera 2 / Camera 4

	1	2	3	4	5	6	7	8
1	588	609	649	610	642	433	0	459
2	579	558	629	551	588	571	0	520
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	2	2	0	1	0	2
8	0	0	0	0	0	0	0	0

Camera 3 / Camera 4

	1	2	3	4	5	6	7	8
1	855	791	937	910	840	762	0	747
2	551	539	611	536	559	548	0	504
3	0	0	0	0	0	0	0	0
4	835	693	886	881	767	697	0	694
5	774	849	972	759	889	848	0	796
6	0	0	0	0	0	0	0	0
7	323	224	525	357	228	274	0	291
8	0	0	0	0	0	0	0	0

Figure 5-20: The number of pairs of simulated trajectories, which are in different camera views, belong to activities i and j $(i, j = 1, \ldots, 8)$, and whose temporal extents are close. Here, $\lambda = 6$ and $T = 0$.

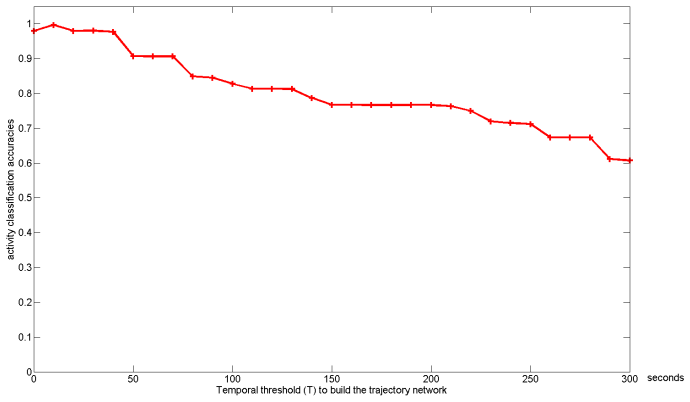


Figure 5-21: The accuracies of classifying trajectories into different activities when the temporal threshold T change from 0 to 300 seconds. Here, $\lambda = 40$ seconds.

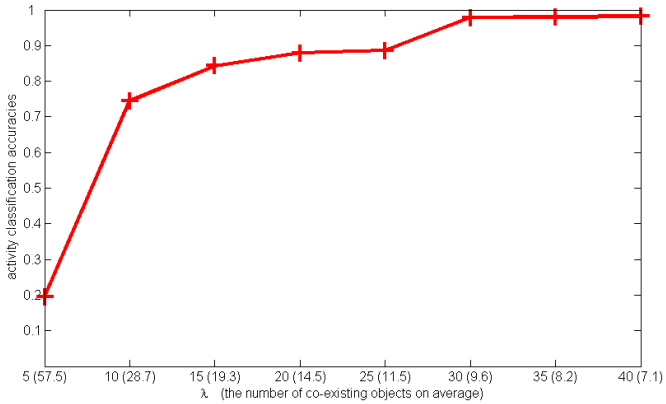


Figure 5-22: Accuracies of correspondence on the simulated data. Solve the correspondence problem of trajectories observed in the views of camera 1 and camera 4. λ varies from 5 seconds to 40 seconds.

ries are simulated and they are not in the data set of 8000 trajectories used to learn the activity models. 1000 trajectories are observed in camera view 1 and around 880 trajectories are observed in camera view 4. Some trajectories of activity 7 (see Figure 5-16) observed in camera view 1 have no corresponding trajectories in camera view 4. We simulate different sets of data by changing the parameter λ. The accuracies of correspondence are plotted in Figure 5-22. It achieves very good correspondence accuracy (higher than 97%) when $\lambda \geq 30$. The accuracy drops when the scene is busier because of two reasons: (1) the activity models are not well learnt; (2) some objects of the same activity exist around the same time so they cannot be distinguished by activity categories and temporal extents.

5.5 Discussion

The performance of our algorithm depends on the number of edges in the trajectory network. If on average a trajectory is connected to a large number of other trajectories, which means that there are many false correspondence candidates, the models of activities cannot be well learnt. The number of edges increases because of two reasons: the scene is busy or the temporal threshold T is large. A large T allows a large transition gap between camera views. So if a scene is busy, the transition gaps between cameras has to be small, which limits the topology of camera views in some

sense. In this work, only temporal information is used to build the trajectory network. That is why the algorithm is sensitive to how busy the scene is. Some other features, such as appearance, can also be used to eliminate some edges. If two objects observed in different camera views are poorly matched by appearance, their trajectories are not connected by an edge even though their temporal extents are close. Thus activity models may be well learnt even in a busy scene. However, in this case the problem of matching appearance across camera views has to be addressed. It is a direction of our future study.

In our clustering method, the number of clusters K has to be manually chosen. Some nonparametric models such as Hierarchical Dirichlet Processes [140] can learn the number of clusters from data. They could be used to improve our clustering method in the future work.

Chapter 6

Tractography Segmentation

Having explored applications to activity analysis in far-field visual surveillance, we will present how the technology of learning motion patterns can also be applied to tractography segmentation from DT-MRI in this chapter. As explained in Section 1.2, DT-MRI is used to visualize and quantify the organization of white matter in the human brain in vivo. Tractography connect local diffusion measurements into global fiber trajectories. In neurological studies of white matter using tractography it is often important to anatomically cluster fiber trajectories into meaningful bundles. This technology is called tractography segmentation. Clustering fiber trajectories has some similarity with clustering trajectories of objects in visual surveillance. So our approach developed for trajectory analysis in a single camera view can also be applied to tractography segmentation.

We use Dual-HDP to cluster fibers and learn the models of bundles from a training set without supervision. A fiber is treated as a document. The points on a fiber trajectory are treated as words. The 3D space of the brain is quantized into voxels. If we need to analyze a new subject, we use Dynamic Dual-HDP to cluster fibers from new subjects. The models of bundles learnt from training data are used as priors, and models are adapted to new data. Optionally, if the symmetry across hemispheres is considered, we can do bilateral clustering as in [107]. Let $u_{ji} = (x_{ji}, y_{ji}, z_{ji})$ be the 3D coordinate of point i on fiber j. Assuming that the brain is aligned and $x = 0$ is the midsagittal plane, we set observed 3D coordinates as $\vec{u}_{ji} = (|x_{ji}|, y_{ji}, z_{ji})$ ignoring

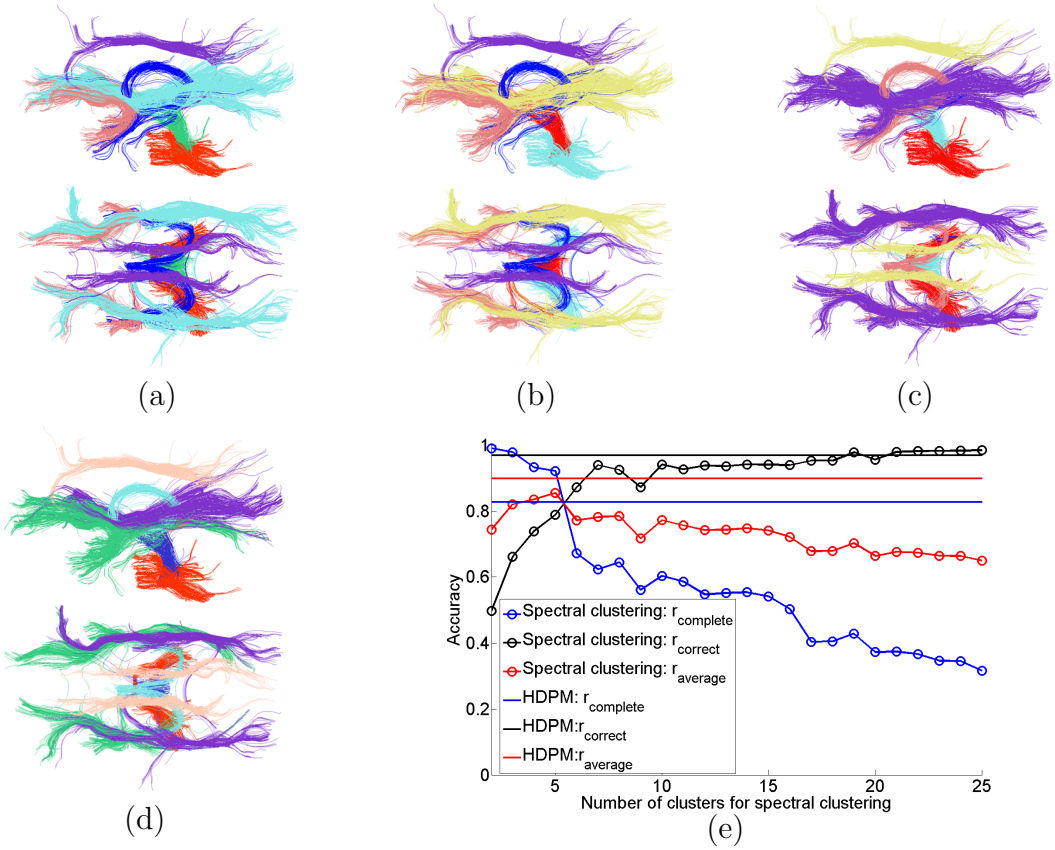


Figure 6-1: Compare the results of two clustering approaches with the ground truth on a data set with 3,152 fibers. Two views are plotted for each result. (a) Ground truth. (b) Our approach. (c) Spectral clustering when the number of clusters is 6. (d) Spectral clustering when number of clusters is 7. (e) The accuracies of completeness and correctness of spectral clustering and our approach (HDPM).

the signs of x coordinates. Thus, learnt models of bundles are symmetric to the midsagittal reflection.

6.1 Experimental Results

We evaluate our approach on multiple data sets. The spatial range of the whole brain is roughly $200 \times 200 \times 200$. The size of voxels is $11 \times 11 \times 11$. We do bilateral clustering. Running on a computer with 3GHz CPU, it takes around one minute to cluster 1,000 fibers and around four hours to cluster 60,000 fibers.

The first data set has 3,152 fibers with ground truth. They are manually labeled to six anatomical structures. Figure 6-1 (a)-(d) plots the clustering results of our

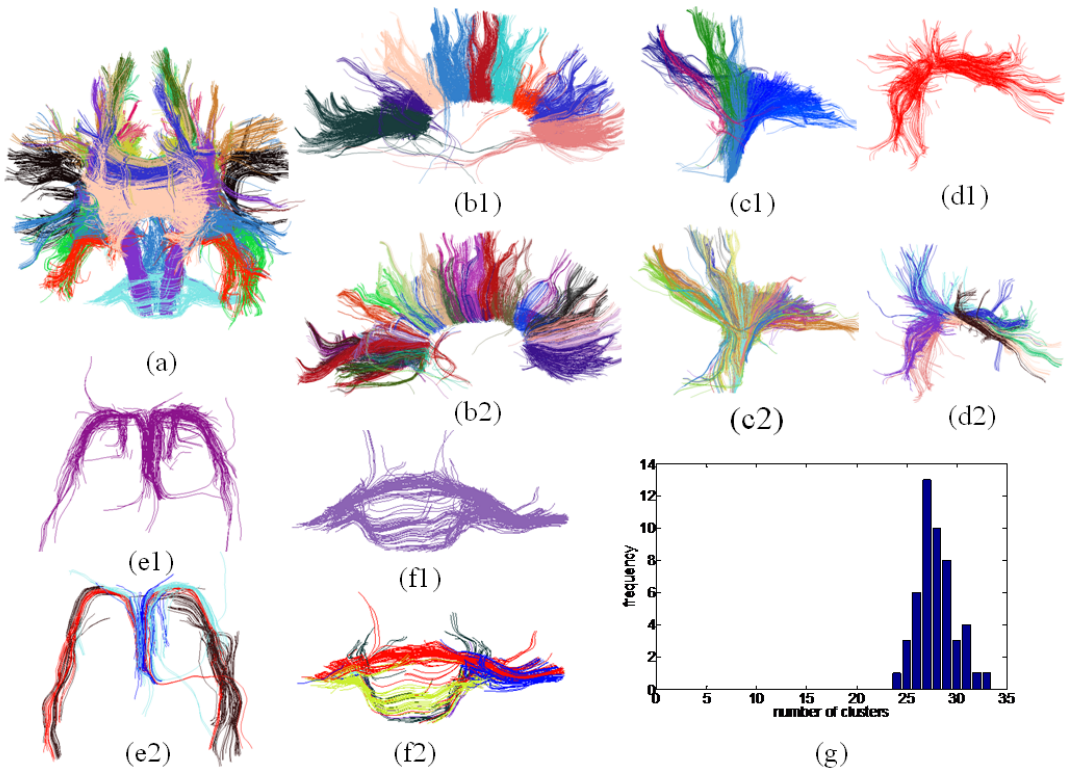
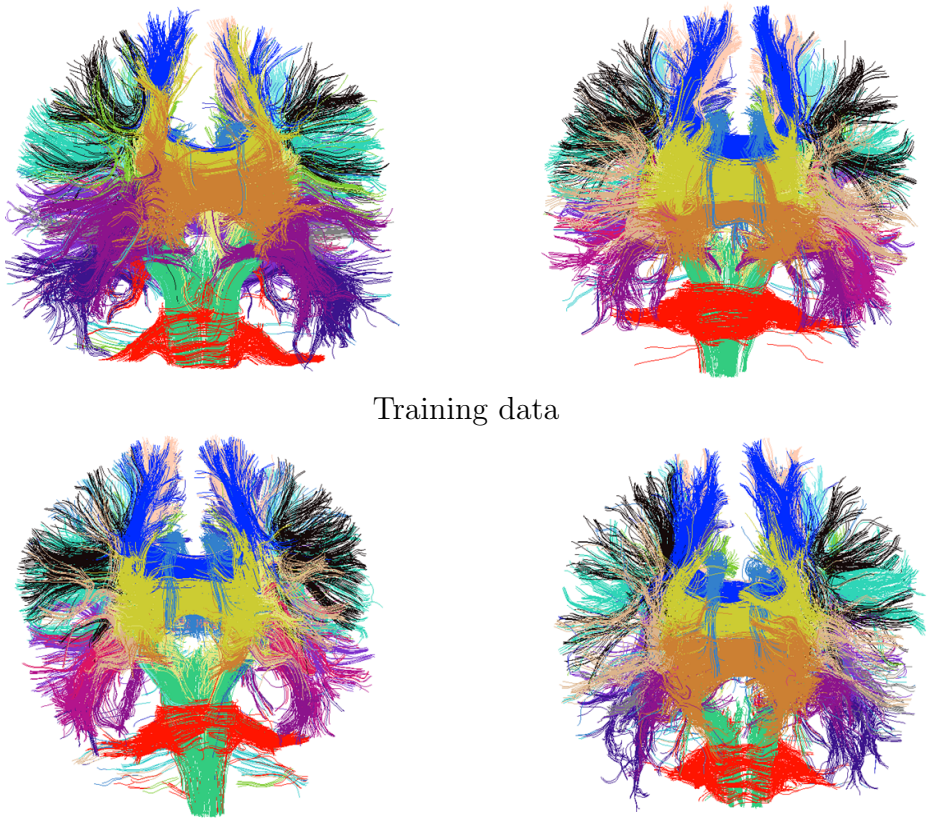


Figure 6-2: Compare results of our approach and the approach proposed in [107], in which experts manually merged the clusters from spectral clustering to obtain anatomical structures. (a) Clustering all the fibers using our approach. (b1)-(f1) show the obtained anatomical structures by merging clusters from our approach (totally 27 clusters). (b2)-(f2) show the obtained anatomical structures by merging clusters from spectral clustering (totally 200 clusters). Colors are used to distinguish clusters. (g) plots the frequency of the numbers of clusters learnt by our approach when running 50 trials of Gibbs sampling with random initializations.

approach and a spectral clustering approach, compared with the ground truth. Colors are used to distinguish clusters. Since clusters may be permuted in different results, the meaning of colors is not consistent across different results. The spectral clustering approach uses the mean of closest distances as the distance measure, which was found the most effective in previous studies [101, 107]. The clustering result of our approach is close to the ground truth. Although the correct number of clusters has been set, two anatomical structures are merged in the result of the spectral clustering approach. A few outlier fibers form a small cluster. As the number of clusters increases to 7, the two anatomical structures still cannot be separated, instead, another structure splits into two clusters.

There are two important aspects, called *correctness* and *completeness*, to be considered when comparing a clustering result with the ground truth [101]. Correctness implies that fibers of different anatomical structures are not clustered together. Completeness means that fibers of the same anatomical structures are clustered together. Putting all the fibers into the same cluster results in 100% completeness and 0% correctness. Putting every fiber into a singleton cluster results in 100% correctness and 0% completeness. To measure correctness, we randomly sample $5,000$ pairs of fibers which are in different anatomical structures according to the ground truth and calculate the accuracy ($r_{correct}$) that they are also in different clusters according to the clustering result. To measure completeness, we randomly sample $5,000$ pairs of fibers which are in the same anatomical structures and calculate the accuracy ($r_{complete}$) that they are also in the same clusters. $r_{average} = (r_{correct} + r_{complete})/2$ is also computed. The accuracies of our approach and spectral clustering are plotted in Figure 6-1 (e). As we increase the number of clusters from 2 to 25, the correctness of spectral clustering increases and its completeness decreases. Its best $r_{average}$ is found when the number of clusters is five, which is close to the ground truth, and it is lower than $r_{average}$ of our approach. The correctness of our approach is almost consistently better than spectral clustering until spectral clustering chooses more than 20 clusters. The completeness of our approach is significantly better than spectral clustering when the number of clusters of spectral clustering is larger than 5.

We compare our approach with the approach proposed in [107] on a larger data set with $12,420$ fibers. In [107], fibers were first grouped into a large number of clusters (200) and then experts merged these clusters to obtain anatomical structures. In this data set there are 10 anatomical structures. Our approach clusters these fibers to 27 clusters. We also manually merge them to these 10 anatomical structures, however its takes much less effort than [107] since the number of clusters is smaller. Figure 6-2 shows some of the anatomical structures obtained by the two approaches. 83.2% fibers have consistent anatomical labels according to the two results. To evaluate how our approach is sensitive to initialization, we run 50 trials of Gibbs sampling with random initializations. Figure 6-2 (g) plots the frequency of the numbers of clusters



Training data

Testing data

Figure 6-3: Cluster fibers across multiple subjects.

learnt from data.

Figure 6-3 shows the results of clustering fibers across multiple subjects. The training data has $63,751$ fibers of two subjects. The models of bundles are learnt from all these fibers. The testing data has $61,572$ fibers of two subjects.

Chapter 7

Limitations and Future Work

Previous chapters show that our approaches achieve promising results on activity analysis in far-field visual surveillance and tractography segmentation in medical imaging. This chapter discusses the limitations of this thesis work and future directions we are interested in investigating.

7.1 Low-level features

For activity analysis in visual surveillance, we use the locations and moving directions of motions as low-level visual features since they are more reliable and easier to compute in far-field settings. While we have demonstrated the effectiveness of our models in a variety of visual surveillance tasks, including more complicated features is expected to further boost the discrimination power of our models. For example, if a pedestrian is walking along the path of vehicles, just based on positions and moving detections his motions cannot be distinguished from those of vehicles and this activity will not be detected as abnormality. If a car drives extremely fast, it will not be detected as abnormal either. Other features, such as appearance and speed, are useful in these scenarios. However, the size of the codebook will increase as more features are included and thus the computational cost will increase.

7.2 Design of "Documents" for Activity Analysis in Crowded Scenes

In Chapter 3, the information on the co-occurrence of moving pixels is critical for our methods to separate atomic activities. One moving pixel tends to be labeled as the same atomic activity as other moving pixels happening around the same time. This information is encoded into the design of video clips as documents. We uniformly divide the long video sequence into short video clips. This "hard" division may cause some problems. The moving pixels happening in two successive frames might be divided into two different documents. By intuition, one moving pixel should receive more influence from those moving pixels closer in time. However, in our models, moving pixels that fall into the same video clip are treated in the same way, no matter how close they are. In [149], we proposed a model allowing random assignment of words to documents according to some prior which encodes temporal information. If two moving pixels are temporally closer in space, they have a higher probability to be assigned to the same documents.

7.3 Temporal Logic

In this work, we do not model activities, interactions and global behaviors with complicated temporal logic. However the atomic activities and global behaviors learnt by our framework can be used as units to model more complicated activities, interactions and global behaviors considering temporal logic using HMM [108, 19], dynamic Bayesian network [62], Petri nets [47] and temporal interval logic [3].

7.4 Jointly Model Activities and Appearance in Multiple Camera Views

As explain in Chapter 5, to the best of our knowledge, we are the first to use activity models to do correspondence. Actually there are some potential benefits if we jointly

model activities and appearance under a hierarchical Bayesian model. Under our current model, the trajectories on the network are connected simply based on temporal information. If on average a trajectory is connected to a large number of other trajectories, which means that there are many false correspondence candidates, the models of activities cannot be well learnt. Some edges on the trajectory network can be removed through matching the appearance of objects. This can make the network sparser and learning easier. On the other hand, objects have different appearance in different camera views. We want to learn an appearance transform function across camera views [65]. Activity models can provide some training examples to learn this transform function through solving the correspondence problem. It will be very interesting to jointly model activities and the appearance transformation functions under a more complicated hierarchical Bayesian model.

7.5 Guide Tractography Segmentation

Our tractography segmentation approach is unsupervised. However, our Bayesian models are very flexible to include knowledge from experts as priors. In future work, we plan to incorporate anatomical information in the model to guide tractography segmentation. For example, experts first label some regions as initialization of anatomical structures and then our Bayesian models expand the regions through clustering fiber trajectories.

7.6 Inference

In this work, Gibbs sampling is used to do inference on hierarchical Bayesian models. Although the efficiency of Gibbs sampling has been improved by integrating out some hidden variables, the inference is still slow for some applications and it lacks theoretical justification on the convergence of Gibbs sampling. For example, under Dual-HDP it takes 12 hours to cluster moving pixels and video clips within an 1.5 hours video sequence and it takes 6 hours to cluster $40,453$ trajectories of objects

collected over one week. Recently some more efficient inference approaches, such as variational inference [17], and parallel sampling [4], have been proposed and applied to Dirichlet process mixture models and HDP. In the future work, we will study how to improve the inference of our models using these schemes.

Chapter 8

Conclusion

In this thesis, we use hierarchical Bayesian models to learn motion patterns in visual surveillance and medical imaging based on the co-occurrence of feature values. Different approaches are proposed for activity analysis under different scenarios. It depends on the number of camera views and crowdedness of scenes. Some technology of learning motion patterns developed in visual surveillance can also be used to tractography segmentation which clusters fibers generated from DT-MRI into anatomically meaningful bundles. Three hierarchical Bayesian models, Dual-HDP, dynamic Dual-HDP and a trajectory network based hierarchical Bayesian model, are proposed for activity analysis in crowded scenes without tracking objects, trajectory analysis in a single camera view and in multiple camera views, and tractography segmentation. Dual-HDP is used to jointly model atomic activities and global behaviors in crowded scenes, to cluster trajectories in a single camera view, and to cluster fibers in tractography segmentation. Dynamic Dual-HDP is used to update the models of activities over time and cluster fibers of new subjects. The trajectory network based hierarchical Bayesian model is used to cluster trajectories in multiple camera views.

Under a Bayesian framework, activity analysis and tractography segmentation tasks are formulated in a principled way. While many existing activity analysis approaches relied on predefined rules or simple probabilistic models and had difficulty modeling complicated activities, our hierarchical Bayesian models structure dependency among a large number of variables to model complicated activities in crowded

and sparse scenes with a single camera view and with multiple camera views. Various knowledge and constraints can be nicely added into a Bayesian framework as priors. As examples, by adding a smoothness constraint as a prior, our model can cluster trajectories in multiple cameras. Using activities models learned from historical data as priors, we can dynamically update the models of activities over time. These tasks are difficult to nonBayesian approaches. Our nonparametric Bayesian models automatically learn the numbers of clusters of moving pixels, video clips and trajectories driven by data instead of manually specifying them as many existing approaches did. All our approaches are unsupervised requiring less human labeling effort.

If the scene is crowded and it is difficult to track objects, we model activities directly from moving pixels without tracking objects. Atomic activities and complicated global behaviors are jointly modeled at different hierarchical levels using Dual-HDP based on the temporal co-occurrence of feature values. It has a better performance than modeling atomic activities and global behaviors separately or sequentially. Co-occurring activities can be separated without supervision. Moving pixels are clustered into atomic activities and a long video sequence is segmented into different types of global behaviors. With atomic activities as middle-level representations, we can query interactions of interest between object in an easy way. Abnormal video clips and moving pixels are detected with probabilistic explanation. This approach has the limitation that it does not work well if all the video clips have similar combinations of atomic activities especially when the monitored area is large and activities in the scene have more temporal overlap.

If the scene is sparse, we first track objects and then cluster trajectories of objects into activity categories and learn the models of paths using Dual-HDP based on the identity co-occurrence of feature values. Daul-HDP has low space complexity than many distance based trajectory clustering methods since it does not require computing the similarity matrix. The time complexity of our approach can be further improved by using variational inference and parallel computing in the future work. It is more robust to tracking errors. Dynamic Dual-HDP uses the models learnt from historical data as priors to update the models of activities over time. It can better explain

activities at different time. It clusters trajectories incrementally and does not have to keep old data in the memory. So it has lower space and time complexities than Dual-HDP. It property is very important if we need to cluster a huge visual surveillance data set collected over months or even years. These approaches does not work well when the scene are crowded with many occlusions and the identity co-occurrence information cannot be reliably obtained.

In order to monitor activity in a large area, video streams from multiple camera views have to be used. By adding a smoothness constraint on the distributions of trajectories over activities as a prior, our hierarchical Bayesian model clusters trajectories in multiple camera views without tracking objects across camera views. It uses both temporal co-occurrence and identity co-occurrence of feature values. It does not require inference on the topology of camera views and does not require solving the challenging correspondence problems. It assumes that the topology of camera views is arbitrary. The camera views can have overlap or no overlap. After the models of activities have been learned without supervision, they can be used to match trajectories of the same object observed in different camera views.

Some of these techniques developed in visual surveillance can be applied to medical imaging where some issues related to learning motion patterns arise. As an example, Dual-HDP and dynamic Dual-HDP are used to cluster fiber trajectories in tractography segmentation. Under dynamic Dual-HDP, models of bundles learned from training data are used as priors to cluster fibers from new subjects. Comparing with existing approaches, our approach has advantages that it can cluster larger scale data sets without subsampling them and automatically decides that number of bundles.

Our approaches are evaluated on multiple large scale visual surveillance and medical imaging data sets. They achieve promising results compared with existing approaches.

These hierarchical Bayesian models exploring co-occurrence of low-level features at multiple levels can also be applied to other fields such as language processing, object recognition and scene categorization.

Appendix A

Gibbs Sampling for Dual-HDP

In the appendix, we will explain how to do Gibbs sampling in the Dual-HDP model as described in Section 3.2.6. The sampling procedure is implemented in two steps. In the first step, given the cluster assignment $\{c_j\}$ of documents is fixed, we sample the word topic assignment $\mathbf{z}$, mixtures π_0 and π_c on topics. It follows the Chinese Restaurant Process (CRP) Gibbs sampling scheme as described in [140], but adding more hierarchical levels. In CPR, restaurants are documents, customers are words, and dishes are topics. All the restaurants share a common menu. The process can be briefly described as following (see more details in [140]).

- When a customer i comes to restaurant j, he sits at one of the existing tables t, and eats the dishes served on table t, or takes a new table t_{new}.

- If a new table t_{new} is added to restaurant j, it orders a dish from the menu.

Since we are modeling clusters of documents, we introduce "big restaurants", which are clusters of documents. The label of document cluster c_j associates restaurant j to big restaurant c_j. The CRP is modified as following.

- If a new table t_{new} needs to be added in restaurant j, we go to the big restaurant c_j and choose one of the existing big tables r in c_j. t_{new} is associated with r, and serves the same dish as r.

- Alternatively, the new table t_{new} may take a new big table r_{new} in the big restaurant c_j. If that happens, r_{new} orders a dish from the menu. This dish will be served on both r_{new} and t_{new}.

Following this modified CRP, given $\{c_j\}$, $\mathbf{k}$, π_0 and $\{\pi_c\}$ can be sampled. It is a straightforward extension of the sampling scheme in [140] to more hierarchical levels.

In order to sample $\{c_j\}$ and generate the clusters of documents, given $\mathbf{z}$, π_0, and $\{\pi_c\}$, we add an extra process.

- When a new restaurant j is built, it needs to be associated with one of the existing big restaurants or a new big restaurant needs to be built and associated with j. It is assumed that we already know how many tables in restaurant j and dishes served at every table.

Let m^t_{jk} be the number of tables in restaurant j serving dish z and $m^t_{j\cdot}$ be the number of tables in restaurant j. To sample c_j, we need to compute the posterior,

$$p(c_j|\{m^t_{jk}\}, \mathbf{c}^{-j}, \{\pi_c\}, \pi_0) \propto p(\{m^t_{jk}\}|c_j, \mathbf{c}^{-j}, \{\pi_c\}, \pi_0)p(c_j|\mathbf{c}^{-j}, \{\pi_c\}, \pi_0) \tag{A.1}$$

where $\mathbf{c}_{-j}$ is the cluster labels of documents excluding document j. c_j could be one of the existing clusters generated at the current stage, i.e. $c_j \in \mathbf{c}^{old}$. In this case,

$$p(m^t_{jk}|c_j, \mathbf{c}^{-j}, \{\pi_c\}, \pi_0) = \quad p(m^t_{jk}|\pi_{c_j}) = \binom{m^t_{j\cdot}}{m^t_{j1} \cdots m^t_{jK}} \prod_{k=1}^{K} \pi_{c_j k}^{m^t_{jk}} \tag{A.2}$$

where K is the number of word topics allocated at the current stage. And,

$$p(c_j|\{\pi_c\}, \mathbf{c}^{-j}, \pi_0) = \frac{n_{c_j}}{M - 1 + \mu} \tag{A.3}$$

where n_{c_j} is the number of documents assigned to cluster c_j.

c_j could also be a new cluster, i.e. $c_j = c^{new}$. In this case,

$$
\begin{aligned}
p(\{m_{jk}^t\}|c_j = c^{new}, \mathbf{c}^{-j}, \{\pi_c\}, \pi_0) &= \int p(\{m_{jk}^t\}|\pi_{new}) p(\pi_{new}|\pi_0) d\pi_{\pi_{new}} \\
&= \binom{m_{j\cdot}^t}{m_{j1}^t \cdots m_{jK}^t} \int \prod_{k=1}^{K} \pi_{new,k}^{m_{jk}^t} \frac{\Gamma(\pi_{0u} + \sum_{k=1}^{K} \pi_{0k})}{\pi_{0u} \prod_{k=1}^{K} \pi_{0k}} \pi_{new,u}^{\pi_{0,u}-1} \prod_{k=1}^{K} \pi_{new,k}^{\pi_{0k-1}} d\pi_{new} \\
&= \binom{m_{j\cdot}^t}{m_{j1}^t \cdots m_{jK}^t} \frac{\Gamma(\alpha)}{\prod_{k=1}^{K} \Gamma(\alpha \cdot \pi_{0k})} \cdot \frac{\prod_{k=1}^{K} \Gamma(\alpha \cdot \pi_{0k} + m_{jk}^t)}{\Gamma(\alpha + m_{j\cdot}^t)} \qquad \text{(A.4)}
\end{aligned}
$$

And,

$$
p(c_j = c^{new}|\{\pi_c\}, \mathbf{c}^{-j}, \pi_0) = \frac{\mu}{M - 1 + \mu} \qquad \text{(A.5)}
$$

So we have,

$$
\begin{aligned}
& p(c_j = c|\{m_{jk}^t\}, \mathbf{c}^{-j}, \{\pi_l\}, \pi_0) \\
\propto \; & \frac{u_c}{u_\cdot + \mu} \prod_{k=1}^{K} \pi_{ck}^{m_{jk}^t}, c \in \mathbf{c}^{old} \\
& \frac{\mu}{u_\cdot + \mu} \frac{\Gamma(\alpha)}{\prod_{k=1}^{K} \Gamma(\alpha \cdot \pi_{0k})} \cdot \frac{\prod_{k=1}^{K} \Gamma(\alpha \cdot \pi_{0k} + m_{jk}^t)}{\Gamma(\alpha + m_{j\cdot}^t)}, c = c^{new} \qquad \text{(A.6)}
\end{aligned}
$$

Bibliography

[1] R. Adelino and S. Ferreira. A dirichlet process mixture model for brain mri tissue classification. *Medical Image Analysis*, 11:169–182, 2006.

[2] S. Ali and M. Shah. A lagrangian particle dynamic approach for crowd flow segmentation and stability analysis. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2007.

[3] J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 39:123–154, 1984.

[4] A. Asuncion, P. Smyth, and M. Welling. Asynchronous distributed learning of topic models. In *Proc. Neural Information Processing Systems Conf.*, 2008.

[5] S. Atev, H. Arumugam, O. Masaoud, R. Janardan, and N. P. Papanikolopoulos. A vision-based approach to collision prediction at traffic intersections. *IEEE Trans. on Intelligent Transportation Systems*, 6:416–423, 2005.

[6] S. Atev, O. Masaoud, and R. J. N. Papanikolopoulos. A collision prediction system for traffic intersections. In *Porc. IEEE Conf. Intelligent Robots Systems*, 2005.

[7] S. Atev, O. Masoud, and N. Papanikolopoulos. Learning traffic patterns at intersections by spectral clustering. In *Porc. IEEE Conf. Intelligent Robots Systems*, 2006.

[8] D. Ayers and M. Shah. Monitoring human behavior from video taken in an office environment. *ivc*, 19:833–846, 2001.

[9] A. Basharat, A. Gritai, and M. Shah. Learning object motion patterns for anomaly detection and improved object detection. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2008.

[10] F. Bashir, W. Qu, A. Khokhar, and D. Schonfeld. Hmm-based motion recognition system using segmented pca. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2005.

[11] M. Bennewitz, W. Burgard, and C. Grzegorz. Utilizing learned motion patterns to robustly track persons. In *Proc. IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2003.

[12] D. Biliotti, G. Anotonini, and J. P. Thiran. Multi-layer hierarchical clustering of pedestrian trajectories for automatic counting people in video sequences. In *Proc. IEEE Workshop on Motion and Video Computing*, 2005.

[13] N. D. Bird, Masoud O., N. P. Papanikolopoulos, and A. Isaacs. Detection of loitering individuals in public transportation area. *IEEE Trans. on Intelligent Transportation Systems*, 6:167–177, 2005.

[14] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[15] M. Blank, L. Gorelick, Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Proc. Int'l Conf. Computer Vision*, 2005.

[16] D. Blei and J. Lafferty. Dynamic topic models. In *Proc. Int'l Conf. Machine Learning*, 2006.

[17] D. M. Blei and M. I. Jordan. Variational inference for dirichlet process mixtures. *Journal of Bayesian Analysis*, 1:121–144, 2006.

[18] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[19] M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *IEEE Trans. on PAMI*, 22:844–851, 2000.

[20] M. Brown and D. Lowe. Recognising panoramas. In *Proc. Int'l Conf. Computer Vision*, 2003.

[21] A. Brun, H. Knutsson, H. J. Park, M. E. Shenton, and C. F. Westin. Clustering fiber traces using normalized cuts. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention*, 2004.

[22] D. Buzan, S. Sclaroff, and G. Kollios. Extraction and clustering of motion trajectories in video. In *Proc. Int'l Conf. Pattern Recognition*, 2004.

[23] Q. Cai and J. K. Aggarwal. Tracking human motion in structured environments using a distributed-camera system. *IEEE Trans. on PAMI*, 21:1241–1247, 1996.

[24] F. Caron, M. Davy, and A. Doucet. Generalized polya urn for time-varying dirichlet process mixtures. In *Proc. of Uncertainty in Artificial Intelligence*, 2007.

[25] R. T. Collins, A. J. Lipton, H. Fujiyoshi, and T. Kanade. Algorithms for cooperative multisensor surveillance. *Proceedings of IEEE*, 89:1456–1477, 2001.

[26] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L. Wixson. A system for video surveillance and monitoring. Technical report, Carnegie Mellon University, Tech. Rep., CMU-RI-TR-00-12, 2000.

[27] G. Dalley, X. Wang, and E. Grimson. Event detection using an attention-based tracker. In *Proc. IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2007.

[28] A. Datta, M. Shah, N. Da, and V. Lobo. Person-on-person violence detection in video data. In *Proc. Int'l Conf. Pattern Recognition*, 2002.

[29] J. W. Davis and A. F. Bobick. The representation and recognition of action using temporal templates. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 1997.

[30] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39:1–38, 1977.

[31] J. Dever, N. V. Lobo, and M. Shah. Automatic visual recognition of armed robbery. In *Proc. Int'l Conf. Pattern Recognition*, 2002.

[32] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proc. of ACM Special Interest Group on Knowledge Discovery and Data Mining*, 2001.

[33] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. 42:143–157, 2001.

[34] Z Ding, J. C. Gore, and A. W. Anderson. Classification amd quantification of neuronal fiber pathways using diffusion tensor mri. *Magnetic Resonance in Medicine*, 49:716–721, 2003.

[35] D. B. Dunson, N. Pillai, and J. H. Park. Bayesian density regression. *Journal of the Royal Statistical Society: Series B*, 69:163–183, 2006.

[36] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2005.

[37] T. S. Ferguson. A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1:209–230m, 1973.

[38] J. Fernyhough, A. Cohn, and D. Hogg. Generation of semantic regions from image sequences. In *Proc. European Conf. Computer Vision*, 1996.

[39] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE Trans. on PAMI*, 30:267–282, 2008.

[40] E. B. Fox, D. S. Choi, and A. S. Willsky. Nonparameteric bayesian methods for large scale multi-target tracking. In *Proceedings of Asilomar Conference on Signals, Systems, and Computers*, 2006.

[41] Z. Fu, W. Hu, and T. Tan. Similarity based vehicle trajectory clustering and anomaly detection. In *Proc. IEEE Int'l Conf. Image Processing*, 2005.

[42] G. Galati, M. Ferri, P. Mariano, and F. Marti. Advanced integrated architecture for airport ground movements surveillance. In *Radar Conference*, 1995.

[43] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[44] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, 2004.

[45] G. Gennari and G. D. Hager. Probabilistic data association methods in visual tracking of groups. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2004.

[46] G. Gerig, S. Gouttard, and S. Corouge. Analysis of brain white matter via fiber tract modeling. In *Proc. of IEEE Engineering in Medicine and Biology*, 2004.

[47] N. Ghanem, D. Dementhon, D. Doermann, and L. Davis. Representation and recognition of events in surveillance video using petri net. In *CVPR Workshop*, 2004.

[48] N. Gheissari, T. B. Sebastian, J. Rittscher, and R. Hartley. Person reidentification using spatiotemporal appearance. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2006.

[49] D. Greenhill, J. P. Renno, J. Orwell, and G. A. Jones. Learning semantic landscape: Embedding scene knowledge in object tracking. *Real Time Imaging*, 11:186–203, 2005.

[50] J. E. Griffin and M. F. J. Steel. Order-based dependent dirichlet processes. *Journal of the American Statistical Association*, 101:179–194, 2006.

[51] T. L. Griffiths and M. Steyvers. Finding scientific topics. In *Proc. of the National Academy of Sciences of the United States of America*, 2004.

[52] P Gurdjos and P. Sturm. Methods and geometry for plane-based self-calibration. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2003.

[53] T. Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence*, 1999.

[54] S. Honggeng and R. Nevatia. Multi-agent event recognition. In *Proc. Int'l Conf. Computer Vision*, 2001.

[55] J. W. Hsieh, Yum S. H., Y. S. Chen, and W. Hu. Automatic traffic surveillance system for vehicle tracking and classification. *IEEE Trans. on Intelligent Transportation Systems*, 7:175–187, 2006.

[56] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Trans. on Systems, Man, Cybernetics-Part C: Applications and Reviews*, 34:334–352, 2004.

[57] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank. A system for learning statistical motion patterns. *IEEE Trans. on PAMI*, 28:1450–1464, 2006.

[58] W. Hu, X. Xiao, D. Xie, T. Tan, and S. Maybank. Traffic accident prediction using 3-d model-based vehicle tracking. *IEEE Trans. on Vehicular Technology*, 53:677–694, 2004.

[59] W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Maybank. Semantic-based visual surveillance. *IEEE Trans. on Image Processing*, 16:1168–1181, 2007.

[60] W. Hu, D. Xie, T. Tan, and S. Maybank. Learning activity patterns using fuzzy self-organizing neural network. *IEEE Trans. on Systems, Man, Cybernetics-Part B*, 34:1618–1626, 2004.

[61] T. Huang and S. Russell. Object identification in a bayesian context. In *Proc. of Int'l Joint Conf. Artificial Intelligence*, 1997.

[62] S. S. Intille and A. F. Bobick. A framework for recognizing multi-agent action from visual evidence. In *Proc. National Conf. Artificial Intelligence*, 1999.

[63] Li. J., S. Gong, and T. Xiang. Scene segmentation for behavior correlation. In *Proc. European Conf. Computer Vision*, 2008.

[64] O. Javed, Z. Rasheed, K. Shafique, and M. Shah. Tracking across multiple cameras with disjoint views. In *Proc. Int'l Conf. Computer Vision*, 2003.

[65] O. Javed, K. Shafique, and M. Shah. Appearance modeling for tracking in multiple non-overlapping cameras. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2005.

[66] S. Jbabdi, M. W. Woolrich, and T.E.J. Behrens. Multiple-subjects connectivity-based parcellation using hierarchical dirichlet process mixture models. *NeuroImage*, 44:373–384, 2009.

[67] L. Jiao, Y. Wu, G. Wu, E. Y. Chang, and Y. F. Wang. Anatomy of a multicamera video surveillance system. *ACM Multimedia Systems*, 210:144–163, 2004.

[68] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. In *Proc. British Machine Vision Conference*, 1995.

[69] L. Jonasson, P. Hagmann, J. P. Thiran, and V. J. Wedeen. Fiber tracts of high angular resolution diffusion mri are easily segmented with spectral clustering. In *International Society for Magnetic Resonance in Medicine*, 2005.

[70] I. Junejo and H. Foroosh. Trajectory rectification and path modeling for video surveillance. In *Proc. Int'l Conf. Computer Vision*, 2007.

[71] I. Junejo, O. Javed, and M. Shah. Multi feature path modeling for video surveillance. In *Proc. Int'l Conf. Pattern Recognition*, 2004.

[72] S. Kamijo, H. Koo, X. Liu, K. Fujihira, and Sakauchi. Development and evaluation of real-time video surveillance system on highway based on semantic hierarchy and decision surface. In *Proc. of IEEE International Conference on System, Man and Cybernetics*, 2005.

[73] S. Kamijo, M. Matsushita, K. Ikeuchi, and M. Sakauchi. Traffic monitoring and accident detection at intersections. *IEEE Trans. on Intelligent Transportation Systems*, 1:108–118, 2000.

[74] J. Kang, I. Cohen, and G. Medioni. Continuous tracking within and across camera streams. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2003.

[75] R. Kaucic, A. Perera, G. Brooksby, J. Kaufhold, and A. Hoogs. A unified framework for tracking through occlusions and across sensor gaps. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2005.

[76] Y. Ke, R. Suckthanlar, and M. Hebert. Event detection in crowded videos. In *Proc. Int'l Conf. Computer Vision*, 2007.

[77] E. Keogh and M. Pazzani. Scaling up dynamic time scaling up dynamic time. In *Proc. of ACM Special Interest Group on Knowledge Discovery and Data Mining*, 2000.

[78] V. Kettnaker and R. Zabih. Bayesian multi-camera surveillance. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 1999.

[79] S. Khan and M. Shah. Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *IEEE Trans. on PAMI*, 25:1355–1360, 2003.

[80] R. Khoshabeh, T. Gandhi, and M. M. Trivedi. Multi-camera based traffic flow characterization and classification. In *Proc. IEEE Conf. Intelligent Transporation Systems*, 2007.

[81] S. Kim and P. Smyth. Hierarchical dirichlet processes with random effects. In *Proc. Neural Information Processing Systems Conf.*, 2006.

[82] H. W. Kuhn. Variants of the hungarian method for assignment problems. *Naval Research Logistics Quarterly*, 3:253–258, 1956.

[83] P. Kumar, S. Ranganath, W. Hu, and K. Sengupta. Framework for real-time behavior interpretation from traffic video. *IEEE Trans. on Intelligent Transportation Systems*, 6:43–53, 2005.

[84] I. Laptev and T. Lindeberg. Space-time interest points. In *Proc. Int'l Conf. Computer Vision*, 2003.

[85] L. Lee, R. Romano, and G. Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Trans. on PAMI*, 22:758–768, 2000.

[86] J. Li, S. Gong, and T. Xiang. Global behavior inference using probabilistic latent semantic analysis. In *Proc. British Machine Vision Conference*, 2008.

[87] X. Li, W. Hu, and W. Hu. A coarse-to-fine strategy for vehicle motion trajectory clustering. In *Proc. Int'l Conf. Pattern Recognition*, 2006.

[88] H. M. Liao, D. Chen, C. Su, and Tyan H. Real-time event detection and its application to surveillance systems. In *Proc. IEEE Symp. Circuits and Systems*, 2006.

[89] C. Lin and Z. Ling. Automatic fall incident detection in compressed video for intelligent homecare. In *Proc. IEEE Int'l Conf. Computer Communications and Networks*, 2007.

[90] J. Lin, M. Vlachos, E. Keogh, and Dunopulous. Iterative incremental clustering of time series. *Advances in Database Technology*, 2:106–122, 2004.

[91] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proc. Int'l Joint Conf. Artificial Intelligence*, pages 674–680, 1981.

[92] S. MacEachern, A. Kottas, and A. Gelfand. Spatial nonparametric bayesian models. Technical report, Institute of Statistics and Decision Sciences, Duke University, 2001.

[93] M. Maddah, W. E. L. Grimson, S. K. Warfield, and W. M. Wells. A unified framework for clustering and quantitative analysis of white matter fiber tracts. *Medical Image Analysis*, 12:191–202, 2008.

[94] M. Maddah, W. M. Wells III, S. K. Warfield, C. F. Westin, and W. E. L. Grimson. Probabilistic clustering and quantitative analysis of white matter fiber tracts. In *Proc. Information Processing in Medical Imaging*, 2007.

[95] M. Maddah, L. Zollei, W. E. L. Grimson, and W. M. Wells III. Modeling of anatomical information in clustering of white matter fiber trajectories using dirichlet distribution. In *MMBIA*, 2008.

[96] D. Makris and T. Ellis. Path detection in video surveillance. *Image and Vision Computing*, 20:859–903, 2002.

[97] D. Makris and T. Ellis. Automatic learning of an activity-based semantic scene model. In *Proc. IEEE Conf. Advanced Video and Signal Based Surveillance*, 2003.

[98] D. Makris, T. Ellis, and J. Black. Bridging the gaps between cameras. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2004.

[99] S. Messelodi, C. M. Modena, and M. Zanin. A computer vision system for the detection and classification of vehicles at urban road intersections. *Pattern Analysis and Applications*, 8:17–31, 2005.

[100] W. Millesi, M. J. Truppe, F. Watzinger, A. Wagner, and R. Ewers. Image guided surgery extended by remote stereotactic visualization. In *Lecture Notes in Computer Science*. Springer, 1997.

[101] B. Moberts, A. Vilanova, and J. W. Jake. Evaluation of fiber clustering methods for diffusion tensor imaging. In *Proceedings of IEEE Visualization*, 2005.

[102] B. T. Morris and M. M. Trivedi. A survey of vision-based trajectory learning and analysis for surveillance. *IEEE Trans. on Circuits and Systems for Video Technology*, 18:1114–1127, 2008.

[103] A. Naftel and S. Khalid. Classifying spatialtemporal object using unsupervised learning in the coefficient features spaces. *ACM Multimedia Systems*, 12:227–238, 2006.

[104] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Proc. Neural Information Processing Systems Conf.*, 2002.

[105] J. C. Niebles, H. Wang, and F. Li. Unsupervised learning of human action categories using spatial-temporal words. In *Proc. British Machine Vision Conference*, 2006.

[106] P. Nillius, J. Sullivan, and S. Carlsson. Multi-target tracking - linking identities using bayesian network inference. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2006.

[107] L. J. O'Donnell and C. F. Westin. Automatic tractography segmentation using a high-dimensional white matter atlas. *IEEE Trans. on Medical Imaging*, 26:1562–1575, 2007.

[108] N. Oliver, B. Rosario, and A. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Trans. on PAMI*, 22:831–843, 2000.

[109] S. K. Pang, J. Li, and S. J. Godsill. Models and algorithms for detection and tracking of coordinated groups. In *Proceedings of Aerospace Conference*, 2008.

[110] C. Piciarelli and G. L. Foresti. On-line trajectory clustering for anomalous events detection. *Pattern Recognition Letters*, 2006:1835–1842, Pattern Recognition Letters.

[111] F. Porikli. Learning object trajectory patterns by spectral clustering. In *Proc. IEEE Int'l Conf. Multimedia and Expo*, 2004.

[112] F. Porikli and T. Haga. Event detection by eigenvector decomposition using object and frame features. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition Workshop*, 2004.

[113] A. Rahimi, B. Dunagan, and T. Darrell. Simultaneous calibration and tracking with a network of non-overlapping sensors. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2004.

[114] C. Rao, A. Yilmaz, and M. Shah. View-invariant reprerepresentation and recognition of actions. *International Journal of Computer Vision*, 50:203–226, 2002.

[115] P. Remagnino, S. A. Velastin, G. L. Foresti, and M. Trivedi. Novel concepts and challenges for the next generation of video surveillance systems. *Machine Vision and Applications*, 18:135–137, 2007.

[116] L. Ren, D. B. Dunson, and L. Carin. The dynamic hierarchical dirichlet process. In *Proc. Int'l Conf. Machine Learning*, 2008.

[117] A. Rodriguez, D. B. Dunson, and A. E. Gelfand. The nested dirichlet process. Technical report, Working Paper 2006-19, Duke Institute of Statistics and Decision Sciences., 2006.

[118] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2006.

[119] M. S. Ryoo and J. K. Aggarwal. Hierarchical recognition of human activities interacting with objects. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2007.

[120] I. Saleemi, K. H. Shafique, and M. Shah. Probabilistic modeling of scene dyanmics for applications in visual surveillance. *IEEE Trans. on PAMI*, 2008.

[121] A. Saunier, T. Sayed, and C. Lim. Probabilistic collision prediction for vision-based automated road safety analysis. In *Proc. IEEE Conf. Intelligent Transporation Systems*, 2007.

[122] H. Schfitze and C. Silverstein. Projections for efficient document clustering. In *Proc. of ACM Special Interest Group on Information Retrieval*, 1997.

[123] T. N. Schoepflin and D. J. Dailey. Dynamic camera calibration of roadside traffic mangement cameras for vehicle speed estimation. *IEEE Trans. on Intelligent Transportation Systems*, 4:90–98, 2003.

[124] J. Sethuraman. A constructive definition of dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[125] Y. Shan, H. Sawhney, and R. Kumar. Unsupervised learning of discriminative edge measures for vehicle matching between non-overlapping cameras. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2005.

[126] Y. Shan, H. Sawhney, and R. Kumar. Vehicle identification between non-overlapping cameras without direct feature matching. In *Proc. Int'l Conf. Computer Vision*, 2005.

[127] E. Shechtman and M. Irani. Space-time behavior based correlation. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2005.

[128] Y. A. Sheikh and M. Shah. Trajectory association across multiple airborne cameras. *IEEE Trans. on PAMI*, 30:361–367, 2008.

[129] J Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on PAMI*, 22:888–905, 2000.

[130] J. S. Shimony, A. Z. Snyder, N. Lori, and T. E. Conturo. Automated fuzzy clustering of neuronal pathways in diffusion tensor tracking. In *International Society for Magnetic Resonance in Medicine*, 2002.

[131] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *Proc. Int'l Conf. Computer Vision*, 2005.

[132] P. Smith, N. V. Lobo, and M. Shah. Temporalboost for event recognition. In *Proc. Int'l Conf. Computer Vision*, 2005.

[133] N. Srebro and S. Roweis. Time-varying topic models using dependent dirichlet processes. Technical report, Department of Computer Science, University of Toronto, 2005.

[134] C. Stauffer. Estimating tracking sources and sinks. In *IEEE Workshop on Event Mining*, 2003.

[135] C. Stauffer and E. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. on PAMI*, 22:747–757, 2000.

[136] C. Stauffer and K. Tieu. Automated multi-camera planar tracking correspondence modeling. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2003.

[137] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky. Describing visual scenes using transformed objects and parts. *International Journal of Computer Vision*, 77:291–330, 2007.

[138] E. B. Sudderth, A. Torralba, Freeman W. T., and A. S. Willsky. Learning hierarchical models of scenes, objects, and parts. In *Proc. Int'l Conf. Computer Vision*, 2005.

[139] N. Sumpter and A. J. Bulpitt. Learning spatio-temporal patterns for predicting object behavior. *Image and Vision Computing*, 18:697–704, 2000.

[140] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet process. *Journal of the American Statistical Association*, 2006.

[141] Y. W. Teh, K. Kurihara, and Welling M. Collapsed variational inference for hdp. In *Proc. Neural Information Processing Systems Conf.*, 2007.

[142] B. Thirion, A. Tucholka, M. Keller, and P. Pinel. High level group analysis of fmri data based on dirichlet process mixture models. In *Proc. Information Processing in Medical Imaging*, 2007.

[143] K. Tieu, G. Dalley, and E. Grimson. Inference of non-overlapping camera network topology by measuring statistical dependence. In *Proc. Int'l Conf. Computer Vision*, 2005.

[144] B Triggs. Camera pose and calibration from 4 or 5 known 3d points. In *Proc. Int'l Conf. Computer Vision*, 1999.

[145] G. Unal, A. Yezzi, S. Soatto, and G. Slabaugh. A variational approach to problems in calibration of multiple cameras. *IEEE Trans. on PAMI*, 29:1322–1338, 2007.

[146] H. Veeraraghavan, O. Maoud, and N. Papanikolopoulos. Computer vision algorithms for intersection monitoring. *IEEE Trans. on Intelligent Transportation Systems*, 4:78–89, 2003.

[147] M. Vlachos, G. Kollios, and D. Gunopulos. Discovering similar multidimensional trajectories. In *Proc. IEEE Conf. Data Engineering*, 2002.

[148] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. Tu. Shape and appearance context modeling. In *Proc. Int'l Conf. Computer Vision*, 2007.

[149] X. Wang and E. Grimson. Spatial latent dirichlet allocation. In *Proc. Neural Information Processing Systems Conf.*, 2007.

[150] X. Wang, K. T. Ma, G. Ng, and E. Grimson. Trajectory analysis and semantic region modeling using a nonparametric bayesian model. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2008.

[151] X. Wang, X. Ma, and E. Grimson. Unsupervised activity perception by hierarchical bayesian models. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2007.

[152] X. Wang, X. Ma, and E. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *IEEE Trans. on PAMI*, 31, 2009.

[153] X. Wang, K. Tieu, and E. Grimson. Learning semantic scene models by trajectory analysis. In *Proc. European Conf. Computer Vision*, 2006.

[154] X. Wang, K. Tieu, and E. Grimson. Correspondence-free multi-camera activity analysis and scene modeling. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2008.

[155] X. Wang, K. Tieu, and E. Grimson. Correspondence-free activity analysis and scene modeling in multiple camera views. *IEEE Trans. on PAMI*, 2009.

[156] Y. Wang, T. Jiang, M. S. Drew, Z. Li, and G. Mori. Unsupervised discovery of action classes. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2006.

[157] Y. Xia, A. U. Turken, S. L. Whitfield-Gabrieli, and J. D. Gabrieli. Knowledge-based classification of neuronal fibers in entire brain. In *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention*, 2005.

[158] T. Xiang and S. Gong. Video behaviour profiling and abnormality detection without manual labelling. In *Proc. Int'l Conf. Computer Vision*, 2005.

[159] T. Xiang and S. Gong. Beyond tracking: Modelling activity and understanding behaviour. *International Journal of Computer Vision*, 67:21–51, 2006.

[160] W. Yan and D. A. Forsyth. Learning the behavior of users in a public space through video tracking. In *Proc. IEEE Workshop Applications of Computer Vision*, 2005.

[161] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2001.

[162] J. Zhang, Z. Ghahramani, and Y. Yang. A probabilistic model for online document clustering with application to novelty detection. In *Proc. Neural Information Processing Systems Conf.*, 2004.

[163] Z. Zhang, K. Huang, and T. Tan. Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In *Proc. Int'l Conf. Pattern Recognition*, 2006.

[164] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2004.

[165] X. Zhu, Z. Ghahramani, and J. Lafferty. Time-sensitive dirichlet process mixture models. Technical report, School of Computer Science Carnegie Mellon University, 2005.