UC Riverside
UC Riverside Electronic Theses and Dissertations

Title
Graphical Models for Wide-Area Activity Analysis in Continuous Videos

Permalink
https://escholarship.org/uc/item/5572m8z1

Author
Nayak, Nandita Miyar

Publication Date
2014

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Graphical Models for Wide-Area Activity Analysis in Continuous Videos

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Nandita M. Nayak

December 2014

Dissertation Committee:

Dr. Amit K. Roy-Chowdhury, Chairperson
Dr. Christian R. Shelton
Dr. Eamonn Keogh
Dr. Victor B. Zordan

Copyright by
Nandita M. Nayak
2014

The Dissertation of Nandita M. Nayak is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

This thesis represents my academic experience which includes the vast amount of ideas and guidance which I received from the people I came in touch with. I would like to start by expressing my deepest gratitude to my advisor Dr. Amit Roy-Chowdhury. He inspired me to pursue this topic and led me through it on a daily basis. He was there for me in every ups and downs of my academic endeavor throughout my graduate life. I was able to achieve my academic goals due to all the support and guidance I received from him.

I am also grateful to my committee members for their useful suggestions. In particular, I would like to thank Prof. Christian Shelton for helping me with the conceptual issues I had in my Ph.D. research. I learnt most of the background materials I needed for my research from the courses taught by Prof. Roy-Chowdhury and Prof. Shelton who are also amazing teachers.

I would also like to express my gratitude to my colleagues in the lab for their ideas and discussions which contributed to my knowledge in the field. In particular, I would like to thank Yingying Zhu for collaborating with me in my work. She helped with the experiments and in formulating the solution for various challenges during my research. I would also like to thank Katya Mkrtchyan, Ramya Srinivasan, Anirban Chakraborty, Elliot Staudt and Shu Zhang for sharing ideas and helping me in understanding concepts better.

A sound foundation for my graduate studies was laid during my masters course at Indian Institute of Science. I would like to thank Prof. K. R. Ramakrishnan for mentoring me in computer vision and for inspiring me to take up research in the subject. This academic endeavor would not have been possible but for his encouragement.

Words cannot express the level of my gratitude to my mother Nirmala Nayak and my father Narendranath Nayak for all their support and guidance. They inspired me to pursue a Ph.D. and stood by me through the thick and thin of times. I am also ever grateful to my husband Manjunath Prabhu for his unconditional support and guidance that made my graduate studies possible. My entire family has supported and encouraged me to pursue my dreams. My son Samridh Prabhu deserves a final mention for making life so much more beautiful with his presence.

Acknowledgment of previously published or submitted materials: The text of this dissertation, in part or in full, is a reprint of the material as is appears in three previously published or submitted papers that I first authored. The co-author Dr. Amit K. Roy-Chowdhury, listed in all three publications, directed and supervised the research which forms the basis for this dissertation. The papers are, as follows.

1. ' Vector field analysis for multi-object behavior modeling ', published at Image and Visual Computing, Oct 2013.

2. 'Exploiting Spatio-temporal Scene Structure for Wide-Area Activity Analysis in Unconstrained Environments', published at IEEE Transactions on Information Forensics and Security, May 2013. My co-author Yingying Zhu contributed to the analysis and writing.

3. 'Learning a Sparse Dictionary of Video Structure for Activity Modeling', accepted at International Conference on Image Processing (oral), 2014.

4. 'Simultaneous Tracking and Recognition of Activities on Sparse Hierarchical Graphs',

submitted at IEEE Transactions on Image Processing. My co-author Yingying Zhu contributed to the analysis and writing.

To my family, for all the love and support.

ABSTRACT OF THE DISSERTATION

Graphical Models for Wide-Area Activity Analysis in Continuous Videos

by

Nandita M. Nayak

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, December 2014
Dr. Amit K. Roy-Chowdhury, Chairperson

Activity analysis is a field of computer vision which has shown great progress in the past decade. Starting from simple single person activities, research in activity recognition is moving towards more complex scenes involving multiple objects and natural environments. The main challenges in the task include being able to localize and recognize events in a video and deal with the large amount of variation in viewpoint, speed of movement and scale.

Surveillance videos typically consist of wide areas being monitored through a static camera. Often, they contain long duration sequences of activities which occur at different spatio-temporal locations and can involve multiple people acting simultaneously. Many times, the activities have contextual relationships with one other. Although context has been studied in the past for the purpose of activity recognition to a certain extent, the use of context in recognition of activities in such challenging environments is relatively unexplored. The primary focus of the work is in recognition of activities in continuous videos.

We discuss three methods of activity recognition in continuous videos. In the first,

we demonstrate the different components of analysis involved in labeling activities in wide-area continuous videos, such as elimination of background noise, identification of motion patterns which correspond to interesting activities and the task of activity modeling. We propose to do this using an optical flow based framework. We discuss the limitations of this work, which can be overcome with the addition of context.

Next, we propose a context-based approach for activity recognition using graphical models. We assume that the location of activities are identified using existing techniques. The task of the graphical model is therefore to label these identified regions using context. Given a collection of videos and a set of weak classifiers for individual activities, the spatio-temporal relationships between activities are represented as probabilistic edge weights in a Markov random field. This model provides a generic representation for an activity sequence that can extend to any number of objects and interactions in a video. We show that the recognition of activities in a video can be posed as an inference problem on the graph. We conduct experiments on the publicly available VIRAT dataset to demonstrate the improvement in recognition accuracy using our proposed model as opposed to recognition using state-of-the-art features on individual activity regions.

Next, we present a unified framework to track multiple people, as well localize and label their activities, in complex long-duration video sequences. To do this, we focus on two aspects - the influence of tracks on the activities performed by the corresponding actors and the structural relationships across activities. We propose a two-level hierarchical graphical model which learns the relationship between tracks, relationship between tracks and their corresponding activity segments, as well as the spatiotemporal relationships across activity

segments. Such contextual relationships between tracks and activity segments are exploited at both the levels in the hierarchy for increased robustness.

Finally, we suggest how the structure learning can be performed in a graphical model which performs activity recognition. While a continuous video consists of several activities, the contextual relationships between these activities are relatively sparse. We propose a method which aims to discover these sparse relationships using an L1-regularization based automatic structure discovery of a graphical model representing the video. Sparsity is imposed on the edges of the graph so as to model a sparse set of relationships.

Contents

List of Figures **xiv**

List of Tables **xx**

1 Introduction **1**
1.1 Activity Recognition . . . 1
1.1.1 Challenges in Wide-Area Activity Analysis . . . 2
1.1.2 Motivation for the Use of Context in Wide-area Activity Analysis . . . 4
1.2 Related Work . . . 6
1.2.1 Activity Recognition Methods . . . 6
1.2.2 Context-Based Activity Recognition . . . 8
1.3 Contributions of the Work . . . 11
1.4 Organization of the Thesis . . . 12

2 Multi-person Activity Recognition in Wide-area Videos **13**
2.1 Introduction . . . 13
2.1.1 Main Contributions . . . 16
2.1.2 Related Work . . . 17
2.2 Overview of Proposed Approach . . . 19
2.3 Streakline Representation of Motion Patterns . . . 21
2.3.1 Identification of Motion Regions . . . 23
2.3.2 Helmholtz Decomposition of Flow Field . . . 26
2.3.3 Motion Regions Using the Helmholtz Decomposition . . . 28
2.4 Segmentation of Motion Patterns . . . 29
2.4.1 Time Segmentation of Streaklines . . . 30
2.4.2 Segmentation of Streaklines in Space . . . 32
2.5 Activity Modeling and Recognition . . . 35
2.5.1 Comparison of Average Preshapes . . . 36
2.5.2 Subspace Analysis . . . 36
2.5.3 Overall Distance Computation . . . 37
2.6 Experiments . . . 39

2.6.1 Dataset . . . 39
2.6.2 Results on UT Interaction Data . . . 40
2.6.3 Results on the VIRAT Data . . . 43
2.6.4 Analysis of the Results . . . 45
2.7 Conclusion . . . 47

3 Context Modeling in Continuous Videos Using Graphical Models 49
3.1 Introduction . . . 49
3.1.1 Contributions . . . 50
3.1.2 Related Work . . . 51
3.1.3 Definitions . . . 53
3.2 Overview . . . 53
3.3 Graphical Representation of Activities . . . 55
3.3.1 Potential Functions . . . 56
3.3.2 Training . . . 59
3.3.3 Inference . . . 59
3.4 Experiments and Results . . . 60
3.4.1 Dataset . . . 60
3.4.2 Pre-processing . . . 62
3.4.3 Methodology . . . 64
3.4.4 Results on UCLA office dataset . . . 65
3.4.5 Results on VIRAT dataset . . . 66
3.5 Conclusion . . . 68

4 Hierarchical Graphical Model For Simultaneous Tracking, Localization And Recognition Of Activities 72
4.1 Introduction . . . 72
4.1.1 Contributions . . . 75
4.1.2 Related Work . . . 76
4.2 Overview . . . 77
4.3 Hierarchical MRF (HMRF) Model . . . 78
4.3.1 Computation of Potential Functions of HMRF . . . 80
4.3.2 Training . . . 83
4.4 Inference on the HMRF . . . 84
4.4.1 Bottom-up Inference: From Tracks to Activities . . . 84
4.4.2 Top-down Inference: From Activities to Tracks . . . 85
4.4.3 Bi-directional Processing for Tracking and Activity Recognition . . . 87
4.5 Experiments and Results . . . 88
4.5.1 Dataset . . . 88
4.5.2 Analysis of the Results . . . 89
4.5.3 Tracking Results on VIRAT Release 2 . . . 90
4.6 Conclusion . . . 92

5 Structure Discovery Using L1-regularized Learning In Graphical Models 95
5.1 Introduction . . . 95
5.1.1 Related Work . . . 98
5.2 Overview . . . 99
5.3 L1-regularized Graphical Model for Activity Recognition . . . 101
5.3.1 Standard L1-regularization of Parameters: . . . 101
5.3.2 Group L1-regularization of Parameters: . . . 103
5.4 Experiments . . . 105
5.4.1 Methodology . . . 106
5.4.2 Analysis of the Results . . . 108
5.5 Conclusion . . . 112

6 Conclusion and Future Work 114
6.1 Thesis Summary . . . 114
6.2 Future Work . . . 116

Bibliography 118

List of Figures

1.1 a) An example scene from a surveillance video in a parking lot demonstrating that different different activities happen together and can influence each other. “Open door”, “unloading vehicle” and “approaching vehicle” are related since they pertain to the same vehicle. Other objects in the scene can cause background clutter making the recognition task challenging. b) the spatio-temporal relationship between two activities in a video. 5

2.1 The figure shows sample frames of the VIRAT dataset used for recognition. The first figure shows a person loading a trunk, the second figure shows a person entering a vehicle and the third figure shows a person closing a trunk. We notice other people in the scene adding to background clutter. Lighting changes and shadows add noise to the data. 15

2.2 The figure shows the overall framework of the proposed method. 20

2.3 The figure shows the streaklines for people opening a trunk in two videos. The circled region shows the similarity in the activity captured by the streaklines. 22

2.4 Decomposition of a flow field: The figure shows a sample flow field and its decomposition into the irrotational and solenoidal components. The critical points are marked in red on each image. Figure a) shows the original flow field; Figure b) is the original flow field marked with regions containing critical points. We notice that the critical point in region 3 is an attracting focus and the critical points in region 1 and 2 are repelling nodes; Figure c) represents the solenoidal component of original flow; and d) represents the irrotational component of original flow. We can see that the irrotational field has no rotational component and the solenoidal field is divergence free (purely rotational). 25

2.5 The figure shows the extraction of motion regions from streaklines. Figure a) shows the streaklines of the action "open trunk". Figure b) shows the corresponding motion field. The critical points of the motion field are marked in red. The streaklines extracted using these critical points are shown in Figure c) and constitute the motion regions of the video. 30
2.6 Examples of time segmentation of streaklines using the Helmholtz decomposition. The first row shows a sample frame and the second row displays the time segmented streaklines. Each segment is marked in a different color. The critical points are marked in blue. 31
2.7 The figure shows the streaklines and the clusters for activities in the VIRAT dataset. The clusters are marked with different colors. 33
2.8 Examples of retrieved results for the UT Interaction dataset. 41
2.9 The figure shows the accuracy of recognition using the UT Interaction data and comparison with previous methods. The activities are: - 1 - Shake Hands, 2 - Hug, 3 - point, 4 - Punch, 5 - Kick, 6 - Push 42
2.10 Example of results for the VIRAT dataset showing some true positives and false negatives for actions close trunk, enter vehicle and unloading. The false negatives are marked in red. 44
2.11 The figure shows the recognition accuracy for the VIRAT dataset. The activities are: 1 - loading, 2 - unloading, 3 - open trunk, 4 - close trunk, 5 - enter vehicle, 6 - exit vehicle 46
3.1 Figure shows the illustration of our proposed method. Training involves modeling the pairwise spatio-temporal relationships between different activity regions which are provided in annotations as mentioned in Section 3.3.1. For a test video, activity regions are identified using the method presented in Section 3.4.2. Using the potentials from training data and observation potentials as described in Section 3.3.1, the node labels are inferred (Section 3.3.3). 54
3.2 Figure shows the Markov random field constructed over a spatio-temporal volume for an activity sequence. Shown in the figure are the activity regions which form the observation variables y. The baseline classifier output forms the observation potential. The labels of the activities which have to be predicted constitute the hidden nodes x. The edges of the graph are learnt iteratively. 57

3.3 The figure shows some examples of segmentation of activity regions. The obtained segmentation is marked in green while the true segmentation is marked in red. 63
3.4 The figure shows the precision and recall obtained on the UCLA office dataset and its comparison with the Bag-Of-Features baseline classifier and SFG [1]. The activities are: 1 - enter room, 2 - exit room, 3 - sit down, 4 - stand up, 5 - work on laptop, 6 - work on paper, 7 - throw trash, 8 - pour drink, 9 - pick phone, 10 - place phone down. 65
3.5 Figure a) shows the accuracy of our method with the VIRAT release 1 dataset for six activities and its comparison with the Bag-of-Words and SFG [1] approach. The activities are: 1 - loading, 2 - unloading, 3 - open trunk, 4 - close trunk, 5 - enter vehicle, 6 - exit vehicle. Figure b) shows the increase in performance with structure improvisation. 66
3.6 The Figure shows the confusion matrix on VIRAT release 1 data. a)Result of applying the baseline classifier BOW to the data. b) Result of applying BOW+context on the data. c) Result of SFG baseline classifier. d) Result of SFG + context. The activities are: 1 - loading, 2 - unloading, 3 - open trunk, 4 - close trunk, 5 - enter vehicle, 6 - exit vehicle. The corresponding increase in recognition accuracy is evident from the graph. 69
3.7 The figure shows the precision and recall obtained on the VIRAT release 2 dataset and its comparison with the Bag-Of-Features and SFG approaches. The activities are: 1 - person loading an object to a vehicle, 2 - person unloading an object from a vehicle, 3 - person opening a vehicle trunk, 4 - person closing a vehicle trunk, 5 - person getting into a vehicle, 6 - person getting out of a vehicle; 7 - person gesturing, 8 - person running, 9 - carrying load, 10 - entering facility, 11 - exiting facility. 70
3.8 The comparison of the prior probabilities which are the output of the baseline classifiers with the posterior probabilities which is the output of our algorithm for a set of six activities. The output of our algorithm is seen to have a more well defined peak (less uncertainty) as compared to the baseline classifier. For the last two, it is seen that the addition of context corrects an incorrect classification. The activities in order are: 1 - person loading an object to a vehicle, 2 - person unloading an object from a vehicle, 3 - person opening a vehicle trunk, 4 - person closing a vehicle trunk, 5 - person getting into a vehicle, 6 - person getting out of a vehicle 71

4.1 Figure demonstrates the bi-directional processing of videos for integrated tracking and activity recognition. The bottom-up (or feedforward) processing involves detection and recognition using an initial set of tracks along with low level features and spatiotemporal context between activities. The top-down (or feedback) processing involves correcting the tracklet associations using the obtained labels. 73
4.2 Figure shows the illustration of our proposed method. Given a continuous video with computed tracklets, a set of tracks and activity segments are initialized. An HMRF model is built over the tracklets and segments. Edge potentials are learned on the annotated training data. Inference on this graphical model provides a revised set of labels for the activities which can be fed back into the system to regenerate the tracks and rebuild the HMRF. The procedure is repeated until a stop criterion is reached. The tracks and labels of all segments are provided as output. 77
4.3 Figure shows a typical HMRF over an activity sequence. Tracklets are extracted from a continuous video and form lower level nodes. Using an initial set of tracks, a segmentation of tracklets is performed to obtain activity segments. These form the higher level nodes. Edges model relationships between potentially associated tracklets, tracklets and their corresponding activity segments, and the spatiotemporal context information between activity segments. The node potentials and edge potentials are marked in the graph. 79
4.4 The figure shows the precision and recall obtained on the VIRAT release 1 dataset with our approach. Comparison has been shown to the performance of baseline classifier BOW [2] as well as Zhu et al [3]. The activities are listed in Section 4.5.1. 91
4.5 The figure shows the precision and recall obtained on the VIRAT release 2 dataset with our approach. Comparison has been shown to the performance of baseline classifier BOW [2] as well as Zhu et al [3]. The activities are listed in Section 4.5.1. 92

4.6 The figure shows two examples where tracking is improved with the addition of context. The top row shows the tracking results without activity context while the bottom row shows the result with the addition of feedback. Red and green signify different tracks in each case. In the first case, it is seen that the track was wrongly terminated due to occlusion in the absence of context. In the second case, the tracklet association error was corrected with the addition of context. 93
5.1 A continuous video can consist of multiple activities. The challenge in context modeling using graphical models is to arrive at a structure which effectively models the contextual relationships between activities. In this chapter, we propose an L1-regularized learning of the graphical model which performs an automatic structure discovery on the graph. 97
5.2 Figure shows the illustration of our proposed method. Given a continuous video with computed tracklets, a set of tracks and activity segments are initialized. An HMRF model is built over the tracklets and segments. Edge potentials are learned on the annotated training data. Starting with a dense graph, L1-regularized structure learning gives a sparse set of edges. Inference on this graphical model provides a revised set of labels for the activities which can be fed back into the system to regenerate the tracks and rebuild the HMRF. The procedure is repeated until a stop criterion is reached. The tracks and labels of all segments are provided as output. 100
5.3 A few examples of activities which were incorrectly detected using a dense graphical model ($\lambda = 0$) and correctly discovered after the L1-regularized parameter learning. The advantage of learning a sparse graph is better representation of contextual information. 105
5.4 The figure shows the sparse contextual relationships discovered by L1-regularized learning on VIRAT 1 dataset. The figure on the left shows the fully connected model assumed before parameter learning. The next figure shows the sparse relationships obtained after parameter learning. The edges corresponding to parameters which are set to zero have been deleted from the graph. The bar graph on the right shows the histogram of obtained sparse parameters. . . 106
5.5 The figure shows the precision and recall obtained on the VIRAT release 1 dataset with our approach. Comparison has been shown to the performance of baseline classifier BOW [2] as well as Zhu et al [3]. The activities in order are: Person entering vehicle, person exiting vehicle, person opening trunk, person closing trunk, person loading vehicle and person unloading vehicle. 107

5.6 For an activity sequence from VIRAT release 1 containing 5 activities, we show the initial dense hierarchical Markov random field model constructed on the sequence (left) and the corresponding sparse graphical model obtained after L1-regularized learning of parameters (right). 109
5.7 The figure shows the precision and recall obtained on the VIRAT release 2 dataset with our approach. Comparison has been shown to the performance of baseline classifier BOW [2] as well as Zhu et al [3]. The activities in order are: Person entering vehicle, person exiting vehicle, person opening trunk, person closing trunk, person loading vehicle, person unloading vehicle, person carrying an object, person gesturing, person running, entering and exiting a facility. 110
5.8 For an activity sequence from VIRAT release 2 containing 7 activities, we show the initial dense hierarchical markov random field model constructed on the sequence (left) and the corresponding sparse graphical model obtained after L1-regularized learning of parameters (right). 112
5.9 The figure shows the sparse structure of the graph discovered by L1-regularized learning on 11 activities of VIRAT 2 dataset. Figure a) shows the fully connected graph assumed before parameter learning. Figure b) shows the sparse graph obtained after parameter learning. The edges corresponding to parameters which are set to zero have been deleted from the graph. Figure c) shows the histogram of the learned parameters $\mathbf{w}$. From the histogram, it can be seen that $\mathbf{w}$ is sparse. The activity labels are the same as in Figure 5.7. 113

List of Tables

4.1 Overall precision and recall values of methods BOW, Gaur et. al[1], Zhu et. al [3] and our approach for the VIRAT release 1 dataset. 90
4.2 Precision and recall values of methods BOW, Amer et. al[4] and Zhu et. al [3] and our approach for the VIRAT release 2 dataset. 93
4.3 Precision and recall values of methods BOW, Amer et. al[4] and Zhu et. al [3] and our approach for the VIRAT release 2 dataset. 93
5.1 Overall precision and recall values of methods BOW, Gaur et. al[1], Zhu et. al [3] and our approach for the VIRAT release 1 dataset. 108
5.2 Precision and recall values of methods BOW, Amer et. al[4] and Zhu et. al [3] and our approach for the VIRAT release 2 dataset. 112

Chapter 1

Introduction

1.1 Activity Recognition

Activity recognition is the task of interpretation of the activities of objects in video over a period of time. The goal of an activity recognition system is to extract information on the movements of objects and/or their surroundings from the video data so as to conclude on the events and context in the video in an automated manner. In a simple scenario where the video is segmented to contain only one execution of a human activity, the objective of the system is to correctly classify the activity into its category, whereas in a more complex scenario of a long video sequence containing multiple activities, it may also involve the detection of the starting and ending points of all occurring activities in the video[5].

Although there is no formal classification of activities into different categories, for the sake of understanding, activities can be divided into simple and complex activities based on the complexity of the recognition task [6]. An activity which involves a single person

and lasts only a few seconds can be termed as a simple activity. Such video sequences are generally recorded in a constrained environment with very little variation or extraneous noise. Some examples of simple activities are running, walking, waving, etc. Although it is uncommon to find such data in the real world, these video sequences are useful in the learning and testing of new models for activity recognition. Popular examples of such activities are found in the Weizmann [2] and KTH [7] datasets.

The focus of this work however, is on wide-area activity analysis in surveillance scenarios. This is a good example of complex activity recognition. In this case, we are dealing with more realistic environments where people enter and exit the scene continuously. The number of people in the scene at any instant cannot be easily predicted. Since we are also dealing with a wide-area, we can have activities occurring at different viewpoints from the static camera. The activities can involve person-person interactions or person-object interactions. In addition, we deal with continuous videos here. This means that more than one activity can take place in a video. The scene also contains some occlusion and background clutter. Some examples of wide-area activity datasets are the UT-Interaction dataset [8], the VIRAT dataset [9], the UCLA dataset [10] and the UCR videoweb dataset [11].

1.1.1 Challenges in Wide-Area Activity Analysis

Wide-Area activity analysis is a challenging task for several reasons. Any activity recognition system is efficient only if it can deal with changes in pose, lighting, viewpoint and scale. These variations increase the dimensionality of the problem. These problems are

prevalent to a greater degree when it comes to wide-area activity analysis. There is a large amount of structural variation in a complex activity, therefore the dimension of the feature space is high. The feature-space also becomes sparser with the dimension, thus requiring a larger number of samples to build efficient class-conditional models thus bringing in the Curse of Dimensionality [12]. Issues of scale, viewpoint and lighting also get harder to deal with for this reason.

Most of the simple activity recognition systems in the past had been tested on sequences recorded in a noise free controlled environment. Although these systems might work reasonably well in such an environment, they may not work in a real world environment which contains noise and background clutter. This problem is more prominent in a wide-area activity recognition system since there are multiple motions in the scene and they can easily be confused with the clutter.

Another challenge in wide-area motion analysis is the presence of multiple activities occurring in a continuous manner. Although many approaches can deal with noise with sufficient training data, there are difficulties in recognizing continuous activities with complex temporal structures, such as an activity composed of concurrent sub-events. Therefore many methods are more suited for modeling sequential activities rather than concurrent ones [13]. In addition, as stated in [13], as an activity gets more complex, many existing approaches need a greater amount of training data, preventing them from being applied to highly complex activities.

1.1.2 Motivation for the Use of Context in Wide-area Activity Analysis

A long-term wide-area surveillance video usually consists of multiple people entering and exiting the scene over a period of time. Therefore, it is hard to predict the number of activities occurring in the scene and the number of people involved in those activities. This variability in the number of actors and the number of action executions within the sequence is what we term as an "unconstrained" environment in this thesis. Most existing activity recognition algorithms focus on the region where an activity occurs while ignoring the contextual information in the surroundings. Such methods place assumptions on the number of objects, scale and viewpoint of the scene and may not be equally effective in more challenging environments. Often, it has been found that examining the surroundings of an activity under consideration in a scene can provide useful clues about the activity. This information obtained from the surroundings is termed as "context" of the activity.

In this thesis, we propose to incorporate spatial and temporal context across activities in a continuous video into our model. The modeling of this contextual information along with a traditional activity recognition system can provide improved recognition rates in challenging environments.

Most existing activity recognition approaches aim at recognizing atomic activities or a single interaction in a short video clip. Real world videos tend to have a large amount of intra-class variation as well as clutter and noise which makes the recognition task difficult. Therefore, although the standard recognition methods can be applied here, it is difficult to obtain a high accuracy results with existing classifiers. A typical example of an outdoor wide area scene is shown in Figure 1.1 a). The different challenges in recognition are marked on

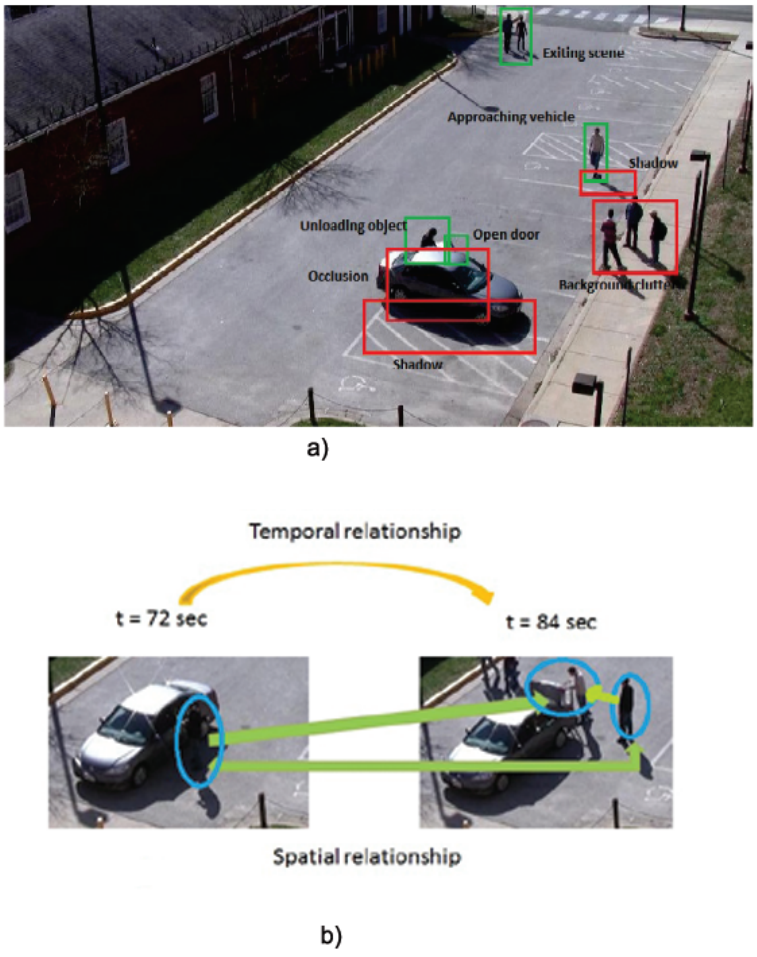

Figure 1.1: a) An example scene from a surveillance video in a parking lot demonstrating that different different activities happen together and can influence each other. "Open door", "unloading vehicle" and "approaching vehicle" are related since they pertain to the same vehicle. Other objects in the scene can cause background clutter making the recognition task challenging. b) the spatio-temporal relationship between two activities in a video.

the figure. Figure 1.1 b) shows the spatial and temporal relationships between two activities in a video.

The presence of multiple activities, however, also imply that we now have more information available to us about the scene as a whole, as compared to a small clip containing a single atomic activity. Activities in a video are often related to each other. For example, the fact that a person opens the trunk of a car makes it very likely that he might place or retrieve an object from the trunk. In addition, if we knew that the person had just exited a facility, it is more likely that he will place an object rather than retrieve it. Therefore, the occurrence of one activity can provide us a context which can be used to recognize another related activity. In this work, we wish to demonstrate a method to model this context and utilize the information to recognize activities in a complex video.

1.2 Related Work

1.2.1 Activity Recognition Methods

A popular approach to activity recognition has been the use of local interest points. Each interest point has a local descriptor to describe the characteristics of the point. Motion analysis is thus brought about by the analysis of feature vectors. Some researchers used spatial interest points to describe a scene [14] [15] [16]. Such approaches are termed as local approaches [17]. Over time, researchers described other robust spatio-temporal feature vectors. SIFT (scale invariant feature transform) [18] and STIP (space time interest points) [19] are commonly used local descriptors in videos. A more recent approach is to combine

multiple features in a multiple instance learning (MIL) framework to improve accuracy [20].

Another approach to action recognition is global analysis of the video [17]. This involves a study of the overall motion characteristics of the video. Many of these methods use optical flow to represent motion in a video frame. One example of this method is in [21]. These approaches often involve modeling of the flow statistics over time. Optical flow histograms have commonly been used to compute and model flow statistics like in [22] which demonstrates the use of optical flow histograms in the analysis of soccer matches. In some other cases, human activities have been represented by 3-D space-time shapes where classification is performed by comparing geometric properties of these shapes against training data [2] [23].

Methods which have been used for modeling activities can be classified as non-parametric, volumetric and parametric time-series approaches [12]. Non-parametric approaches typically extract a set of features from each frame of the video. Non parametric approaches could involve generation of 2D templates from image features [24] [25], 3D object models using shape descriptors or object contours [2] or manifold learning by dimensionality reduction methods such as PCA, locally linear embedding (LLE) [26] and Laplacian eigenmaps [27]. Parametric methods involve learning the parameters of a model for the action using training data. These could involve Hidden Markov Models (HMM) [28] and linear [29] and non-linear dynamical systems [30]. Volumetric methods of action recognition perform sub volume matching or use filter banks for spatio-temporal filtering [31]. Some researchers have used multiple 2D videos to arrive at a 3D model which is then used for view invariant action recognition [32].

1.2.2 Context-Based Activity Recognition

As compared to a simple activity recognition system, the inherent structure and semantics of complex activities require higher-level representation and reasoning methods [12].There have been different approaches used to analyze complex activities. One common approach has been the use of graphical models. Graphical models encode the dependencies between random variables which in many cases are the features which represent the activity. These dependencies are studied with the help of training sequences. Some examples of graphical models commonly used are Belief networks (BNs), Hidden Markov Models (HMMs) and Petri nets. Belief networks and Dynamic Belief Networks (DBNs) are graphical models that encode complex conditional dependencies between a set of random variables which are encoded as local conditional probability densities. These have been used to model two person interactions like kicking, punching, etc by estimating the pose using Bayesian networks and the temporal evolution using Dynamic Bayesian networks [33] [34]. A grid based belief propagation method was used for human pose estimation in [35]. Graphical models often model activities as a sequential set of atomic actions. A statistical model is created for each activity. The likelihood of each activity is given by the probability of the model generating the obtained observations [13].

A popular approach for modeling complex activities has been the use of stochastic and context free grammars. It is often noticed that a complex large-scale activity often can be considered as a combination of several simple sub-activities that have explicit semantic meanings [36]. Constructing grammars can provide useful insights in such cases. These methods try to learn the rules describing the dynamics of the system. These often involve

hierarchical approaches which parallel language grammars in terms of construction of sentences from words and alphabets. A typical example is when the activity recognition task is split into two steps. First, bottom-up statistical method can be used to detect simple sub-activities. Then the prior structure knowledge is used to construct a composite activity model [37]. In another instance, context free grammars in [38] followed a hierarchical approach where the lower-levels are composed of HMMs and Bayesian Networks, whereas the higher level interactions are modeled by context free grammars [12]. More complex models like Dependent Dirichlet Process-Hidden Markov Models (DDP-HMMs) have the ability to jointly learn co-occurring activities and their time dependencies [39].

Knowledge and logic based approaches have also been used in complex activity recognition [12]. Logic based approaches construct logical rules to describe the presence of an activity. For instance, a hierarchical structure could be used by defining descriptors of actions extracted from low-level features through several mid-level layers. Next, a rule based method is used to approximate the probability of occurrence of a specific activity by matching the properties of the agent with the expected distributions for a particular action [40]. Recently, the use of visual cues to detect relations among persons have been explored in a social network model [41].

Description based methods try to identify relationships between different actions such as "before", "after", "along with", etc. The algorithm described in [42] is one such method which uses spatio-temporal feature descriptors. The Bag of Words approach [43] disregards order and tries to model complex activities based on the occurrence probabilities of different features. Attempts have been made to improve on this idea by identifying neigh-

borhoods which can help in recognition [44] and by accommodating pairwise relationships in the feature vector to consider local ordering of features [45]. Hierarchical methods have also been proposed which build complex models by starting from simpler ones and finding relationships between them [46].

Many of these approaches require either tracking body parts, or contextual object detection, or atomic action/primitive event recognition. Sometimes tracks and precise primitive action recognition may not be easily obtained for complex/interactive activities since such scenes frequently contain occlusions and clutter. Spatio-temporal feature based approaches, like [47], hold promise since no tracking is assumed. The statistics of these features are then used in recognition schemes [43]. Recently, spatial and long-term temporal correlations of these local features were considered and promising results shown. The work in [17] models the video as a time-series of frame-wide feature histograms and brings the temporal aspect into picture. A matching kernel using "correlograms" was presented in [48], which looked at the spatial relationships. A recent work [13] proposes a match function to compare spatio-temporal relationships in the feature by using temporal and spatial predicates, which we will describe in detail later.

Often, there are not enough training videos available for learning complex human activities; thus, recognizing activities based on just a single video example is of high interest. An approach of creating a large number of semi-artificial training videos from an original activity video was presented in [49]. A self-similarity descriptor that correlates local patches was proposed in [50]. A generalization of [50] was presented in [51], where spacetime local steering kernels were used.

1.3 Contributions of the Work

The objective of this work is to design algorithms which can recognize activities in wide-area continuous videos. The study has been carried out in three parts.

We start by proposing a flow-based activity recognition system which identifies regions of interest in a wide-area video and models them using a combination of shape matching and subspace analysis [52, 53]. This system can scale up to activities involving multiple actors [54]. However, activities are analyzed individually and assumed to be independent of other activities in the scene.

Wide-area continuous videos often contain activities which are potentially related to each other and might influence each other. Therefore, as a next step, we explore the contextual relationships in wide-area continuous videos using graphical models [55]. With the assumption that the locations of activities in a video are pre-determined using existing approaches, we then show how a Markov random field can be built using the nodes of the graph as activities and edges representing their spatiotemporal relationships.

While such a graph is successful in modeling context, it is often found that existing activity classifiers are not always accurate in localization of activities. There is also the task of tracking which is challenging in wide-area videos in the presence of multiple targets. As a next step, we explore relationships between tracks and activity segments. We also do not assume fixed location of activities. We propose a bi-directional approach where we demonstrate the influence of tracks on activities and vice-versa for simultaneous tracking, localization and recognition of activities.

Finally, we also propose a novel approach to learn the structure of the graphical

model automatically. Given a set of training data, the optimum structure for the graphical model is chosen using an L1-based regularization of the parameters of the graph. In each case, experiments are conducted on recent wide-area surveillance datasets and improvement in results is demonstrated.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows. We present a method for multi-person activity recognition in wide-area videos in Chapter 2. This work assumes activities to be independent of each other and models them individually. We discuss the limitations of this assumption and motivate the use of spatiotemporal context in activity recognition. Next, we present a graphical model based approach for activity recognition in Chapter 3. Here, the activity regions are computed in the pre-processing stage and the graphical model is used to learn the contextual relationships. In Chapter 4, we extend the graphical model based approach to the case where activity locations as well as tracks are not assumed to be constant. Finally, we discuss the structure learning of graphical models using L1-regularized parameter estimation in Chapter 5. We conclude the thesis in Chapter 6 with a summary of the work and discussion of future directions.

Chapter 2

Multi-person Activity Recognition in Wide-area Videos

2.1 Introduction

Natural videos usually consist of multiple motion patterns generated by objects moving at arbitrary speeds and time intervals. They could have multiple events occurring simultaneously at arbitrary viewpoints and varying scales. The analysis of such videos can be termed as complex activity recognition. Recognition of complex activities often involves dealing with features distributed in a high dimensional space due to a higher amount of intra class variations. Algorithms dealing with such sequences should be robust to background clutter, noise and changes in viewpoint and scale. Most of the traditional activity recognition algorithms, such as [56] [2] [57], work with simpler datasets like [7] [2] which place assumptions on the number of objects, scale and viewpoint of the scene. However,

in real world situations it is hard to encounter such videos. Therefore, there is a need for algorithms which can handle the structure and semantics of complex activities.

A scene is a collection of moving pixels. Optical flow provides a natural representation for this motion. It represents the pixel-wise motion from one frame to the next; therefore, it captures the spatial and temporal dynamics in a video. Since a complex activity involves multiple motion patterns, it is useful to separate the motion patterns before modeling them, to reduce the search space. One way of doing this would be to compute tracks. However, it is not always feasible to compute accurate tracks in real world videos. The use of optical flow would eliminate this need for computing tracks. The problem of separation of motion pattern reduces to the problem of segmentation of optical flow. Although prone to the same inaccuracies as tracks, optical flow is computed for every pixel in the video. It is therefore, a more statistically reliable indicator in the presence of noise. These factors motivate us to use optical flow as the input features for our recognition algorithm.

In this work, we recognize activities by analyzing the underlying pixel-wise motion using optical flow. Each region in a video where the pixels exhibit similar motion is said to constitute a motion pattern. Individual motion patterns are considered as "events" which can be identified by segmenting the flow patterns. This motion pattern could be due to one or more objects in the scene. An activity is represented as a collection of motion patterns. Optical flow is represented using streaklines which are obtained by integrating the flow over time. The activity in a video could be composed of multiple such motion patterns, which are assumed to be correlated. Therefore, the overall match score between two videos is obtained by matching the individual motion patterns. The streaklines which constitute

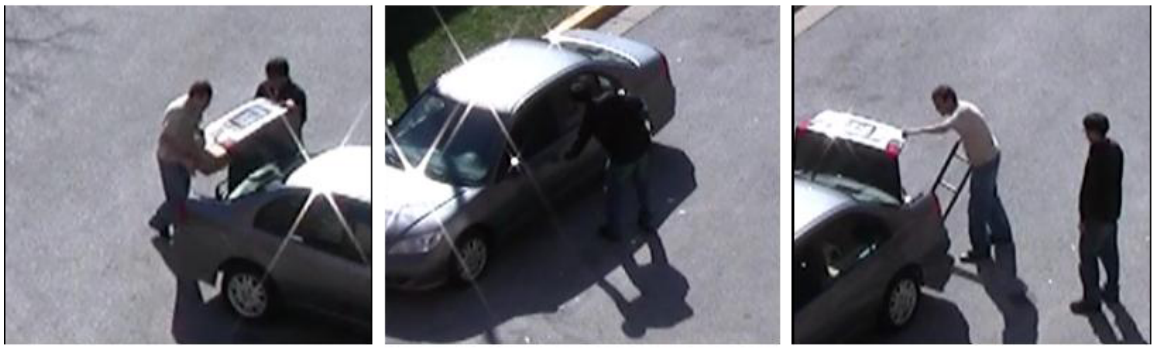

Figure 2.1: The figure shows sample frames of the VIRAT dataset used for recognition. The first figure shows a person loading a trunk, the second figure shows a person entering a vehicle and the third figure shows a person closing a trunk. We notice other people in the scene adding to background clutter. Lighting changes and shadows add noise to the data.

a motion pattern can be identified using their average shape vectors and spatio-temporal variation with respect to the average shape. This variation is modeled using a collection of linear subspaces which capture their spatio-temporal variation in a low dimensional representation. These patterns can be matched by a combination of shape comparison and subspace analysis. We validate the robustness of our algorithm by experimenting on two realistic outdoor datasets. We do not place any assumptions on the number of motion patterns in the scene. The proposed method can be used across a wide range of activities with varying scales and viewpoints. Some sample frames of the data used for activity recognition are shown in Figure 2.1.

These are the definitions of some of the commonly used terminology in the chapter:

Motion Pattern - A spatio-temporal region in a video in which all pixels exhibit similar motion. Each motion pattern is considered as an **"event"** in the video.

Activity - The action which is to be recognized in a video. An activity is composed of one

or more events.

Streakline - The locus of all points in a video which have passed through a particular pixel.

Particle - An abstraction of a point on a streakline.

Motion Region - Those streaklines in the video which correspond to distinctive motion and are used for recognition.

2.1.1 Main Contributions

The salient features of this chapter are:

1. We provide a unified framework for activity recognition in wide-area videos. The proposed system can perform a bottom-up analysis starting from pixel wise motion to identifying motion regions in the volume to segmentation and modeling of these regions. Some state-of-the-art methods like [1] and [42], which deal with similar datasets, explore spatio-temporal information at a feature level. Our method on the other hand explores spatio-temporal information at a global level. This has the advantage that we can segment out different events occurring in the video and then model them in a single framework, unlike these competing methods which would build a model over all interest points in the video (or we will have to use a different segmentation algorithm). Thus, we propose a framework based on the analysis of flow that is able to handle the entire image analysis pipeline - from the low level to the high level processing.

2. Another contribution of this work lies in the use of optical flow for multi-object behav-

ior analysis. Unlike previous methods which utilize optical flow in the form of motion statistics [17], we model the actual dynamics of flow rather than using histograms which do not retain the spatial and temporal information. Therefore, we provide a framework for representation and comparison of complex activities using optical flow.

3. Although we have built upon the work in [58] which uses streaklines and Helmholtz decomposition for crowd segmentation, there are several differences in our work as compared to theirs in the modeling and in the application of streaklines. First, the objective of the proposed method in [58] is to segment a video into different regions exhibiting similar motion, whereas our objective is to explicitly model every motion pattern in a video for the purpose of activity recognition. In [58], the authors propose a method to perform a space segmentation of the streaklines at every frame, whereas we deal with spatio-temporal segmentation of the entire volume. We compute the distance between critical points to identify time segments of motion patterns. In [58], the Helmholtz decomposition is again used to compute a divergence factor, which is then used to identify abnormal activities. Here, we use the Helmholtz decomposition to identify the regions which are of interest to us for the purpose of modeling and recognizing activities. Therefore, we have extended the method in [58] to work not just on crowded environments but also in videos which contain sparse motion.

2.1.2 Related Work

A major thrust of research in complex activity recognition has been in the selection of features and their representations. Different representations have been used in activity

recognition, most of which can broadly be classified as local or global representations [12]. Local representations like [57] [56] identify small spatio-temporal regions in the video as the regions of interest. The spatial and temporal modeling of activities is then performed in the recognition stage. Global representation like [59] [17] on the other hand, model the scene as a whole. These representations often span a larger spatio-temporal volume, so the spatial and temporal information is captured in the features themselves. Methods such as [60] and [42] use STIP-based features for recognition of complex activities. The recognition is then performed by modeling relationships between these features in a complex graph based or histogram based framework. We hypothesize that representing motion patterns using optical flow is more intuitive than using spatio-temporal features since the spatio-temporal information is embedded in the flow. This therefore, is a global representation. Also, unlike previous global methods which use histograms of optical flow, we explicitly model the spatial and temporal evolution of flow.

Optical flow has widely been used in the past for activity recognition. It serves as an approximation of the true motion of objects projected onto the image plane [12]. Optical flow has predominantly been used in features like space-time interest points (STIP) [19] as a part of the feature descriptor. The time series of histogram of optical flow has been modeled as a non-linear dynamical system using Binet-Cauchy kernels in [17]. Optical flow histograms have also been used to analyze the motion of individual players in soccer videos [22]. Most of such approaches utilize the statistics of optical flow for recognition rather than the flow itself. This removes the spatio-temporal structure from the flow. They also assume that the flow belongs to one object in the scene. Optical flow has been extensively

used in crowd motion analysis. Dense crowd motion analysis and segmentation has been performed using optical flow in [61]. Helmholtz decomposition has been used to segment different motions in crowd scenes by streakline computation in [58]. In contrast to the above trends, we show how flow-based methods can be used in the analysis of multi-object scenes with sparse motion.

2.2 Overview of Proposed Approach

The overall algorithm is described in Figure 2.2. The goal of our algorithm is to model the activity in a video as a combination of motion patterns. There are two components to the algorithm - identification of motion patterns and modeling and comparison of motion patterns.

The identification of motion pattern involves identifying regions in the video which correspond to useful motion and segmenting these regions into individual motion patterns. These regions of interest are termed as motion regions. We start by computing the optical flow at each time instant. Optical flow is highly susceptible to noise which can result in spurious patterns which are difficult to analyze. Therefore, we work with streaklines which are obtained by integrating optical flow over time. Motion regions are then identified as the streaklines which show a significant amount of motion. We demonstrate a framework based on the Helmholtz decomposition of a vector field to extract these regions.

Once we identify the streaklines which correspond to the motion regions in a video, motion patterns are recognized by performing a space-time clustering on these streaklines. We demonstrate a method of identifying time segments of streaklines using the Helmholtz

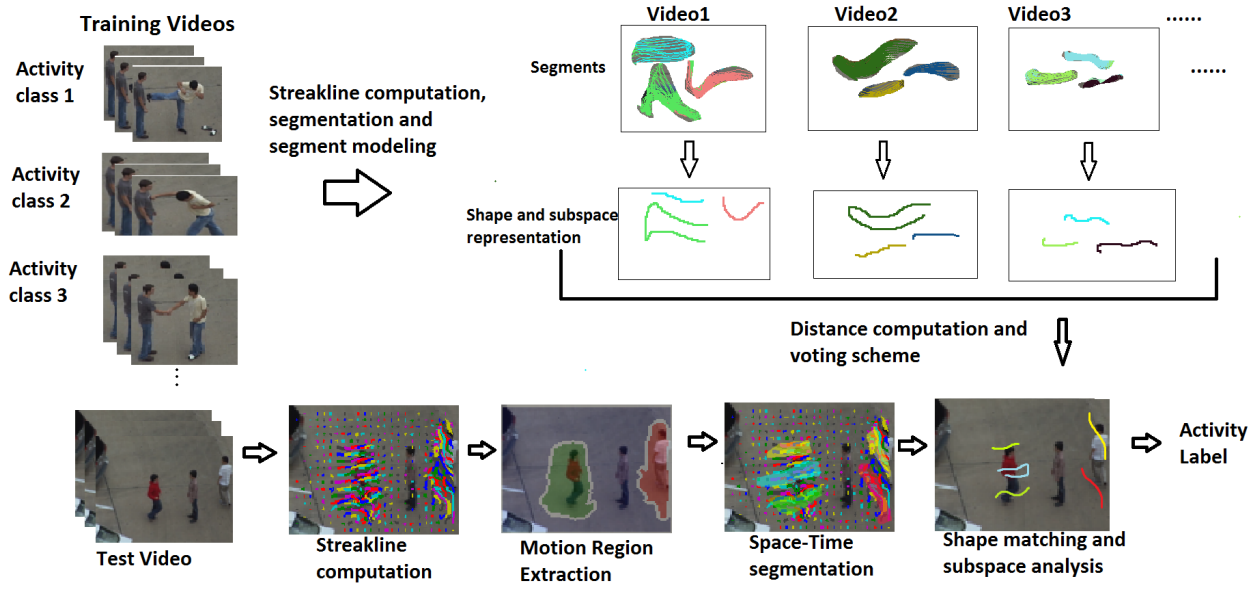

Figure 2.2: The figure shows the overall framework of the proposed method.

decomposition. We further perform a space segmentation by running a clustering algorithm. After the segmentation step, each space-time segment is considered as an individual motion pattern in the video.

After identifying the motion patterns, we need to model them and define a distance measure to compare motion patterns across videos. We compute the average preshape of the streaklines and a linear subspace representation for the spatio-temporal variation about the average preshape for each group of streaklines constituting a motion pattern. Given a set of videos for training and a test video, we compute the models for all the training data. The test data is matched to each of the training data by a combination of shape matching and subspace matching algorithm. The final match score is obtained by a time warping over

the time segments. The test video is classified using a N-nearest neighbor classification.

2.3 Streakline Representation of Motion Patterns

The first step of our algorithm is to represent a video using streaklines. Streaklines are a concept derived from fluid dynamics to represent a time-varying flow field. Suppose we inject a set of particles in the flow field continuously at certain points in the field, the path traced by these particles are called streaklines.

More formally, a streakline is defined as the locations of all particles that passed through a particular point over a period of time. It can be computed by initializing a set of particles at every time instant in the field and propagating them forward with time according to the flow field at that instant. This results in a set of paths, each belonging to one point of initialization. It can be shown that the streakline representation has advantages over other representations like streamlines and pathlines in being able to capture changes in the field as well as in smoothness of the resulting representation. Given a video with n pixels per frame for a duration of N frames, we compute streaklines $s_1, ..., s_n$ where $s_i = [X_i, Y_i]^T$, $X_i = [x_{i,1}, x_{i,2}, ..., x_{i,N}]^T$, $Y_i = [y_{i,1}, y_{i,2}, ..., y_{i,N}]^T$, $s_i \in \mathbb{R}^{2N}$ for $i = 1, 2, ..., n$. Every point on the streakline $(x_{i,t}, y_{i,t})$ corresponds to a particle p initialized at pixel i at time instant t.

The particle p is initialized at the i^{th} pixel of the frame at time instant t. For the subsequent frames, the particle is propagated from its old position $(x_{i,t}^{old}, y_{i,t}^{old})$ to its new position $(x_{i,t}^{new}, y_{i,t}^{new})$ using the particle advection.

Figure 2.3: The figure shows the streaklines for people opening a trunk in two videos. The circled region shows the similarity in the activity captured by the streaklines.

$$\begin{aligned} x_{i,t}^{new} &= x_{i,t}^{old} + u(x_{i,t}^{old}, y_{i,t}^{old}) \\ y_{i,t}^{new} &= y_{i,t}^{old} + v(x_{i,t}^{old}, y_{i,t}^{old}) \end{aligned} \tag{2.1}$$

where $u(x, y)$ and $v(x, y)$ are the X and Y components of the instantaneous optical flow at position (x, y).

Streaklines are ideally suited for motion analysis in video. Because they are computed over a larger interval of time as compared to optical flow, they are more robust to noise and easier to analyze than optical flow. They capture the pixel-wise spatio-temporal information in a video. Similar activities will result in similar streaklines, therefore modeling and comparison of streaklines can be used for activity classification. Figure 2.3 illustrates the streaklines for similar activities being performed in different scenes. We notice that the streaklines look similar in the circled region.

2.3.1 Identification of Motion Regions

Motion in a video is often sparse. In most natural videos, motion is confined to small regions in the video. Since we compute streaklines at every pixel in each time frame, the size of the computed data is the same as the number of pixels in the video. To reduce the computational space and increase efficiency, we first need to reduce the size of the data. This can be done by identifying regions of meaningful motion in the video. We refer to such regions as "motion regions".

There are several ways by which we could identify the motion regions in a video. For example, in [58], the authors perform segmentation on the whole volume and then eliminate small insignificant segments. However, this may not be computationally efficient,

especially if the meaningful regions are small compared to the whole volume. Also, for our purpose, we do not need to identify every single streakline which represents motion. We are interested in those regions in the spatio-temporal volume which are most distinctive for the purpose of recognition. The Helmholtz decomposition has widely been used in the past to recognize distinctive points in a vector field. We utilize this concept derived from fluid dynamics to recognize motion regions.

The Helmholtz decomposition is a concept derived from physics, which states that any smooth field can be uniquely decomposed into an irrotational component and a solenoidal component. The extrema of these components are termed as critical points. In particular, the extrema of the irrotational field occur at regions of high divergence and convergence. Therefore, these would be the distinctive regions of the flow field that we are interested in modeling. Since optical flow is highly transient, we propose to use a flow field, which we call the "motion field" derived from the streaklines to compute the Helmholtz decomposition. We compute an aggregate flow by averaging the value of flow over a set of k frames. This aggregate flow represents the average motion which each pixel has undergone. Next, we apply a smoothing function over this field to make it differentiable. The resultant field is known as the motion field $\mathbf{F}$.

In this section, we will explain in detail, the computation of motion regions from the motion field using the Helmholtz decomposition.

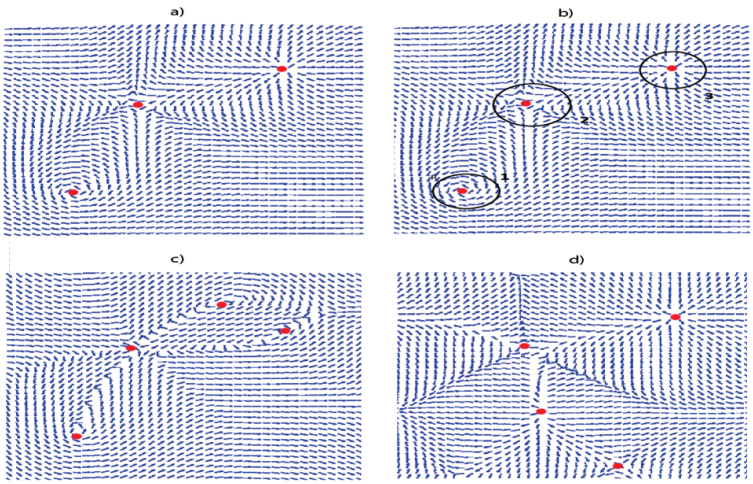

Figure 2.4: Decomposition of a flow field: The figure shows a sample flow field and its decomposition into the irrotational and solenoidal components. The critical points are marked in red on each image. Figure a) shows the original flow field; Figure b) is the original flow field marked with regions containing critical points. We notice that the critical point in region 3 is an attracting focus and the critical points in region 1 and 2 are repelling nodes; Figure c) represents the solenoidal component of original flow; and d) represents the irrotational component of original flow. We can see that the irrotational field has no rotational component and the solenoidal field is divergence free (purely rotational).

2.3.2 Helmholtz Decomposition of Flow Field

The Helmholtz decomposition theorem states that any arbitrary vector field which is assumed to be differentiable can be decomposed into a curl free (irrotational) component and a divergence free (solenoidal) component [62], i.e.,

$$\mathbf{F} = \mathbf{F}_{sol} + \mathbf{F}_{irr}, \tag{2.2}$$

where $\mathbf{F}$ is the overall field, $\mathbf{F}_{sol}$ represents the solenoidal component and $\mathbf{F}_{irr}$ represents the irrotational component, $\mathbf{F} \in \mathbb{R}^{m \times n}$ where $m \times n$ is the video frame size.

Since $\mathbf{F}_{sol}$ is divergence free, we have $\nabla \mathbf{F}_{sol} = 0$. Similarly, since $\mathbf{F}_{irr}$ is curl free, we have $\nabla \times \mathbf{F}_{irr} = 0$. We can also define a scalar potential ϕ and a vector potential $\mathbf{A}$ such that

$$\mathbf{F} = -\nabla\phi + \nabla \times \mathbf{A} \tag{2.3}$$

We see an illustration of the Helmholtz decomposition of a vector field in Figure 2.4. We notice that the first component is purely a rotational field whereas the second component is purely divergent. Below, we will illustrate the extraction of regions of interest from the motion field using this decomposition.

Computing the Flow Field Components

According to the Helmholtz decomposition, the motion field is composed of an irrotational and solenoidal component. We also mentioned that the motion field can be expressed in terms of a scalar potential (ϕ) and a vector potential ($\mathbf{A}$). We can obtain the irrotational and solenoidal components of the motion field from the scalar and vector

potentials respectively. We will follow the technique described in [62] to solve for the scalar and vector potentials. The scalar potential can be obtained by projecting onto the curl-free component and solving the following variational problem:

$$\arg\min_{\phi} \int_{\Lambda} \|\mathbf{F} + \nabla\phi\|^2 dA, \Lambda \subset \Re^2 \tag{2.4}$$

where Λ is the image domain under consideration and A is the area. It can be shown that the solution to ϕ is obtained by solving the following Poisson equation[62]:

$$\nabla.\mathbf{F} = \nabla^2\phi \tag{2.5}$$

$$\mathbf{F} + \nabla\phi.\hat{n} = 0 \text{ in } \partial\Lambda \tag{2.6}$$

where $\hat{n}$ is the unit outward normal to the boundary $\partial\Lambda$.

A similar formulation can be derived for the vector potential. The solenoidal component can be solved using the following variational problem.

$$\arg\min_{\mathbf{A}} \int_{\Omega} \|\mathbf{F} - (\nabla \times \mathbf{A})\|^2 dA, \Omega \subset \Re^3, \tag{2.7}$$

the optimum solution of which is obtained by the following PDE formulation:

$$\nabla \times \mathbf{F} = \nabla \times \nabla \times \mathbf{A} = \nabla(\nabla.\mathbf{A}) - \nabla^2\mathbf{A} \tag{2.8}$$

$$\mathbf{F} - (\nabla \times \mathbf{A}) \times \hat{n} = 0 \text{ in } \partial\Omega. \tag{2.9}$$

Here $\hat{n}$ is the unit outward normal to the boundary $\partial\Omega$. Since we have the the curl ($\nabla\times$) to be an operator in three dimensions, for an arbitrary $\mathbf{A}$ we need to extend the two dimensional field $\mathbf{F}$ to 3D by setting the z-component to zero.

On solving the above equations, we obtain the scalar potential ϕ and the vector potential $\mathbf{A}$. The irrotational and solenoidal components of the flow field are accordingly

obtained as

$$\mathbf{F}_{irr} = \nabla\phi \tag{2.10}$$

$$\mathbf{F}_{sol} = \nabla \times \mathbf{A} \tag{2.11}$$

2.3.3 Motion Regions Using the Helmholtz Decomposition

The irrotational component of the Helmholtz decomposition carries useful information about the sources and sinks of the motion field. These sources and sinks are a result of motion in a video, therefore they can be used to identify regions of motion in the video. The sources and sinks are also known as critical points. A point $C(x_0, y_0)$ is defined as a singular/critical point of the vector field if $C(x_0, y_0) = (0, 0)^T = 0$ and $C_1(x, y) \neq 0$ for any other point C_1 with coordinates $x \neq x_0, y \neq y_0$ in the neighborhood of (x_0, y_0).

Consider a point $\mathbf{v}(x, y)$in the irrotational field in 2D given by

$$\mathbf{v}(x, y) = \begin{bmatrix} u(x, y) \\ \\ v(x, y) \end{bmatrix}$$

The Jacobian matrix of the irrotational field at a point (x, y) on the field denoted by J_v is given by

$$J_{\mathbf{v}} = \begin{bmatrix} u_x & u_y \\ \\ v_x & v_y \end{bmatrix}$$

where u_x and v_x are the partial derivatives of u and v with respect to x and u_y and v_y are the partial derivatives of u and v with respect to y. The determinant of the Jacobian at (x, y) is denoted as $|J_{\mathbf{v}}|$. The critical points are identified by finding those points in the field where u and v are zero, but $|J_{\mathbf{v}}| \neq 0$. The critical points of the vector field and its

components from Helmholtz decomposition are marked in Figure 2.4.

As mentioned before, the critical points of the irrotational field occur in regions of high convergence and divergence in the field. Intuitively, these would be the most distinctive regions of the motion field, and therefore, we would want to model the streaklines which correspond to these regions. Therefore, we define a motion region as a set of streaklines which pass within a small distance of a critical point. Here, we set the distance as 5 pixels for a frame size of 150×200, however, this distance can be modified based on the resolution of the video. An example of the motion regions identified using critical points is shown in Figure 2.5.

2.4 Segmentation of Motion Patterns

The motion information in a video is contained in the form of motion patterns. Each video could contain multiple motion patterns, each said to correspond to an "event". These motion patterns vary in time durations as well as in space. Activity recognition in such videos requires modeling of the motion patterns as well as studying the spatio-temporal relationships between them. We perform activity recognition in two steps - identification of motion patterns and modeling of motion patterns.

An activity in a video can be composed of one or more motion patterns. Since we are dealing with complex, real-world scenarios, there could also be motion patterns which are introduced by background clutter or noise. To make our algorithm robust to these factors, we do not place any assumptions on the number or locations of motion patterns in the scene. Our next task therefore, is to identify motion patterns. Because we represent a

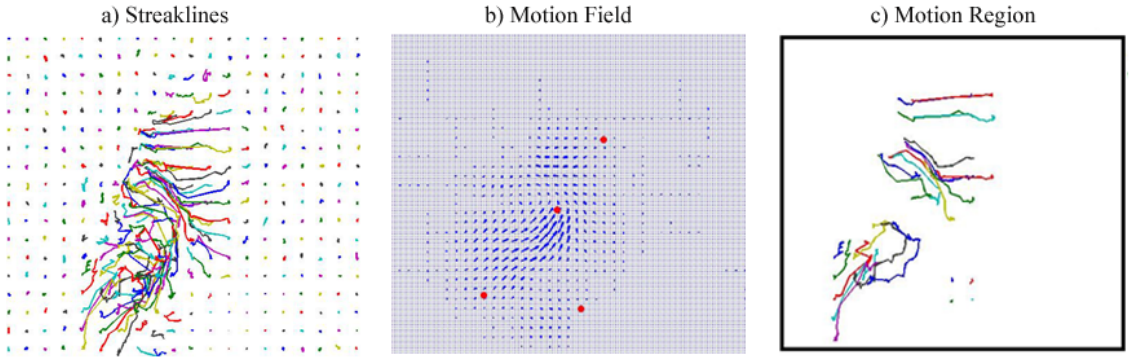

Figure 2.5: The figure shows the extraction of motion regions from streaklines. Figure a) shows the streaklines of the action "open trunk". Figure b) shows the corresponding motion field. The critical points of the motion field are marked in red. The streaklines extracted using these critical points are shown in Figure c) and constitute the motion regions of the video.

video as a group of streaklines, the task of identification of motion pattern is performed by a segmentation of streaklines. We segment the streaklines both in time and space domain.

2.4.1 Time Segmentation of Streaklines

We propose that the critical points extracted using Helmholtz decomposition can also be used for time segmentation of streaklines. This is based on the observation that whenever there is not much change in the motion pattern from one time instant to another, the location of critical and their characteristics do not change much. On the other hand, when a new motion pattern originates, a new critical point emerges, or when an existing motion pattern ends, a critical point disappears. Therefore, by associating the critical points from one frame to the next, we can identify the start and end points of motion patterns. Each critical point is associated with a motion region. Therefore, a motion region exists

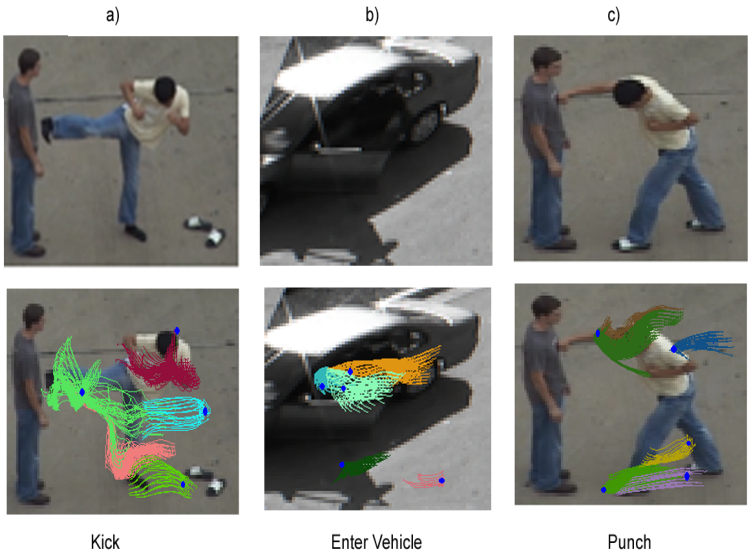

Figure 2.6: Examples of time segmentation of streaklines using the Helmholtz decomposition. The first row shows a sample frame and the second row displays the time segmented streaklines. Each segment is marked in a different color. The critical points are marked in blue.

in the duration in which the corresponding critical point is observed. To associate critical points from one frame to the next, we use the following distance measure as described in [63].

Every singular point $C(x, y) = (u(x, y), v(x, y))^T$ is mapped to a circular coordinate system $(\gamma(x, y), r(x, y))$ given by

$$\cos\gamma = \frac{u_x + v_y}{sqrt(u_x + v_y)^2 + (v_x - u_y)^2} \tag{2.12}$$

$$\sin\gamma = \frac{v_x - u_y}{sqrt(u_x + v_y)^2 + (v_x - u_y)^2} \tag{2.13}$$

$$r = \frac{1}{2} + \frac{u_x v_y - v_x u_y}{u_x^2 + u_y^2 + v_x^2 + v_y^2} \tag{2.14}$$

where u_x, u_y, v_x, v_y are elements of the Jacobian of the singular point C denoted by J_C. The similarity measure between two singular points is given by the Euclidean distance between

them in the (γ, r) plane as defined in Equation (2.15).

$$d_c(C_i, C_j) = \sqrt{r_1^2 + r_2^2 - 2r_1 r_2 \cos(\gamma_1 - \gamma_2)} \tag{2.15}$$

Therefore, we compute the critical points in every frame and compute the distance between critical points from one frame to the next using Equation (2.15). It is seen that the critical points that arise due to the same event in adjacent frames have a very small distance and can therefore be associated. Whenever a new critical point arises, a new event is said to begin and when the critical point disappears, an event ends. The streaklines that belong to the motion region associated with the critical point in the time interval in which a critical point is observed is said to constitute the time segment. Figure 2.6 shows some examples of time segmentation using our algorithm.

2.4.2 Segmentation of Streaklines in Space

Each video segment could be made up of more than one motion pattern. Each motion pattern could correspond to one object in a scene, or a part of an object in the scene. We therefore, perform a clustering of streaklines in space such that the streaklines in each individual cluster exhibit similar motion. Each cluster is said to belong to one motion pattern or event in the video. To perform a segmentation of the motion patterns, we will first transform the streaklines into a shape space. The shape representation of streaklines is given below:

Shape Representation of Streaklines: Consider a streakline $s \in \mathbb{R}^{2k}$ in a time segment of length k.

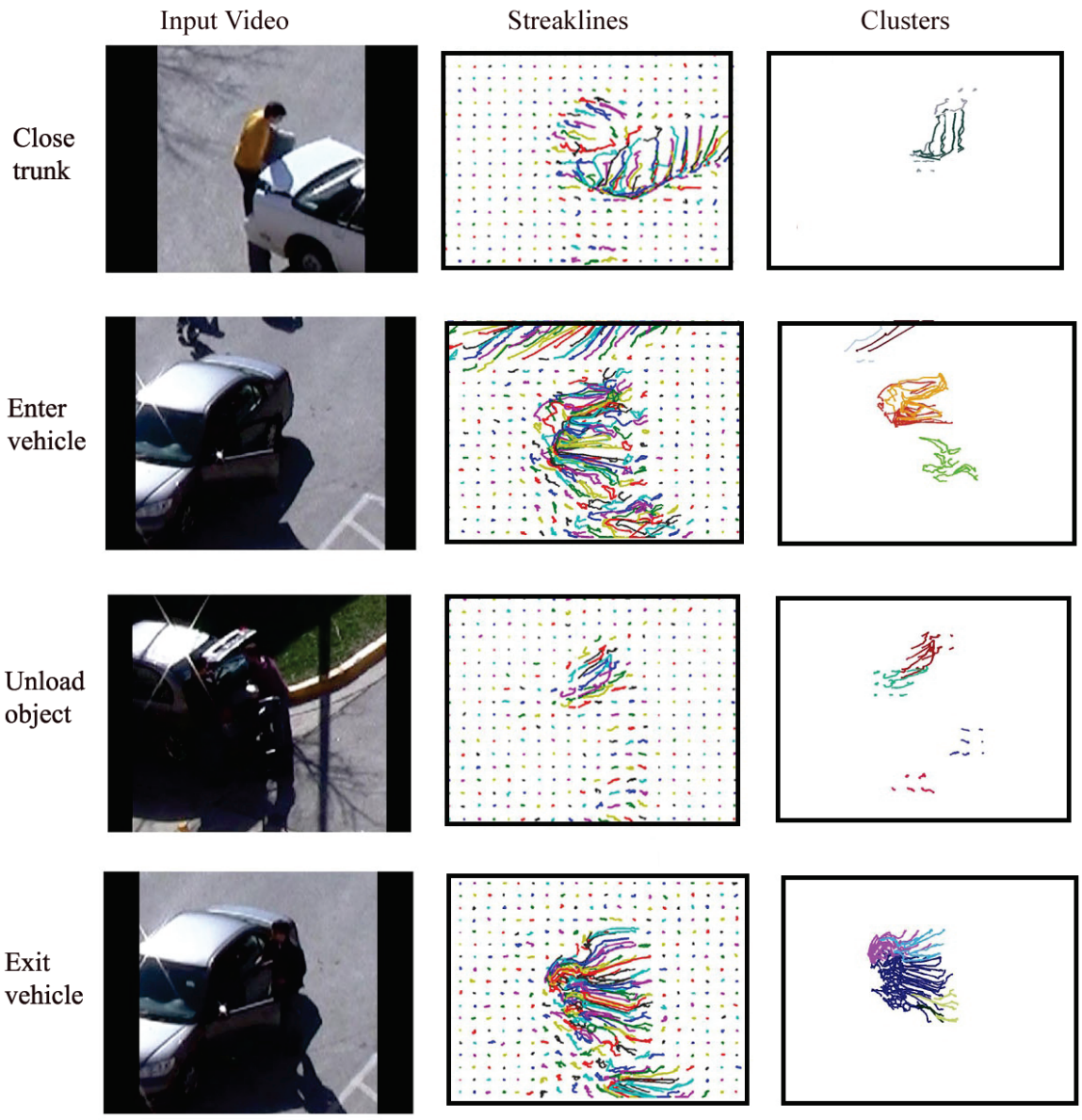

Figure 2.7: The figure shows the streaklines and the clusters for activities in the VIRAT dataset. The clusters are marked with different colors.

$$s = \begin{bmatrix} x_1 & x_2 & . & . & x_k & y_1 & y_2 & . & . & y_k \end{bmatrix}^T$$

Next, we remove the scaling and translation from s to obtain a normalized vector c. This is done by subtracting the mean from s and scaling to unit norm.

$$c = \frac{Ps}{\|Ps\|}, \tag{2.16}$$

where $P = I_{2k} - \frac{1}{k} I_{D_{2k}}$, I_{2k} being the $2k \times 2k$ identity matrix and $I_{D_{2k}}$ is the $2k \times 2k$ matrix given by $I_{D_{2k}} = \begin{bmatrix} 1_{k\times k} & 0_{k\times k} \\ 0_{k\times k} & 1_{k\times k} \end{bmatrix}$, where $1_{k\times k}$ is a $k \times k$ matrix of ones. This normalized vector is independent of translation and scale and is called a preshape vector of a collection of points [64].

Extraction of Motion patterns: Suppose a video is made up of p underlying motion patterns $[M^1, M^2, ..., M^p]$. Let each motion pattern M^i, $i\epsilon\{1, 2, .., p\}$ contain n^i pre-shape vectors $[c_1^i, c_2^i, s_3^i, ..., c_{n^i}^i]$, where $c_j^i \in \mathbb{R}^{2k\times 1}$. Let $[\tilde{M}^1, \tilde{M}^2, ..., \tilde{M}^p]$ be our estimates of the motion pattern. Estimation of the motion patterns can be performed in a clustering framework. Here, we use the average preshape of the motion pattern as the representative model for clustering. The average preshape of a motion pattern M^i is given by

$$\bar{c}^i = \sum_{j=1}^{n^i} c_j^i \tag{2.17}$$

The projection error between a preshape c and an average preshape $\bar{c}^i$ of motion pattern M^i is calculated as the square of the Euclidean norm of their distance, $\|c - \bar{c}^i\|^2$. Therefore, a preshape $c \in M^i$ if

$$\|c - \bar{c}^i\|^2 \leq \|c - \bar{c}^{i'}\|^2, \forall i' \neq i \tag{2.18}$$

where we are using the Euclidean norm as an approximation of the actual pre-shape vector norm (Procrustus distance). The above clustering problem can be solved using a standard k-means clustering framework.

Because we do not want to assume the number of motion patterns in the video, we set a threshold on the model residue ε_{thresh} and compute p such that

$$\sum_{i=1}^{p}\sum_{j=1}^{n^i} \|c_j^i - \bar{c}^i\|^2 \leq \varepsilon_{thresh} \tag{2.19}$$

Some examples of space segmentation are shown in Figure 2.7.

2.5 Activity Modeling and Recognition

In the previous section, we computed a set of motion patterns as well as the average preshape of each motion pattern. This average preshape provides us with the mean path traced by the object or part of the object which is involved in the event. The average preshape $\bar{c}^i$ of motion pattern M^i therefore, can be used to model the motion pattern. Apart from the average preshape, the streaklines can be characterized by their spatio-temporal evolution. To make this evolution independent of its location, we model the evolution as the variation of the preshapes of a motion pattern about the average preshape. Each motion pattern M^i contains n^i preshapes of length k^i. The spatio-temporal evolution of preshapes can be modeled by examining the linear subspaces along which there is maximum variation in the data. This can be achieved by a subspace analysis of the data. The task of activity classification requires a comparison between the average preshape as well as the similarity between their subspaces. In this section, we will explain these steps in detail.

2.5.1 Comparison of Average Preshapes

Consider two preshape vectors c^i and c^j of motion patterns M^i and M^j. To compare c^i and c^j, we first need to ensure that they are of the same length. This is done by resampling the preshape vectors to a length l. Here, l can be a constant or a function of the duration of the time segment. The distance between the resampled preshape vectors can be measured by the full Procrustes distance [64] which is the Euclidean distance between the Procustes fit of the preshapes $\bar{c}^i$ and $\bar{c}^j$. The Procustes fit $(\beta, \theta, (a + jb)$ is chosen to minimize the distance given by

$$d_s^{(i_1,i_2)} = \|\bar{c}^i - \bar{c}^j \beta \exp^{j\theta} - (a + jb)1_l\|, \tag{2.20}$$

where β is the scale, θ is the rotation and $(a + jb)$ is the translation, 1_l is the l dimensional column vector of ones. Since the preshapes have already been normalized, the estimated scale $\beta \approx 1$ and the estimated translation $(a + jb) \approx 0$. The rotation will be obtained as $\theta = \arg(c^{i\mathrm{T}} c^j)$.

2.5.2 Subspace Analysis

Let the preshapes constituting a motion pattern M^i be $C^i = c^i_j, j = 1..n^i$, where n^i is the number of streaklines in M^i. Since the average preshape captures the average motion in M^i, we wish to model the spatio-temporal variation in the motion pattern M^i using subspace analysis. We use the preshape vector c^i to compute a linear subspace representation for M^i. A linear subspace representation for C^i can be computed by a

principal component analysis of the covariance matrix of C^i given by

$$R^i = \frac{1}{n^i} \sum_{j=1}^{n^i} (c_j^i - \bar{c}^i)(c_j^i - \bar{c}^i)^T \tag{2.21}$$

where R^i is the covariance matrix. We choose the first r eigenvectors $V_1^i, V_2^i ... V_r^i$ of R^i as the orthogonal vectors for the low dimensional representation of C^i. The value of r is chosen experimentally.

The similarity between the subspace representation of motion patterns M^i and M^j is given as the sum of the r principal angles between the corresponding subspaces [65], i.e.

$$d_\theta^{(i,j)} = \sum_{m=1}^{r} \arccos(V_m^{i\mathrm{T}} V_m^j). \tag{2.22}$$

2.5.3 Overall Distance Computation

The total distance between a training and test video is computed as follows: For the training sequences, it is assumed that the motion patterns pertaining to the training activity have been identified and modeled. For the test sequence, there could be a different number of motion patterns. For every motion pattern in the training data, we find the closest motion pattern in the test data. This distance is computed as follows:

Consider a training video with n_r motion patterns and a test video with n_T motion patterns. The distance between a motion pattern M^i in the training video and M^j in the test video is given by the weighted average

$$d(i,j) = w_1 d_s^{(i,j)} + w_2 d_\theta^{(i,j)}, , \tag{2.23}$$

where d_s and d_θ are the shape and subspace distances given in Equations (2.20) and (2.21).

w_1 and w_2 are the weights which are set such that the overall distance lies in the range $0-1$. These weights are determined using training data. For each motion pattern M^i, we choose the best match as that motion pattern in the test video which has the least distance D^i. The total distance between a training and a test video is given by the sum of the best match distances for all motion patterns, i.e.,

$$D = \sum_{i=1}^{n_r} D^i \tag{2.24}$$

We use a k-nearest neighbor classifier for recognition of activities. i.e. considering the k closest training clips, the activity is classified as that category to which most of the k neighbors correspond. Therefore, the steps in recognition of activities using our algorithm are as follows:

1. For each training video v, compute the motion patterns $M^1, M^2...M^{p_v}$. Model each motion pattern M^i using the average preshape $\bar{c^i}$ and r eigenvectors $V_1^i, V_2^i, ..V_r^i$.

2. For the given test video t, compute the motion patterns and the model for each motion pattern. The distance between every motion pattern M^i in the training video and M^j in the test video is computed using Equation (2.23).

3. For each motion pattern M^i, the least distance D^i with a test motion pattern is chosen as the best match.

4. The total distance between two videos is given by the sum of distances of the best match between their motion patterns.

5. Compute the distance between every training video and the test video. The activity

in the test video is classified using a k-nearest neighbor classifier.

2.6 Experiments

To validate our approach, we perform experiments on two publicly available complex datasets. Each of these datasets involve outdoor scenes and multiple actors interacting in the presence of noise and background clutter.

2.6.1 Dataset

The first set of experiments are conducted on the UT Interaction dataset [8]. This dataset consists of high resolution video of two actors performing actions such as handshake, kicking, hugging, pushing, punching and pointing. Each task is performed by 10 actors in outdoor environments. Each video is of a duration of aproximately 3 seconds. Often there are people walking or performing other activities in the background, causing background clutter. We test our method on this dataset to validate the use of our method for analysis of articulated motion. We demonstrate and compare our results with three previous methods which use the same dataset.

The second set of experiments were conducted on the VIRAT dataset. The VIRAT public dataset [9] contains activities involving people-people and people-vehicle interactions. The people-vehicle activities include person opening and closing the trunk, person entering and exiting a vehicle and person loading and unloading objects from the vehicle. Often, there are other people moving in the scene causing background clutter. There is variation in the scale as well as orientation of objects in the dataset. Often, there are shadows or

occlusions leading to a high amount of noise in the scene.

As mentioned before, the critical points of the irrotational field occur in regions of high convergence and divergence in the field. Intuitively, these would be the most distinctive regions of the motion field, and therefore, we would want to model the streaklines which correspond to these regions. Therefore, we define a motion region as a set of streaklines which pass within a small distance of a critical point.

2.6.2 Results on UT Interaction Data

Our method performed well on the UT Interaction data. The videos are of different lengths and the activities are performed from two different viewpoints. We computed streaklines over the entire video. The motion regions were found to be concentrated around the limbs of the persons involved due to the nature of the activities. It was found that most of the activities were composed of two to three events. For example, the "pointing" action is composed of the person raising his hand and then lowering it. Similarly, "shaking hands" is composed of two people approaching each other, shaking their hands and dispersing. The spatial segmentation separated out the articulated motion in the video. Some examples of retrieved results are shown in Figure 2.8. We use a leave one out strategy for activity recognition. 9 out of 10 sequences were used for training and the remaining for testing. It was found that the performance of our method on the UT Interaction dataset was similar to the other state of the art methods like [42] and [1]. We achieved an overall recognition accuracy of 72.0%, while [42] achieves an accuracy of 70.8% and [1] achieves an accuracy of 70.6%. The method worked well on activities like hug and shake hands where the motion patterns

Figure 2.8: Examples of retrieved results for the UT Interaction dataset.

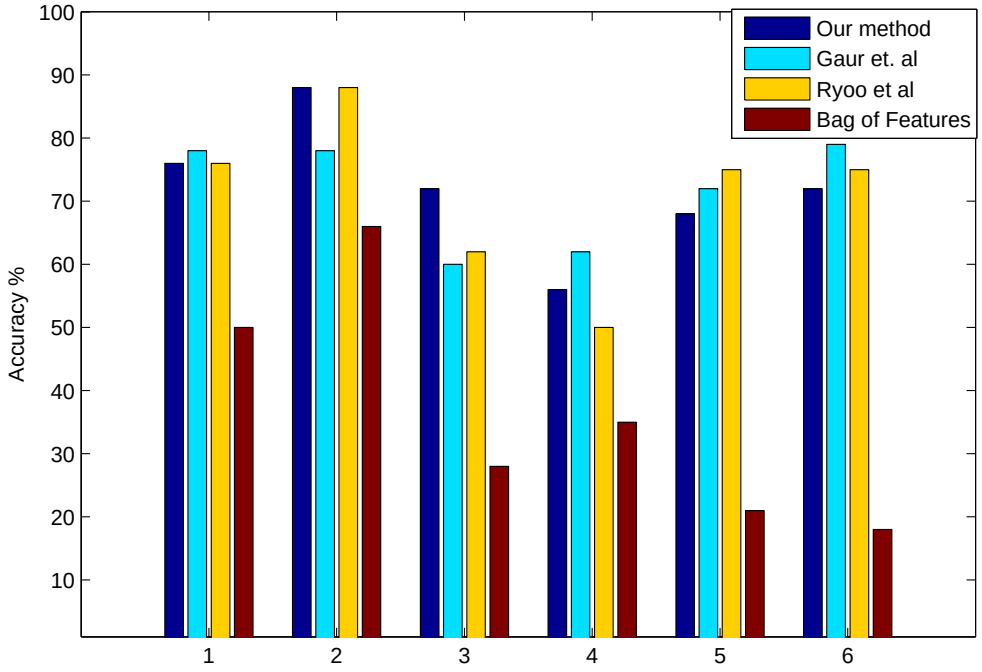

Figure 2.9: The figure shows the accuracy of recognition using the UT Interaction data and comparison with previous methods. The activities are: - 1 - Shake Hands, 2 - Hug, 3 - point, 4 - Punch, 5 - Kick, 6 - Push

were highly distinguishable. The performance for activities like punch and kick were slightly lower since the events were similar to each other. The comparison of our method to other previous STIP-based approaches is shown in Figure 2.9. It can be seen that on an average, our method performs as well as other previous STIP-based methods and better than Bag of Features. The advantage of our method as compared to these previous methods is that the spatio-temporal relationships in a STIP-based method have to be explicitly modeled using graphs or other complex structures. Therefore, as the activities get more complex, the graph gets more complex and the computation increases exponentially. Whereas in our

method, the spatio-temporal relationships are embedded in the streaklines, therefore the computational cost is linear with respect to the number of streaklines in a motion pattern. Moreover, we provided a unified bottom-up analysis framework starting from the low-level features (streaklines), segmenting them into individual regions of interest, identifying events and modeling activities as a combination of events. This is unlike other competing methods which consider the entire volume as a set of features and models them, or requires different tools to do the low level processing (which are not dealt with in detail in those papers). This has been give in more detail in Section 2.6.4.

2.6.3 Results on the VIRAT Data

The experiments conducted on the VIRAT data test the robustness of our approach to the presence of clutter and variations in scale. The generation of normalized preshape vectors from streaklines handles the difference in scale. The rotation invariant shape comparison handles the changes in viewpoint to some extent. Activities like closing and opening trunk consisted of one event, the other activities often consisted of two or three events. For example, entering a vehicle is composed of opening the door, sitting inside the vehicle and closing the door. The space segmentation separated individual objects in the scene and helped in the elimination of background clutter.

A leave one out strategy in conjunction with the N-nearest neighbor was used for classification. The results were compared to that using [1]. The results are shown in Figure 2.11. It can be seen that our results are comparable to other state of the art methods here also. However, as mentioned before, our system presents an entire end-to-end pipeline for

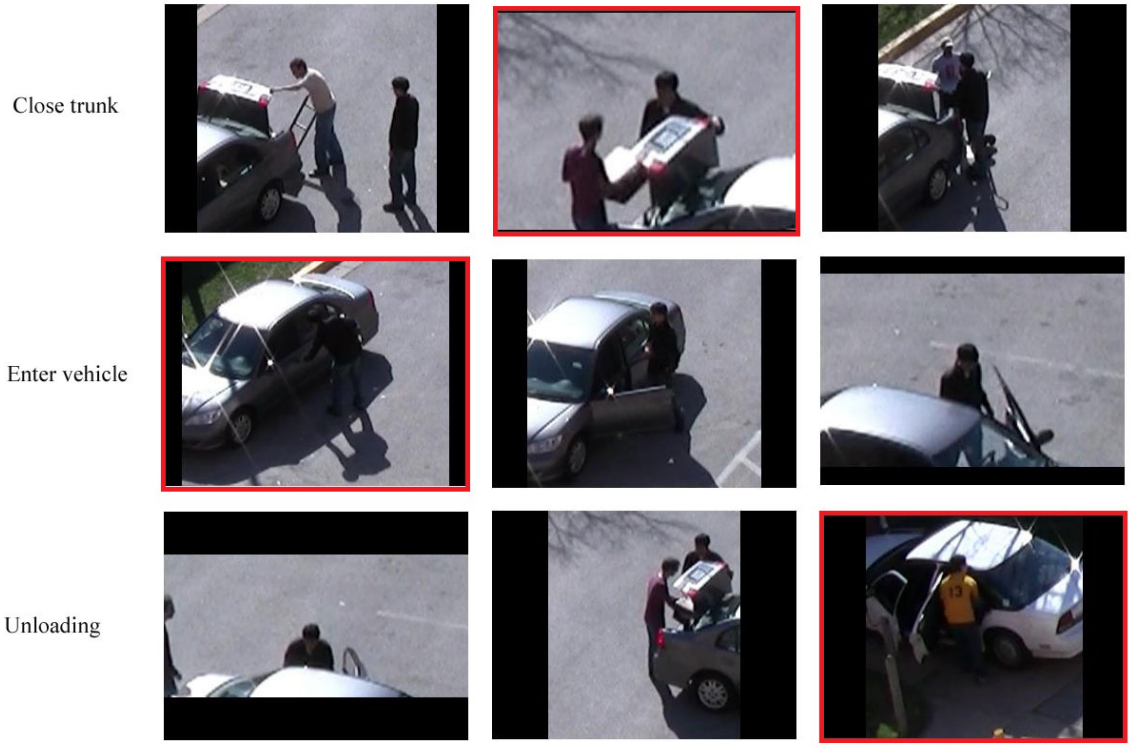

Figure 2.10: Example of results for the VIRAT dataset showing some true positives and false negatives for actions close trunk, enter vehicle and unloading. The false negatives are marked in red.

image analysis and is computationally efficient as discussed in Section 2.6.4. The method performed well in recognizing multi-person activities like people walking together and people approaching each other. The accuracy of recognition for loading and unloading was lower since the events are similar to those in entering and exiting vehicles. Some examples of videos retrieved are show in Figure 2.10. The erroneous results are marked in red. In the first row, it is seen that the second example contains a person carrying an object, and was confused with unloading. This example failed to be retrieved. Similarly, shadows and occlusions have caused false negatives in the second and third rows.

2.6.4 Analysis of the Results

As seen from Figures 2.9 and 2.11, the performance of our method is comparable to that of other state of the art methods. However, the advantage of our method is that, unlike previous methods which try to analyze activities at the feature level, we propose a global approach to activity recognition. This facilitates a bottom-up analysis of a video, where we begin with the streaklines over the entire video, then individual motion patterns, then model and compare these motion patterns. Therefore, our method provides an end-to-end system which computes a set of features, segments out different events and defines a distance measure over them. This is unlike other methods like [42] and [1], where segmentation is not an integral part of the method and has to be performed separately before the activity modeling and recognition can be done.

There is also the advantage of computational efficiency in the modeling and comparison using our algorithm. For a STIP-based method, for example in [1], a graph is

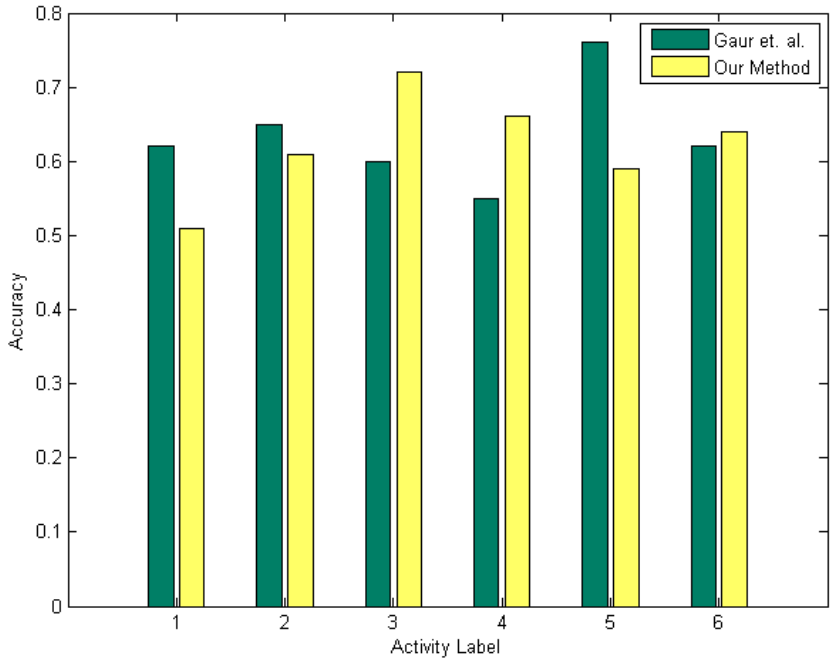

Figure 2.11: The figure shows the recognition accuracy for the VIRAT dataset. The activities are: 1 - loading, 2 - unloading, 3 - open trunk, 4 - close trunk, 5 - enter vehicle, 6 - exit vehicle

matched for every time segment in the test video to every time segment in the training video. The time complexity for matching a graph with V nodes and E edges is known to be $O(V^2E)$. Since the number of edges for a completely connected graph with V nodes is of the order of V^2, we can expect the time complexity of algorithms like [1] to increase exponentially with the number of feature points/nodes. In comparison, consider a motion pattern with N streaklines. Our method computes the mean shape vector for each motion pattern. This requires $O(N)$ operations. Comparison of mean shape vectors using the Procrustes distance is a $O(1)$ operation. It can be shown that the subspace analysis to compute the first k eigenvectors of N streaklines of length p is $O(Nkp)$. Therefore, for a motion pattern with N streaklines, the overall computational cost of modeling and comparison is proportional to $O(N)$, i.e. the complexity increases linearly with the number of streaklines in a motion pattern.

2.7 Conclusion

In this work, we proposed a flow-based system for activity recognition in wide-area videos. We modeled activities as a collection of motion patterns. We demonstrated the use of streaklines to represent and model these motion patterns. The Helmholtz decomposition was used to identify regions of useful motion which were analyzed further. The segmentation of streaklines can be used to separate motion patterns and model them individually. We also showed a method for computing the similarity between two videos using these models. Experiments were conducted on multi-object scenes with a high amount of noise and clutter.

While the proposed system was fairly successful in separating out different activity

patterns in a wide-area scene, the modeling of motion patterns is performed individually. Single person activities are treated the same as multi-person activities. Also, the relationships across activities are not taken into consideration in the model. In reality, continuous videos contain activities which are likely to influence each other. Modeling these relationships can substantially improve the recognition results. For example, in Section 2.6.3, we mentioned that unloading was often confused with carrying a load. While occlusion makes it difficult to distinguish between these two activities, the preceding activity can help in distinguishing the two. Unloading is often preceded by opening a trunk, whereas closing the trunk is often followed by carrying a load. Or carrying a load is followed by opening the trunk. This example demonstrate the limitations of the proposed approach, which can be overcome with the modeling of context. In the next chapter, we will propose a graphical model based approach for context modeling in continuous videos.

Chapter 3

Context Modeling in Continuous Videos Using Graphical Models

3.1 Introduction

The use of context is actively being explored in computer vision today. The use of any data in the video which does not directly correspond to the object or activity being analyzed can be termed as context. Consider a wide area surveillance scene consisting of multiple actors performing a series of activities. Unlike sports videos which are governed by a fixed set of rules, these videos are unconstrained and contain a variable number of objects and activities. By unconstrained, we mean that the activities might be related but do not unfold according to set rules. In such long duration sequences, we can expect that several activities would influence each other causally while some others might occur independently. However, inferring these causalities is not trivial due to the presence of multiple actors. Also,

tracking in such sequences can be challenging due to the presence of clutter and occlusion. In this work, we propose to model the spatio-temporal context between individual activities in a long duration sequence using a Markov random field. Since the number of actors can vary from one sequence to another, we propose to construct the graphical model which is specific to a test sequence.

The key idea behind our approach is that, if two activities are related, they can be expected to occur within a small spatio-temporal vicinity. The spatial separation, temporal separation and the association frequency of these activities can therefore be modeled as context for recognition of these individual activities. Given a collection of videos and a set of baseline classifiers for atomic activities, we wish to learn the spatio-temporal relationships between atomic activities and model them. The relationships are learnt from the training data.

We discuss a context based model for activity recognition in continuous videos. We propose a Markov random field model on the activity nodes, with the edge potentials modeling the spatio-temporal relationships between them. The baseline classifiers (which are assumed to provide a weak classification) give us the node potentials. An inference on this MRF will help us estimate the activities in the sequence.

3.1.1 Contributions

The main contributions of this chapter are the following.

1. We propose a generalized formulation for modeling the contextual relationships between activities in the presence of multiple actors and when they are acting simultane-

ously in the scene. They could be interacting with each other or acting independently.

2. We take a probabilistic approach to modeling relationships between activities. We model the spatial relationships, temporal relationships as well as the association frequencies into the potential functions of a random field model. Inference on this graph gives us the estimate of the categories to which each activity corresponds. We perform experiments on realistic videos containing multiple activities spread over space and time with high amount of clutter and noise.

3.1.2 Related Work

Graphical models are commonly used to encode relationships in video analysis. A grid based belief propagation method was used for pose estimation in [35]. Stochastic and context free grammars have been used to model complex activities in [38]. Co-occurring activities and their dependencies have been studied using Dependent Dirichlet Process - Hidden Markov Models (DDP-HMMs) in [39]. In our work, we propose a Markov random field framework which can handle varying number of actors and activities.

Spatio-temporal relationships have played an important role in the recognition of complex activities. Methods such as [42]and [66] explore spatio-temporal relationships at a feature level. The spatial and temporal relationships between space-time interest points have been encoded as "feature graphs" in [1]. Although such methods have been applied to multiple activities occurring simultaneously, it may not be practical to construct such graphs over long term video sequences and do not explore the relationships across activities. Complex activities were represented as spatio-temporal graphs representing multi-scale

video segments and their hierarchical relationships in [67]. Most of these papers focus on the modeling of low level features for recognition. Variable length Hidden Markov models are used to identify activities with high amount of intra class variabilities in [68]. In this thesis, we have modeled the spatio-temporal relationships between different activities which form a higher level representation.

Some of the previous approaches such as [69] assume a known structure of the graph for context representation. Models such as the AND-OR graphs or other tree structures have been suggested in the past [10] [70] for modeling sports sequences and office environments. These models however, are more suited for structured environments where there are a set of rules governing the behavior of people such as in sports, or where the number of objects/activities or the combinations of sub-activities are limited as in an office environment. Applying such models to unconstrained sequences can be laborious due to the exponential number of combinations of activities which have to be learnt here to construct such models. Similarly, papers such as [71] use context to infer a collective activity using single person activities. In such sequences however, it is assumed that all participating persons/objects contribute to the collective activity. Whereas, in a typical surveillance scenarios, different actors may or may not be interacting with each other, therefore such models cannot be directly applied here. The authors in [72] deal with recognizing a single activity over multiple cameras by topology inference, person re-identification and global activity interpretation. Here, we are dealing with a set of different activities which may or may not be correlated, therefore a Markov random field is a more suited model to capture these complex spatio-temporal relationships. Social roles for hierarchical representation of activ-

ities in sports videos is explored in [73]. Most of this work deals with short duration videos or with videos with a pre-defined structure such as sports videos. We propose to define the structure of the graph on the test sequence rather than use a pre-defined structure.

3.1.3 Definitions

- **Action** - A region of uniform motion in the video, one or more of which form a meaningful activity.
- **Activity** - A meaningful event in the video which we wish to identify. Our objective is to assign every activity a class label in the range $c_1..c_N$.
- **Activity region** - A spatio-temporal volume in which the activity takes place. An activity region A_i is represented by its spatial and temporal centroids s_i and t_i.
- **Activity Sequence** - A set of activities which occur in close proximity with each other and can have causal influences on each other. Each activity sequence is modeled as a graphical model and evaluated.

3.2 Overview

An overview of our MRF model for activity recognition in activity sequences is shown in Figure 3.1. Given a long-term video, the goal of our approach is to estimate the category to which individual activity belongs. We assume that we have some training videos available, each of which have one or more sequences of activities occurring in different spatio-temporal regions. Each spatio-temporal region where a potential activity takes place

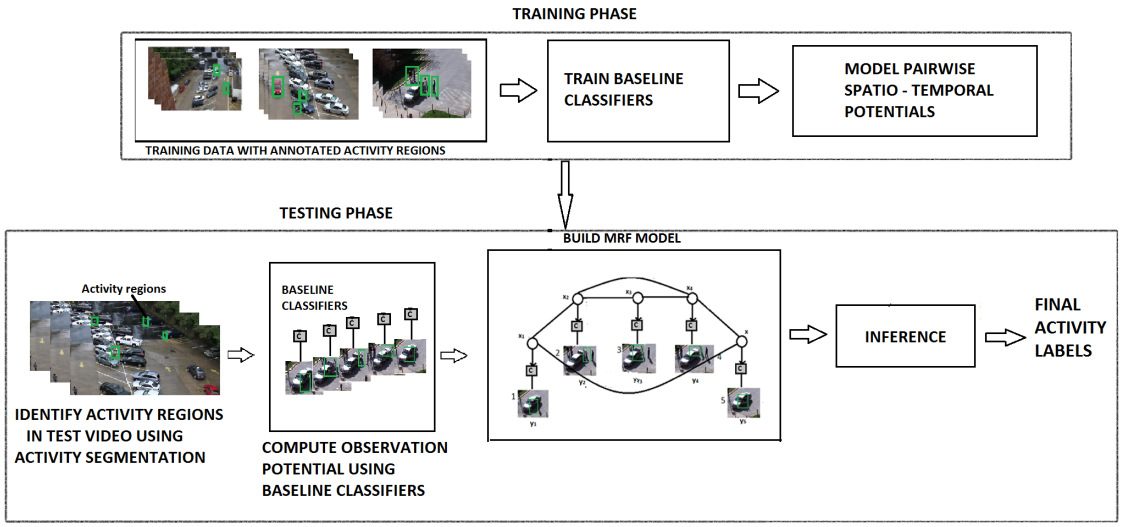

Figure 3.1: Figure shows the illustration of our proposed method. Training involves modeling the pairwise spatio-temporal relationships between different activity regions which are provided in annotations as mentioned in Section 3.3.1. For a test video, activity regions are identified using the method presented in Section 3.4.2. Using the potentials from training data and observation potentials as described in Section 3.3.1, the node labels are inferred (Section 3.3.3).

is termed as an activity region. We also have available a set of baseline classifiers $C = \{c_1, c_2..c_N\}$, which can output a probability of an activity region y belonging to a particular class c_i, i.e. $P(c_i|y)$.

A typical surveillance video, such as a parking lot video (shown in Figure 1.1) contains several activities occurring simultaneously or in succession in different portions of the scene. The number of objects, people and activities change from one video sequence to another. Having identified the activity regions in a video using the baseline classifiers, and having clustered them into sequences which are potentially related to each other, we explore three key aspects to improve the accuracy of recognition: 1) The relationship in the spatial locations of activities, 2) the relationship in the temporal locations of activities and 3) the probability of association of two given activities, i.e., the probability that one

activity might occur in the vicinity of another.

These concepts are modeled by a Markov random field (MRF). Since the MRF is used to model the context information across activities, we choose the nodes of the MRF as atomic activities rather than pixels or image regions, as is commonly done in image segmentation. Each edge represents the spatio-temporal context between the activity nodes that it connects. The node potentials are obtained using the likelihood of the activities given by the baseline classifiers. The edge potentials are learnt from the training data. We perform inference on the resulting MRF to estimate the activities in the test sequence. We conduct experiments on the UCLA office dataset containing indoor office sequences and the publicly available VIRAT dataset containing parking lot videos.

3.3 Graphical Representation of Activities

The goal of our algorithm is to model the space-time relationships between the activities in a scene using a Markov random field. The MRF is an undirected graph $G = (V, E)$, with a set of nodes V and a set of edges E. Given a video sequence to be recognized, we first construct an MRF over all probable related activities in the sequence. Each node denotes an activity and an edge represents the spatio-temporal relationship between two activities. There are a set of observations $Y = \{y_1..y_n\}$ and a set of hidden variables $X = \{x_1..x_n\}$ for a sequence of n activities. An observation node y_i denotes the image observation of an activity, which are the features computed over an activity region. The output of the baseline classifiers for each activity is used to compute an observation potential. A hidden

node denotes an atomic activity to be estimated. A node x_i can be defined as

$$x_i = (c_i, s_i, t_i), \tag{3.1}$$

where x_i denotes a node, c_i denotes the activity class to which it belongs, s_i is its spatial location and t_i denotes its temporal location. The MRF is given by

$$\Psi = \frac{1}{Z} \prod_{i,j \in E} \mathbf{w_{st}^{ij}} \psi_{st}(x_i, x_j) \prod_{i \in V} \mathbf{w_o^i} \psi_o(x_i, y_i), \tag{3.2}$$

where Ψ is the overall potential. Here, we assume that the MRF factors over the edges. There are two kinds of potentials associated with the graph. $\psi_{st}(x_i, x_j)$ is the edge potential which is the spatio-temporal relation between two hidden nodes connected by an edge and $\psi_o(x_i, y_i)$ is the observation potential of a node. $\mathbf{w_o}$ and $\mathbf{w_{st}}$ are the node and edge weights respectively. Z is the normalization constant. The illustration of our proposed graphical model is shown in Figure 4.3.

3.3.1 Potential Functions

The node observation potentials and the spatio-temporal edge potentials are defined as given below.

Observation Potential

The observation potential or the node potential is the evidence of the activity obtained from the video data. These are obtained from the image observations of the activities which are the baseline classifiers. We have one baseline classifier per activity class, the output of which is the probability of the given activity belonging to a particular

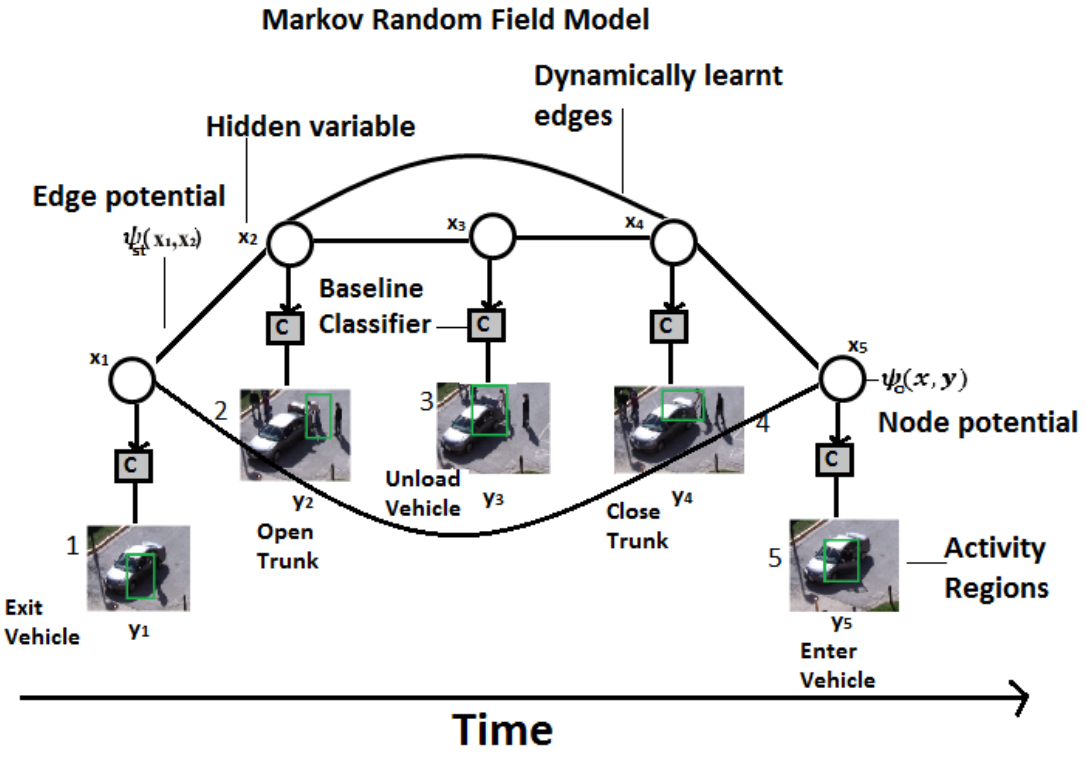

Figure 3.2: Figure shows the Markov random field constructed over a spatio-temporal volume for an activity sequence. Shown in the figure are the activity regions which form the observation variables y. The baseline classifier output forms the observation potential. The labels of the activities which have to be predicted constitute the hidden nodes x. The edges of the graph are learnt iteratively.

category. We use a Bag-Of-Features approach over space-time interest points [2] as our baseline classifiers due to its popularity for recognition of atomic activities. Specifically, space-time interest points based on Harris and Forstner operators are computed over the training set. A feature vector is generated for each point. During training, a codebook is build by clustering and quantizing these features. Each category of activity is modeled as a distribution over this vocabulary. The interest points are computed over the test video and regions with significant number of points from the vocabulary are said to be the

activity regions, denoted as the observation variables y_i. A discriminative classifier such as a multiclass SVM classifier is used to compute the probability of an activity region belonging to a particular category $P(c_j|y_i)$. These probabilities are learnt jointly over the training data. The observation potential is therefore defined as

$$\psi_o(x_i, y_i) = p(x_i|y_i, C), \tag{3.3}$$

where ψ_o is the observation potential, y_i is the observation variable and C is the set of baseline classifiers. It is to be noted that any other set of features or algorithm can also be used for the baseline classifiers.

Spatio-temporal Potential

The spatio-temporal potential is defined on edges connecting the activity variables in the graph. Actions which are within a spatio-temporal distance of each other are assumed to be related to each other. There are three components to this potential: the spatial component, the temporal component and the association component. The spatial component models the probability of an activity belonging to a particular category given its spatial configuration with its neighbor. Similarly, the temporal component models the probability of an activity belonging to a particular category given its temporal distance with its neighbor. The association component is the probability of two activities being within a pre-defined spatio-temporal vicinity of each other. The spatial and temporal components are modeled as normal distributions whose parameters μ_s, σ_s, μ_t and σ_t are computed using the training data. The spatial component is given by

$$\psi_s(x_i, x_j) = \mathcal{N}_{sd}(\|s_i - s_j\|^2; \mu_s(c_i, c_j), \sigma_s(c_i, c_j)), \tag{3.4}$$

$$\psi_t(x_i, x_j) = \mathcal{N}_{td}(\|t_i - t_j\|^2; \mu_t(c_i, c_j), \sigma_t(c_i, c_j)). \tag{3.5}$$

where $\mu_s(c_i, c_j)$,$\sigma_s(c_i, c_j)$, $\mu_t(c_i, c_j)$ and $\sigma_t(c_i, c_j)$ are the parameters of the distribution of relative spatial and temporal positions of the activities, given their categories. The association probability f_{ij} is computed as a ratio of the number of times an activity category c_j has occurred in the vicinity of activity category c_i to the total number of times the category c_i has occurred. Therefore, the spatio-temporal potential is given by

$$\psi_{st}(x_i, x_j) = f_{ij}\psi_s(x_i, x_j)\psi_t(x_i, x_j) \tag{3.6}$$

3.3.2 Training

Training involves learning the edge weights for the graphical model. This is done by optimizing Equation 3.3 over the training data. Due to the log linear nature of the model, this is a convex optimization problem. For each training instance, the potential functions are obtained as explained above. The edge potentials are obtained by a pseudo-negative log likelihood optimization over loopy belief propagation.

3.3.3 Inference

Inference in a graphical model involves computing the marginal probabilities of the hidden or unknown variables given an evidence or an observed set of variables. We choose the belief propagation method for estimation of parameters. Since there are loops in our model, the loopy belief propagation is used. Although this algorithm is not guaranteed to converge, it has shown excellent empirical performance [74].

At each iteration, a node sends messages to its neighbor. All nodes are updated

based on the messages from their neighbors. Consider a node $x_i \in V$ with a neighborhood $N(x_i)$. The message sent by a node $x_i \in V$ to its neighbor $x_j \in V, (x_i, x_j) \in E$ can be given as

$$m_{x_i,x_j}(x_j) = \alpha \int_{x_i} \psi_{st}(x_i, x_j)\psi_o(x_i, y_i) \prod_{x_k \in N(x_i)} m_{x_k,x_i}(x_i)dx_i \tag{3.7}$$

The marginal distribution of each activity region is given by

$$p(x_i) = \alpha\psi_o(x_i, y_i) \prod_{x_j \in N(x_i)} m_{x_j,x_i}(x_i) \tag{3.8}$$

The activity label which has the highest marginal distribution is assigned to the region.

The overall algorithm of our approach is presented in Algorithm 2.

3.4 Experiments and Results

3.4.1 Dataset

The goal of our approach is to model activity context in continuous videos, therefore, we perform experimentation on long duration realistic videos. Traditional datasets like Weizmann [2] and KTH [7] cannot be used to validate our system. Some other datasets like [8] contain long unsegmented video, but these activities are not related to each other and the sequence is not a realistic one. Therefore, we evaluate our system on two challenging datasets containing long duration activities: 1)The UCLA office dataset and 2)The publicly available VIRAT ground dataset [9].

The UCLA office dataset [10] consists of indoor and outdoor videos of single and two-person activities. Here, we perform experiments on the lab scene containing close to 35 minutes of video captured with a single fixed camera in a room. We work on 10 single person

Algorithm 1 Algorithm for labeling activities in a test sequence using our context model

Input:	$\mathcal{S}_{\mathcal{R}} = \{V_1 \dots V_{N_{\mathcal{R}}}\}$	Set of training videos containing activity annotations
	An activity sequence $\mathcal{Y}_f$ containing n activities occurring in close spatio-temporal vicinity $\{y_1 \dots y_n\}$.	
Output:	Labels of activities $\{x_1 \dots x_n\}$	
Training:	Train baseline classifiers $c_1 \dots c_N$ for N activities and model the spatio-temporal potential $\psi_{st}(x_i, x_j)$ between all pairs of activities using annotated training videos using Eqn (3.6).	

Testing:

1. Identify activity regions using the activity segmentation algorithm.
2. Compute observation potential $\psi_o(x_i, y_i)$ for each activity segment given by the baseline classifiers using Eqn (3.3).
3. Initialize graph $\mathcal{G}$ containing n observation variables representing activity regions and n hidden variables representing the activity labels.
4. Run inference to generate posteriors;
5. Compute labels from posteriors and output labels.

activities: Enter lab, exit lab, sit down, stand up, work on laptop, work on paper, throw trash, pour drink, pick phone receiver and place receiver down. There is very little variation in viewpoint, occlusion and scale here. The first half of the data is used for training and the second half for testing. Each activity occurs 6 to 15 times in the dataset.

The VIRAT dataset is a state-of-the-art activity dataset with many challenging characteristics, such as wide variation in the activities and a high amount of occlusion and clutter. It consists of surveillance videos of 11 scenes with different scales of resolution. These are parking lot videos involving single vehicle activities, person and vehicle interactions, and people interactions. There are also some group activities. This dataset consists

of scenes captured on a single camera although the viewpoint can differ from one scene to the next. In any scene, the activities can occur at different orientations depending on the location. However, since these are wide-area videos, persons of interest are usually far away from the camera, the change in spatio-temporal distance with camera view is considered negligible. We have used parking lot scenes $VIRAT_S_0000$, $VIRAT_S_0401$ and $VIRAT_S_0502$ for the first set of experiments and all data for the second set. The length of the videos vary between $2-15$ minutes and containing up to 30 activities in a video. For every scene, the first half is used for training and the second half for testing.

We perform two sets of experiments on the VIRAT dataset, one on Release 1 and the other on Release 2 of the data. For Release 1, there are 6 activities which are annotated: Person entering vehicle, person exiting vehicle, person opening trunk, person closing trunk, person loading vehicle and person unloading vehicle. In release 2, additional 5 activities have been added: person carrying an object, person gesturing, person running, entering and exiting a facility. For release 1, we have provided comparison with the baseline Bag-of-Words classifier as well as the state-of-the-art String of Feature Graphs [1] method. For release 2, we show comparison with the baseline Bag-of-Words classifier.

3.4.2 Pre-processing

To label meaningful activities in a long-duration wide area video, the first step is to identify the spatio-temporal location of activities. We call this step as "activity segmentation". The video is first divided into overlapping time windows of fixed duration. Activity region computation is performed on windows of three scales. Here each window consists of

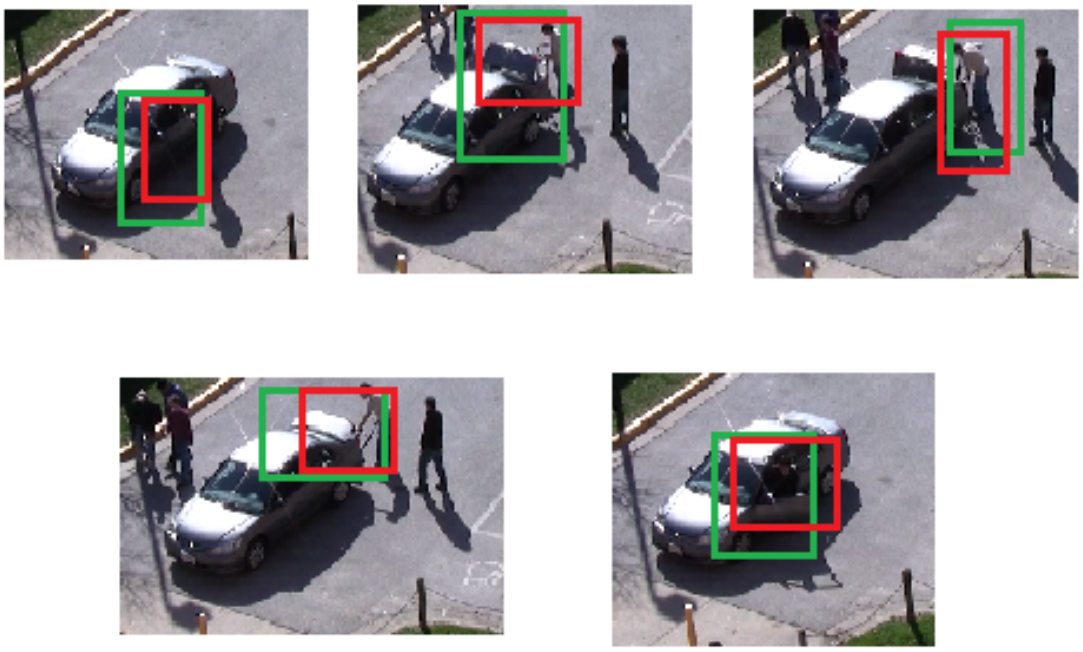

Figure 3.3: The figure shows some examples of segmentation of activity regions. The obtained segmentation is marked in green while the true segmentation is marked in red.

30, 60 and 120 frames with an overlap of half the number of frames. Feature points are computed for each time window which contains a track and the time window is spatially clustered into as many regions as the number of tracks in the window. Here, we use the Space-Time Interest Points (STIP) [2] as our features.

For each time window, the baseline classifiers are used to assign a probability of the window belonging to a particular activity. All activities which do not correspond to the set of "interesting" activities are considered as "background activities". We also train a baseline classifier for background activities. For each set of overlapping windows, the window which has achieved the highest probability is chosen as the activity region. All

regions which correspond to background activities are eliminated. Recognition is carried out on the remaining activity regions. An example of activity segments identified in a sequence is shown in Figure 3.3. A limitation of this approach is that, when the segmentation algorithm fails to detect an activity segment, it is eliminated from further processing, thereby missed detections are not corrected.

3.4.3 Methodology

We used a randomly selected set of half the data for training and the other half for testing. During the training, we assume that the activity segmentation and the activity labels are available to us. We normalize all distances with respect to the scale of the video to make the approach invariant to scale. A threshold was set on the spatio-temporal distance between activities to determine if the relationship between them has to be modeled. We used the distance threshold as a bounding box of 4 times the average dimensions of the person in the scene and a time threshold of 20 seconds. These values have been fixed experimentally. The graphical model is constructed on individual activity sequences. Classification over an entire activity sequence is carried out using the proposed method. For each activity region in the sequence, the baseline classifier is applied to generate the observations. A graph is constructed based on the spatio-temporal distances between activities. Inference is carried out on the graph using the MRF parameters computed during training. Labels are assigned to each activity region based on the posterior probabilities.

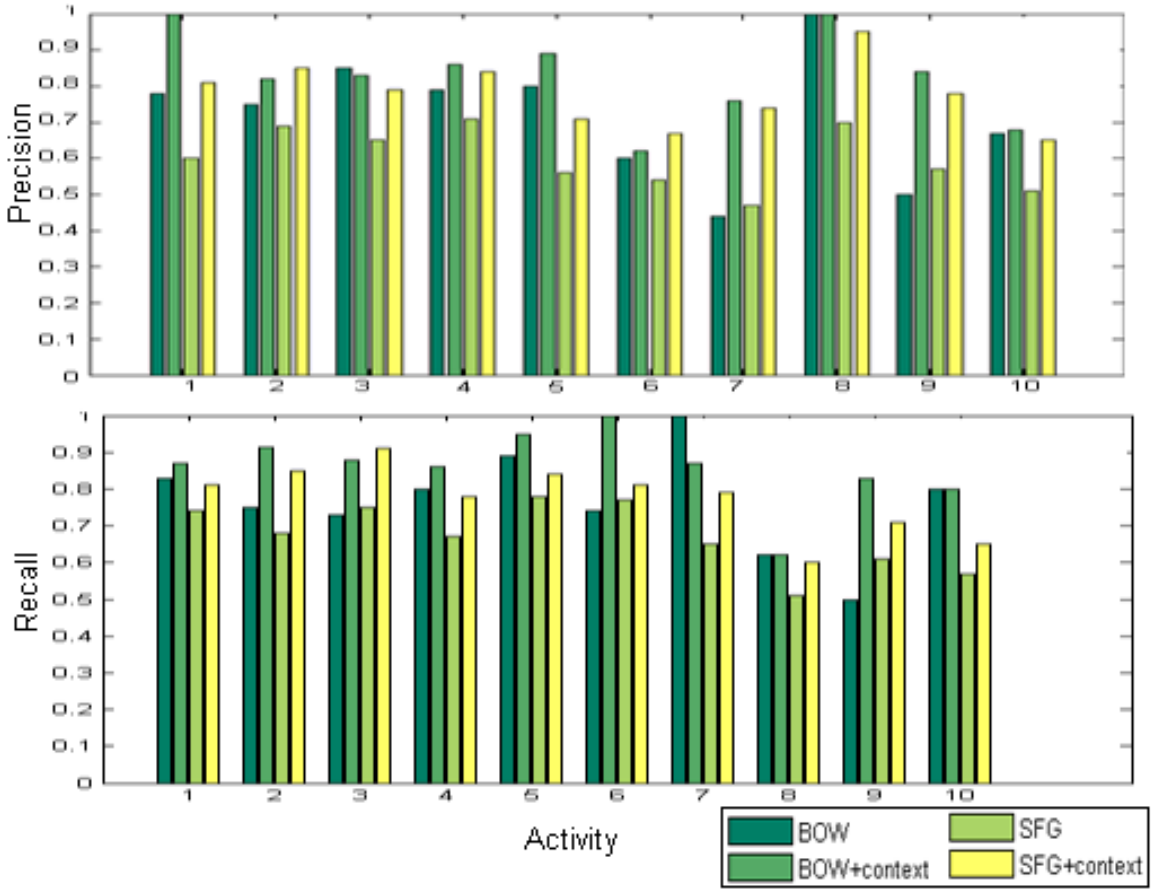

Figure 3.4: The figure shows the precision and recall obtained on the UCLA office dataset and its comparison with the Bag-Of-Features baseline classifier and SFG [1]. The activities are: 1 - enter room, 2 - exit room, 3 - sit down, 4 - stand up, 5 - work on laptop, 6 - work on paper, 7 - throw trash, 8 - pour drink, 9 - pick phone, 10 - place phone down.

3.4.4 Results on UCLA office dataset

For the UCLA dataset, we consider only single person activities in a high resolution video with little variation in viewpoint and occlusion. Although this dataset has been used for experimentation in [10], the events which have been classified for the lab data and the accuracy of recognition for each event have not been provided by the authors. Therefore, we provide comparison to the baseline methods used here, which is the Bag-of-Words and the

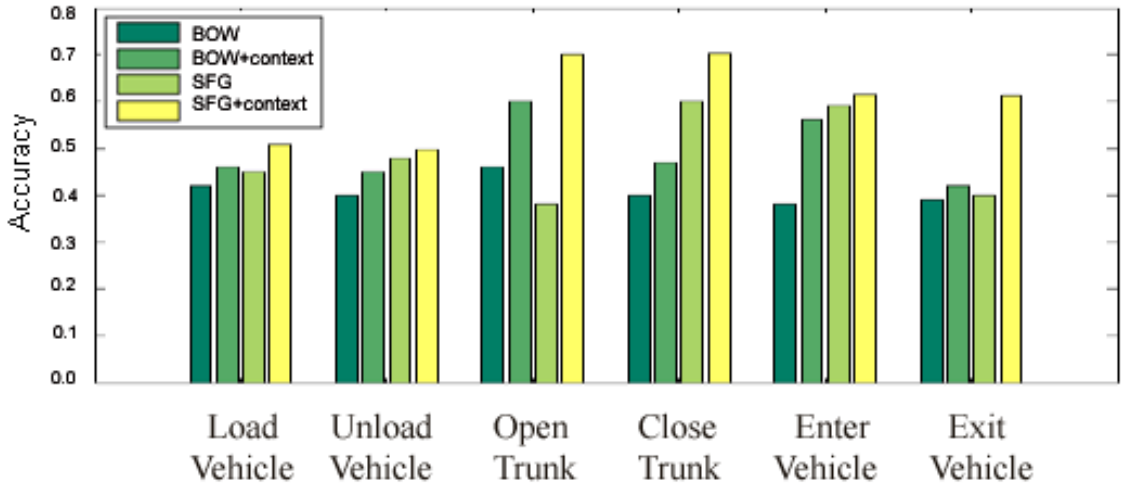

Figure 3.5: Figure a) shows the accuracy of our method with the VIRAT release 1 dataset for six activities and its comparison with the Bag-of-Words and SFG [1] approach. The activities are: 1 - loading, 2 - unloading, 3 - open trunk, 4 - close trunk, 5 - enter vehicle, 6 - exit vehicle. Figure b) shows the increase in performance with structure improvisation.

SFG [1]. In both cases, it can be seen that the addition of context improves performance. The Bag-of-Words classifier gives an overall accuracy of 75.4%.For some activities, the BOW classifier was able to identify all instances, therefore no further improvement was possible. An overall accuracy of 86.7% was achieved with the addition of context. For the SFG method, an overall accuracy of 62.3% was achieved while the addition of context gave an accuracy of 77.9%. The values of precision and recall for BOW and BOW+context, SFG and SFG+context are shown in Figure 3.4.

3.4.5 Results on VIRAT dataset

We compare our approach with two well known approaches: the Bag-of-Words approach and the String of Feature Graphs (SFG) approach which is a recent method that

provides state-of-the-art performance on multi-object data in realistic videos. This method also models spatio-temporal relations at the feature level.

For the VIRAT release 1 data, we demonstrate our method using the BOW as well as SFG as baseline classifiers in Figure 3.5. We have also shown the results of the baseline classifiers for comparison. We can see that our method performs better than the SFG method in most cases. An overall accuracy of 40% was obtained using BOW and 51.3% was obtained using the SFG method. The usage of our method on BOW resulted in an overall accuracy of 52.4% while the usage of our method on SFG resulted in an accuracy of 61.5%. The accuracy of recognition for activities "loading" and "unloading" was found to be slightly lower than the rest since they involve similar gestures. The confusion matrix for the 6 activities using BOW. BOW+context, SFG and SFG + context is shown in Figure 3.6.

The second set of experiments was conducted on the release 2 of VIRAT dataset. This dataset contains five additional activities - person carrying load, gesturing, running, entering and exiting facility. These activities add some additional context information to the data. We provide the precision-recall values for each activity as well as the comparison with Bag-Of-Features and SFG approaches in Figure 3.7. Here also, we find that the addition of context helps in better recognition in both cases. The overall accuracy of BOW+context was 52.6% while BOW had an accuracy of 41.3%. The overall accuracy of SFG was 37.8% while the overall accuracy of SFG+context was 46.4%. It was seen that the activities "enter vehicle" and "load vehicle" were often confused with each other in the absence of context. But the use of context tells us that if a person opens the trunk, he is likely to load it,

whereas if the person opens a door, he is likely to enter it. This contextual information was captured by our model and brought about a an improvement in the performance. It was seen that the method shows an improvement over the baseline classifiers in the case of partial occlusion as well as noise due to shadows and clutter.

In Figure 3.8, we illustrate the difference between the output of the baseline classifier and our algorithm for different activities like enter vehicle, exit vehicle, open and close trunk. It can be seen that the output of our algorithm has a more well defined peak in probability, which in turn means less uncertainty in prediction as compared to the baseline classifier. This shows that the confidence of classification can be increased with the use of context. In the last two cases, the addition of context corrects an incorrect classification (represented by the highest probability).

3.5 Conclusion

In this chapter, we have proposed an approach to model continuous activities in wide-area scenes using graphical models. We have shown that the spatio-temporal relationships between different activities in a scene can be used as context in the recognition of activities. We illustrated a scheme based on graphical models used to learn the spatio-temporal relationships from training video. We inferred the most probable set of labels for the activities in the test video given their spatio-temporal configurations and observation potentials generated from weak classifiers.

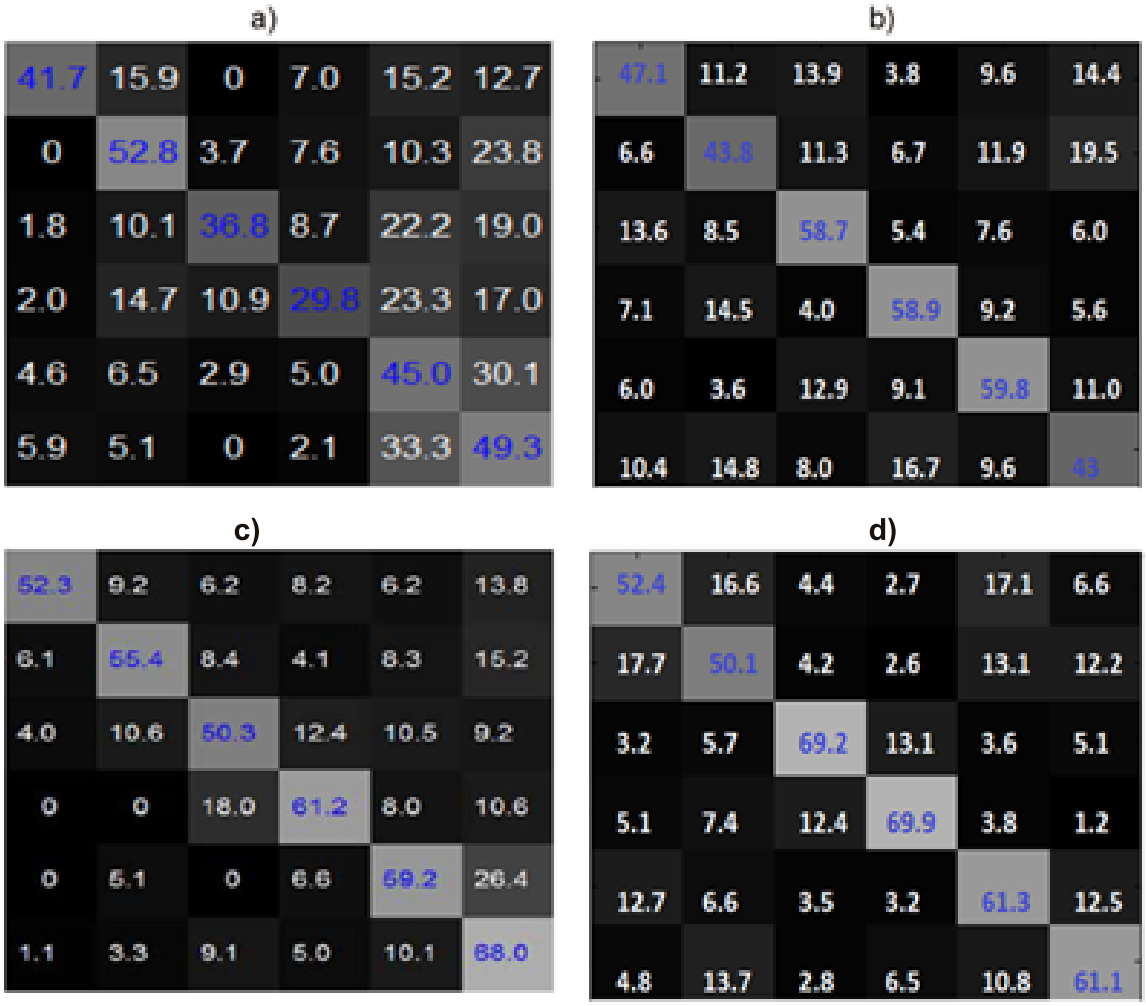

Figure 3.6: The Figure shows the confusion matrix on VIRAT release 1 data. a)Result of applying the baseline classifier BOW to the data. b) Result of applying BOW+context on the data. c) Result of SFG baseline classifier. d) Result of SFG + context. The activities are: 1 - loading, 2 - unloading, 3 - open trunk, 4 - close trunk, 5 - enter vehicle, 6 - exit vehicle. The corresponding increase in recognition accuracy is evident from the graph.

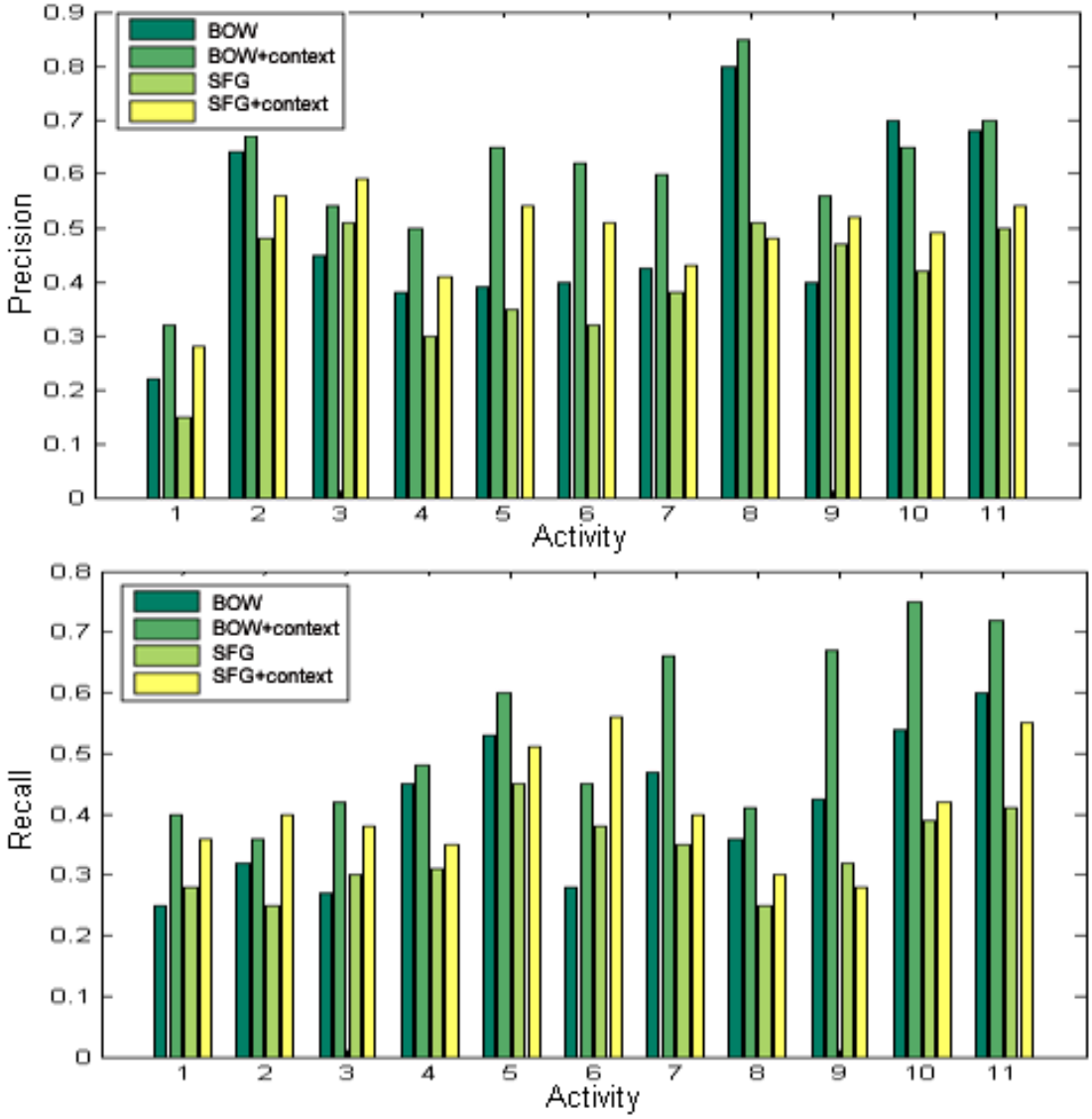

Figure 3.7: The figure shows the precision and recall obtained on the VIRAT release 2 dataset and its comparison with the Bag-Of-Features and SFG approaches. The activities are: 1 - person loading an object to a vehicle, 2 - person unloading an object from a vehicle, 3 - person opening a vehicle trunk, 4 - person closing a vehicle trunk, 5 - person getting into a vehicle, 6 - person getting out of a vehicle; 7 - person gesturing, 8 - person running, 9 - carrying load, 10 - entering facility, 11 - exiting facility.

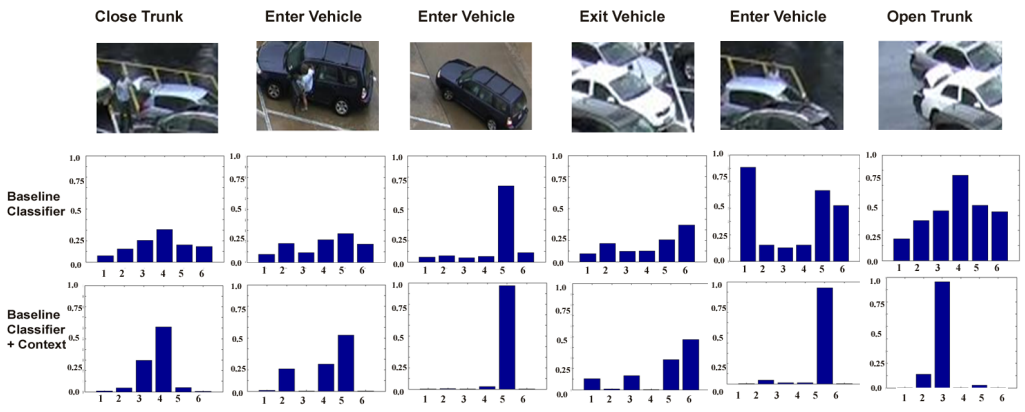

Figure 3.8: The comparison of the prior probabilities which are the output of the baseline classifiers with the posterior probabilities which is the output of our algorithm for a set of six activities. The output of our algorithm is seen to have a more well defined peak (less uncertainty) as compared to the baseline classifier. For the last two, it is seen that the addition of context corrects an incorrect classification. The activities in order are: 1 - person loading an object to a vehicle, 2 - person unloading an object from a vehicle, 3 - person opening a vehicle trunk, 4 - person closing a vehicle trunk, 5 - person getting into a vehicle, 6 - person getting out of a vehicle

Chapter 4

Hierarchical Graphical Model For Simultaneous Tracking, Localization And Recognition Of Activities

4.1 Introduction

A continuous video consists of two inter-related components: 1) tracks of the persons in the video and 2) localization and labels of the activities of interest performed by these actors. Activity analysis of continuous videos involves solving both the tracking as well as recognition problems. In the past, most research on video analysis has treated these two problems separately. However, in the context of continuous videos, such as surveil-

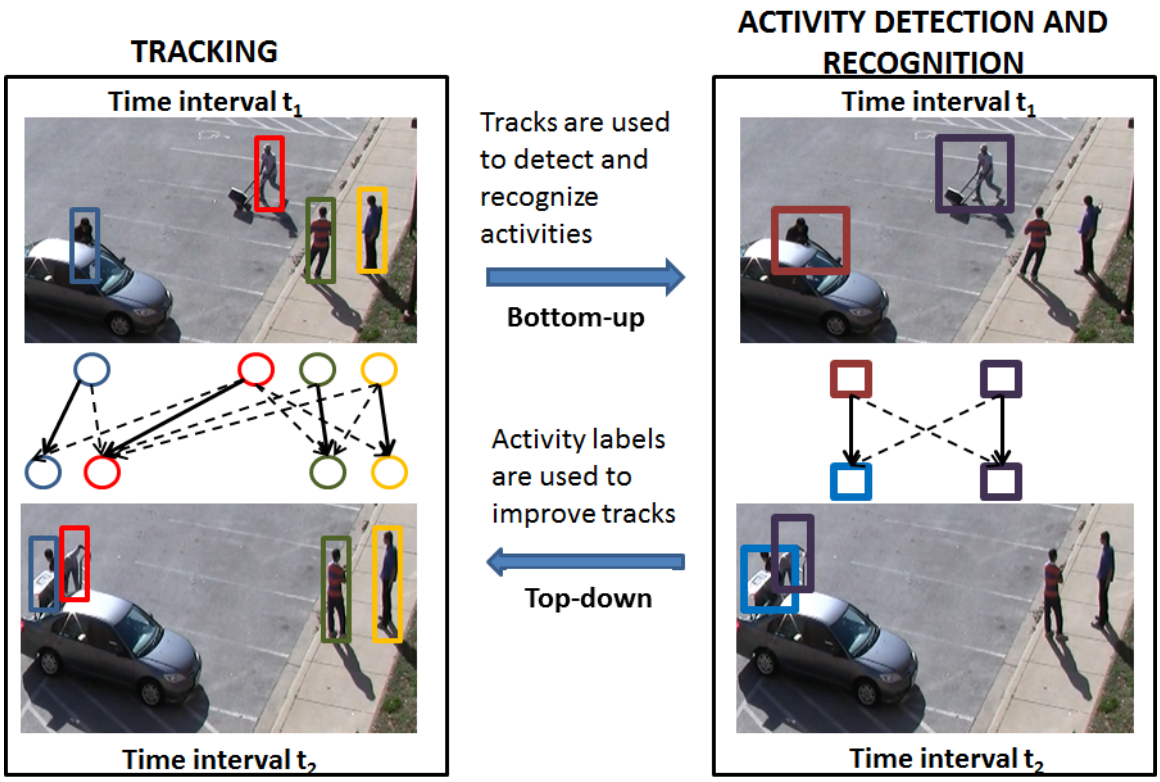

Figure 4.1: Figure demonstrates the bi-directional processing of videos for integrated tracking and activity recognition. The bottom-up (or feedforward) processing involves detection and recognition using an initial set of tracks along with low level features and spatiotemporal context between activities. The top-down (or feedback) processing involves correcting the tracklet associations using the obtained labels.

lance or sports videos, the solution to one problem can help in finding the solution to the other. Knowing the tracklet associations can help in better detection and recognition of activities. Similarly, information about the location and labels of activities in a scene can help in determining the movement of people in the scene. Therefore, we propose a method which performs the two tasks in an integrated framework modeling contextual relationships between tracks as well as activities using graphical models.

Most early approaches for activity recognition focused on modeling and representation of single person activities. However, while dealing with more complex scenarios of multiple person activities or continuous videos, it has been widely acknowledged that, in addition to the features themselves, the structural information between sets of features and/or objects, often termed as context, plays an important role in discriminating between activities. While graphical models are commonly used to encode such structural relationships [42, 4, 66, 71, 10], the question of how to arrive at the ideal structure for this graphical model still remains unsolved.

Research in the area of biological vision has shown that, the human visual system employs a bi-directional (top-down as well as bottom-up) reasoning in analyzing and interpreting data of multiple resolutions [75]. This has been found to be particularly helpful in correcting errors due to false detections or noise. Applying these concepts to the analysis of continuous videos, we consider the task of obtaining recognition scores using tracks as a bottom-up (or feedforward) approach, while the task of correcting tracks using obtained recognition labels is treated as top-down (or feedback) processing. We alternate between both these steps to result in a bi-directional algorithm that can help in increasing

the accuracy of both these tasks.

4.1.1 Contributions

The main contribution of our work described in this chapter is to propose a framework for simultaneous tracking, localization and labeling of activities in continuous videos, by integrating bottom-up and top-down processing along with automatic structure learning. Our approach can handle a varying number of actors and activities. In order to achieve this, we propose the following steps:

1. In the feedforward processing, the tracks are used to detect regions in the video where interesting activities are taking place. The activities in these detected regions are then recognized. The detection and recognition of activities is carried out simultaneously using a 2-stage hierarchical Markov random field (HMRF). The lower level nodes model relationships across tracklets, while the higher level nodes model information across activities, also known as inter-activity context. A feedback processing is then carried out, in which the recognition results are then used to correct errors in tracking. An illustration of the bi-directional computational framework in a continuous video is shown in Figure 4.1.

2. We use an expectation-maximization formulation to alternate between the two steps in a bi-directional framework to arrive at a solution for both these tasks.

4.1.2 Related Work

Reviews of related work in tracking and recognition can be found in [12][76]. We focus only on those that consider the problem of simultaneous localization and recognition. Simultaneous localization and classification of scenes in broadcast programs has been researched in the past in [77] but these scenes have distinctive breaks unlike continuous activity sequences. Localization and classification of single-person activities with distinctive breaks was performed in [78]. Activity localization and labeling of single person activities was also demonstrated in [79] but the system used concatenated short duration sequences which lack contextual information. We perform localization and classification on multi-person sequences in continuous videos and also explore the integration of tracking into the framework.

Simultaneous activity recognition and tracking has been studied in the context of interacting objects. The relations between interacting targets obtained from activity recognition is used in the tracking process using a relational dynamic Bayesian network in [80]. Simultaneous recognition of a collective activity and tracking of the multiple targets involved is performed in [81]. However, these only deal with the motion relations between interacting persons and not across activities of the same person. They also do not look into the bi-directional processing in an EM framework.

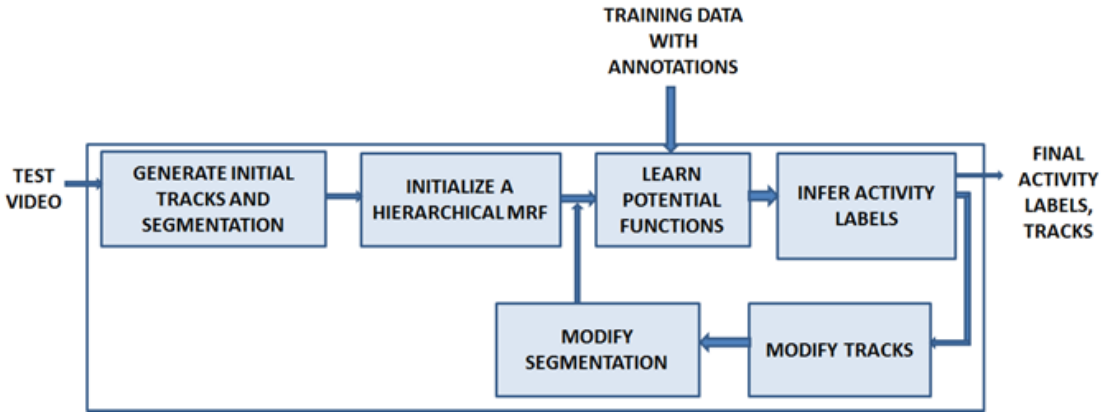

Figure 4.2: Figure shows the illustration of our proposed method. Given a continuous video with computed tracklets, a set of tracks and activity segments are initialized. An HMRF model is built over the tracklets and segments. Edge potentials are learned on the annotated training data. Inference on this graphical model provides a revised set of labels for the activities which can be fed back into the system to regenerate the tracks and rebuild the HMRF. The procedure is repeated until a stop criterion is reached. The tracks and labels of all segments are provided as output.

4.2 Overview

The illustration of our proposed method is shown in Figure 4.2. The association, consistency and spatiotemporal potentials are learned from the annotated training data. For a given test video, we assume that we have with us a set of tracklets, which denote short duration segments of tracks which are assumed to be accurate. It is assumed that, each tracklet belongs to a single activity.

To begin with, we generate a set of match hypotheses for tracklet association and a likely set of tracks. An observation potential is computed for each tracklet using the features computed at the tracklet. Tracklets are grouped into activity segments using a standard baseline classifier such as multiclass SVM or motion segmentation.

Next, we construct a two-level Markov random field using the tracklets and activity

segments. The first level nodes correspond to the tracklets and the second level nodes correspond to the activity segments. One or more tracklets can correspond to the same activity segment. Edges model relationships between nodes of the same level as well as nodes at different levels. This structure incorporates the context information between adjacent tracklets as well as across activity segments.

The dense HMRF has edges connecting each node to all nodes within a certain spatiotemporal range from the node. This gives us the initial graph on which we perform the learning and recognition. The node features and edge features for the potential functions are computed from the training data.

Inference on this graph provides the posterior probabilities for all nodes using information available at two resolutions. The tracking is repeated with the new set of activity segments. The graph is rebuilt using this new structure and the procedure is repeated. Convergence is said to be achieved when the node labels and tracks do not change from one iteration to the next.

The output of the algorithm is a set of tracks, segments and the labels assigned to each segment.

4.3 Hierarchical MRF (HMRF) Model

Consider a video to consist of a set of p tracklets resulting in tracks T. The tracklets can be grouped into a set of q activity segments along the tracks. We design a 2-level hierarchical MRF with nodes $X = X_t \bigcup X_a$, where the lower level nodes $X_t = \{x_{t_1}, x_{t_2}...x_{t_p}\}$ correspond to tracklets and the higher level nodes $X_a = \{x_{a_1}, x_{a_2}...x_{a_q}\}$ correspond to

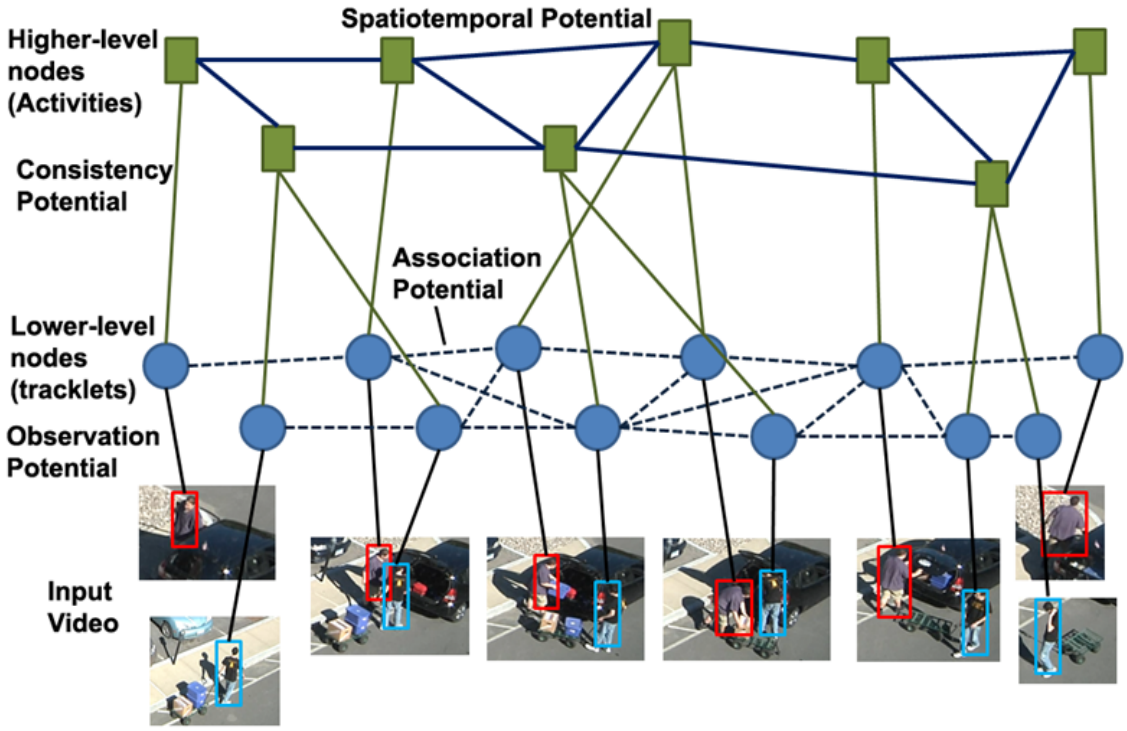

Figure 4.3: Figure shows a typical HMRF over an activity sequence. Tracklets are extracted from a continuous video and form lower level nodes. Using an initial set of tracks, a segmentation of tracklets is performed to obtain activity segments. These form the higher level nodes. Edges model relationships between potentially associated tracklets, tracklets and their corresponding activity segments, and the spatiotemporal context information between activity segments. The node potentials and edge potentials are marked in the graph.

activity segments. The set of observed features obtained for a node x_i is denoted as y_i. There are three kinds of edges in the graph: edges connecting adjacent tracklets which belong to a valid track hypothesis, edges connecting tracklets to their corresponding activity segments, and edges connecting activity segments which are within a specified spatiotemporal distance of each other. A typical HMRF constructed over a continuous video is shown in Figure 4.3.

The overall energy function of the HMRF is given by

$$E(X_t, X_a, T) = \frac{1}{Z}\exp(-\Psi(X_a, X_t, T)), \tag{4.1}$$

$$\begin{aligned}\Psi(X_a, X_t, T) = &\sum_{x_{t_i}} \mathbf{w_o^{t_i}}\psi_o(x_{t_i}, y_{t_i}) + \sum_{x_{a_i}} \mathbf{w_o^{a_i}}\psi_o(x_{a_i}, y_{a_i}) \\ &+ \sum_{x_{t_i}} \sum_{x_{t_j} \in N(x_{t_i})} \mathbf{w_a^{t_i,t_j}}\psi_a(x_{t_i}, x_{t_j}) \\ &+ \sum_{x_{t_i}} \sum_{x_{a_j} \in N(x_{t_i})} \mathbf{w_c^{t_i,a_j}}\psi_c(x_{t_i}, x_{a_j}) \\ &+ \sum_{x_{a_i}} \sum_{x_{a_j} \in N(x_{a_i})} \mathbf{w_{st}^{a_i,a_j}}\psi_{st}(x_{a_i}, x_{a_j}),\end{aligned} \tag{4.2}$$

where $\psi_o(.)$ is the observation potential computed over both levels, $\psi_a(.)$ is the association potential, $\psi_c(.)$ is the consistency potential and $\psi_{st}(.)$ is the spatiotemporal context potential of the HMRF. Here, $\mathbf{w_o}$ is the model parameter for the observation potentials and $\mathbf{w_a}$, $\mathbf{w_c}$ and $\mathbf{w_{st}}$ are the corresponding model parameters for the edges of the graphical model, represented using similar superscripts. It is to be noted that for a multi-state model such as in this case, with the nodes taking n states, each edge parameter is a matrix of n^2 elements.

4.3.1 Computation of Potential Functions of HMRF

We will now describe the four kinds of potential functions mentioned above in detail.

Observation Potential

Each node of the graph (lower or higher level) is associated with an observation potential. At the lower level the observation potential is obtained from the image features

associated with the tracklet corresponding to the node, while at the higher level, it is obtained from the image features of *all* the tracklets that link to the higher level node. Here, we utilize space-time interest points [2] as well as object attributes to learn a multi-class SVM classifier in a Bag-of-Words formulation. This is also referred to as the baseline classifier. The observation potential of a node x_i is therefore defined as

$$\psi_o^c(x_i, y_i) = -\log(P(x_i = c|y_i)), \tag{4.3}$$

where ψ_o^c is the observation potential for a node x_i (tracklet or activity segment) and y_i is its observed feature descriptor. It is to be noted that any other set of features or algorithms can also be used for the baseline classifiers.

Association Potential

The association potential is defined on the edges connecting tracklets which are hypothesized to be associated with each other. The association potential models the likelihood of association of two tracklets by measuring the compatibility of activities taking place in the two tracklets. The association potential for two tracklets belonging to activity class c_a and c_b is given by

$$\psi_a(x_{t_i}, x_{t_j}) = \mathbf{I}(x_{t_i}, x_{t_j}), \tag{4.4}$$

where $\mathbf{I}(a, b)$ is an indicator function which returns 1 if the features belonging to tracklet a and the features belonging to the class b map to the same activity label and 0 otherwise.

Consistency Potential

The consistency potential is defined on the edges connecting tracklets to their corresponding activity segments. This potential function models the compatibility in the hierarchy between the lower level nodes and the higher level nodes which contain the same spatio-temporal region. The consistency potential is given by

$$\psi_c(x_{t_i}, x_{a_j}) = \exp(-k_{ij})\mathbf{I}(x_{t_i}, x_{a_j}), \tag{4.5}$$

where k_{ij} is the difference in the observation potentials of x_{t_i} and x_{a_j}. $\mathbf{I}(.)$ is the indicator function which returns 1 if a tracklet belongs to the same activity class as the activity segment to which it corresponds and 0 otherwise.

Spatio-temporal Context Potential

The spatio-temporal context potential is defined on edges connecting the action segments in the graph. Actions which are within a spatio-temporal distance of each other are assumed to be related to each other. There are three components to this potential: the spatial component, the temporal component and the frequency component.

The spatial and temporal components are modeled as normal distributions whose parameters μ_s, σ_s, μ_t and σ_t are computed using the training data. The spatial and temporal centroid of x_{a_i} and x_{a_j} is given by (s_i, t_i) and (s_j, t_j). The spatial component models the probability of an activity belonging to a particular category given its spatial configuration with its neighbor. The spatial potential is defined as

$$\psi_s(x_{a_i}, x_{a_j}) = N_{sd}(\|s_i - s_j\|^2; \mu_s(c_i, c_j), \sigma_s(c_i, c_j)), \tag{4.6}$$

Similarly, the temporal component models the probability of an activity belonging to a particular category given its temporal distance with its neighbor. The temporal potential is defined as

$$\psi_t(x_{a_i}, x_{a_j}) = N_{td}(\|t_i - t_j\|^2; \mu_t(c_i, c_j), \sigma_t(c_i, c_j)). \tag{4.7}$$

where $\mu_s(c_i, c_j)$,$\sigma_s(c_i, c_j)$, $\mu_t(c_i, c_j)$ and $\sigma_t(c_i, c_j)$ are the parameters of the distribution of relative spatial and temporal positions of the activities, given their categories.

The frequency component is the probability of two activities being within a predefined spatio-temporal vicinity of each other. The association probability $F(a_i, a_j)$ is computed as a ratio of the number of times an activity category c_j has occurred in the vicinity of activity category c_i to the total number of times the category c_i has occurred. Therefore, the spatio-temporal potential is given by

$$\psi_{st}(x_{a_i}, x_{a_j}) = F(a_i, a_j)\psi_s(x_{a_i}, x_{a_j})\psi_t(x_{a_i}, x_{a_j}). \tag{4.8}$$

4.3.2 Training

As in the previous chapter, training involves learning the edge weights for the graphical model. This is done by optimizing Equation 4.2 over the training data. Due to the log linear nature of the model, this is a convex optimization problem. For each training instance, the potential functions are obtained as explained above. The edge potentials are obtained by a pseudo-negative log likelihood optimization over loopy belief propagation.

4.4 Inference on the HMRF

Given an initial set of tracks, activity labels are obtained by inference on the HMRF using the learned parameters. Inference on a graphical model involves computing the marginal probabilities of the hidden or unknown variables given an evidence or an observed set of variables. There are two steps in our inference algorithm which are alternated in an EM framework to obtain the solution to the tracking and activity recognition problems. Using a set of pre-computed tracks, we obtain a set of activity labels in a bottom-up inference strategy. Next, using the obtained activities, tracks are re-computed in a top-down processing. These steps are explained in detail below.

4.4.1 Bottom-up Inference: From Tracks to Activities

Inference is the task of estimating labels of activities using the computed parameters. Due to the loopy nature of the graph, an exact solution is intractable. We consider an approximate objective to solve this optimization. A pseudo-likelihood function is computed by replacing the likelihood with univariate conditionals. A grouping of consecutive actions taking the same activity labels gives activity regions. Output of the algorithm is the labels of activities and the structure of the graphical model.

We choose the belief propagation method for inference on the graph. At each iteration, a node sends messages to its neighbor. All nodes are updated based on the messages from their neighbors. Consider a node $x_i \in V$ with a neighborhood $N(x_i)$. The message $m_{x_i,x_j}(x_j)$ sent by a node $x_i \in V$ to its neighbor $x_j \in V, (x_i, x_j) \in E$ can be given

as

$$m_{x_i,x_j}(x_j) = \alpha \int_{x_i} \Psi(x_i, x_j)\Psi_o(x_i, y_i) \prod_{x_k \in N(x_i)} m_{x_k,x_i}(x_i)dx_i \tag{4.9}$$

Here $\Psi(x_i, x_j)$ is taken as the association, consistency or spatiotemporal potential depending on the level of the nodes which it connects. We solve the inference problem starting with the lower level nodes and propagate the message to the higher level nodes. The marginal distribution of each activity region is given by

$$p(x_i) = \alpha\psi_o(x_i, y_i) \prod_{x_j \in N(x_i)} m_{x_j,x_i}(x_i) \tag{4.10}$$

The spatio-temporal region is said to belong to that category which has the highest marginal probability.

We use the loopy belief propagation algorithm due to its proven excellent empirical performance [74]. However, other variational inference methods such as the mean-field approximation can also be used for inference.

4.4.2 Top-down Inference: From Activities to Tracks

Tracks are to be formed by associating non-overlapping tracklets. Knowledge about the activities a person conducts in a given time interval can help in estimating his position and thereby the tracklet association. Therefore, in addition to the cost due to feature similarities, the compatibility of two tracklets given the activities that are being performed by the actor in the spatiotemporal region represented by the tracklets, given by the association potential, is utilized in the tracklet association algorithm.

The tracklet association is posed as a min-cost network problem as given in [82]. For a set of tracklets $t_1, t_2...t_n$, a set of m tracks $T_1, T_2, .., T_m$ are to be identified, such that, each track contains one or more tracklets. This can be accomplished by finding a set of m possible paths between two tracklets t_i and t_j, given by $h_{ij}^1, h_{ij}^2, .., h_{ij}^m$ known as the match hypotheses. Each hypothesis is associated with a cost of matching, given by d_{ij}^k. The tracks are defined as a matching function $T(f)$ of a set of binary flow variables f, estimated as

$$\begin{aligned}\hat{f} =& \underset{f}{\text{argmin}} P(f, X_t) = \underset{f}{\text{argmin}} \sum_i d_{en} f_{en,i} + \sum_i d_{ex} f_{i,ex} \\ &+ \sum_{ij} d_{ij} f_{ij} + \sum_{ij} w_a^{c_i,c_j} \psi_a^{(c_i,c_j)}(x_{t_i}, x_{t_j}) f_{ij}\end{aligned} \tag{4.11}$$

Here, f represents the set of binary flow variables indicating whether the tracklet i is an entry point $f_{i,en}$ of a track, exit point $f_{i,ex}$ of a track or a transition f_{ij} to another tracklet. Therefore, $f_{en,i}, f_{ex,i}, f_{ij} \in \{0, 1\}$. Every node can either be an entry node, an exit node, or be associated with a neighboring tracklet j. Therefore, $f_{en,i} + \sum_j \sum f_{ji} = 1, f_{i,ex} + \sum_j \sum_j f_{ij} = 1$.

The first and second constraints are binary constraints that model the cost associated with the image or motion features for inflow and outflow, given by d_{en} and d_{ex} respectively, the third constraint d_{ij} models the cost of association of two tracklets based on image or motion similarities. This matching cost is given as a weighted combination of distance between the color histograms of the tracklets and the spatiotemporal distance between them. The fourth term models the association cost of two tracklets t_i and t_j performing actions c_i and c_j and models the compatibility between activities performed by the tracklets. This term integrates the information from the higher-level activity nodes to the

inference of tracks.

The match hypotheses for a set of tracklets can be found using the K-shortest path algorithm [81]. An initial set of tracks are computed using just the binary constraints. Activity segmentation is conducted on these set of tracks by running a baseline classifier on the tracklets and grouping adjacent tracklets of a track belonging to the same activity into a single activity segment. The HMRF is constructed on these tracklets and activity segments. Using the obtained labels from recognition, the cost matrix is updated and the tracks are re-computed. The algorithm is repeated with the modified tracks.

4.4.3 Bi-directional Processing for Tracking and Activity Recognition

As explained in Section 4.3, the activity labels of the HMRF can be obtained by maximizing the energy function $E(X_t, X_a, T)$ in Equation 4.2, or in other words, minimizing $\Psi(X_t, X_a, T)$, i.e.

$$\begin{aligned} \hat{X} &= \underset{X_t, X_a}{\operatorname{argmax}} E(X_t, X_a, T) \\ &= \underset{X}{\operatorname{argmin}} \Psi(X_t, X_a, T) \end{aligned} \tag{4.12}$$

This is dependent on knowing the tracks T which are used to compute the nodes and edges of the graph as seen from Equation 4.2. Alternately, the track association problem utilizes the association potential which requires the activity labels assigned to the tracklets as can be seen from Equation 4.11. We can see that both X and T are dependent on each other. We propose to solve the tracking and activity recognition problems simultaneously. Since both X and T are unknown, this can be solved as an expectation maximization problem by

iterating between two steps.

E-Step: The expectation step computes the conditional expectation of the node labels $X^{(p)}$ given the parameters of the HMRF and the current estimation of the tracks given by $f^{(p)}$. This can be shown to be obtained as the posterior probabilities of the graphical model given by $X^{(p)} = \underset{X}{\operatorname{argmin}} \Psi(X_t, X_a, T(f^{(p)}))$. This can be solved as described in Section 4.4.1.

Maximization Step: The maximization step revises the flow parameters given the current node labels. We recompute the spatiotemporal context potential between the tracklets for all hypotheses and recompute the flow variables as $f^{(p+1)} = \underset{f}{\operatorname{argmin}} P(f, X_t^{(p)})$. This can be solved as described in Section 4.4.2.

The overall algorithm of our proposed method is explained in Algorithm 2.

4.5 Experiments and Results

4.5.1 Dataset

As in the previous chapter, we again perform experiments on the VIRAT public dataset. We perform two sets of experiments, one on Release 1 and the other on Release 2 of the data. For Release 1, there are 6 activities which are annotated: 1 - loading, 2 - unloading, 3 - open trunk, 4 - close trunk, 5 - enter vehicle, 6 - exit vehicle. In release 2, additional 5 activities have been added: 7 - person carrying an object, 8 - person gesturing, 9 - person running, 10 - person entering facility and 11 - person exiting facility.

During training, we normalize all distances with respect to the scale of the video to

make the approach invariant to scale. A threshold was set on the spatio-temporal distance between activities to initiate the dense graph. We used the distance threshold as a bounding box of 4 times the average dimensions of the person in the scene and a time threshold of 20 seconds. These values have been fixed experimentally. The graphical model is constructed on individual activity sequences.

To evaluate the accuracy of activity recognition, if there is more than a 40% overlap in the spatiotemporal region of a detected activity as compared to the ground truth and the labeling corresponds to the ground truth labeling, the recognition is assumed to be correct. Some examples of data which were correctly identified using our approach while incorrectly identified using a dense graphical model are shown in Figure 5.3.

4.5.2 Analysis of the Results

The classification results on VIRAT release 1 data is shown in Figure 5.5 and the results on VIRAT release 2 data is shown in Figure 5.7. The overall precision and recall values for VIRAT release 1 and comparison with approaches [1] and [3] is provided in Table 5.1. For release 2, in addition to providing comparison with BOW, we also provide comparison against two recent approaches [4] and [3]. Authors in [3] utilize spatiotemporal context, while the authors in [4] utilize sum-product networks on low level features to localize foreground objects and label activities. However, both these approaches only deal with the labeling problem. Our results are comparable to those in [4] and [3]. Although the overall accuracy is slightly lower with our approach, note that we consider the more challenging problem of joint labeling and tracking of activities in our approach which is

necessary for continuous videos. Table 4.2 shows the overall precision and recall values on VIRAT release 2 data for these approaches. Figures 5.5 and 5.7 show comparison with [3] only since the recognition scores of individual activities are not given in [4].

4.5.3 Tracking Results on VIRAT Release 2

Two examples of tracking results are shown in Figure 4.6. In the first case, we have a sequence of activities performed by a single person in the presence of occlusion. While the absence of context terminates the track due to the presence of occlusion, the presence of feedback detects that a trunk has been opened and it is very likely that the same person would close the trunk. Therefore, track is not terminated. Similarly, the second example shows two persons loading a trunk. While there is an error in the tracks in the absence of feedback, it is seen that the addition of feedback takes into account the fact that the person getting out of the vehicle is very likely to enter the vehicle (as often witnessed in the training data) and corrects the tracks.

For a qualitative evaluation of tracking using our approach, there is no prior research which has provided results on tracking that we can compare with. Also, datasets that have been popular in the tracking community do not present activity recognition results. Therefore, we provide tracking results against the ground truth (GT) in Table 4.3.

Method	**BOW**	**Gaur[1]**	**Zhu[3]**	**HMRF**
Precision	47.2	51.6	61.7	62.6
Recall	45.8	57.8	62.9	62.7

Table 4.1: Overall precision and recall values of methods BOW, Gaur et. al[1], Zhu et. al [3] and our approach for the VIRAT release 1 dataset.

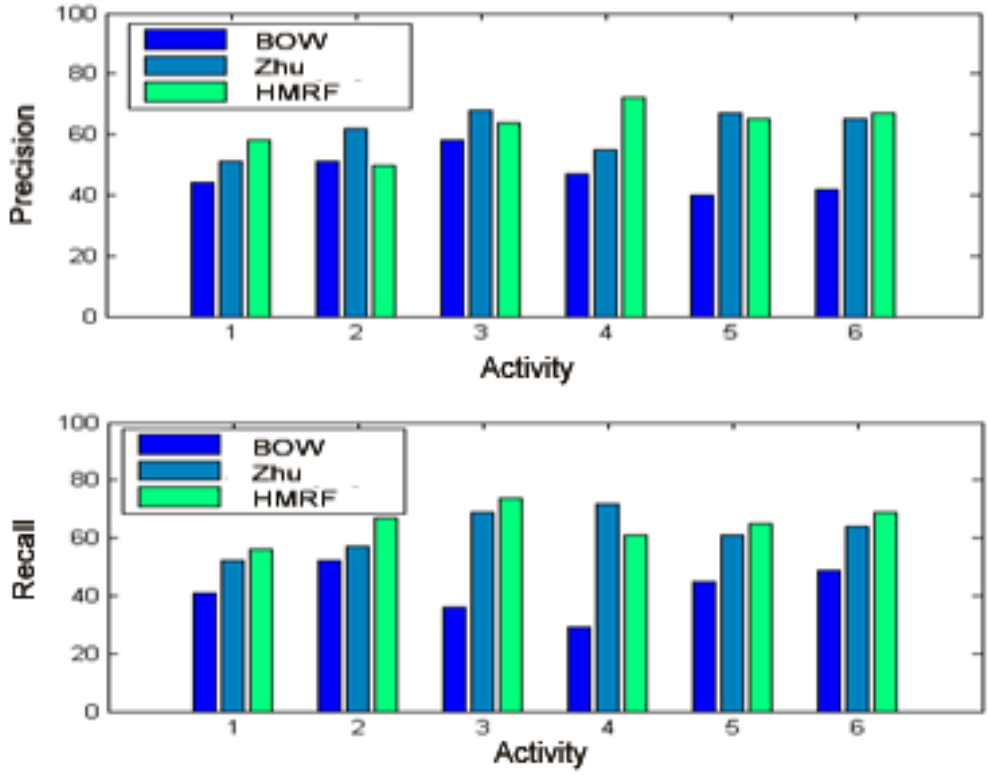

Figure 4.4: The figure shows the precision and recall obtained on the VIRAT release 1 dataset with our approach. Comparison has been shown to the performance of baseline classifier BOW [2] as well as Zhu et al [3]. The activities are listed in Section 4.5.1.

We compile the tracking results over 150 trajectories. The metrics used for measuring the tracking accuracy are: Mostly tracked: more than 80% of the track is correctly tracked; Mostly lost (ML) 20% or less tracked; Fragmented tracks (FT) Single track split into multiple IDS; ID switches (IDS) Switch between multiple tracks. It can be seen that there is a clear improvement in the tracking performance with the addition of bi-directional tracking.

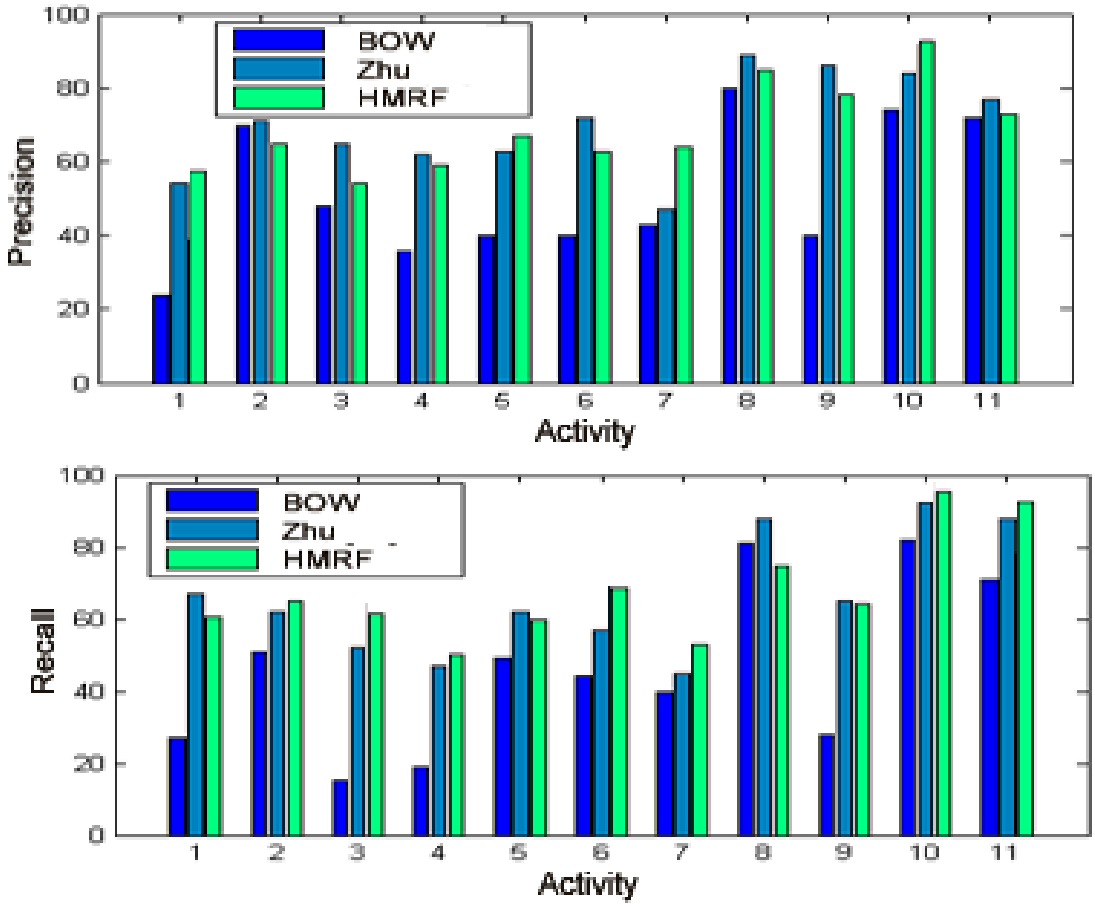

Figure 4.5: The figure shows the precision and recall obtained on the VIRAT release 2 dataset with our approach. Comparison has been shown to the performance of baseline classifier BOW [2] as well as Zhu et al [3]. The activities are listed in Section 4.5.1.

4.6 Conclusion

In this chapter, we have presented a method which can perform tracking, localization and recognition of activities in continuous sequences in an integrated framework. The biologically inspired bi-directional processing is shown to be effective in improving the accuracy of tracking as well as activity recognition.

Method	BOW	Amer[4]	Zhu[3]	HMRF
Precision	50.3	72	71.5	70.4
Recall	52	70	73.1	74.5

Table 4.2: Precision and recall values of methods BOW, Amer et. al[4] and Zhu et. al [3] and our approach for the VIRAT release 2 dataset.

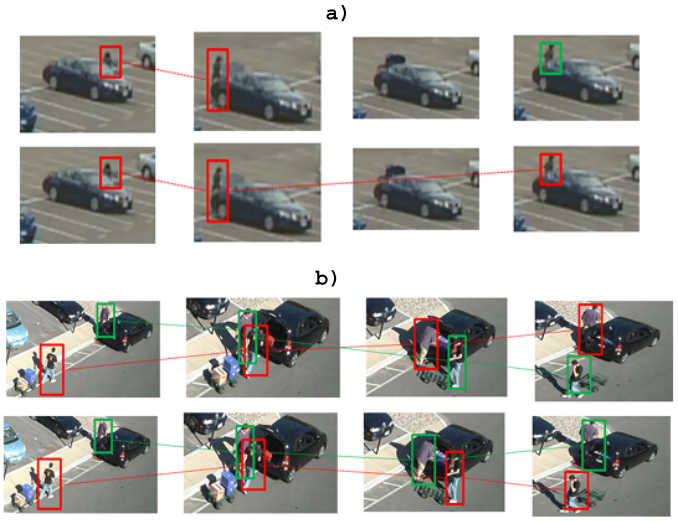

Figure 4.6: The figure shows two examples where tracking is improved with the addition of context. The top row shows the tracking results without activity context while the bottom row shows the result with the addition of feedback. Red and green signify different tracks in each case. In the first case, it is seen that the track was wrongly terminated due to occlusion in the absence of context. In the second case, the tracklet association error was corrected with the addition of context.

Metric	One-Step Tracking	Bi-directional processing
GT	150	150
MT	105	121
ML	19	13
FT	36	14
IDS	35	22

Table 4.3: Precision and recall values of methods BOW, Amer et. al[4] and Zhu et. al [3] and our approach for the VIRAT release 2 dataset.

Algorithm 2 Algorithm for integrated tracking, localization and labeling of activities in a test sequence using HMRF.

Input:	$\mathcal{S}_{\mathcal{R}} = \{V_1 \dots V_{N_{\mathcal{R}}}\}$	Set of training videos containing activity annotations
	A continuous test video containing one or more activities.	
Output:	Labels of activities $\{x_{a_1} \dots x_{a_q}\}$ and tracks $T_1 \dots T_k$	.
Training:	Train baseline classifiers $c_1 \dots c_N$ for N activities and model the association potential $\psi_a(x_{t_i}, x_{t_j})$, consistency potential $\psi_c(x_{t_i}, x_{a_j})$ and spatio-temporal potential $\psi_{st}(x_{a_i}, x_{a_j})$ between all pairs of activities using annotated training videos. Train model parameters w.	
Initial tracking:	Generate hypotheses on tracklets and get an initial estimates of tracks $f^{(1)}$.	

Testing:

1. Tracklets form the lower level node $\{x_{t_1}, x_{t_2} ... x_{t_p}\}$. Run baseline classifiers to compute labels l_{old} for all nodes and initial activity segmentation using current tracks $\{x_{a_1}, x_{a_2}, ... x_{a_q}\}$.

2. Compute observation potential $\psi_o(x_{t_i}, y_{t_i})$ for each tracklet and $\psi_o(x_{a_i}, y_{a_i})$ activity segment using the baseline classifiers.

3. Initialize hierarchical MRF G containing p tracklets and q activity segments.

4. **E-Step:** Run inference to generate posteriors and labels for all nodes l_{new}.

5. **M-Step:** Recompute association potential using current labels l_new. Solve Equation 4.11 using the revised potential and recompute tracks $f^{(new)}$.

6. Compute new localization using f_{new}. Rebuild graph.

7. Repeat the EM algorithm until $l_{old} = l_{new} \parallel n_{iter} = max_{iter}$

8. Output tracks $T(f)$ and current labels for $\{x_{a_1}, x_{a_2}, ... x_{a_q}\}$.

Chapter 5

Structure Discovery Using L1-regularized Learning In Graphical Models

5.1 Introduction

Most early approaches for activity recognition focused on modeling and representation of single person activities. However, while dealing with more complex scenarios of multiple person activities or continuous videos, it has been widely acknowledged that, in addition to the features themselves, the structural information between sets of features and/or objects, often termed as context, plays an important role in discriminating between activities. Researchers have proposed different ways to represent the structure in a video. Graphical models are commonly used to encode such structural relationships [42, 4, 66, 71, 10].

What is the ideal structure of this graphical model? Given that the useful contextual relationships in the video are sparse as compared to the total number of elements, how do we discover this sparse structure? The objective of this chapter is to propose a method to automatically learn this structure.

The effectiveness of a graphical model depends on the structure as well as the parameters chosen for the model. In the case of unconstrained videos such as surveillance videos, the graph structure varies with the number of people and activities in the video. Therefore, choosing the right set of edges for the graph is a challenging task. Prior approaches such as [83] have fixed the graph apriori or used dense graphs as an alternative. The disadvantage of a dense graph is that the number of parameters to be estimated in the model grows exponentially with the number of edges. This makes the computation of parameters statistically inefficient and the model inaccurate. In addition, it may not be practical to fix the graph in some applications. Here, we perform structure discovery as a part of the parameter learning process such that the resulting graph has a sparse set of edges.

Sparsity has widely been used in different applications where it is advantageous to have a small set of parameters that effectively model the data. In continuous videos with a variable number of activities and people, the total number of possible contextual relationships can be exponential in the number of activities. However, in reality, the number of activities which are actually related to each tends to be a small subset of all possible relationships. For example, two people in the scene may be acting independently and may not influence actions performed by each other. Similarly, a preceding action may

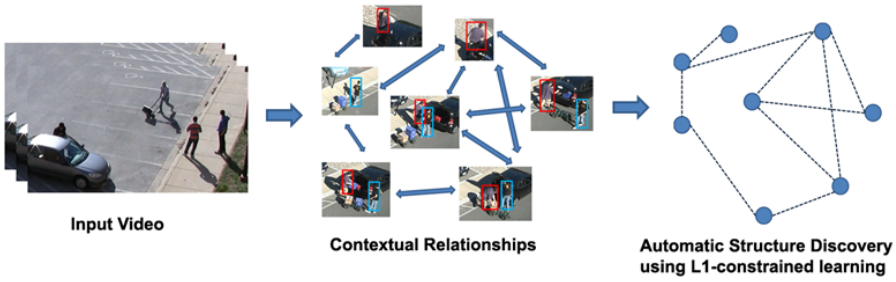

Figure 5.1: A continuous video can consist of multiple activities. The challenge in context modeling using graphical models is to arrive at a structure which effectively models the contextual relationships between activities. In this chapter, we propose an L1-regularized learning of the graphical model which performs an automatic structure discovery on the graph.

provide sufficient context to the next action, while the other relationships may not be as significant. Therefore, by learning a sparse set of parameters, and in turn a sparse graph, we can effectively retain those contextual relationships which tend to influence the recognition scores to a greater extent, while also reducing the computational complexity involved in solving a dense graph.

L1-regularized learning is a useful tool to select a sparse set of features which represent a particular data. Different methods of sparse dictionary learning such as deep Boltzmann machines [84], stacked auto-encoders and sparse coders [85] have been used to represent image data in the context of object recognition. These concepts have been extended to video data in approaches such as 3D convolutional neural networks [86] and independent subspace analysis [87]. Such approaches have demonstrated competitive performances in classification. However, most computer vision approaches which have used L1-regularized optimization have only explored sparsity in feature representation and not in structure representation. The main novelty of this work is to extend the concept of

sparse feature learning to estimating a sparse set of relationships between events in continuous videos.

We model activities in video using features as well as the inter-activity structural relationships. Starting with a dense graph, a sparsity constraint is imposed on the edges as well as the features. The resultant graph models the most common contextual relationships between activities, while also choosing the optimal parameters and features in an automatic and computationally efficient manner. We demonstrate the sparsity of the graphical model obtained, thereby showing the ability of the algorithm to discover contextual relationships.

5.1.1 Related Work

Recently, sparse coding techniques have gained popularity in the field of activity recognition. A 3D convolutional network learned over a fixed set of input frames to represent the video was proposed in [86]. A cascading system of independent subspace analysis and spatial pooling was used to learn a set of local features. These features were classified using K-means vector quantization and χ^2 kernel in [87]. Action attributes are modeled using a sparse dictionary based representation in [88]. Anomaly detection is performed by measuring the encoding error of features learned using sparse coding in [89]. Recent methods like [90, 91] use group sparsity for feature learning. These methods primarily focus on representation of features at different scales in a video [92]. We on the other hand, use a sparse learning approach for automatic structure discovery and parameter learning in graphical models.

In applications such as activity recognition, the structure of the graph is difficult to

determine. Prior approaches have either used fixed graphical models such as in [83] [71] or built graphs of known structures such as and-or graphs [10]. Recent approaches such as [3] have tackled this problem by using a greedy forward search to determine the best possible graph, thereby making the learning and inference very intensive. The idea of learning an optimal set of structural relationships in a L1-regularized learning framework is the novelty of our approach, and can be extended to these other applications where structure can play a crucial role.

5.2 Overview

The illustration of our proposed method is shown in Figure 5.2.

We start with the same formulation of the problem as in the previous chapter. Pre-processing consists of computing tracklets and computing low level features such as space-time interest points in the region around these tracklets. Tracking involves association of one or more tracklets to tracks. Activity localization can now be defined as a grouping of tracklets into activity segments and recognition can be defined as the task of labeling these activity segments.

To begin with, we generate a set of match hypotheses for tracklet association and a likely set of tracks. An observation potential is computed for each tracklet using the features computed at the tracklet. Tracklets are grouped into activity segments using a standard baseline classifier such as multiclass SVM or motion segmentation.

Next, we construct a two-level Markov random field using the tracklets and activity segments. The first level nodes correspond to the tracklets and the second level nodes

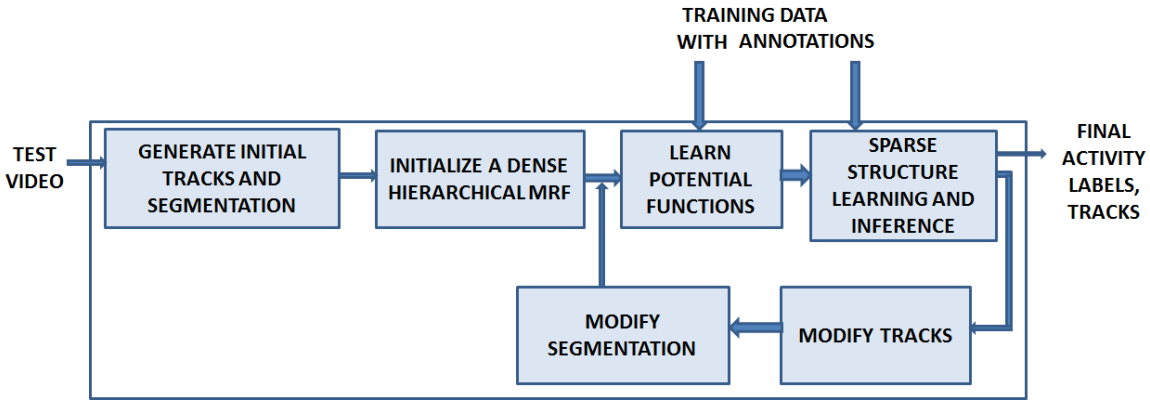

Figure 5.2: Figure shows the illustration of our proposed method. Given a continuous video with computed tracklets, a set of tracks and activity segments are initialized. An HMRF model is built over the tracklets and segments. Edge potentials are learned on the annotated training data. Starting with a dense graph, L1-regularized structure learning gives a sparse set of edges. Inference on this graphical model provides a revised set of labels for the activities which can be fed back into the system to regenerate the tracks and rebuild the HMRF. The procedure is repeated until a stop criterion is reached. The tracks and labels of all segments are provided as output.

correspond to the activity segments. One or more tracklets can correspond to the same activity segment. Edges model relationships between nodes of the same level as well as nodes at different levels. This structure incorporates the context information between adjacent tracklets as well as across activity segments.

The dense HMRF has edges connecting each node to all other nodes within a certain spatiotemporal range. This gives us the initial graph on which we perform the learning and recognition.

The node features and edge features for the potential functions are computed from the training data. There are two tasks to be performed on the graph - choosing an appropriate structure and learning the parameters of the graph. Both these steps can be performed simultaneously by posing the parameter learning as an L1-regularized optimization. The

sparsity constraint on the HMRF ensures that the resulting parameters are sparse, thus capturing the most critical relationships between the objects. The parameters which are set to zero denote the edges which have been deleted from the graph. The non-zero parameters denote the parameters of edges retained after automatic structure discovery.

Inference on this graph provides the posterior probabilities for all nodes using information available at two resolutions. The activity labels are used in a top-down fashion to recompute the tracks. Activity segmentation on the recomputed tracks gives us a new set of nodes on which structure learning and inference is then repeated. Convergence is said to be achieved when the node labels and tracks do not change from one iteration to the next.

The output of the algorithm is a set of tracks, segments and the labels assigned to each segment.

5.3 L1-regularized Graphical Model for Activity Recognition

5.3.1 Standard L1-regularization of Parameters:

The graphical model consists of a set of potentials and a set of parameters. As described in Equation 4.2, there are three kinds of parameters described on the edges of the graph - $\mathbf{w_a}$, $\mathbf{w_c}$ and $\mathbf{w_{st}}$. The overall parameter vector of the model is therefore formed by concatenating the weight vectors of all the potential functions, given by $\mathbf{w} = [\mathbf{w}_a^T, \mathbf{w}_c^T, \mathbf{w}_{st}^T]^T$. We begin by computing the potential functions on a densely connected graphical model. While we could use a fully connected graphical model, here, we assume that nodes within

a specified spatiotemporal distance can influence each other contextually. Therefore, we build a graph where every node is connected to all nodes within a specified spatiotemporal distance. The structure discovery using L1-regularization of parameters is carried out as given below.

We wish to learn the structure of a sparsely connected graph, which represents the contextual relationships in the data. We propose to do this by an setting a sparsity condition on the parameters. A sparse set of parameters also results in a sparse set of edges, since setting a parameter to zero sets the corresponding potentials of the energy function to 1. We also set a sparsity constraint on the node parameters for effective feature coding. The non-zero node parameters would specify the sparse node features which are chosen to model the activities. The non-zero edge parameters would specify the edges which encode important contextual information between activities. This is done by imposing a restriction on the L1-norm of the parameter vector. For a set of m training instances and n nodes in the graph, the L1-regularized learning problem can be given by

$$\begin{aligned} F = \min_{\mathbf{w}} - \sum_{k=1}^{m} [\sum_{i=1}^{n} [\mathbf{w}_o \psi_o(x_o, y_o) + \sum_{j \in N(i)} \mathbf{w}_{ij} \psi_e(x_i, x_j)]] \\ + m \log Z(w) + \lambda |\mathbf{w}|_1 \end{aligned} \tag{5.1}$$

Here, λ is the regularization parameter which decides the sparsity of the resultant solution. This poses the structure learning as an optimization problem. This is a useful formulation for learning the graphical model since it does not impose any constraint on the structure and is also much faster than the search based method of edge addition/deletion.

5.3.2 Group L1-regularization of Parameters:

In the above formulation, each node can take as many states as the number of meaningful activities in the data. For multi-state nodes representing n activities, the potential function can take n^2 values for each edge. Each edge is therefore represented by an edge parameter $\mathbf{w}$ which is composed of a matrix of n^2 elements, given by w_{ij}, where $i, j \in \{1, 2, .., n\}$

We want to learn the edges of a graphical model, each edge parameter representing the joint distribution of a node given the neighbor. The edge is reduced to zero only if *all* elements of the edge are set to zero. This is achieved by the L2-regularization of the n^2 elements over each edge. However, each non-zero edge in this case tends to have all parameters set to non-zero elements. To introduce sparsity for the elements of the non-zero edges, we introduce the l1-regularization over this function. This leads us to the group l1-regularization, which is defined as the l1-regularization of l2-norm of $\mathbf{w}$. Since there are three kinds of edge potentials in the graph, we form three regularization factors for the three sets of edges with the flexibility to choose three different regularization parameters. The optimization function therefore reduces to

$$\begin{aligned} F = \min_{\mathbf{w}} & -\sum_{k=1}^{m}[\sum_{i=1}^{n}[\mathbf{w}_o\psi_o(x_o, y_o) + \sum_{j\in N(i)} \mathbf{w}_{ij}\psi_e(x_i, x_j)]] \\ & + m \log Z(w) + \lambda_c \sum_{i\in E_c} \sum_{j\in N(i)} \|\mathbf{w_c}\|_2 + \lambda_a \sum_{i\in E_a} \sum_{j\in N(i)} \|\mathbf{w_a}\|_2 \\ & + \lambda_{st} \sum_{i\in E_{st}} \sum_{j\in N(i)} \|\mathbf{w_{st}}\|_2 \end{aligned} \tag{5.2}$$

This function can be viewed as a sum of a differentiable convex function and a convex regularizer. We solve this using the Barizilai-Borwein spectral projection method

[93]. This method views the equation as a constrained optimization problem, with a series of group constraints. In the group regularization, the constraint given in the form of $\sum_{\mathbf{g}} \lambda_g \mathbf{w}_g$, replaces the non-differentiable regularizer with a linear function. The function is solved using a variant of the projected-gradient method with a variable step size. Therefore, we now have a smooth optimization problem over a convex set.

The spectral projection method solves for the parameters in an iterative manner. In each iteration, the value of the parameters is changed in the direction of the projection of the current values on the function space, i.e.,

$$\mathbf{w_{k+1}} = \mathcal{P}_{\int}(\mathbf{w_k} - \alpha \nabla F(\mathbf{w_k})), \tag{5.3}$$

where $\mathcal{P}_{\int}$ represents a Euclidean projection and α is the step size. For details of the spectral projection method, please refer [93]. The final solution introduces sparsity for the edges of the graph using the L1 constraint on the groups, as well as within each group by minimizing the total number of parameters.

Inference is carried out on the test video by using the parameters computed. Due to the loopy nature of the graph, an exact solution is intractable. We consider an approximate objective to solve this optimization. A pseudo-likelihood function is computed by replacing the likelihood with univariate conditionals. Convergence was achieved using the mean-field approximate inference. A grouping of consecutive actions taking the same activity labels gives activity regions. Output of the algorithm is the labels of activities and the structure of the graphical model.

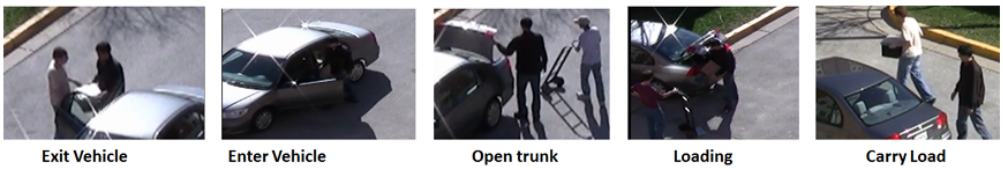

Figure 5.3: A few examples of activities which were incorrectly detected using a dense graphical model ($\lambda = 0$) and correctly discovered after the L1-regularized parameter learning. The advantage of learning a sparse graph is better representation of contextual information.

5.4 Experiments

We perform two sets of experiments on the VIRAT dataset, one on Release 1 and the other on Release 2 of the data. For Release 1, there are 6 activities which are annotated: Person entering vehicle, person exiting vehicle, person opening trunk, person closing trunk, person loading vehicle and person unloading vehicle. In release 2, additional 5 activities have been added: person carrying an object, person gesturing, person running, entering and exiting a facility.

For VIRAT release 1, we provide comparison with recent approaches [1] and [3]. For release 2, in addition to providing comparison with BOW, we also provide comparison against two recent approaches [4] and [3]. Authors in [1] model structural relationships between features. Authors in [3] utilize spatiotemporal context, while the authors in [4] utilize sum-product networks on low level features to localize foreground objects and label activities. While a fixed graph is used for modeling in [4] and [1], the authors in [3] perform a greedy forward search to determine the graph.

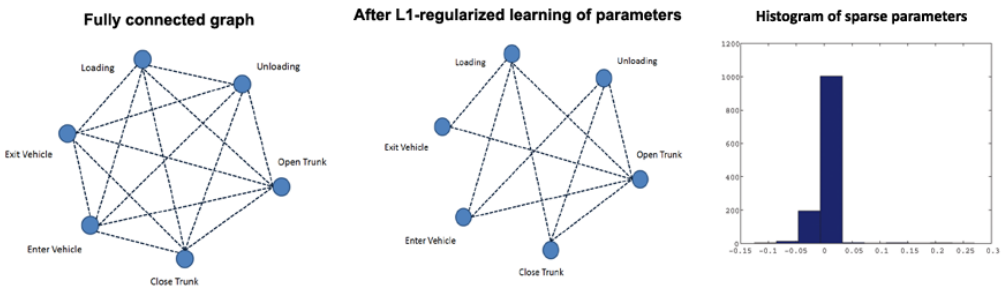

Figure 5.4: The figure shows the sparse contextual relationships discovered by L1-regularized learning on VIRAT 1 dataset. The figure on the left shows the fully connected model assumed before parameter learning. The next figure shows the sparse relationships obtained after parameter learning. The edges corresponding to parameters which are set to zero have been deleted from the graph. The bar graph on the right shows the histogram of obtained sparse parameters.

5.4.1 Methodology

We used a randomly selected set of half the data for training and the other half for testing. During the training, we assume that the activity regions and the activity labels are available to us. We normalize all distances with respect to the scale of the video to make the approach invariant to scale. The graphical model is constructed on individual activity sequences. The regularization parameters experimentally determined where $\lambda_c = 3$, $\lambda_a = 3$, $\lambda_{st} = 4$.

To evaluate the accuracy of activity recognition, if there is more than a 40% overlap in the spatiotemporal region of a detected activity as compared to the ground truth and the labeling corresponds to the ground truth labeling, the recognition is assumed to be correct. Some examples of data which were correctly identified using our approach while incorrectly identified using a dense graphical model are shown in Figure 5.3.

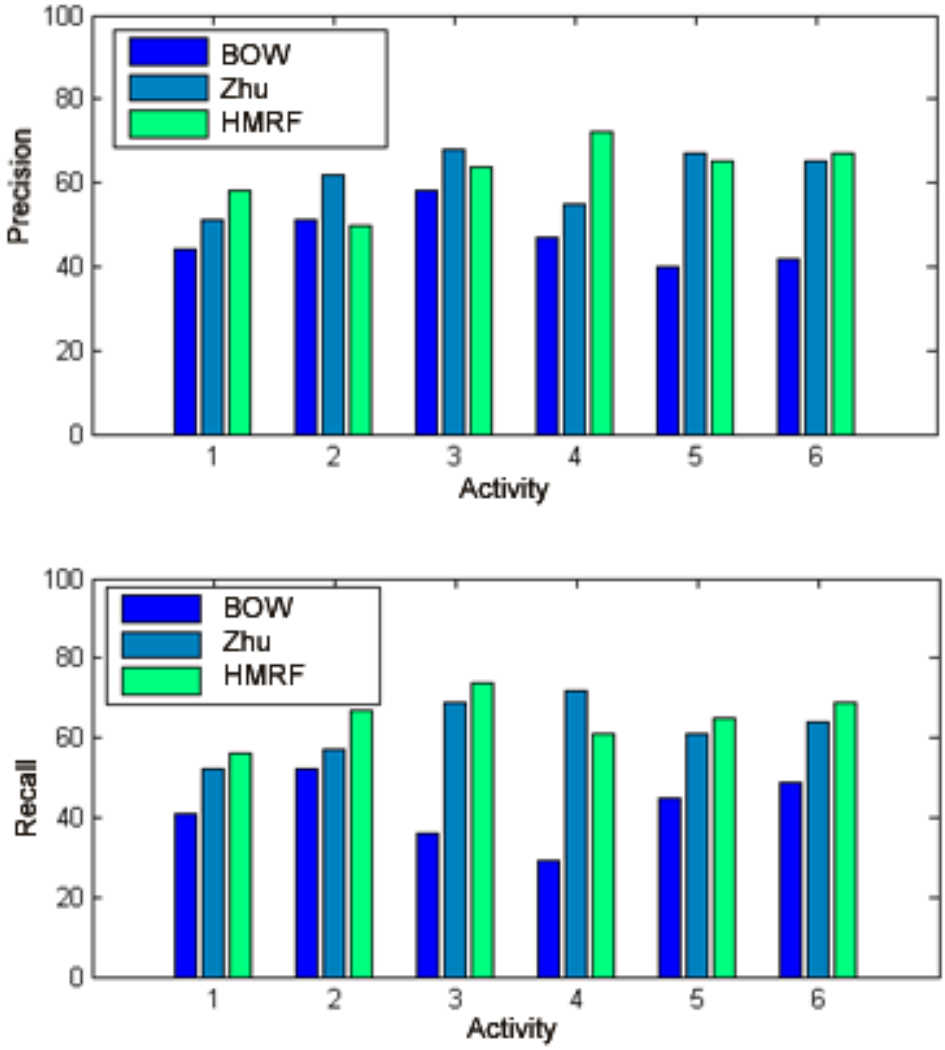

Figure 5.5: The figure shows the precision and recall obtained on the VIRAT release 1 dataset with our approach. Comparison has been shown to the performance of baseline classifier BOW [2] as well as Zhu et al [3]. The activities in order are: Person entering vehicle, person exiting vehicle, person opening trunk, person closing trunk, person loading vehicle and person unloading vehicle.

5.4.2 Analysis of the Results

The sparse structure discovered by the L1-regularized learning for the parameters w_{ij} of an edge for VIRAT release 1 is shown in Figure 5.4. The structure represents the contextual relationships modeled in a parameter $\mathbf{w_k}$. It can be seen that 9 relationships out of 15 possible combinations of 6 activities were retained. In addition, it was also observed that the connections learnt could be intuitively justified as the contextual relationships between activities that are often observed in the training data. For example, loading and unloading is often related to opening and closing the trunk. These edges of the graph were retained, while some others, such as the edge connecting loading to unloading was deleted. Only about 32.9% of the parameters were non-zero in the resulting model. The histogram of computed parameters is shown in Figure 5.4 c).

We also demonstrate the structure discovery in an activity sequence from VIRAT release 1 in Figure 5.6. A dense graphical model constructed over a sequence of 5 activities is shown on the left and the corresponding sparse graphical model obtained using our approach is shown on the right. The dense graphical model was constructed by adding an edge between every two nodes which had a spatio-temporal distance of less than half the maximum separation between activities in the sequence. After L1-regularization, those

Method	**BOW**	**Gaur[1]**	**Zhu[3]**	**Our Approach**
Precision	47.2	51.6	61.7	64.8
Recall	45.8	57.8	62.9	64.1

Table 5.1: Overall precision and recall values of methods BOW, Gaur et. al[1], Zhu et. al [3] and our approach for the VIRAT release 1 dataset.

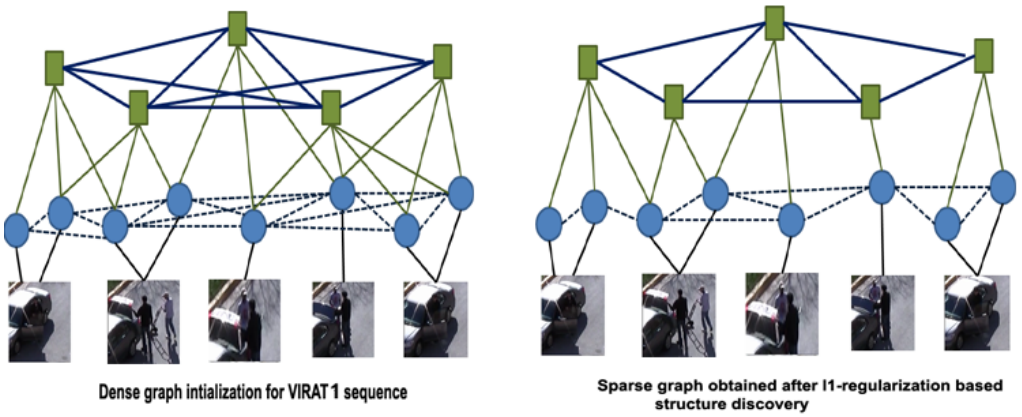

Figure 5.6: For an activity sequence from VIRAT release 1 containing 5 activities, we show the initial dense hierarchical Markov random field model constructed on the sequence (left) and the corresponding sparse graphical model obtained after L1-regularized learning of parameters (right).

edges whose parameters have been set to zero were deleted resulting in the sparse graph.

The classification results on VIRAT release 1 data is shown in Figure 5.5. The overall precision and recall values for VIRAT release 1 and comparison with recent approaches [1] and [3] is provided in Table 5.1. It can be seen that the performance of our approach is better than the recent state-of-the-art methods for most activities. The overall performance is also better. This improvement can be attributed to the improvement in structure, which captures the relationships across activities effectively.

Similarly, we compute the graphical model and the activity recognition scores for VIRAT release 2 dataset consisting of 11 activities. The precision and recall values obtained are shown in Figure 5.7. It can be seen that our approach performed better than the current state-of-the-art methods. The overall accuracy of our method and other recent approaches is shown in Table 5.2. For one sequence of activities containing 7 meaningful activities, the initial dense HMRF before L1-regularized learning and the output of our algorithm, which

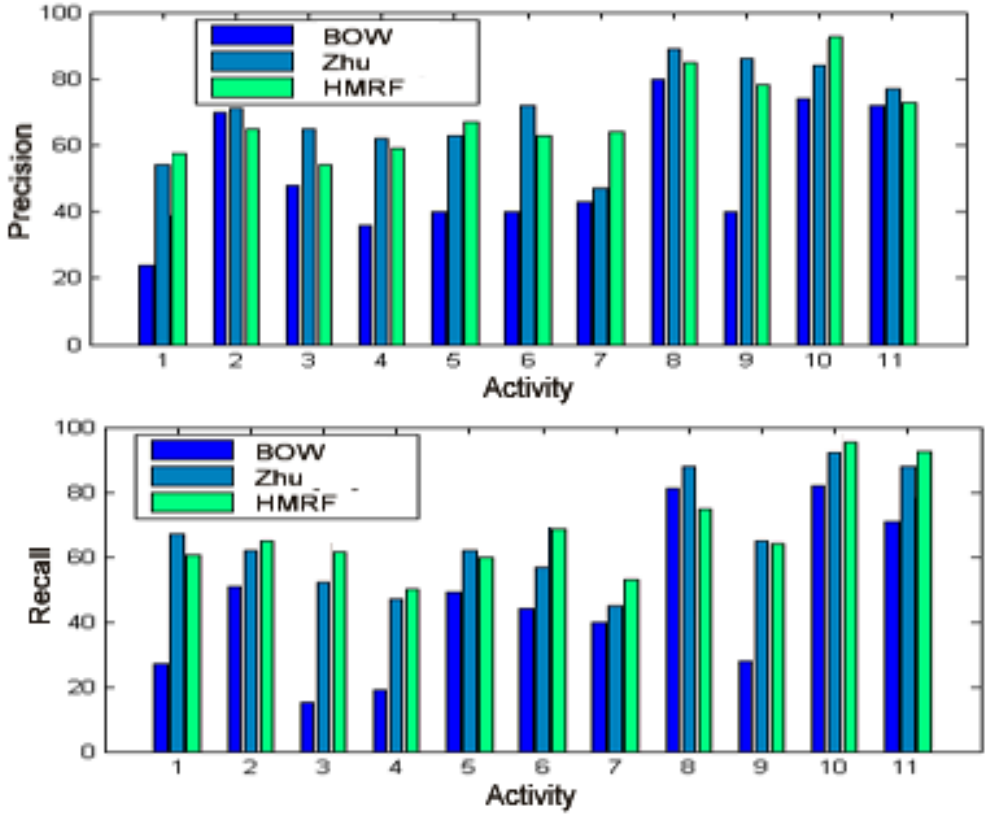

Figure 5.7: The figure shows the precision and recall obtained on the VIRAT release 2 dataset with our approach. Comparison has been shown to the performance of baseline classifier BOW [2] as well as Zhu et al [3]. The activities in order are: Person entering vehicle, person exiting vehicle, person opening trunk, person closing trunk, person loading vehicle, person unloading vehicle, person carrying an object, person gesturing, person running, entering and exiting a facility.

is the resulting sparse graph are shown in Figure 5.8. About 31.3% of the parameters were retained after the L1-regularized learning.

For the 11 activities of VIRAT release 2, we demonstrate the contextual relationships captured in the parameter matrix $\mathbf{w}$ in Figure 5.9. Again, it was seen that our approach captured the contextual relationships which seemed most intuitive. For example, the activity running was mostly associated with people entering/exiting the facility or exiting a vehicle and opening a trunk. These relationships are seen in the resulting graph. The histogram of the computed parameters is also shown in Figure 5.9 c). From the histogram, it is evident that the parameters are very sparse, thereby eliminating edges of the graph.

Computational Time: It is well known that inference on a graphical model with loops is an NP-hard problem and can be tractable only with a bounded tree width [94]. While it can be solved in polynomial time in the size of the structure for select low-tree width graphs, in our case, we have an unbounded tree-width with multiple states that makes exact theoretical calculations on computational complexity very difficult. Also, the structure of the graph varies depending on the sequence. However, it can be said that, with the reduction in the number of edges and the introduction of sparsity, the tree-width as well as the number of loops in the graph is very likely to reduce, thereby achieving a speed-up in the performance. Experimentally, we run the approach on the dense graph (setting all values of λ to zero) and compare the taken for inference on the same set of activities using the sparse graph. Values were computed for 30 sequences containing 5 nodes on matlab in Intel(R) Core(TM) i3 CPU @2.27GHZ . It was found that inference on the dense graph took 0.1248 seconds while the inference on the sparse graph with roughly 30% of the edges

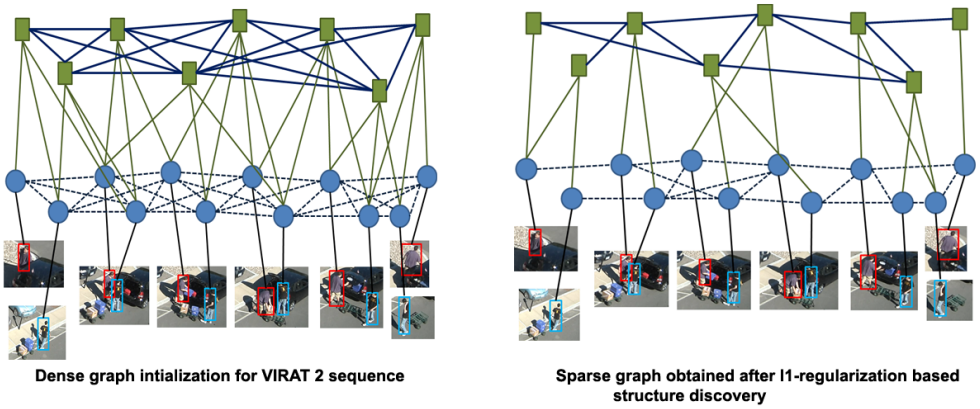

Figure 5.8: For an activity sequence from VIRAT release 2 containing 7 activities, we show the initial dense hierarchical markov random field model constructed on the sequence (left) and the corresponding sparse graphical model obtained after L1-regularized learning of parameters (right).

took only 0.0312 seconds. This clearly shows the improvement in speed due to sparsity. In summary, not only do we achieve higher accuracy in the graph discovery process, we do so with an order of magnitude less computational time.

5.5 Conclusion

In this chapter, we have demonstrated a method which can automatically discovery the contextual relationships between activities in continuous sequences. We have demonstrated that the L1-regularized learning of parameters is a good substitute to alter-

Method	**BOW**	**Amer[4]**	**Zhu[3]**	**Our approach**
Precision	50.3	72	71.5	72.6
Recall	52	70	73.1	74.2

Table 5.2: Precision and recall values of methods BOW, Amer et. al[4] and Zhu et. al [3] and our approach for the VIRAT release 2 dataset.

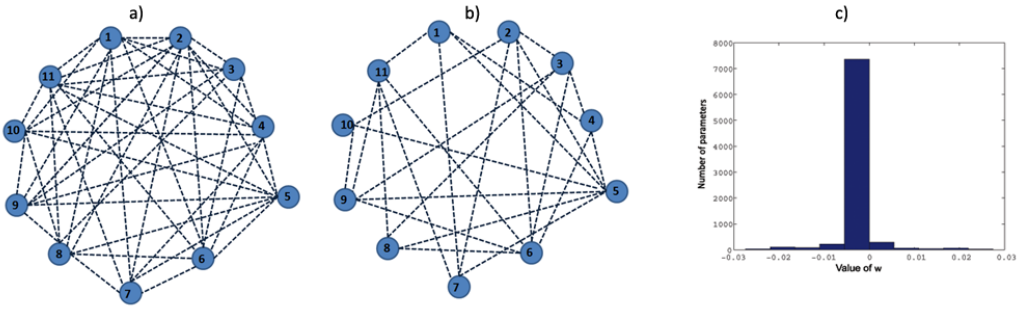

Figure 5.9: The figure shows the sparse structure of the graph discovered by L1-regularized learning on 11 activities of VIRAT 2 dataset. Figure a) shows the fully connected graph assumed before parameter learning. Figure b) shows the sparse graph obtained after parameter learning. The edges corresponding to parameters which are set to zero have been deleted from the graph. Figure c) shows the histogram of the learned parameters $\mathbf{w}$. From the histogram, it can be seen that $\mathbf{w}$ is sparse. The activity labels are the same as in Figure 5.7.

nate methods such as greedy forward search. The resulting graph was sparse and intuitively picked those edges which were effectively improved the recognition scores. We demonstrated an improvement in recognition accuracy as well as inference time using our approach. This method can be extended to other applications which utilize graphical models for context representation.

Chapter 6

Conclusion and Future Work

6.1 Thesis Summary

In this thesis, we have proposed novel algorithms for detection and recognition of activities in wide-area continuous videos. The core idea of the work is to utilize spatio-temporal contextual information obtained from the current and neighboring activities to improve the accuracy of localization and recognition. The contributions can be broken down in the following manner.

In chapter 2, we demonstrated a system to recognize and label activities in multi-person wide-area videos. We introduced the different stages of processing involved in such videos, namely the identification of motion patterns, localization and recognition of activities. We showed that optical flow is a useful tool to model such activities. Elimination of noise was carried out by integrating flow to obtain streaklines. Clustering of streaklines achieved identification of motion patterns. Localization and labeling of activities was

carried out using shape matching and subspace analysis. However, this method treated activities individually. It was seen that some activities were confused with each other due to occlusion or similarities in motion patterns. We proposed that such limitations could be overcome if activity context was taken into consideration.

In chapter 3, we have demonstrated the use of graphical models in discovering the contextual relationships between activities. We have shown that the spatial and temporal neighborhood of an activity provides contextual information which can be used to localize and label activities. This information was modeled using a graphical model. The experiments showed that the use of context improved the recognition scores significantly as opposed to using just the baseline classifiers.

In chapter 4, we discussed the relationship between tracks and activities. Tracks provide contextual cues regarding the location of an activity. We also showed that the knowledge about the activities in a continuous video sequence can aid in the tracklet association for estimation of tracks. We proposed a bi-directional approach to solve both these problems simultaneously. We showed that the results obtained were comparable to the case where tracks were assumed to be accurate or activities were localized beforehand.

Finally, we proposed a method to estimate the structure of a graphical model using sparse learning. Since the contextual relationships in a scene are sparse as compared to the total number of activities, a better estimation of the structure of the graphical model can aid in faster computation and can help in arriving at a more accurate model. L1-regularized parameter estimation was suggested to achieve the same.

6.2 Future Work

This work on activity analysis in wide-area continuous videos has many possible future directions. In this work, we considered that a video sequence is given to us and activities are to be recognized offline. However, it would be interesting if the inference could be carried out real-time. The problem would therefore reduce to the prediction of future activities given the past activities. The generative nature of a Markov random field is useful for this purpose. The activity recognition system we have proposed can also be extended to learn a semantic map of activities in wide-area continuous videos.

One of the major limitations of the current work is that, it assumes sufficient amount of training data, based on which the model is built. The training data is assumed to have all feasible combinations of activity sequences. In a typical wide-area video, such a condition often does not hold. Typically, there is a limited amount of training data to begin with, although from time to time, there is a possibility of more data being added into the system. Therefore, online learning of the model can be a good future direction to this work. This addition can make the system more robust.

Although we present an approach to learn the structural relationships for activity recognition using L1-regularization, this idea can be generalized across applications. The importance of structural relationships has been acknowledged in a wide range of applications. The authors in [83], [71] explore structural relationships across agents performing an activity. Structural information in video is used to aid segmentation in [77]. Structure has widely been used in the field of object recognition to model relationships between object parts [95] and to discover new object categories in [96]. Recently, scene structure

was modeled to learn correlation between object classes in [97]. The idea of learning structural relationships in a L1-regularization based framework can be extended to these other applications where structure can play a crucial role.

Bibliography

[1] U. Gaur, Y. Zhu, B. Song, and A. Roy-Chowdhury, String of feature graphs analysis of complex activities, in *International Conference on Computer Vision*, 2011.

[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, Actions as space-time shapes, in *International Conference on Computer Vision*, 2005.

[3] Y. Zhu, N. Nayak, and A. Roy-Chowdhury, Context-aware modeling and recognition of activities in video, in *Computer Vision and Pattern Recognition*, 2013.

[4] M. R. Amer and S. Todorovic, Sum-product networks for modeling activities with stochastic structure, in *Computer Vision and Pattern Recognition*, 2012.

[5] J. Aggarwal and Q. Cai, Computer Vision and Image Understanding (1999).

[6] N. M. Nayak, R. Sethi, B. Song, and A. K. Roy-Chowdhury, Motion pattern analysis for modeling and recognition of complex human activities, in *Guide to Video Analysis of Humans: Looking at People*, edited by T. Moeslund, A. Hilton, V. Kruger, and L. Sigal, Springer, 2004.

[7] C. Schuldt, I. Laptev, and B. Caputo, Recognizing human actions: A local svm approach, in *Internation Conference on Pattern Recognition*, 2004.

[8] M. Ryoo, C. Chen, J. Aggarwal, and A. Roy-Chowdhury, An overview of contest on semantic description of human activities (sdha) 2010, in *International Conference on Pattern Recognition*, 2010.

[9] S. Oh and et al, A large-scale benchmark dataset for event recognition in surveillance video, in *Computer Vision and Pattern Recognition*, 2011.

[10] M. Pei, Y. Jia, and S.-C. Zhu, Parsing video events with goal inference and intent prediction, in *International Conference on Computer Vision*, 2011.

[11] G. Denina et al., Videoweb dataset for multi-camera activities and non-verbal communication, in *Distributed Video Sensor Networks*, 2010.

[12] P. Turaga, R. Chellappa, V. Subrahmanian, and O. Udrea, IEEE Transactions on Circuits and Systems for Video Technology (2008).

[13] M. Ryoo and J. Aggarwal, International Journal on Computer Vision (2009).

[14] W. Forstner and E. Gulch, ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data (1987).

[15] C. Harris and M. Stephens, A combined corner and edge detector, in *Fourth Alvey Vision Conference*, 1988.

[16] T. Lindeberg, International Journal of Computer Vision (1998).

[17] R. Chaudhary, A. Ravichandran, G. Hager, and R. Vidal, Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions, in *Computer Vision and Pattern Recognition*, 2009.

[18] D. Lowe, Object recognition from local scale-invariant features, in *International Conference on Computer Vision*, 1999.

[19] I. Laptev and T. Lindeberg, Local descriptors for spatio-temporal recognition, in *First International Workshop on Spatial Coherence for Visual Motion Analysis*, 2004.

[20] N. Cinbis and S. Sclaroff, Object, scene and actions: combining multiple features for human action recognition, in *European Conference on Computer Vision*, 2010.

[21] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, International Journal of Computer Vision (1994).

[22] A. Efros, A. Berg, G. Mori, and J. Malik, Recognizing action at a distance, in *International Conference of Computer Vision*, 2003.

[23] A. Yilmaz and M. Shah, Computer Vision and Pattern Recognition (2005).

[24] R. Polana and R. Nelson, International Journal of Computer Vision (1997).

[25] A. Bobick. and J. Davis, IEEE Transaction on Pattern Analysis and Machine Intelligence (2001).

[26] S. Roweis and L. Saul, SCIENCE (2000).

[27] M. Belkin and P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in *Advances in Neural Information Processing Systems*, 2001.

[28] Z. Liu and S. Sarkar, IEEE Transactions on Pattern Analysis and Machine Intelligence (2006).

[29] N. Cuntoor and R. Chellappa, Epitomic representation of human cctivities, in *Computer Vision and Pattern Recognition*, 2007.

[30] B. North, A. Blake, M. Isard, and J. Rittscher, IEEE Transactions on Pattern Analysis and Machine Intelligence (2000).

[31] R. Young and R. Lesperance, Spatial Vision (2001).

[32] P. Natarajan, V. Singh, and R. Nevatia, Learning 3d action models from a few 2d videos for view invariant action recognition, in *Computer Vision and Pattern Recognition*, 2010.

[33] S. Park and J. Aggarwal, Recognition of two-person interactions using a hierarchical bayesian network, in *ACM SIGMM International Workshop on Video Surveillance*, 2003.

[34] Z. Zeng and J. Qiang, Knowledge based activity recognition with dynamic bayesian network, in *European Conference in Computer Vision*, 2010.

[35] M. Lee and R. Nevatia, IEEE Transactions on Pattern Analysis and Machine Intelligence (2009).

[36] Z. Zhang, K. Huang, and T. Tan, Complex activity representation and recognition by extended stochastic grammar, in *Asian Conference on Computer Vision*, 2006.

[37] Y. Ivanov and A. Bobick, IEEE Transactions on Pattern Analysis and Machine Intelligence (2000).

[38] M. Ryoo and J. Aggarwal, Recognition of composite human activities through context-free grammar based representation, in *Computer Vision and Pattern Recognition*, 2006.

[39] D. Kuettel, M. Breitenstein, L. V. Gool, and V. Ferrari, What's going on? discovering spatio-temporal dependencies in dynamic scenes, in *Computer Vision and Pattern Recognition*, 2010.

[40] G. Medioni, R. Nevatia, and I. Cohen, IEEE Transactions on Pattern Analysis and Machine Intelligence (1998).

[41] L. Ding and A. Yilmaz, Learning relations among movie characters: A social network perspective, in *European Conference on Computer Vision*, 2010.

[42] M. Ryoo and J. Aggarwal, Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities, in *International Conference on Computer Vision*, 2009.

[43] H. W. J.C. Niebles and L. Fei-fei, Unsupervised learning of human action categories using spatial-temporal words, in *British Machine Vision Conference*, 2006.

[44] A. Kovashka and K. Grauman, Learning a hierarchy of discriminative space-time neighborhood features for human action recognition, in *Computer Vision and Pattern Recognition*, 2010.

[45] P. Matikainen, M. Hebert, and R. Sukthankar, Representing pairwise spatial and temporal relations for action recognition, in *European Conference on Computer Vision*, 2010.

[46] S. Park, A hierarchical bayesian network for event recognition of human actions and interactions, in *Association For Computing Machinery Multimedia Systems Journal*, 2004.

[47] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, Visual Surveillance and Performance Evaluation of Tracking and Surveillance (2005).

[48] S. Savarese, A. DelPozo, J. C. Niebles, and L. Fei-Fei, Spatial-temporal correlations for unsupervised action classification, in *IEEE Workshop on Motion and Video Computing*, 2008.

[49] M. S. Ryoo and W. Yu, One video is sufficient? human activity recognition using active video composition, in *IEEE Workshop on Motion and Video Computing*, 2011.

[50] E. Shechtman and M. Irani, Matching local self-similarities across images and videos, in *Computer Vision and Pattern Recognition*, 2007.

[51] H. J. Seo and P. Milanfar, Detection of human actions from a single example, in *International Conference on Computer Vision*, 2009.

[52] N. M. Nayak, Y. Zhu, and A. K. Roy-Chowdhury, Image and Vision Computing (2013).

[53] N. M. Nayak, A. T. Kamal, and A. K. Roy-Chowdhury, Vector field analysis for motion pattern identification in video, in *International Conference on Image Processing*, 2011.

[54] N. M. Nayak, B. Song, and A. K. Roy-Chowdhury, Dynamic modeling of streaklines for motion pattern analysis in video, in *MLvMA Workshop at Computer Vision and Pattern Recognition*, 2011.

[55] N. M. Nayak, Y. Zhu, and A. K. Roy-Chowdhury, IEEE Transactions on Information Forensics and Security (2013).

[56] I. Laptev and T. Lindeberg, Space-time interest points, in *International Conference on Computer Vision*, 2003.

[57] Y. Ke, R. Sukthankar, and M. Hebert, Efficient visual event detection using volumetric features, in *International Conference on Computer Vision*, 2005.

[58] R. Mehran, B. Moore, and M. Shah, A streakline representation of flow in crowded scenes, in *European Conference on Computer Vision*, 2010.

[59] A. Bobick and J. Davis, IEEE Transactions on Pattern Analysis and Machine Intelligence (2001).

[60] Y. Zhu, N. M. Nayak, U. Gaur, B. Song, and A. Roy-Chowdhury, Computer Vision and Image Understanding (2013).

[61] M. Hu, S. Ali, and M. Shah, Detecting global motion patterns in complex videos, in *International Conference on Pattern Recognition*, 2008.

[62] P. Ghosh, L. Bertelli, B. Sumengen, and B. Manjunath, IEEE Transactions on Image Processing (2010).

[63] H. Theisel and T. Weinkauf, Vector field metrics based on distance measures of first order critical points, in *International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2002.

[64] I. Dryden and K. Mardia, *Statistical Shape Analysis*, John Wiley and Sons, 1998.

[65] G. H. Golub and C. V. Loan, *Matrix Computations*, The Johns Hopkins University Press, 1996.

[66] Y. Zhang, X. Liu, M.-C. Chang, W. Ge, and T. Chen, Spation-temporal phrases for activity recognition, in *European Conference on Computer Vision*, 2012.

[67] W. Brendel and S. Todorovic, Learning spatiotemporal graphs of human activities, in *International Conference on Computer Vision*, 2011.

[68] K. Tang, L. Fei-Fei, and D. Koller, Learning latent temporal structure for complex event detection, in *Computer Vision and Pattern Recognition*, 2012.

[69] J. Varadarajan, R. Emonet, and J. Odobez, Bridging the past, present and future: Modeling scene activities from event relationships and global rules, in *Computer Vision and Pattern Recognition*, 2012.

[70] A. Gupta, P. Srinivasan, J. Shi, and L. Davis, Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos, in *Computer Vision and Pattern Recognition*, 2009.

[71] W. Choi, K. Shahid, and S. Savarese, Learning context for collective activity recognition, in *Computer Vision and Pattern Recognition*, 2011.

[72] C. C. Loy, T. Xiang, and S. Gong, International Journal of Computer Vision (2010).

[73] T. Lan, L. Sigal, and G. Mori, Social roles in hierarchical models for human activity recognition, in *Computer Vision and Pattern Recognition*, 2012.

[74] Y. Li and R. Nevatia, Key object driven multi-category object recognition, localization and tracking using spatio-temporal context, in *European Conference on Computer Vision*, 2008.

[75] M. Siegel, Journal of Computational Neuroscience (2000).

[76] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, Neurocomputing **74** (2011).

[77] Y. Song, Mcmc-based scene segmentation method using structure of video, in *International Symposium on Communications and Information Technologies*, 2010.

[78] M. Hoai, Z.-Z. Lan, and F. D. Torre, Joint segmentation and classification of human actions in video, in *Computer Vision and Pattern Recognition*, 2011.

[79] C.-Y. Chen and K. Grauman, Efficient activity detection with max-subgraph search, in *Computer Vision and Pattern Recognition*, 2012.

[80] V. K. Singh and R. Nevatia, Visual Computer (2011).

[81] W. Choi and S. Savarese, A unified framework for multi-target tracking and collective activity recognition, in *European Conference on Computer Vision*, 2012.

[82] L. Zhang, Y. Li, and R. Nevatia, Global data association for multi-object tracking using network flows, in *Computer Vision and Pattern Recognition*, 2008.

[83] V. I. Morariu and L. S. Davis, Multi-agent event recognition in structured scenarios, in *Computer Vision and Pattern Recognition*, 2011.

[84] R. Salakhutdinov and G. Hinton, A betterway to pretrain deep boltzmann machines, in *Neural Information Processing Systems*, 2012.

[85] Q. V. Le et al., Building high-level features using large scale unsupervised learning, in *International Conference on Machine Learning*, 2012.

[86] S. Ji, W. Xu, M. Yang, and K. Yu, IEEE Transactions on Pattern Analysis and Machine Intelligence (2012).

[87] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis, in *Computer Vision and Pattern Recognition*, 2011.

[88] Z. J. Q. Qiu and R. Chellappa, Sparse dictionary-based representation and recognition of action attributes, in *International Conference on Computer Vision*, 2011.

[89] B. Zhao, L. Fei-Fei, and E. Xing, Online detection of unusual events in videos via dynamic sparse coding, in *Computer Vision and Pattern Recognition*, 2011.

[90] J. Zheng and Z. Jiang, Learning view-invariant sparse representations for cross-view action recognition, in *International Conference on Computer Vision*, 2013.

[91] J. Luo, W. Wang, and H. Qi, Group sparsity and geometry constrained dictionary learning for action recognition from depth maps, in *International Conference on Computer Vision*, 2013.

[92] N. M. Nayak and A. K. Roy-Chowdhury, Learning a sparse dictionary of video structure for activity modeling, in *International Conference on Image Processing*, 2014.

[93] M. Schmidt, *Graphical Model Structure Learningwith l1-Regularization*, PhD thesis, THE UNIVERSITY OF BRITISH COLUMBIA, Vancouver, 2010.

[94] V. Chandrasekaran, N. Srebro, and P. Harsha, Complexity of inference in graphical models, in *Proceedings of the Twenty-Fourth Conference Annual Conference on Uncertainty in Artificial Intelligence*, 2010.

[95] C. Desai, D. Ramanan, and C. Fowlkes, International Journal of Computer Vision (2011).

[96] Y. J. Lee and K. Grauman, Object graphs for context aware category discovery, in *Computer Vision and Pattern Recognition*, 2011.

[97] N. Zhou, Y. Shen, J. Peng, and J. Fan, Learning inter-related visual dictionary for object recognition, in *Computer Vision and Pattern Recognition*, 2012.

[98] M. Ryoo and J. Aggarwal, Stochastic representation and recognition of high-level group activities: Describing structural uncertainties in human activities, in *Computer Vision and Pattern Recognition Workshops*, 2009.

[99] C. Cedras and M. Shah, Image and Vision Computing (1995).

[100] D. Tran and E. Sorokin, Human activity recognition with metric learning, 2008.

[101] R. David et al., Advanced Video and Signal-Based Surveillance (2010).

[102] N. Oliver, B. Rosario, and A. Pentland, IEEE Transactions on Pattern Analysis and Machine Intelligence (2000).

[103] R. Li and R. Chellappa, Group motion segmentation using a spatio-temporal driving force model, in *Computer Vision and Pattern Recognition*, 2010.

[104] G. H. R. Chaudhry, A. Ravichandran and R. Vidal, Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions, in *Computer Vision and Pattern Recognition*, 2009.

[105] U. Gaur, B. Song, and A. Roy-Chowdhury, Query-based retrieval of complex activities using strings of motion-words, in *IEEE Workshop on Motion and Video Computing*, 2009.

[106] R. Sethi, A. Roy-Chowdhury, and S. Ali, Activity recognition by integrating the physics of motion with a neuromorphic model of perception, in *IEEE Workshop on Motion and Video Computing*, 2009.

[107] Y. Tong, S. Lombeyda, A. Hirani, and M. Desbrun, ACM Transaction on Graphics (2003).

[108] R. Sethi and A. Roy-Chowdhury, ACM International Workshop on Multimodal Pervasive Video Analysis (2010).

[109] P. Anderson, *Nonverbal Communication: Forms and Functions*, Second edition, 2008.

[110] B. Song, T. Jeng, E. Staudt, and A. R. Chowdhury, A stochastic graph evolution framework for robust multi-target tracking, in *European Conference on Computer Vision*, 2010.

[111] G. Peyre, Toolbox differencial calculus - a toolbox to handle differential operators, 2008.

[112] N. Shroff, P. Turaga, and R. Chellappa, Moving vistas: Exploiting motion for describing scenes, in *Computer Vision and Pattern Recognition*, 2010.

[113] S. Ali, A. Basharat, and M. Shah, Chaotic invariants for human action recognition, in *International Conference of Computer Vision*, 2007.

[114] F. Bashir, A. Khokhar, and D.Schonfeld, IEEE Transaction in Image Processing (2005).

[115] C. Garth, X. Tricoche, and G. Scheuermann, Tracking of vector field singularities in unstructured 3d time-dependent datasets, in *IEEE Conference on Visualization*, 2004.

[116] N. Vaswani, A. Roy-Chowdhury, and R. Chellappa, Activity recognition using the dynamics of the configuration of interacting objects, in *Computer Vision and Pattern Recognition*, 2003.

[117] A. Veeraraghavan, A. K. Roy-chowdhury, and R. Chellappa, IEEE Transactions on Pattern Analysis and Machine Intelligence (2005).

[118] Y. M. A. Bissacco, A. Chiuso and S. Soatto, Computer Vision and Pattern Recognition (2005).

[119] A. K. R.-c. A. Veeraraghavan and R. Chellappa, Internation Conference on Pattern Recognition (2005).

[120] K. Cock and B. Moor, Systems and Control Letters (2002).

[121] R. Martin, IEEE Transactions on Signal Processing (2000).

[122] G. Granlund and H. Knutsson, *Signal processing for computer vision*, Kluwer, 1995.

[123] P. Viola and M. Jones, Computer Vision and Pattern Recognition (2001).

[124] A. Kale et al., IEEE Transactions on Image Processing (2004).

[125] S. Joo and R. Chellappa, Computer Vision and Pattern Recognition Workshop (2006).

[126] M. Leordeanu and M. Hebert, A spectral technique for correspondence problems using pairwise constraints, in *International Conference of Computer Vision*, 2005.

[127] H. Sakoe and S. Chiba, IEEE Transactions on Acoustics Speech and Signal Processing (1978).

[128] F. Jiang, J. Yuan, S. Tsaftaris, and A. Katsaggelos, Computer Vision and Image Understanding (2011).

[129] I. Wersborg, T. Bautze, F. Born, and K. Diepold, A cognitive approach for a robotic welding system that can learn how to weld from acoustic data, in *Computational Intelligence in Robotics and Automation*, 2009.

[130] Y. Benezeth, P. Jodoin, V. Saligrama, and C. Rosenberger, Computer Vision and Pattern Recognition (2009).

[131] D. Makris and T. Ellis, IEEE Transactions on Systems, Man and Cybernetics (2005).

[132] H. Liu, R. Feris, V. Krueger, and M. Sun, EURASIP Journal on Image and Video Processing (2010).

[133] N. Vaswani, A. Roy-Chowdhury, and R. Chellappa, IEEE Transactions on Image Processing (2005).

[134] U. Gaur, Complex activity recognition using string of feature graphs, Master's thesis, University of California, Riverside, CA, USA, 2010.

[135] G. Willems, T. Tuytelaars, and L. Gool, An efficient dense and scale-invariant spatio-temporal interest point detector, in *European Conference on Computer Vision*, 2008.

[136] K. Mikolajczyk and C. Schmid, IEEE Transactions on Pattern Analysis and Machine Intelligence (2005).

[137] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto, International Journal of Computer Vison (2003).

[138] A. Bosch, A. Zisserman, and X. Munoz, Image classification using random forests and ferns, in *International Conference on Computer Vision*, 2007.

[139] N. Dalal and B. Triggs, Histogram of oriented gradients for fast human detection, in *Computer Vision and Pattern Recognition*, 2005.

[140] A. Gupta, J. Shi, and L. Davis, A shape aware model for semi-supervised learning of objects and its context, in *Neural Information Processing Systems*, 2008.

[141] B. Yao and L. Fei-Fei, Modeling mutual context of object and human pose in human-object interaction activities, in *Computer Vision and Pattern Recognition*, 2010.

[142] A. Gupta, A. Kembhavi, and L. Davis, Pattern Analysis and Machine Intelligence (2009).

[143] A. Oliva and A. Torralba, The role of context in object recognition, in *Trends in Cognitive Science*, 2007.

[144] D. Ramanan, Learning to parse images of articulated objects, in *Neural Information Processing Systems*, 2006.

[145] J. K. Aggarwal and M. S. Ryoo, ACM Computing Surveys (2011).

[146] T. Lan, Y. Wang, W. Yang, S. N. Robinovitch, and G. Mori, Pattern Analysis and Machine Intelligence (2011).

[147] A. Prest, C. Schmid, and V. Ferrari, Weakly supervised learning of interactions between humans and objects, Technical report, INRIA, 2010.

[148] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Morgan Kaufmann Publishers Inc., 1988.

[149] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky, Nonparametric belief propagation, in *Computer Vision and Pattern Recognition*, 2003.

[150] R. Benmokhtar and I. Laptev, Inria-willowat trecvid 2010: Surveillance event detection, in *TREC Video Retrieval Evaluation*, 2010.

[151] S. Vijayanarasimhan and K. Grauman, Active frame selection for label propagation in videos, in *European Conference on Computer Vision*, 2012.

[152] R.-X. Gao, T.-F. Wu, S.-C. Zhu, and N. Sang, Bayesian inference for layer representation with mixed markov random field, in *International Conference on Energy minimization methods in computer vision and pattern recognition*, 2007.

[153] N. Plath, M. Toussaint, and S. Nakajima, Multi-class image segmentation using conditional random fields and global classification, in *International Conference on Machine Learning*, 2009.

[154] D. Gong, G. Medioni, S. Zhu, and X. Zhao, Kernelized temporal cut for online temporal segmentation and recognition, in *European Conference on Computer Vision*, 2010.

[155] Y. Zhu, N. Nayak, and A. Roy-Chowdhury, IEEE Journal on Selected Topics in Signal Processing, Special Issue on Anomalous Pattern Discovery (2013).

[156] A. Coates and A. Y. Ng, The importance of encoding versus training with sparse coding and vector quantization, in *International Conference on Machine Learning*, 2011.

[157] M. Gonen and E. Alpaydin, Journal of Machine Learning Research (2011).

[158] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, Sequential Deep Learning for Human Action Recognition, in *2nd International Workshop on Human Behavior Understanding*, 2011.

[159] X. Zhang, H. Zhang, and X. Cao, Action recognition based on spatial-temporal pyramid sparse coding, in *International Conference on Pattern Recognition*, 2012.

[160] G. Taylor, R. Fergus, Y. LeCun, and C. Bregler, Convolutional learning of spatio-temporal features, in *European Conference on Computer Vision*, 2010.

[161] H. Lee, Tutorial on deep learning and applications, in *NIPS 2010 Workshop on Deep Learning and Unsupervised Feature Learning*, 2010.

[162] J. Yangy, K. Yuz, Y. Gongz, and T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in *Computer Vision and Pattern Recognition*, 2009.

[163] C.-C. Chang and C.-J. Lin, ACM Transactions on Intelligent Systems and Technology **2** (2011).

[164] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, Discriminatively trained deformable part models, Release 4, http://people.cs.uchicago.edu/ pff/latent-release4/.

[165] N. Dalal and B. Triggs, Histograms of oriented gradients for human detection, in *CVPR*, 2005.

[166] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2nd edition, 2006.

[167] C. Desai, D. Ramanan, and C. C. Fowlkes, Discriminative models for multi-class object layout, in *International Journal of Computer Vision*, 2011.

[168] C. H. Teo, Q. Le, A. Smola, and S. V. N. Vishwanathan, A scalable modular convex solver for regularized risk minimization, in *SIGKDD*, pages 727–736, 2007.