UNIVERSITY OF CALIFORNIA
RIVERSIDE

Towards Sparse Modeling of Multi-Object Interactions in Video

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

Doctor of Philosophy

in

Electrical Engineering

by

Yingying Zhu

December 2014

Dissertation Committee:

Dr. Amit K. Roy-Chowdhury, Chairperson
Dr. Ertem Tuncel
Dr. Wei Ren

The Dissertation of Yingying Zhu is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

Foremost, I would like to thank my supervisor Dr. Amit K. Roy-Chowdhury for sharing his knowledge, giving me the freedom to take a diverse set of courses, constantly trying to improve my research work and pushing me to work hard. I would like to thank my other dissertation committee members Dr. Wei Ren, and Dr. Ertem Tuncel for their help and advice during my pursuit of a Ph.D. degree at UCR. I am deeply grateful for the time and energy you have spent on the dissemination of knowledge.

I would also like to thank all the people who helped me during my studies at UCR, including Dr. Stefano Lonardi, Dr. Eamonn Keogh, and Dr. Anastasios Mourikis, Dr. Mathew Barth and all the members in the Video Computing Group. I would like to thank Dr. Bi Song, Shu Zhang, Nandita Nayak, Elliot Staudt, Ramya Malur Srinivasan, Mahmudul Hasan, Dr. Ahmed Tashrif Kamal and Dr. Chong Ding for valuable discussions and cooperation in research.

I would like to extend my thanks to the Bourns College of Engineering, University of California, Riverside, for their generous support. I would like to thank the National Science Foundation and for their grant NSF grant (IIS-0712253 and IIS-1316934), ONR grant (N00014-12-1-1026) and Mayachitra Inc. for their DARPA STTR award (W31P4Q-11-C0042), to Dr. Amit K. Roy-Chowdhury, which partially supported my research.

Finally and most importantly, I would like to thank my mother and father for giving me life. I would like to thank my mother for taking care of me and my younger sister so well. I would like to thank my husband for never giving up on me. You all give me the reason to live a strong life.

Acknowledgment of previously published, accepted or submitted materials: The text of this dissertation, in part or in full, is a reprint of the material as appears in previously published, accepted and submitted papers which are listed below. The co-author Dr. Amit K.

Roy-Chowdhury, directed and supervised the research which forms the basis for this dissertation.

1. Y. Zhu, N. Nayak, U. Gaur, B. Song, A. Roy-Chowdhury, Modeling Multi-object Interactions using String of Feature Graphs. In Computer Vision and Image Understanding (CVIU), 2013.

2. Y. Zhu, N. Nayak, A. Roy-Chowdhury, Context-Aware Activity Recognition and Anomaly Detection in Video, In IEEE Journal of Selected Topics in Signal Processing (J-STSP), 2013.

3. Y. Zhu, N. Nayak, A. Roy-Chowdhury, Context-Aware Activity Modeling using Hierarchical Conditional Random Fields, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), DOI 10.1109/TPAMI.2014.2369044.

4. Y. Zhu, A. K. Roy-Chowdhury, Automatic Discovery of Sparse Contextual Relationships for Context-Aware Visual Recognition, International Journal of Computer Vision (IJCV), 2014, (submitted).

5. U. Gaur, Y. Zhu, B. Song, and A. Roy-Chowdhury, String of feature graph analysis of complex activities, In International Conference on Computer Vision (ICCV), 2011.

6. Y. Zhu, N. M. Nayak, A. Roy-Chowdhury, Context-Aware Modeling and Recognition of Activities in Video, In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2013.

7. Y. Zhu, A. Roy-Chowdhury, Graphical Models for Context-Aware Analysis of Continuous Videos. GLOBALSIP, 2013.

ABSTRACT OF THE DISSERTATION

Towards Sparse Modeling of Multi-Object Interactions in Video

by

Yingying Zhu

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

In this dissertation, we develop intelligent methodologies for the modeling and recognition of activities in continuous videos. Videos usually consist of activities involving interactions between multiple actors. Recognition of such activities requires modeling the spatio-temporal relationships between the actors and their individual variabilities. We propose the generalized framework of "String-of-Feature-Graph" that implicitly quantifies the spatial and temporal relationships between interacting objects through modeling spatio-temporal relationships between local motion features. Furthermore, activities related in space and time rarely occur independently and can serve as the context for each other. Thus, rather than modeling only feature-level context, we also implicitly or explicitly model the contextual relationships between activities. Specifically, we utilize probabilistic graphical models, in a max-margin framework, to jointly model and recognize related activities in space and time using motion and various context features within and between actions and activities. We call these models are context-aware graphical models.

When such models are discriminatively trained, redundant features that are highly correlated with each other are usually used. Sparse features are likely to be preferred in such situations because: 1) when model features are sparse, it would be more efficient and effective to estimate the parameters; 2) intrinsic and contextual attributes as well as association rules of inter-dependent objects are usually sparse. Thus, we develop a sparse modeling framework, building upon the proposed context-aware graphical models and group $l1$-regularization, to enhance the efficiency and accuracy for activity recognition. The proposed framework is general enough to work for the recognition of any type of inter-dependent visual objects, such as visual activities and image objects.

In real activity recognition applications, not all types of activities are known to us or exist in the training examples. Approaches that aim to detect the abnormal activities which are different from the known or training activities in certain aspect are in need. As an extension of the proposed context-aware models for activity recognition, we further work on the detection of anomalous activities. We define three types of anomalous activities with abnormal motion and/or context behaviors. With the learned context-aware graphical model from normal activities, we utilize statistical inference methods for the detection of anomalous activities whose motion and context patterns deviate from the learned patterns. Our studies advance computer vision through demonstrated benefits of using the proposed approaches over the state-of-the-art works.

Contents

List of Figures xi

List of Tables xvii

1 Introduction 1
- 1.1 Related Works 4
  - 1.1.1 Recognition of Complex Activity 4
  - 1.1.2 Exploring Context for Activity Recognition 5
  - 1.1.3 Sparse Modeling for Visual Recognition 8
  - 1.1.4 Anomalous Activity Recognition 10
- 1.2 Main Contribution 11
- 1.3 Outline 12

2 Feature-level Contextual Modeling - SFG 14
- 2.1 Introduction 14
  - 2.1.1 Overview of Proposed Framework 15
    - 2.1.1.1 Feature Descriptor 15
    - 2.1.1.2 Adaptive Feature Selection 17
  - 2.1.2 Contributions of Present Work 19
- 2.2 String of Feature Graph 20
  - 2.2.1 Model Development 20
  - 2.2.2 Spatio-temporal Matching of SFGs 21
    - 2.2.2.1 Matching Two Feature-Graphs 21
  - 2.2.3 Dynamic Time Warping of SFGs 23
    - 2.2.3.1 Subsequence DTW for Continuous Video 24
- 2.3 Special Cases 25
  - 2.3.1 Track-Based SFG 26
    - 2.3.1.1 Track-Based Feature Graph Matching 29
  - 2.3.2 STIP-Based SFG 30
    - 2.3.2.1 STIP-Based Feature Graph Matching 31
- 2.4 Adaptive Feature Selection 31
  - 2.4.1 Switched Dynamic System Model 32
  - 2.4.2 Switching Between Activity Modes 33
- 2.5 Experiments 34
  - 2.5.1 Preprocessing 35
  - 2.5.2 STIP-based SFG Results 37

2.5.2.1 Classification-Based Recognition Results 38
2.5.2.2 Query-Based Retrieval Results 40
2.5.3 Track-Based SFG Results 42
2.5.3.1 Performance Comparison 42
2.5.3.2 Classification-Based results 44
2.5.3.3 Query-Based results 44
2.5.4 Adaptive Feature Selection 46
2.6 Conclusion 50

3 Higher-Level Context Modeling - graphical models 52
3.1 Introduction 52
3.1.1 Overview of the Framework 53
3.1.2 Contributions of Present Work 55
3.2 Struct-SVM 56
3.2.1 Feature Descriptors 56
3.2.2 Model Development 59
3.2.3 Model Learning and Inference 61
3.2.3.1 Learning Model Parameters 61
3.2.3.2 Inference 63
3.3 Structural Model 64
3.3.1 Feature Descriptors 65
3.3.2 Model Development 66
3.3.3 Model Learning and Inference 67
3.3.3.1 Learning Model Parameters 67
3.3.3.2 Inference 69
3.4 Hierarchical-CRF 69
3.4.1 Feature Descriptors 72
3.4.2 Model Development 75
3.4.2.1 Action-Based Linear-Chain CRF 77
3.4.2.2 Incorporating Higher Order Potentials 77
3.4.2.3 Incorporating Inter-Activity Potentials 79
3.4.3 Model Learning and Inference 79
3.4.3.1 Learning Model Parameters 80
3.4.3.2 Inference 84
3.5 Computational Analysis 84
3.6 Experiments 86
3.6.1 Datasets 86
3.6.1.1 UCLA Dataset 86
3.6.1.2 VIRAT Ground Dataset 88
3.6.2 Preprocessing 90
3.6.3 Structural Model 91
3.6.3.1 Recognition Results on VIRAT Release 1 92
3.6.3.2 Recognition Results on VIRAT Release 2 93
3.6.4 Hierarchical-CRF 95
3.6.4.1 Recognition Results on UCLA Dataset 96
3.6.5 Recognition Results on VIRAT Release 1 98
3.6.6 Recognition Results on VIRAT Release 2 99
3.7 Conclusion 101

4 Learning Sparse Graphical Representation 103
4.1 Introduction . . . 103
4.1.1 Sparse Modeling Approach . . . 105
4.1.2 Contributions of The Present Work . . . 106
4.2 Formulation of Sparse Models . . . 107
4.2.1 Model Features . . . 107
4.2.2 The Context-Aware CRF Model . . . 108
4.2.3 Sparse Modeling . . . 111
4.3 Sparse Model Learning and Inference . . . 113
4.3.1 Optimization Algorithm . . . 114
4.3.1.1 Iterative Soft-Thresholding Sub-Routine . . . 114
4.3.2 Speeding Up the Optimization Procedure . . . 117
4.3.3 Model Inference with Top-K Greedy Search . . . 118
4.4 Experiments . . . 119
4.4.1 Activity Recognition . . . 120
4.4.1.1 UCLA Dataset . . . 120
4.4.1.2 Preprocessing . . . 120
4.4.1.3 Experimental Results . . . 121
4.4.2 Object Recognition . . . 125
4.4.2.1 PASCAL VOC Dataset . . . 126
4.4.2.2 Preprocessing . . . 126
4.4.2.3 Experimental Results . . . 128
4.5 Conclusion . . . 129

5 Anomalous Activity Detection 131
5.1 Introduction . . . 131
5.1.1 Overview of the Framework . . . 132
5.1.2 Contributions of The Present Work . . . 133
5.2 The Algorithm . . . 134
5.2.1 Anomaly Definitions . . . 136
5.2.1.1 Point Anomaly . . . 136
5.2.1.2 Contextual Anomaly . . . 138
5.2.1.3 Collective Anomaly . . . 138
5.3 Experiments . . . 140
5.3.1 Preprocessing . . . 140
5.3.2 Results . . . 142
5.3.2.1 Point Anomaly . . . 142
5.3.2.2 Contextual Anomaly . . . 143
5.3.2.3 Collective anomaly . . . 144
5.4 Conclusion . . . 146

6 Discussion 148

Bibliography 150

A Publication List 160

List of Figures

2.1 Representative frames of the datasets used in this work. Note that the videos contain multiple actors performing activities simultaneously, sometimes in the presence of irrelevant subjects. 15

2.2 Activity modeling: Local features are computed from the video and grouped together to form feature collections. Temporally ordered strings of these local feature collections is termed as “string of feature-graphs” (SFGs). 17

2.3 (Left) Local feature-graphs are matched using the graph-based spectral technique in Section 3.1. (Right) The feature-graph matching scores thus generated are used in DTW matching of the two videos. which are represented by strings of feature graphs, to account for difference in speed and execution. 18

2.4 Representative frames of global and local activities recognized in this work. The first frame shows a person approaching a vehicle (global activity). Low resolution motion features are suitable for recognition of such actions. The second frame shows a person loading a trunk. This is a local activity which can be recognized by examining high resolution motion features in the region marked in red (showed in the third frame). 19

2.5 Examples of NDG and NMG descriptors. The left column shows the sample images for a vehicle-backup (top) and a vehicle-u-turn (bottom) (only regions of interest are shown). The next two columns show the corresponding NDG and NMG descriptors. 28

2.6 Example of RD and SRD of two tracks. The images show sample frames of two people walking together (top) and person leaving a vehicle (bottom) (only regions of interest are shown). The graph on the left shows the raw relative distance between the two tracks and the exponential fitting result in each case. The graph on the right shows the derivative of smoothed relative distance (SRD) in each case. 29

2.7 Overall flow of the activity recognition system using adaptive feature selection in the SFG framework. 34

2.8 Estimated conditional probability distributions of the four motion attributes given the activity mode 37

2.9 Recognition accuracy on the UT-Interaction dataset by using voting scheme on top of SFG model. 39

2.10 The recognition accuracy of our method with respect to number of query examples. It can be seen that when number of query example decreases, the performance of our method does not drop significantly. 39

2.11 (a): Confusion matrix for VIRAT dataset. Nearest neighbor classifier and eight-leave-one-out cross-verification are used. (1:person loading an object to a vehicle, 2:person uploading 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). Most misclassifications are inside activity group (1, 2, 3, and 4) and activity group (5 and 6). (b): ROC of 7-NN classifier on VIRAT dataset (Bag of feature: Laptev et al. [100]+7NN). 40

2.12 (a): Recognition accuracy and false positives on 9 activities from the UCR VideoWeb dataset in a query-based retrieval framework. Standard deviation in performance (accuracy) for different queries is marked on the bars. (b): Retrieval results: The left column depicts the query videos and the other three columns are the best matches on UCR VideoWeb dataset. The bounding boxes of the sub-graphs that best match the feature graphs of the query video are shown. A blue dash box represents an incorrect match. 43

2.13 Retrieval results: The left column depicts the query videos and the other three columns are the best matches on UT-Interaction dataset. 44

2.14 (a): Average performance of different kinds of motion descriptors on recognizing activities characterized by individual tracks. 7 car-backup, 16 car-turn, and 2 car-uturn. (Parameter of ROC: distance threshold, the same threshold is used for all kinds of activities, training can be applied to find the best distance threshold for each kind of activity). (b): Average performance of track-based SFG system on VIRAT Testing Dataset. For each run, only one training instance is used for each kind of activity, the rest are treated as testing instances. While the algorithm achieves high recognition performance, the object identifer further enhance the performance. Joint NDG-NMG and joint RD-SRD descriptors are used. . . 45

2.15 (a): Average performance of track-based SFG system on VIRAT Testing Dataset with object identifier. Only one query instance is used for each query. (b): Examples of query results on VIRAT testing Dataset. The left column depicts the query tracks involved in the targeted activity and the other three columns are the best matches on part of VIRAT testing dataset. 46

2.16 Switching results on an example video sequence are shown. One sample image for each activity is shown. Each cyan bar in the figure indicates the recognition result from the adaptive feature selection and compares it to the ground truth (blue bar). The length of the bar indicates the duration of the recognized activity. Red bounding box indicates track features are selected, and purple indicates STIP features are selected for the SFG model in adaptive feature selection. The results show that the system is able to automatically switch between different features. 48

3.1 (i) Example images from a wide area video (interesting activities happening within about 2 minutes are shown). Activities in the same color in each video happen in the same local spatio-temporal region. Activity classes are listed in Fig. 1 in the supplementary material. (a). For the indices, the first index denotes the temporal order of the activity in the region, while the second number denotes the activity class, e.g., 1-6 means the activity belongs to class 6 and is the first activity that happens in this video volume. (ii) Example of context in activity recognition. A person of interest is located by red bounding box, surrounding objects are located by bounding boxes of other colors, and the circles in purple indicate the motion regions of the activities. 53

3.2 The image shows one frame of 'person unloading an object from a vehicle'. In the image, moving objects are the person and the vehicle, and the person is in the rear of the vehicle. So, for this frame, $g_1(n) = [1 \quad 0 \quad 0]$ and $g_2(n) = [0 \quad 1 \quad 0]$, where n is the frame number of this image in the activity. 58

3.3 (a) The image shows an example of inter-activity spatial relationship. The red circle indicates the motion region of a_i at this frame while the purple rectangle indicates the activity region of a_j. Assume SC is defined by quantizing and grouping $r_s(n)$ into three bins: $r_s(n) \leq 0.5$ (a_i and a_j is at the same spatial position at the n^{th} frame of a_i), $0.5 < r_s(n) < 1.5$ (a_i is near a_j at the n^{th} frame of a_i), and $r_s(n) \geq 1.5$ (a_i is far away from a_j at the n^{th} frame of a_i). In the image, $r_s(n) > 1.5$, so, $sc_{ij}(n) = [0 \quad 0 \quad 1]$. (b) The image shows one example of inter-activity temporal relationship. The n^{th} frame of a_i occurs before a_j. So, $tc_{ij}(n) = [1 \quad 0 \quad 0]$. 59

3.4 The left graph shows the video representation of an activity set with n motion segments and m activities (for testing, m needs to be determined by inference of the learned structural model). The right graph shows the graphical representation of our model. The gray nodes in the graph are the feature observations and the white nodes are the model variables. The dashed lines indicate that the connections between activity labels and the observations of action segments are not fixed, i.e., the structure of connections is different for different activity sets. 65

3.5 The left graph shows the video representation of an activity set with n motion segments and m candidate activities. The right graph shows the graphical representation of our Hierarchical-CRF model. The white nodes are the action variables and the gray nodes in the graph are the hidden activity variables. Note that observations associated with the model variables are not shown for clear representation. 71

3.6 12Illustration of CRF models for activity recognition. (a): Action-based Linear-Chain CRF; (b): Action-based higher-order CRF model (with latent activity variables); (c): Action-based two-layer HCRF. Note that all the observations for the random variables are omitted for compactness. One action segment denotes a random variable in the action layer, whose value is the activity label for the action segment. A colored circle denotes a random variable in the activity layer, whose value is the label for its connected clique. As shown in (a), in the action layer, action segments that belong to the same trajectory are modeled as a linear-chain CRF. Then, hidden activity-level variables with action-activity edges (in light blue) are added for each action cliques to form higher-order CRF as shown in (b). An activity nodes and its associated action nodes have a same color. Finally, pair-wise activity edges (in red) are added to form the proposed two-layer HCRF model. 76

3.7 Detected objects of interest in the UCLA office scene. 87

3.8 Subsets of context attributes used for the development of intra-activity context features for UCLA Dataset (the superscripts indicates the correspondence between the subsets and the objects). 88

3.9 Examples of agent-object interactions detected from image. 88

3.10 Subsets of context attributes used for the development of intra-activity context features. 89

3.11 Examples of detected intra-activity context features. The example images are shown with detected high-level image features. Object in red bounding box is a moving person; object in blue bounding box is a static vehicle; object in orange bounding box is a moving object of other kind; object in black bounding box is a bag/box on the agent. 90

3.12 Recognition Results for VIRAT Release 1. (a): Confusion matrix for the baseline classifier; (b): Confusion matrix for our approach using motion and intra-activity context features; (c): (b): Confusion matrix for our approach using motion and intra- and inter- activity context features. 92

3.13 Precision and recall (in parenthesis) for the six activities defined in VIRAT Release 1. Baseline: NDM+SVM; Our method (1): the proposed structural model with motion feature and intra-activity context feature; our method (2): the proposed structural model with motion feature, intra-activity and inter-activity context features. Note that SVM+BOW works on video clips; while other methods work on continuous videos. 93
3.14 Example activities (from VIRAT Release 1) correctly recognized by baseline classifier (top), incorrectly by baseline classifier but corrected using intra-activity context (middle), and incorrectly recognized by baseline classifier and intra-activity context, but rectified using inter-activity context (bottom). 94
3.15 Precision and recall (in parenthesis) for different methods (averaged across activities). 94
3.16 Precision (a) and recall (b) for the eleven activities defined in VIRAT Release 2. 95
3.17 Examples (from VIRAT Release 2) in the bottom row show the effect of context features in correctly recognizing activities that were incorrectly recognized by the baseline classifier, while other examples of the same activities correctly recognized by the baseline classifier are shown in the top row. 95
3.18 Precision (a) and recall (b) for the ten activities in UCLA Office Dataset. The activities are defined in Section 3.6.1.1. HCRF is the short of Hierarchical-CRF. 97
3.19 Overall and average per-class accuracy for different methods on UCLA Office Dataset. The BOW+SVM method is tested on video clips, while other results are in the framework of our proposed action-based CRF models upon automatically detected action segments. HCRF is the short of Hierarchical-CRF. 97
3.20 Precision (a) and recall (b) for the six activities defined in VIRAT Release 1. . . 99
3.21 Example activities (defined in VIRAT Release 1) correctly recognized by action-based linear-chain CRF (top), incorrectly by linear-chain CRF but corrected using higher-order CRF with intra-activity context (middle), and incorrectly recognized by higher-order CRF, but rectified using action-based hierarchical CRF with inter-activity context (bottom). 100
3.22 Average accuracy for the six activities defined in VIRAT Release 1. Note that SVM+BOW works on video clips; while other methods work on continuous videos. Note that BOW+SVM works on video clip while others work on continuous video. 100
3.23 Precision (a) and recall (b) for the eleven activities defined in VIRAT Release 2. 101
3.24 Average accuracy (in recall) for different methods. 101

4.1 Illustration on the proposed sparse modeling for context-aware visual recognition. Starting with candidate contextual features represented by a dense graph, our goal is to automatically learn an optimum sparse graphical model, with both sparse contextual features and as well as a sparse graphical structure, capturing the informative contextual patterns within and between object classes. Node that the two object classes may be connected by multi-edges, representing different types of relationships. 104
4.2 Sample image of UCLA office scene with objects of interest marked on the image. 121
4.3 Subsets of context attributes used for the development of Intra-CF for UCLA Dataset. “A” denotes the agent. 121
4.4 Precision (top) and recall (bottom) for the ten activities in UCLA Office Dataset. 123
4.5 (a): The learned graphical structure using the proposed GSM. 124

4.6 Sample results of activity localization and labeling. Top: results of using GSM^2; Bottom: results of using $l1$-CRF and $l2$-CRF. The labeling results using $l1$-CRF and $l2$-CRF are the same in this case. Note the different structures of inter-relationships among the detected activities. Red bounding box indicates the mislabeled activity. 125

4.7 Comparison of our approach with the state-of-the-art object detectors on VOC 2007. GSM^2: the proposed group sparse modeling approach with $l2$-norms within all parameter groups. 126

4.8 Learned sparse graph for the context-aware object recognition task. 127

4.9 Example test images. For each column, the top image shows the detection results of $l1$-CRF; the middle one shows the detection results of $l2$-CRF; and the bottom one shows the detection results of our group sparse modeling using l2 norms for within group parameter regularization. In each image, objects with red labels are incorrectly labeled. 128

5.1 An example that demonstrates the importance of context in activity recognition. Motion region surrounding the person of interest is located by red circle, interacting vehicle is located by blue bounding box. 132

5.2 Example of probability density functions of $d_x^a(y^{opt}, j), \quad j = 1,2$ and 3 for normal activities and the corresponding distances of a point anomaly (indicated by red circle). The first ten activities in Fig. 1 in the supplementary material are considered as normal activities, and used to train the structural model. For the point anomaly detected $y^{opt} = 6$. 137

5.3 Example of probability density functions of $d_{g_k}^a(y^{opt}, j)$, $k = 2$, $j = 2,3$ and 4, for normal activities and the corresponding distances of a contextual anomaly (indicated by red circle). The first ten activities in Fig. 1 in the supplementary material are considered as normal activities and used to train the structural model. For the contextual anomaly detected $y^{opt} = 1$. Intra-activity context subset G_2 is defined in Sec. 3.4.1. 139

5.4 (a): Example of probability density functions of $d_{sc}^a(({y_i}^{opt}, {y_j}^{opt}), (i', j'))$ for normal activities and the corresponding distances of a collective spatial anomaly (indicated by red circle). For the detected collective anomaly a_i and a_j ${y_i}^{opt} = 2$, ${y_j}^{opt} = 5$. (b): Example of probability density functions of $d_{tc}^a(({y_i}^{opt}, {y_j}^{opt}), (i', j'))$ for normal activities and the corresponding distances of a collective temporal anomaly (indicated by red circle). For the detected collective anomaly a_i and a_j ${y_i}^{opt} = 5$, ${y_j}^{opt} = 6$ (In both cases, the first ten activities in Fig. 1 in the supplementary material are considered as normal activities and used to train the structural model). 141

5.5 Recognition Results for VIRAT Release 2. (a): Confusion matrix for BOW+SVM baseline classifier; (b): Confusion matrix for our approach using BOW+SVM as the baseline classifier. The activities considered are listed in Fig. 1 in the supplementary material 143

5.6 ROC curves for point anomaly detection 144

5.7 (a, c, e): Examples of normal activities; (b, d, f): Examples of detected contextual anomalies. First row: Person getting into a vehicle usually occurs in the parking area (a), and the anomaly is detected when it happen in an area not for parking (b). Second row: Person exiting a facility happens at a normal exit (c), whereas an anomaly is detected when the person exits from a door that is rarely used (d). Third row: A person gesturing far from a vehicle is normal in our dataset (e), whereas in (f) the 'gesturing' occurs near the trunk of the vehicle, which is identified as a contextual anomaly. 145

5.8 Example of collective spatial anomaly and collective temporal anomalies. (a, b): we consider two activities - 'person getting into a vehicle' and 'person unloading an object from a vehicle'. For most of the examples in the dataset, when these two activities happen together, the unloading happens from the trunk of the car while the person enters the driver's door. Thus, a collective spatial anomaly is detected when the unloading and entering happen near the same part of the vehicle. (c, d): The example of a 'person getting out of a vehicle' usually occurs before 'person getting into a vehicle', however, in the detected anomaly, 'person getting out of a vehicle' occurs after 'person getting into a vehicle'. 146

List of Tables

2.1 Results of Adaptive Feature Selction. (Recognition accuracy on the VIRAT dataset by using fixed types of feature and adaptive feature selection. N/A means that the activity cannot be recognized by the corresponding feature) 49

2.2 Comparison of computation complexity. (Note that the computation time is given as as the approximate percentage of total computation time using STIP features only. Training overhead includes the time used to construct the training SFGs from the labeled and segmented video clips for track-based and stip-based algorithm, plus the attribute space construction time for adaptive feature selection algorithm) 50

4.1 Comparing combinations of different norms used for parameter groups of contextual features. PG is short for parameter group. 122

4.2 Comparison of different methods in recognition accuracy for the UCLA Office dataset. 123

4.3 Comparing combinations of different norms used for parameter groups of contextual features. PG is short for parameter group. 126

Chapter 1

Introduction

Activity recognition, which grants intelligent systems the ability to sense and analyze what the targeted agents are doing, is an essential task in the field of computer vision and pattern recognition. The applications of visual-based activity recognition are broad as in video indexing systems, surveillance systems, human computer interaction systems and intelligent driving-assistant systems, etc.. Along with the development of machine learning techniques as well as the increasing computational power of intelligent systems, there has been a surge of interest in automatic activity recognition in videos. Activity recognition has been widely studied, but most of the literature has concentrated on relatively simple activities as evidenced in the KTH or Wiezmann datasets [120]. Given the early success of Bag-of-Word (BOW) methods for action classification, the community continue to address the problem of structured localization and recognition of complex human activities.

It has been demonstrated in [80] that context is significant in human visual systems. Videos usually consist of activities involving interactions between multiple actors. Thus, in Chapter 2, we propose the generalized framework of "String-of-Feature-Graph" that implicitly quantifies the spatial and temporal relationships between interacting objects through modeling spatio-temporal relationships between local motion features. The framework can be used for both activity-based video retrieval and activity-based video classification. It works without the need for a large number of training examples and allows matching a short query sequence to a large database through sub-graph and sub-sequence matching.

However, SFG approach is time consuming for both activity querying and activity classification, especially when the training dataset is large. In other words, it is not scalable to large video datasets. Furthermore, the SFG approach does not model the higher-level interactions, which are semantically meaningful to humans. In the tasks of activity recognition in continuous videos, instances to be recognized rarely happen independently and the interdependence between different kinds of instances and their surroundings usually follow certain patterns. Thus, jointly modeling and recognizing related visual objects is expected to improve the recognition accuracy [22, 140]. We call such kinds of activity recognition approaches as context-aware pattern recognition.

Graphical models such as And-Or graph, Conditional Random Field (CRF) and Hidden CRF (HCRF) are frequently used for context-aware pattern recognition tasks, such as image object recognition and activity recognition [22, 104, 107, 119, 126, 130, 140]. These context-aware graphical models have been demonstrated to perform well in the recognition of interdependent image objects by introducing mid-level features or object attributes, as well as exploring the inter-relationships between different image objects [22].

However, developing robust graphical models for activity recognition is more challenging. Previous works on context-aware graphical models for activity recognition focus on the modeling of individual group activities or a sequence of simple activities such as actions [104, 107, 119, 126, 130]. Few work address the problem of jointly modeling and recognition of more complex activities, such as multi-object interactions, in continuous videos. In Chapter 3, we develop several context-aware graphical models that model and recognize activities in videos jointly, exploring motion and various context features within and between activities.

The problem of activity recognition in continuous videos requires two main tasks: to detect motion regions and to label these detected motion regions. The detection and labeling problems can be solved simultaneously as proposed in [76] or separately as proposed in [138, 140]. For the latter, candidate action or activity regions are usually detected before the labeling task. Simply using sliding window methods for the spatio-temporal localization of an activity may lead to low accuracy in activity localization and further reduce the recognition accuracy. Instead, we introduce the concept of action segment as the element of activities, which can be easily obtained by using fixed-length temporal window or using more sophisticated motion segmentation algorithms. With these localized action segments, the problem of activity recognition is then converted to a problem of labeling, that is, to assign each action segment with an optimum activity label. Then, graphical models with activity labels of these action segments as the elementary variables are built for the labeling problem.

When these graphical models are discriminatively trained, redundant features that are highly correlated with each other are usually used. Thus sparse features are likely to be preferred in such situations. On the other hand, in order to achieve high activity recognition performance, the accurate recognition of other visual objects, such as image object, is of equal importance to the recognition of activities themselves. In Chapter 4, we formulate a general framework of sparse modeling for the recognition of inter-dependent visual objects, especially image objects and activities, in videos. This sparse modeling approach automatically learns the optimum graphical model with both sparse features as well as a sparse structure for the particular recognition task at hand.

For many visual recognition tasks, including activity recognition/classification, not all instances to be recognized are of known classes or exist in the training dataset. Anomaly detection is a critical issue for such tasks. As an extension of the developed context-aware graphical models, we address the problem of anomalous activity detection in Chapter 5. Different from previous works that utilize only motion features of individual actions/activities, we also take into account the abnormality of context features within activities as well as the inter-relationships between them for the detection of anomalous activities.

1.1 Related Works

In this section, we briefly review the works related to the approaches developed in this dissertation. For a more complete survey in activity recognition, please refer to [121].

1.1.1 Recognition of Complex Activity

Complex activities usually involve multiple persons interacting with each other or with other objects like buildings and vehicles. The literature on complex activity modeling and recognition can be classified into three categories: graphical, syntactic, and logical approaches [74, 120].

Dynamic Bayesian networks (DBNs), which encode complex conditional dependencies between a set of random variables, is a representative graphical model used for complex activities [38]. The inference method on a CRF-based model proposed in [58, 59] searches through the graphical structure, in order to find the one that maximizes the potential function. Though this inference method is computationally less intensive than exhaustive search, it is still time consuming. As an alternative, greedy search has been used for inference in object recognition [22].

Motivated by grammars in language modeling, syntactic approaches specify how activities can be constructed from action primitives, and use these rules as grammars for visual activity recognition [45, 89].

Logic-based methods form logical rules to express common-sense knowledge to describe activities; for example, [116] represented each logical rule as first-order logic formula. All these approaches rely on either tracking body parts [38, 89], or object detection [38, 116], or atomic action/primitive event recognition [45, 116].

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 for creating a large number of semi-artificial training videos from an original activity video was presented in [93]. A self-similarity descriptor that correlates local patches was proposed in [103]. A generalization of [103] was presented in [101], where space-time local steering kernels were used. These methods require a sliding window through time and space.

Switched dynamical systems have been proposed to compute discrete switches between models in environments which exhibit continuous dynamics and discrete model changes. The authors in [85] cast a switched dynamical system as a Dynamical Bayesian network with applications in human motion analysis. A space dependent Markov chain was used to model the switches between models in [73]. A consensus between multiple classifiers for recognition has been proposed in [52]. These approaches utilize a common feature set and the different classifiers are based on a common framework. In the part of our work that deals with adaptive feature selection, we propose switching between models that utilize different features.

1.1.2 Exploring Context for Activity Recognition

Many existing works exploring context focus on interactions among features, objects and actions [2, 130, 42, 108, 124], environmental conditions such as spatial locations of certain activities in the scene [69], and temporal relationships between activities [71, 110]. Spatio-temporal feature based approaches, like [23], hold more promise since no tracking is assumed.

The statistics of these features are then used in recognition schemes [77]. However, as these approaches are built upon the statistics of extracted local features, spatial and long-term temporal correlations are often ignored.

Few approaches have looked at integrating multiple features for activity recognition. Most of these approaches aim at using different features for a hierarchical analysis of activities. The authors in [25] have shown that both low resolution and high resolution features are needed for the understanding of human actions and interactions. An integrated framework for the recognition of objects and activities using multiple features has been demonstrated in [69]. A combination of space-time cuboids and vocabulary of spin images is used for single person activity recognition in [67]. Few approaches like [25] compute features at multiple resolutions and integrate them. Alternatively, choosing between multiple features is another possible approach.

However, Spatio-temporal constraints across activities in a wide-area scene are rarely considered. The work in [13] models the video as a time-string of frame-wide feature histograms. It does bring the temporal aspect into picture; however the spatial structure information gets lost in the histogram representation. In [32], spatio-temporal relationships are considered by modeling activities as "strings of motion words". However, this method is limited to the availability of the tracks of objects involved. A matching kernel using "correlograms" was presented in [98], which looked at the spatio-temporal proximity among features. A recent work [90] proposed a match function to compare spatio-temporal relationships in the features by using temporal and spatial predicates. By considering the statistics of these relationships, the benefits of spatio-temporal modeling were demonstrated. The number of training videos needed to be large enough to represent the dataset.

Graphical models are commonly used to encode relationships in video analysis. In [31], variable length Markov models were used to learn qualitative spatio-temporal relations relevant to object interactions in the scene. A Dynamic Bayesian network was used to model the temporal evolution in two person activities in [84]. A grid based belief propagation method was used for pose estimation in [63]. Stochastic and context free grammars have been used to model complex activities in [89]. Co-occurring activities and their dependencies have been studied using Dependent Dirichlet Process - Hidden Markov Models (DDP-HMMs) in [55]. Graphical models usually require a good amount of training data.

Motion segmentation and action recognition are done simultaneously in [76]. The proposed algorithm models the temporal order of actions while ignoring the spatial relationships between actions. The work in [110] models a complex activity by a variable-duration hidden Markov model on equal-length temporal segments. It decomposes a complex activity into sequential actions, which are the context of each other. However, it considers only the temporal relationships, while ignoring the spatial relationships between actions. AND-OR graph [3, 39, 104] is a powerful tool for activity representation. It has been used for multi-scale analysis of human activities in [3], α, β, γ procedures were defined for a bottom-up cost sensitive inference of low-level action detection. However, the learning and inference processes of AND-OR graphs become more complex as the graph grows large and more and more activities are learned. In [58, 59], a structural model is proposed to learn both feature level and action level interactions of group members. This method labels each image with a group activity label. How to smooth the labeling results along time is a problem and is not addressed in the paper. Also, these methods aim to recognize group activities and are not suitable in our scenario where activities cannot be considered as the parts of larger activities.

In [10], complex activities are represented as spatiotemporal graphs representing multi-scale video segments and their hierarchical relationships. Existing higher-order models [53, 54, 57, 134] propose the use of higher order potentials that encourage the smoothness of variables within cliques of the graph. Higher-order graphical models have been frequently used in image segmentation, object recognition, etc. However, few works exist in the field of activity recognition. We propose a novel method that explicitly models the action and activity level motion and context patterns with a Hierarchical-CRF model and use them in the inference stage for recognition.

The problem of simultaneous tracking and activity recognition was addressed in [18, 49]. In these works, tracking and action/activity recognition are expected to benefit each other through an iterative process that maximizes a decomposable potential function which consists of tracking potentials and action/activity potentials. However, only collective activities are considered in [18, 49], in which the individual persons of interest have a common goal in terms of activity. Our proposed graphical models address the general problem of activity recognition, when individual persons in the scene may conduct heterogeneous activities.

1.1.3 Sparse Modeling for Visual Recognition

Finally, we review related works on sparse modeling for object recognition, as well as activity recognition, where context is frequently explored. Despite various approaches of context modeling for object and activity recognition, few of the existing works address the optimum selection of sparse contextual features as well as the sparse graph structures. Subset selection and shrinkage methods are two usually ways for sparse modeling or feature selection. Various types of convex relaxation, particularly $l1$-regularization, have proven to be very effective for sparse modeling. There have been many works using element-wise $l1$-regularization for sparse modeling in visual pattern recognition, where the low-level features are usually predefined to us. These works includes face recognition [66, 136], image classification [68, 123, 128] and action/activity recognition [122]. However, few existing works explore inter-relationships between instances to be recognized for visual-based pattern recognition tasks.

[37] introduced $l1$-regularized class-specific dictionaries to enhance the discriminative power of the learned dictionaries. In [11], the proposed action model incorporated the

inter-relationships in the feature level between action classes to improve the classification performance by applying $l1$-regularized dictionary learning sequentially to learn the sparse representation of actions, as well as the sparse correlations between action classes with respect to their low-level features. However, the action model ignored the high-level inter-relationships between action classes, such as spatial and temporal relationships between action/activity classes. These existing works addressed the problem of feature selection using element-wise $l1$-regularization and did not consider the relations between different features. Thus they are very different from our problem where contextual features are naturally grouped together.

Group Lasso has been applied in logistic regression [70] to enable the group sparsity of model parameters in logistic regression. This penalty can achieve group level variable selection and is group-wise invariant to orthogonal transformation like ridge regression [70]. In [30, 105], element-wise $l1$-penalty within parameter groups is introduced to enforce element-wise sparsity within each parameter group. Group Lasso or sparse group lasso can be considered as a group $l1$-penalty with element-wise $l2$ or $l1$ penalty within each parameter group, respectively. Our work is closely related to Group Lasso.

In [28], the mixture regression with element-wise $l1$ regularization and pairwise ∞ regularization was proposed to achieve group sparse representation for visual analysis. The automatically grouped sparse features represents multi-edge connections between images to be annotated. Least square regression with group $l1$-regularization was used in [133] for the learning of sparse image annotations. These works used $l1$-regularization for feature selection of individual object classes without exploring the higher-level inter-relationships between the objects to be recognized or classified.

1.1.4 Anomalous Activity Recognition

Methods on the detection of anomalous activities can be divided into two categories: low-level anomaly detection and high-level anomaly detection. Approaches on low-level anomaly detection identify local spatio-temporal regions that probably contain anomalous patterns of low-level features, before high-level analysis such as object tracking and activity classification is done [6][97][125]. Some other works of this category represent activities as local spatio-temporal regions. Abnormal activities are discovered by modeling the dominant motion patterns of these local regions. In [96], a probabilistic framework was developed to detect local anomalies that have infrequent patterns with respect to their neighbors. In [19], the authors proposed an online algorithm to incrementally learn a sparse dictionary of motion features of normal instances. Spatial-temporal blocks whose motion features can not be reconstructed sparsely from the learned dictionary were identified as anomalies.

Several works have looked at the problem on high-level anomaly detection. These approaches usually identify semantically meaningful activities while detecting anomalies. In [40] activities were represented as bags of event n-grams. Disjunctive sub-classes of an activity class were discovered automatically. An information-theoretic method was used to explain the detected anomalies. In [88][41] activities were represented by Suffix Trees over multiple temporal scales which efficiently extract structure of activities by analyzing their constituent sub-events. An linear-time algorithm was proposed to detect anomalous subsequences of activities which are inconsistent with the learned Suffix Trees. In [48], attribute grammars were built to describe constraints on attributes and syntactic structure of normal events. Events which do not follow the syntax of the learned grammars or whose attributes do not satisfy these constraints were detected as anomalies. Many other works [95][46] were based on trajectories of moving objects in videos. Dominant trajectory clusters were identified and modeled as normal while trajectories which do not fit into the learned models were detected as anomalies. Although these approaches work well in identifying global anomalous activities whose characteristics can be determined by underlying object trajectories, they are not robust for activities involving abnormal local motions, e.g., “person gesturing”.

1.2 Main Contribution

In this dissertation, we focus on the modeling and recognition of activities in continuous videos, exploring various motion and contextual features. The main contribution includes four aspects.

(i) We propose the generalized SFG framework as in [36, 139] for the modeling and recognition of activities in videos, exploring the contextual information among low-level motion features. A dynamic switching scheme is proposed to adaptively choose the optimum type of motion features for activity recognition, which reduces the computational complexity while does not offset the recognition accuracy.

(ii) We propose three graphical models - structural model [140], higher-order CRF [142] and hierarchical-CRF [141] - to explore higher-level context within and between activities for the modeling and recognition of activities in videos.

(iii) We work on the recognition of more general visual objects that are inter-dependent of each other, such as activities in videos and objects on images, exploring the context within individual object and/or inter-dependence between objects. Specifically, by introducing group $l1$-regularization for the feature selection on existing context-aware graphical models, we propose a framework of sparse modeling for the recognition of potentially inter-dependent visual objects to select the optimal set of features, *as well as* the optimal sparse graph structure of the models.

(iv) Different from previous works on anomalous activity detection, which focus on the detection of activities with unknown motion attributes, we consider also anomalous activities that exhibit abnormal contextual attributes as in [138]. Based on types of abnormal attribute an anomalous activity has, we define three kinds of anomalous activities - point anomaly, contextual anomaly and collective anomaly. With the learned context-aware activity model, we how it can be used for the detection of anomalous activities of these three kinds.

1.3 Outline

The rest of the dissertation is organized as follows. In Chapter 2, we present the generalized "String-of-feature" framework, which models an activity and/or an activity sequence as a string of feature graphs and adaptively selects the optimum features for activity recognition, according to simple motion observations, in order to enhance the recognition accuracy while saving computational resource. In chapter 3, we first show how Struct-SVM can be modified for activity recognition. Then, we describe three graphical models for activity modeling and recognition - structural model [140], higher-order CRF [142] and hierarchical-CRF [141]. In Chapter 4, we propose a framework for the detection of anomalous activities in videos, considering both motion and contextual behaviors. In chapter 5, we analyze why sparse modeling is beneficial for many visual recognition tasks. A sparse modeling approach for the recognition of visual object is proposed for feature selection of both intra- and inter-object features, resulting a sparse model that is sparse in both feature level and structural level. Its applications in activity recognition and object recognition are explored in the experiments.Finally, we conclude the dissertation in Chapter 6 with the discussion of the future work.

Chapter 2

Feature-level Contextual Modeling - SFG

2.1 Introduction

The dynamical interactions between objects in a scene can be described using the following characterization: kinesics of individual objects (e.g., walking, running), temporal aspects (e.g., standing in a line), proximics or spatial relationship between objects (e.g., approaching), and haptics, (e.g., shaking hands, exchanging) [4]. Most work in activity recognition has concentrated on analyzing only one of these aspects (predominantly kinesics) as evidenced by the popular activity datasets like KTH [100] and Weizmann [35]. Most video analysis based applications such as surveillance, sports video analysis, content-based search, etc. require effective approaches for modeling and recognition of far more complex activities than these datasets.

Recognition of complex activities requires understanding of spatio-temporal relationships between different objects, in addition to individual variability, cluttered background, viewpoint changes, and other environment induced conditions. Modeling all these parameters proves

Figure 2.1: Representative frames of the datasets used in this work. Note that the videos contain multiple actors performing activities simultaneously, sometimes in the presence of irrelevant subjects.

to be a challenging task. In this work, we focus primarily on modeling and recognition of complex activities that involve multiple interacting objects - people and vehicles(see Fig. 2.1 for examples).

The main challenge that needs to be overcome is to develop a generalized representation of the video that respects the spatio-temporal ordering of local features at different resolution levels, ranging from local image interest points to trajectories of individual objects. To achieve this goal, we build abstract graphs upon features. The spatio-temporal representations combined with graph-based spectral matching techniques provides a powerful framework to model complex activities in video, and an efficient computational strategy is applied to estimate the similarities between them. Our framework is motivated by success of time sequence analysis approaches in speech recognition, but modified in order to capture the spatio-temporal properties of individual actions, the interactions between objects, and speed of activity execution.

2.1.1 Overview of Proposed Framework

2.1.1.1 Feature Descriptor

A video can be thought of as a spatio-temporal collection of primitive features (e.g. STIP features or track features). In order to handle the execution speed and motion variations of activities, we divide the video into small temporal bins. Each bin consists of a graphical structure representing the spatial arrangement of the local low-level features (see Fig. 2.2), which is called a ***feature graph***. Since activities in the video evolve along time, it is natural to represent the video as a "string of feature graphs" (SFGs). Thus the query/training video becomes a string of such graphs, while a test video is also a string of graphs, albeit of a possibly higher complexity.

Then, the problem is, how to match these two strings of graphs. This is cast as a combination of sub-graph matching and time sequence alignment (see Fig. 2.3). The local feature collections are first matched in a graph-theoretic manner, thereby preserving the spatio-temporal relationships between features.

The final match score between the query and test video is a dynamic programming based temporal alignment score between their corresponding SFGs, thus compensating for d-ifferences in speed of execution. By combining local spatial matching with global temporal alignment, we are able to match videos while respecting their spatio-temporal structure of local features. This gives us the ability to recognize activities that involve interactions between multiple objects like people getting into/getting out of a vehicle, following, dispersing, and so on. Our graph matching scheme supports partial matching, i.e., given query examples, similar actions in a testing video can be retrieved even if the testing video contains multiple actions happening simultaneously.

The proposed framework for SFG-based modeling of activities can be implemented using any features which obey spatio-temporal ordering, such as STIP features, cuboid features, or track features. In this model, we use the STIP features and track features to demonstrate the effectiveness of the framework. The STIP-based SFG method can be thought of as a generalization of the scheme in [90] where the spatio-temporal relationships were modeled using

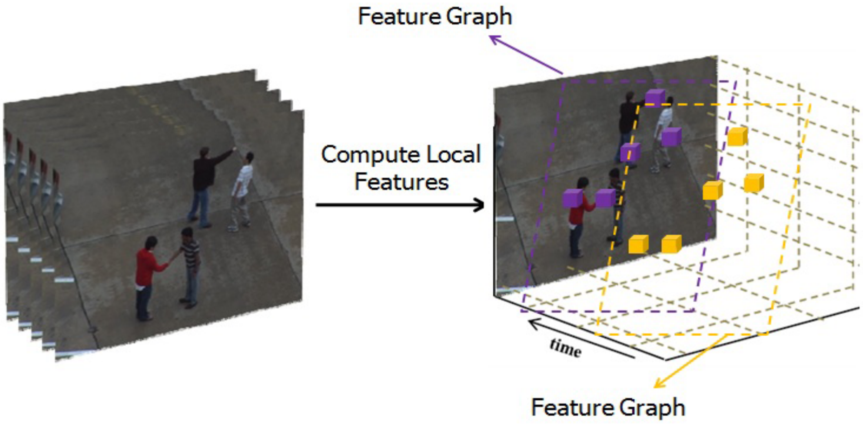


Figure 2.2: Activity modeling: Local features are computed from the video and grouped together to form feature collections. Temporally ordered strings of these local feature collections is termed as "string of feature-graphs" (SFGs).

a collection of rules. Our proposed framework allows a more general structure on the video and does not need to recognize body parts, unlike [83, 89], or primitive activities [45, 116]. Additionally, our feature model tackles action recognition in the two modalities (classification- and query-based). The model is not intrinsically tied to any classification mechanism hence enabling its use in scenarios such as query-based retrieval, i.e. recognition with only a single (or very few) example video(s) of the activity in question. This is a highly desired feature since obtaining multiple training examples for increasingly complex activities is often difficult. We show experimental results on three relatively complex datasets namely the UT-Interaction dataset [92], VIRAT dataset [79] and UCR VideoWeb activity dataset[21]. All these datasets comprise of multiple interactive activities in realistic settings with clutter and changing backgrounds.

2.1.1.2 Adaptive Feature Selection

One of the desired properties of the proposed SFG modeling is adaptive feature selection. Natural videos contain activities of different kinds, some of which are very localized

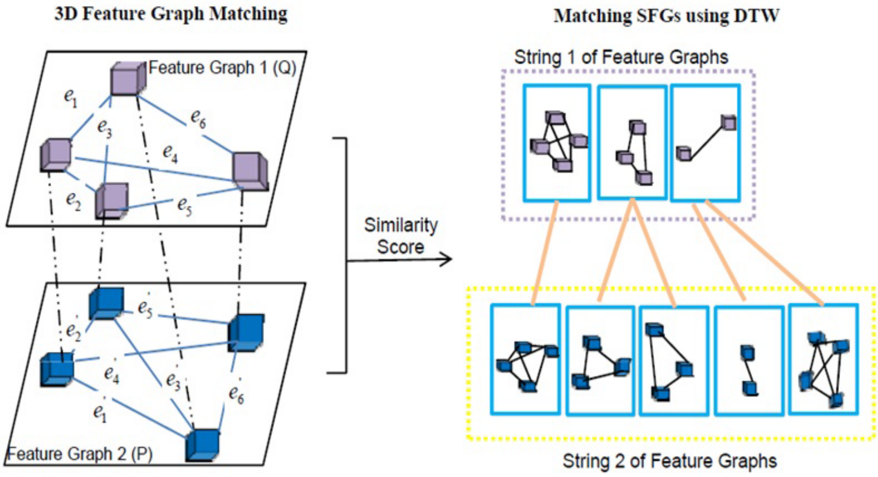


Figure 2.3: (Left) Local feature-graphs are matched using the graph-based spectral technique in Section 3.1. (Right) The feature-graph matching scores thus generated are used in DTW matching of the two videos. which are represented by strings of feature graphs, to account for difference in speed and execution.

(e.g., people shaking hands) while others evolve over wider space-time spans (e.g., two people approaching). The analysis of such “local” and “global” activities requires different kinds of features. For example, the global activity of people approaching can be understood based on the tracks of the individuals (low-resolution features), whereas their handshaking requires more detailed information (higher resolution features). Most existing methods in activity recognition focus only on one level of video resolution and describe features that are relevant only for that scenario [74, 120]. A few papers do combine multiple resolutions in describing features for activity recognition [25, 15]. However, they compute features at multiple resolution over each activity segment and integrate them. Our perspective in this work is to develop a switching system that adaptively selects between different feature types, using only one kind of feature in each time segment. This not only allows us to analyze a variety of activities in natural videos, but also does so in a computationally efficient manner.

Consider the example in Fig. 2.4. The first frame shows a person approaching a vehicle. This can be modeled and recognized using just a single track for the person (assuming

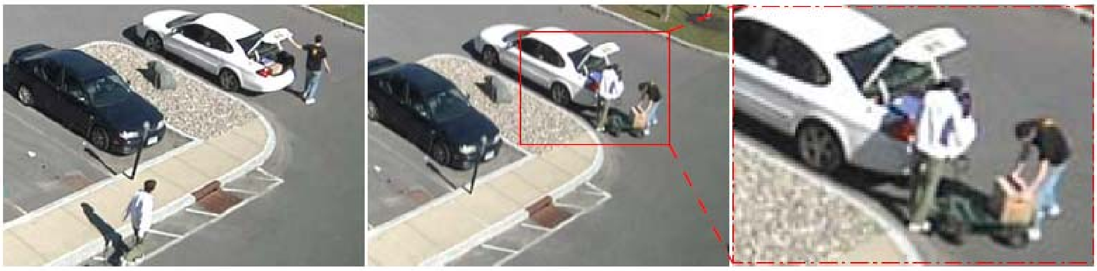


Figure 2.4: Representative frames of global and local activities recognized in this work. The first frame shows a person approaching a vehicle (global activity). Low resolution motion features are suitable for recognition of such actions. The second frame shows a person loading a trunk. This is a local activity which can be recognized by examining high resolution motion features in the region marked in red (showed in the third frame).

that people and vehicles can be detected). However, this is not enough to understand what the person is doing near the vehicle. Higher resolution features are necessary at this stage. This can be done if we can design a system that will automatically switch to a different class of features. It requires developing schemes that will determine the optimum feature describing the right level of motion details and automatically switch between these multi-resolution features. Switching systems, which have been studied widely [109], provide an excellent mechanism to achieve this. Integrated with the SFG modeling of activity, the switching scheme also provides significant computational benefits. Higher resolution features, like those required to recognize person loading an object to a vehicle, are computationally more expensive to extract and analyze than low-resolution trajectories of individuals. The switching scheme allows for varying computational loads depending upon the analysis requirement.

2.1.2 Contributions of Present Work

This chapter makes the following contributions. It proposes a string-based feature representation of activities, the SFG, that respects the spatio-temporal ordering in the scene. It shows how image-based and track-based features can be used in the SFG. It also proposes a switching scheme to automatically choose between the different features, thus reducing computational complexity. Experimental results are shown on state-of-the-art datasets. The main differences of this submission with [36] are as follows: we have shown how the proposed SFG model can be applied to track-based features, in addition to STIP features; we have incorporated a model for switching between different feature types using a switched dynamical system, in order to reduce computation complexity while preserving the recognition accuracy; we have added a significant amount of new experimental results.

2.2 String of Feature Graph

In this section, we describe the framework of "string of feature graphs" modeling of activities in video. In order to take into account of the spatio-temporal properties of individual actions and interactions between objects involved in activity recognition, we represent the features within a time window as a feature graph. Dynamic time warping is applied upon the generated strings of feature graphs in the final recognition, which allows for variations in sampling rates and speed of activity execution.

2.2.1 Model Development

Let us consider a video V of duration T containing a complex activity. V can be represented as a collection of feature points $V = \{f^t_{x,y} | t \in [1,T]\}$ where $f^t_{x,y}$ is a feature point at spatial location x,y and time index t. Matching two videos would involve matching their corresponding feature points in a spatio-temporal order preserving manner. Let us divide the video into N intervals in time $t_0, t_1..t_N$ and let the features contained in a single time interval be collectively denoted as F. Therefore, the video can now be represented as $V = \{F_1, F_2, \dots F_N\}$ where $F_1 = \{f^t_{x,y} | t \in [t_0, t_1)\}$, $F_2 = \{f^t_{x,y} | t \in [t_1, t_2)\}$, etc. Now, the spatio-temporal matching of two videos $V^{(1)}$ divided into N_1 intervals and $V^{(2)}$ divided into N_2 intervals would involve matching their individual feature collections $\{F_i^{(1)}|i = 1\dots N_1\}$ and $\{F_i^{(2)}|i = 1\dots N_2\}$ in a temporal order-preserving fashion, wherein the similarity measure between two feature collections would involve feature content matching as well as geometric structure matching. This representation of a video naturally leads us to a string representation, where local feature collections F form the elements of the string. In order to keep the structure information within each feature collection F, a graphical description is used and F is represented as a feature-graph. Therefore the temporally ordered collection of F forms a string of feature-graphs (SFGs). Fig. 2.2 visually explains the modeling process.

2.2.2 Spatio-temporal Matching of SFGs

As explained earlier, the match score between two videos is the string alignment score between their corresponding SFGs. Since string alignment of any form requires a known method of measuring distance between the characters of the strings, we describe in the following subsections how we a) use a spectral technique to compute similarity between two feature-graphs (feature-graphs being the characters in the SFG strings) and b) use the so computed feature-graph match scores to find the optimal alignment score between two SFGs.

2.2.2.1 Matching Two Feature-Graphs

Computing the similarity between two feature graphs involves matching individual feature-descriptors (i.e., nodes) as well as pairwise neighborhood relationships (i.e., edges).

We represent each feature collection, i.e., each character in the string, as a fully-connected three dimensional graph where feature points form the nodes. Then the feature correspondence problem can be formulated as a graph matching problem by considering the matching between both nodes and edges. Given two such graphs, one being a feature collection from the testing video, P, with n_P nodes, and one being a feature collection from query video, Q, with n_Q nodes, we follow the spectral technique described in [65] to find correspondences between their respective feature points (nodes). This approach avoids the combinatorial explosion inherent to the correspondence problem by formulating it in closed form as a spectral analysis problem on a graph adjacency matrix.

An assignment (i,i') is defined as a correspondence between a pair of nodes from two graphs, where $i \in P$ and $i' \in Q$. For each candidate assignment $a = (i,i')$, there is a distance score between feature i and feature i' associated with it. Let L be a list (with length $n_L = n_P \times n_Q$) of all possible candidate assignments between features of P and Q. Given such a list, let a matrix $\mathbf{M}$ (size $n_L \times n_L$) store the affinities of every possible pair of assignments $(a,b) \in L$. Note that $\mathbf{M}(a,a)$ for $a = (i,i')$ measures how well the feature point i matches the feature point i', and $\mathbf{M}(a,b)$, where $a = (i,i')$ and $b = (j,j')$, describes the relative pair-wise relationships of points (i,j) in P with points (i',j') in Q. We define $d_n(i,i')$ as the distance between the nodes i and i'. It measures the Euclidean distance between the features of nodes i and i'. In order to account for scale, we consider the geometric structure of the graphs based on the angles between the edges in the graph. We define $d_e(\vec{ij},\vec{i'j'})$ as the distance between edges (i,j) and (i',j') based on the angle difference between them. For candidate assignments $a = (i,i')$ and $b = (j,j')$, the elements $\mathbf{M}(a,a)$ and $\mathbf{M}(a,b)$ of matrix $\mathbf{M}$ are defined as

$$\mathbf{M}(a,a) = \begin{cases} \omega_n[1 - d_n(i,i')] & d_n(i,i') \le \tau_n \\ 0 & d_n(i,i') > \tau_n \end{cases}, \tag{2.1}$$

$$\mathbf{M}(a,b) = \begin{cases} \omega_e[1 - d_e(\vec{ij},\vec{i'j'})] & d_e(\vec{ij},\vec{i'j'}) \le \tau_e \\ 0 & d_e(\vec{ij},\vec{i'j'}) > \tau_e \end{cases}, \tag{2.2}$$

where τ_n is a pre-defined maximal distance between two features whose relationship should not be ignored and τ_e is a pre-defined threshold for edge difference. d_n and d_e are normalized between [0,1] and thus τ_n and τ_e are also chosen in that range. ω_n and ω_e are weights of node matching and edge matching, which adjust the relative importance of node similarity and edge similarity in the graph matching.

Now, suppose the length of the query feature graph is n_Q and the length of the testing feature graphs is n_P. Let x be an indicator vector of length $n_Q \times n_P$ such that $x(a) = 1$ if candidate assignment $a = (i, i')$ represents a corresponding pair of nodes and 0 otherwise. We aim to find an optimal solution x^* which maximizes the score

$$x^* = \arg\max_x x^T \mathbf{M} x. \tag{2.3}$$

The solution to the above problem, x^*, gives the optimal correspondence between feature points in P and Q. This solution has to be subject to the mapping constraints required by one-to-one mapping as in [65].

Once we estimate the optimal match, x^*, of two feature collections P and Q, their similarity can be measured by

$$sim(Q,P) = (x^*)^T \mathbf{M} x^*, \tag{2.4}$$

and the distance between them defined as

$$d(Q,P) = 1 - \frac{sim(Q,P)}{sim(Q,Q)}. \tag{2.5}$$

2.2.3 Dynamic Time Warping of SFGs

Recall that an SFG of a video is a time-ordered strings of its feature-graphs. Matching two SFGs should be flexible, in that it should be robust to the different rates at which an activity might occur and also the actual length of the template video and the test video. This can be achieved by time normalizing the two SFGs. The speech recognition community has successfully used a dynamic programming approach termed dynamic time warping (DTW) [94] for non-linear time normalization. We borrow this idea and apply it to flexibly match two SFGs, hence making them robust to speed differences in different instances of the activity.

The aim of DTW is to minimize the local distortion between two sequences by finding an optimal warping function φ. For our case, the local distortion is defined as the sum of local pair-wise distances between their feature collections. Formally, for two SFGs $\mathscr{Q} = \{Q_1 \ldots Q_{N_{\mathscr{Q}}}\}$ and $\mathscr{P} = \{P_1 \ldots P_{N_{\mathscr{P}}}\}$, where $N_{\mathscr{Q}}$ and $N_{\mathscr{P}}$ are the number of characters (i.e. feature graphs) in $\mathscr{Q}$ and $\mathscr{P}$ respectively, the sequence distortion is defined as

$$D_{\varphi}(\mathscr{Q}, \mathscr{P}) = \frac{1}{M_{\varphi}} \sum_{k=1}^{K_{\varphi}} d(Q_{\varphi(k)}, P_{\varphi(k)}) m_k, \tag{2.6}$$

and the distance between the two SFGs can be computed as

$$D(\mathscr{Q}, \mathscr{P}) = \arg\min_{\varphi} D_{\varphi}(\mathscr{Q}, \mathscr{P}). \tag{2.7}$$

Here m_k are the path-weights, and $M_{\varphi} = \sum_k m_k$ is a normalization factor. The details of the solution to this optimization problem can be found in [94]. The entire matching process is pictorially presented in Fig. 2.3.

2.2.3.1 Subsequence DTW for Continuous Video

In real applications, the test video is often a continuous video containing multiple persons performing multiple activities. Given a query video, which often contains only the desired activity, we would want to find a subsequence within the testing video sequence that optimally fits the query sequence, i.e., identify the fragment within the testing video that is most similar to the query. For this purpose, we utilize a variant of DTW – subsequence DTW [72], by releasing the restriction on the boundary condition, as explained below.

Let $\mathscr{Q} = \{Q_1 \ldots Q_{N_\mathscr{Q}}\}$ and $\mathscr{P} = \{P_1 \ldots P_{N_\mathscr{P}}\}$ be two SFGs of the query and testing videos respectively, where $N_\mathscr{P} >> N_\mathscr{Q}$. The goal is to find a subsequence $\mathscr{P}'(a^*,b^*) = \{P_{a^*} \ldots P_{b^*}\}$ with $1 \leq a^* \leq b^* \leq N_\mathscr{P}$ such that

$$(a^*,b^*) = \arg \min_{(a,b):1 \leq a \leq b \leq N_\mathscr{P}} \left(D(\mathscr{Q},\mathscr{P}'(a^*,b^*)\right). \tag{2.8}$$

The indices a^* and b^* can be computed by a small modification of the classical DTW algorithm in the generation of the accumulated cost matrix $\mathbf{C}$ used to describe the cost of aligning two sequences [72]. The goal of DTW is to find the minimal cost path through an accumulated cost matrix. By applying subsequence DTW, it can be shown that $b^* = \arg\min_{b \in [1,N_\mathscr{P}]} \mathbf{C}(N_\mathscr{Q},b)$. $a^* \in [1,N_\mathscr{P}]$ is the maximal index such that path $(a^*,1)$ belongs to the warping path.

It is usually the case that the database contains multiple instances of the activity that are similar to the query example. It is desirable to retrieve all the subsequences of $\mathscr{P}$ that are close to $\mathscr{Q}$ with respect to the DTW distance. This can be achieved by recursively repeating the above process. We present our implementation of matching continuous video using subsequence DTW in Algorithm 1.

2.3 Special Cases

Irrespective of the features used to describe a video, the task of activity recognition requires us to examine the properties of these features as well as their spatial and temporal arrangement. Therefore, although motion features can be very different from each other, we can represent the local spatio-temporal volume (STV) surrounding the targeted activity as an

Algorithm 1 Matching SFG of continuous video through subsequence DTW

```
Input:   $\mathscr{Q} = \{Q_1 \dots Q_{N_\mathscr{Q}}\}$    SFG of the query video
         $\mathscr{P} = \{P_1 \dots P_{N_\mathscr{P}}\}$    SFG of the testing video
         $\tau \in \mathbb{R}$                               cost threshold
Output:  Ranked list of all subsequences of $\mathscr{P}$ that have a DTW distance to $\mathscr{Q}$ below the threshold $\tau$.
  1. Initialize the ranked list to be an empty list.
  2. Construct accumulated cost matrix $\mathbf{C}$ whose elements are defined as
       $\mathbf{C}(n,1) = \sum_{k=1}^{n} d(Q_k, P_1), n \in [1, N_\mathscr{Q}],$
       $\mathbf{C}(1,m) = d(Q_1, P_m), m \in [1, N_\mathscr{P}],$
       $\mathbf{C}(n,m) = \min\{\mathbf{C}(Q_{n-1}, P_{m-1}), \mathbf{C}(Q_{n-1}, P_m), \mathbf{C}(Q_n, P_{m-1})\} + d(Q_n, P_m).$
  3. Define a distance function: $\Delta(b) \triangleq \mathbf{C}(N_\mathscr{Q}, b)$, $b \in [1, N_\mathscr{P}]$.
  4. Determine $b^* \in [1, N_\mathscr{P}]$ that gives minimal $\Delta$.
  5. If $\Delta(b^*) > \tau$ (which means no additional subsequence of $\mathscr{P}$ close to $\mathscr{Q}$ exists), then terminate the procedure.
  6. Compute the corresponding DTW-minimizing index $a^* \in [1, N_\mathscr{P}]$ using standard DTW algorithm, which searches optimal warping path in $\mathbf{C}$ in reverse order of the indices starting with $(N_\mathscr{Q}, b^*)$.
  7. Extend the ranked list by the subsequence $\mathscr{P}'(a^*, b^*)$.
  8. Set $\Delta(b) \triangleq \infty$ for all $b$ within a suitable neighborhood of $b^*$.
  9. Continue with Step 4.
```

SFG, whose edge features define the geometric structure of its node features. In this section, we describe the construction of SFG for track and STIP based features. The main task is to develop suitable node and edge measurement techniques discussed in 2.2.2.1 for the particular motion features.

2.3.1 Track-Based SFG

Activities involving objects exhibiting long-distance motion can be recognized from the global motion trends of the objects and their pattern of interactions [81][33][102]. Some examples of such activities are car u-turn, car turn, people dispersion, and group walking, etc. In this section, we implement the SFG framework in activity recognition based on motion features of tracks. Suppose we have trajectories and identifications of moving objects in the scene. The local STV surrounding each track is an interesting activity region. All the collected features of tracklets within the interesting spatio-temporal volume make up a feature graph.

Track Descriptors In this section, we develop four motion feature descriptors for tracks. The in-plane rotation-invariant descriptors - normalized change of gradient direction(NDG) and normalized change of gradient magnitude (NMG) - capture the global motion pattern of individual tracks. NDG of a track is its absolute change of gradient direction along time normalized by its maximum absolute value. NMG of a tracklet is its change of gradient magnitude along time normalized by the maximum magnitude of gradient. Let $\tilde{t}_i$ be the track of object i, and $p_i(t) = [x_i(t)\ y_i(t)]$ for $t = 1, 2, ...$ be the position of object i at time t. The features of the track i at time t are defined as

$$NDG_i(t) = \frac{|\frac{d}{dt}\arctan(\frac{d_{x_i}(t)}{d_{y_i}(t)})|}{max(|\frac{d}{dt}\arctan(\frac{d_{x_i}(t)}{d_{y_i}(t)})|)}, \tag{2.9}$$

$$NMG_i(t) = \frac{\sqrt{d_{x_i}(t)^2 + d_{y_i}(t)^2}}{max(\sqrt{d_{x_i}(t)^2 + d_{y_i}(t)^2})}, \tag{2.10}$$

where $d_{x_i}(t)$ and $d_{y_i}(t)$ are the instantaneous gradients of object i along x and y axis respectively. It is is easy to prove that both NDG and NMG are in-plane rotation invariant. Fig. 2.5 shows the sample descriptors.

Slope of smoothed relative distance (SRD) of a pair of tracks is the change of their relative distance smoothed along time, which captures the interaction trends between the two tracks. Relative distance of two tracks is obtained first. Break-points, where the trend of interaction changes (e.g. from approaching to dispersing) are detected and used to segment the RD descriptor. Break-points are defined as those local extrema of the relative distance sequence

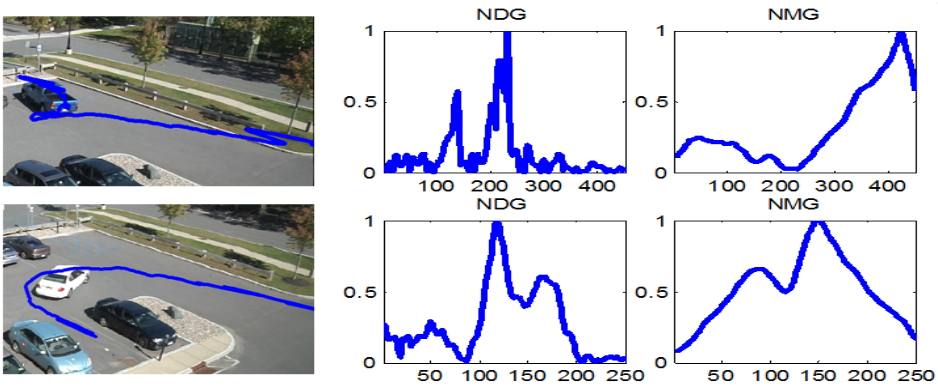


Figure 2.5: Examples of NDG and NMG descriptors. The left column shows the sample images for a vehicle-backup (top) and a vehicle-u-turn (bottom) (only regions of interest are shown). The next two columns show the corresponding NDG and NMG descriptors.

whose distance with the immediate previous extrema is greater than a pre-determined threshold. Exponential curve fitting is utilized to smooth out the segments in the resulting the RD descriptor. Let $\tilde{t_i}$ and $\tilde{t_j}$ be the tracks of object i and j respsectively, and $p_i(t) = [x_i(t)\ y_i(t)]$ and $p_j(t) = [x_j(t)\ y_j(t)]$ for $t = 1, 2, ...$ be the positions of objects i and j at time t. The relative distance of object i and j at time t is $d(t) = \sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2}$. The detected break points $t_1, t_2, ..., t_n$ and the beginning and ending points t_0, t_{n+1} segment the sequence of relative distance of the two objects into $n+1$ segments $rd(k)$ for $k = 0, 1, ..., n$. The *RD* and *SRD* features of tracks of i and j at time t are defined as

$$RD_{(i,j)}(t) = exp\_fit(rd(k)) \quad if \quad t_k < t \leq t_{k+1}, \tag{2.11}$$

$$SRD_{(i,j)}(t) = \frac{RD_{(i,j)}(t)}{dt}, \tag{2.12}$$

where exp_fit refers to fitting an exponential function to the specific rd sequence.

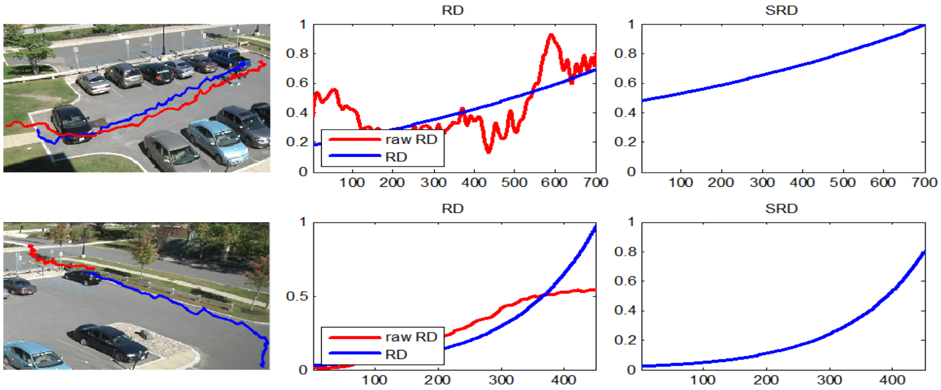


Figure 2.6: Example of RD and SRD of two tracks. The images show sample frames of two people walking together (top) and person leaving a vehicle (bottom) (only regions of interest are shown). The graph on the left shows the raw relative distance between the two tracks and the exponential fitting result in each case. The graph on the right shows the derivative of smoothed relative distance (SRD) in each case.

2.3.1.1 Track-Based Feature Graph Matching

In the feature graph matching, tracks are segmented into tracklets by concatenated equal-length time windows. Each tracklet forms a node in the graph. The node features in the graph are the smoothed motion features of the tracklets. The edge features quantize the interaction between the two underlying objects. It is natural to use the smoothed Euclidean distance between individual track features of two tracklets as the node distance measurement, and the smoothed distance between the interacting features of two pairwise tracklets as the edge distance measurement.

Assume tracklet i belongs to the query video, and tracklet i' belongs to the testing video. Let f_i^{IND} be the concatenated NDG or NMG features of tracklet i, and $f_{i',m}^{IND}$ be the concatenated NDG or NMG features of tracklet i'. Let $f_{\overrightarrow{(i)(i')}}^{SRD}$ be the concatenated SRD between i and i'. For a feature graph Q in the query video and a feature graph P in the testing video, the node distance, edge distance, and elements of similarity matrix defined in Section 2.2.2.1 are specified as

$$d_n(i,i') = \frac{\|f_i^{IND} - f_{i'}^{IND}\|}{s}, \tag{2.13}$$

$$d_e(\vec{ij},\vec{i'j'}) = \frac{\|f^{SRD}_{\vec{(i)(i')}} - f^{SRD}_{\vec{(j)(j')}}\|}{s}, \tag{2.14}$$

where s is the length of a tracklet. When we are interested in only the interaction patterns of tracks involved in activities, ω_n defined in 2.2.2.1 is set to be zero, and only differences in track interactions are considered in the graph matching. When we are only interested in individual motion patterns of objects involved, ω_e is set to be zero, and only node differences are considered in the graph matching.

2.3.2 STIP-Based SFG

Bag-of-Words based on STIP features exhibits promising results in object categorization and semantic video retrieval across several datasets [75]. While the statistics of STIP features may indicate which candidate activity the test video contains, BOW needs large amount of training data to achieve good recognition performance. Also, it is easily understandable that the spatio-temporal arrangements of STIP clusters is essential for activity recognition. For instance, the actions - open a trunk and close a trunk - have very similar statistics of STIP descriptors, but the two are actually very different activities due to the different temporal order of STIP clusters. In this section, we systematically incorporate spatial and temporal information of STIPs in activity recognition model, by implementing the SFG framework on top of STIP features. The proposed method can achieve the same recognition level with much less training data.

2.3.2.1 STIP-Based Feature Graph Matching

To extract spatial-temporal features, we rely on the spatio-temporal interest point (STIP) detector proposed in [61]. The STIPs are detected by finding the center locations of local spatio-temporal volumes, which have large variations along both the spatial and the temporal directions, using a spatio-temporal extension of 2D Harris operator [61]. Then, STIP feature-graphs are constructed following the procedure described in 2.2.1.

In the matching of feature graphs with STIPs as the nodes, it is natural to use the Euclidean distance as the similarity measurement between two nodes, and use the difference between angles of two edges as the similarity measurement between the two edges. A STIP feature f typically consists of a location descriptor f^l, which indicates its 3-D location in the spatio-temporal domain, and a local motion descriptor f^m. Let $f_i = (f_i^l, f_i^m)$ be the STIP feature vector of node i, and $f_{i'}$ be the STIP feature vector of node i'. The distance measurements $d_n(i,i')$ and $d_e(\vec{ij}, \vec{i'j'})$ in Section 2.2.2.1 are specified as

$$d_n(i,i') = 1 - \frac{f_i^m \cdot f_{i'}^m}{\|f_i^m\| \|f_{i'}^m\|}, \tag{2.15}$$

$$d_e(\vec{ij}, \vec{i'j'}) = 1 - e^{-p(1-\frac{[(f_i^l - f_j^l)]\cdot[(f_{i'}^l - f_{j'}^l)]}{\|(f_i^l - f_j^l)\| \|(f_{i'}^l - f_{j'}^l\|)})}. \tag{2.16}$$

2.4 Adaptive Feature Selection

A video can be thought of as a spatio-temporal collection of primitive low resolution features and high resolution features. Recognition of activities can be achieved by different levels of motion details [1]. Low resolution features are often simpler and more sparse than high resolution features. They work better at recognizing activities characterized by global motion pattern, such as vehicle turn, group walking and people dispersion. On the other hand, algorithms based on high resolution features are suitable for recognizing activities in the local mode because more motion details are captured. Although such algorithms can also recognize global activities to some extent, they are often computationally expensive [1]. In order to improve the recognition accuracy while reducing computation complexity, it is important to choose motion features at the right scale of resolution for the recognition task. In this section, we integrate the proposed SFG modeling of activities into a Switched Dynamic System (SDS) to develop a scheme of adaptive feature selection in activity recognition. Our goal is to optimize the recognition accuracy as well as the computational complexity of our system by switching between the two kinds of features.

2.4.1 Switched Dynamic System Model

We propose a SDS model for the switching between activities for complex videos containing both global and local activities. In the SDS, two modes are considered: global mode and local mode corresponding to the global activity and local activity. Each spatio-temporal activity volume is assigned with a mode and the feature used in recognizing the activity is determined accordingly (how to locate these sptio-temporal activity volumes is introduced in Section 2.5.1). Motivated by works in hybrid systems like [87], the SDS model can be specified by the tuple

$$\Psi = \{M, O, \Phi, F^{low}, F^{high}\}, \tag{2.17}$$

where M denotes the modes in the system, O are the observations from which motion features are extracted. Φ is the attribute pattern derived from observations of low resolution motion details, and are used to decide the modes, F^{low} are the low resolution features and F^{high} are the high resolution features.

2.4.2 Switching Between Activity Modes

In statistical pattern recognition methods, modes are also known as pattern classes. Each pattern class consists of different patterns, which can be represented by a vector of quantitative attributes $\Phi = [\phi^1, \phi^2, ..., \phi^a]$ carrying distinguishing information about the patterns [5], where a is the number of informative attributes. Each mode is assumed to have distinguishing distribution of these attributes. Thus the joint distribution of the informative attributes can be used to determine the mode of the observed pattern.

Let O_t be the observations of motion at time step t; the corresponding pattern is defined as $\Phi_t = \Gamma(O_t) = [\phi_t^1, \phi_t^2, ..., \phi_t^a]$, where Γ is the mapping from the observation space to the attribute space. Let $p(M)$ be the prior probability of each mode and $P(\Phi|M)$ be the distribution of attribute vector of a given mode M. Maximum likelihood can be used to determine the modes from the observed attributes. For an observed pattern Φ_t, the mode M_t of the pattern is

$$M_t = \underset{M}{max}[P(\Phi_t|M) \cdot p(M)]. \tag{2.18}$$

To simplify the estimation of probability distributions, we suppose that different types of attributes are independent of each other. A Naive Bayesian network can be applied to decide the underlying model M_t given a certain pattern Φ_t. Let g denote the global mode and l denote the local mode and $p(g)$ and $p(l)$ be the prior probabilities of global and local modes. Let the distributions of the i^{th} attribute given the mode M be $p(\phi_i|M)$. The distribution of the attribute vector given the mode is $\prod_i p(\phi_i|M)$. Thus the mode M_t of pattern Φ_t is

$$M_t = \begin{cases} g & \text{if } \prod_i p(\phi_t^i|g) \cdot p(g) > \prod_i p(\phi_t^i|l) \cdot p(l) \\ l & \text{if } \prod_i p(\phi_t^i|g) \cdot p(g) < \prod_i p(\phi_t^i|l) \cdot p(l) \end{cases} . \tag{2.19}$$

We integrate the SFG method into the SDS model to realize automatic feature selection in activity recognition. Fig. 2.7 shows the overall flow of the proposed recognition system.

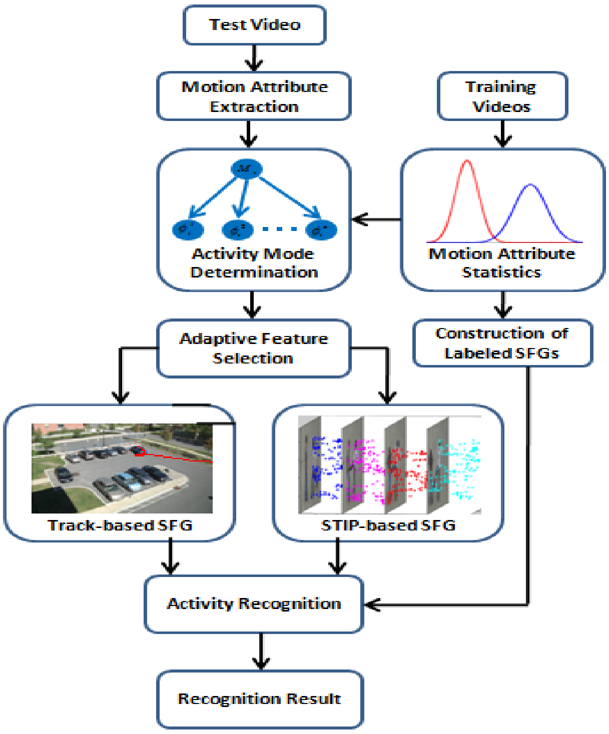

Test Video
Motion Attribute Extraction
Training Videos
Activity Mode Determination
Motion Attribute Statistics
Adaptive Feature Selection
Construction of Labeled SFGs
Track-based SFG
STIP-based SFG
Activity Recognition
Recognition Result

Figure 2.7: Overall flow of the activity recognition system using adaptive feature selection in the SFG framework.

2.5 Experiments

In order to evaluate the efficacy of our method to recognize complex activities involving multi-object interactions, we conducted experiments on three state-of-the-art datasets containing long duration videos and a large scale of complex activities including UT-Interaction dataset [92], VIRAT dataset [79] and UCR VideoWeb activity dataset[21],

UT Interaction dataset [92] is composed of both segmented and unsegmented videos, and include several pairs of interacting people simultaneously executing activities across different background, scale and illumination. The interaction activities which we looked at are shaking hands, hugging, pointing, punching, kicking and pushing. VIRAT dataset is a state-of-the-art activity dataset with many challenging characteristics, such as wide variation in the activities and the clutter in the scene. It consists of surveillance videos of six parking lots with different scales of resolution. The activities in the dataset includes single vehicle activities, person and vehicle interactions, and people interactions. We examine fourteen kinds of activities - global activities including vehicle u-turn, vehicle turn and vehicle backup, people walking together, people gathering, and people dispersion, and local activities including person loading an object to a vehicle, person uploading an object from a vehicle, person opening a vehicle trunk, person closing a vehicle trunk, person getting into a vehicle and person getting out of a vehicle. The portion of the UCR VideoWeb activity dataset [21] we work on (details can be obtained from the authors) involves up to 10 actors interacting in various ways with each other, vehicles and facilities. The activities were: people meetin, people following, vehicles turning, people dispersing, shaking hands, gesturing, waving, hugging and pointing.

In accordance with the motivation of the framework, we work in both classification-based and query-based activity recognition. In the classification-based scheme, testing instances are classified into predefined kinds of activities given multiple training instances of the same kinds using nearest neighbor classifier. The query-based scheme is based on an example video-based retrieval framework wherein the algorithm is provided with one (or, at most, a few, but not enough to build a classifier) video(s) depicting an action of interest. The aim is to retrieve videos which have similar activity as the query video(s) has.

2.5.1 Preprocessing

Object detection and tracking are performed first. We utilize the tracking method developed in [106] to obtain the trajectories of moving objects. Identifications of moving objects (person, vehicle, or others) are obtained using [26], and shadows are excluded by using color histogram. Note that object identification is applied to each trajectory.

Preprocessing of Tracks A weighted moving average filter is applied to the raw tracks in order to smooth out the effect of local outliers on the global motion pattern. Tracks obtained from automatic trackers are often short and contain at most one or two complex activities involving large-scale motion. Observing that stopping is often the sign of the end of an global activity, we detect the stopping events on each track and segment long tracks accordingly. Each track segment is considered as a complete activity agent. A stopping is detected when the variance of the positions of the interesting objects within a temporal window is below a predefined threshold.

Adaptive Feature Selection Background substraction in [106] is used to obtain the binarized silhouettes of moving objects and their bounding boxes. Informative attributes derived from the silhouettes and bounding boxes are used to specify the Naive Bayesian network discussed in Section 2.4.2.

Half of the dataset [79] parking lot 04 is used as training data to select the informative attributes, and the probability distribution of these attributes is obtained using Gaussian mixture model and Expectation Maximization. The selected track-based low resolution attributes are:

(1) Variance of width of bounding box σ_W.

(2) Variance of the area inside the silhouette of moving object, where $Area = H \times W$, H and W are the height and width of the bounding box.

(3) Average velocity of the underlying objects $mean_v$.

(4) Range of the track R, which is defined as R = $max[max(x) - min(x), max(y) - min(y)]$.

The estimated conditional probability distribution of motion attributes given the activity mode is shown in Fig. 2.8.

Activity instances in the training data are labeled and segmented out, and SFGs are

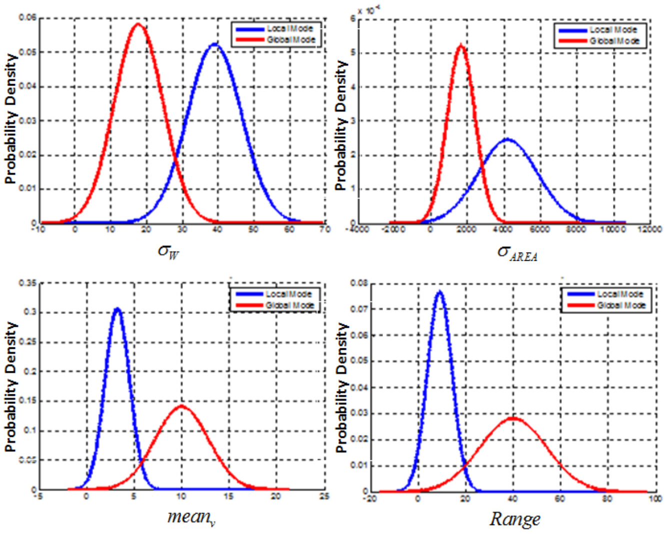


Figure 2.8: Estimated conditional probability distributions of the four motion attributes given the activity mode

constructed for these instances. For the track-SFG, we use joint NDG-NMG and RD-SRD features. Tracks involved in local activities are often very short and have a small range. We consider it is unlikely that track-based low resolution feature can distinguish local activities. So, we do not train track-based SFGs for local activities.

For distance thresholds τ_n and τ_e in (2.1), we determined their optimal values experimentally based on labeled training data and used them in testing. The values of τ_n and τ_e used in the following experiments are 0.4 and 0 for STIP-based SFG and are zeros for track-based SFG. The values may be different in other scenarios.

2.5.2 STIP-based SFG Results

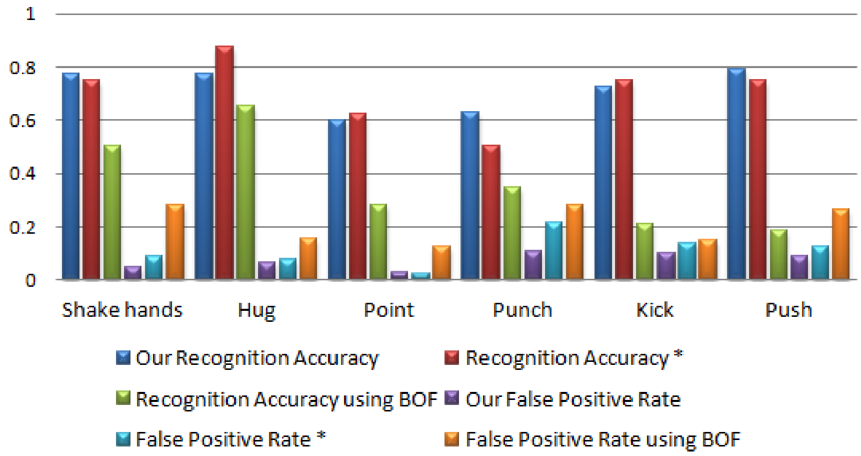
In this part of experiments, we implement STIP-based SFG on UCR VideoWeb Dataset, UT Interaction Dataset, and VIRAT Dataset.

2.5.2.1 Classification-Based Recognition Results

The classification-based recognition performance of the proposed algorithm is first evaluated on the UT Interaction dataset [92]. In order to compare with previous systems, we use an experimental setting similar to [90], which proposed a supervised learning method for the same set of activities on this dataset. We randomly choose two among the ten videos of each class to form the training set and leave out the others for testing.

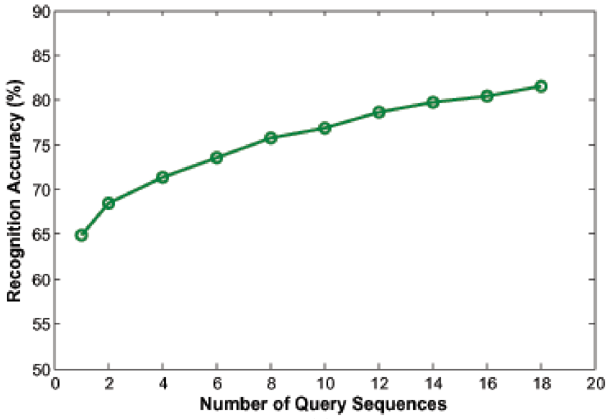
We verify that our system is able to recognize multiple complex activities from continuous videos. We were able to achieve high recognition scores and lower false positive rates. We compare our results with previous methods in Fig. 2.9. Our overall performance on the UT Interaction dataset is superior to Bag-of-Feature approach. Here the results of Bag-of-Feature approach are reported on segmented video clips, while our results and [90] are reported on continuous video. Our results are similar to those in [90] for some activities and better for others. However, our approach can use only a single query to perform recognition as demonstrated in Fig. 2.10 and hence has a wider generalizability. In [92], recognition results of several approaches are reported on the same dataset; the average recognition accuracy is in the range from 0.49 to 0.88. Our performance is comparable to the best performance in [92]. Note that the experiment settings in [92] are slightly different from ours. Their results are reported by leaving one out among a set of ten for testing and using the other 9 for the training, and the videos are segmented, while we use 2 sets as labeled query videos and test on 8. Thus we are achieving results comparable to [92] with much less training data on continuous videos (a significantly harder problem).

In the next experiment, we verify the effectiveness of our system in correctly classifying activities in VIRAT Dataset [79]. We work on the segmented video clips from the portion



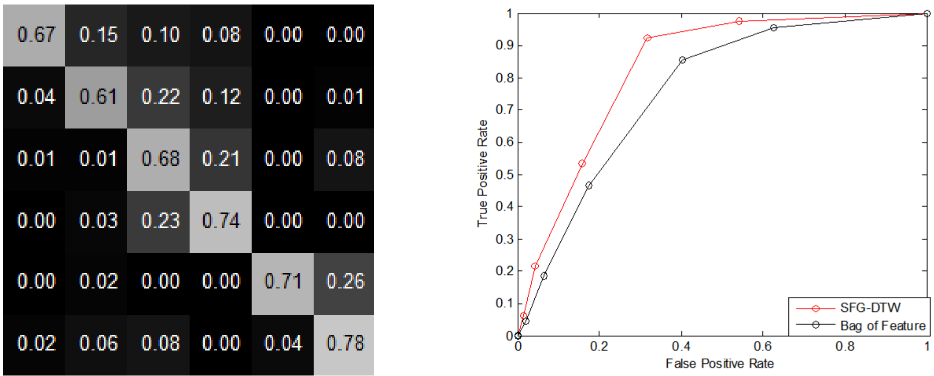
(a)

Figure 2.9: Recognition accuracy on the UT-Interaction dataset by using voting scheme on top of SFG model.



(a)

Figure 2.10: The recognition accuracy of our method with respect to number of query examples. It can be seen that when number of query example decreases, the performance of our method does not drop significantly.



0.67	0.15	0.10	0.08	0.00	0.00
0.04	0.61	0.22	0.12	0.00	0.01
0.01	0.01	0.68	0.21	0.00	0.08
0.00	0.03	0.23	0.74	0.00	0.00
0.00	0.02	0.00	0.00	0.71	0.26
0.02	0.06	0.08	0.00	0.04	0.78

(a) (b)

Figure 2.11: (a): Confusion matrix for VIRAT dataset. Nearest neighbor classifier and eight-leave-one-out cross-verification are used. (1:person loading an object to a vehicle, 2:person uploading 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). Most misclassifications are inside activity group (1, 2, 3, and 4) and activity group (5 and 6). (b): ROC of 7-NN classifier on VIRAT dataset (Bag of feature: Laptev et al. [100]+7NN).

of VIRAT dataset for which annotation is available, because the evaluation of this experiment needs the ground truth of activities. We segmented out the video clips according to the annotation files. Each video clip contains only one execution of the activities of interest. There are forty eight video clips in total, eight instances for each type of activity,in the whole dataset used in this experiment. The results are shown as a confusion matrix and ROC curves in Fig. 2.11.

2.5.2.2 Query-Based Retrieval Results

We compute the DTW aligning cost between the query SFGs of the testing video and each query video containing a specific action and count the instances that the DTW distance is less than a threshold. Based on this number (i.e., number of similar training videos), the system makes a decision on the recognized activity.

To evaluate the performance of the STIP-based SFG method in query-based retrieval, we first work with the UCR VideoWeb activity dataset [21]. We work with video clips from this dataset, report the best matches found by our system and accordingly present and analyze the accuracy/false positive rates.

We proceed by taking a small video clip depicting a complex activity and search the dataset for matches. The STIP features for the query and the dataset videos are computed. The query and dataset videos were uniformly segmented into temporal segments, the feature points in each segment forming a feature graph, and the string of time ordered graphs forming the SFG descriptor. The length of each segment is set to be 20 frames. Next we find the pairwise correspondences between each of the feature collections from the query video with those of dataset videos using the spectral solution in Section 2.2.2.1. We finally perform the DTW matching across the entire query and dataset SFGs (composed of time ordered feature-graphs) based on the local match scores calculated.

The results from this experiment involving query-based activity video retrieval are shown in Fig. 2.12. For each activity class, we chose 3 random videos from the samples of that class to be the query. The results reported here are obtained by averaging across the 3 test cases. Recognition on activities like vehicle turning and shaking hands performed especially well since they continue for longer time periods and hence generate better feature points. On the other hand, activities such as "pointing" happen in a short amount of time and are thus more difficult to recognize. We found that the recognition results obtained based on a single sample video generate higher false positive rates. This is justifiable due the fact that in a single query-based retrieval framework, there is no statistically reliable way to set the acceptance threshold.

We also studied variation in recognition performance of our method with change in query videos. The standard deviation in the scores for different query-videos is marked in Fig. 2.12 (a). In line with our previous argument, short-duration activities such as "pointing" had higher variability. The activity "hug" was confused with background clutter or actors crossing each other.

We show some results of activity retrieval on UCR VideoWeb using one query video in Fig. 2.12 (b). The query videos are shown on the left and the other three columns show the top three best matches. The bounding boxes of the sub-graphs that best match the feature graphs of query video are also shown. This demonstrates the capability of our system in locating the activities of interest in the spatial-temporal video volumes.

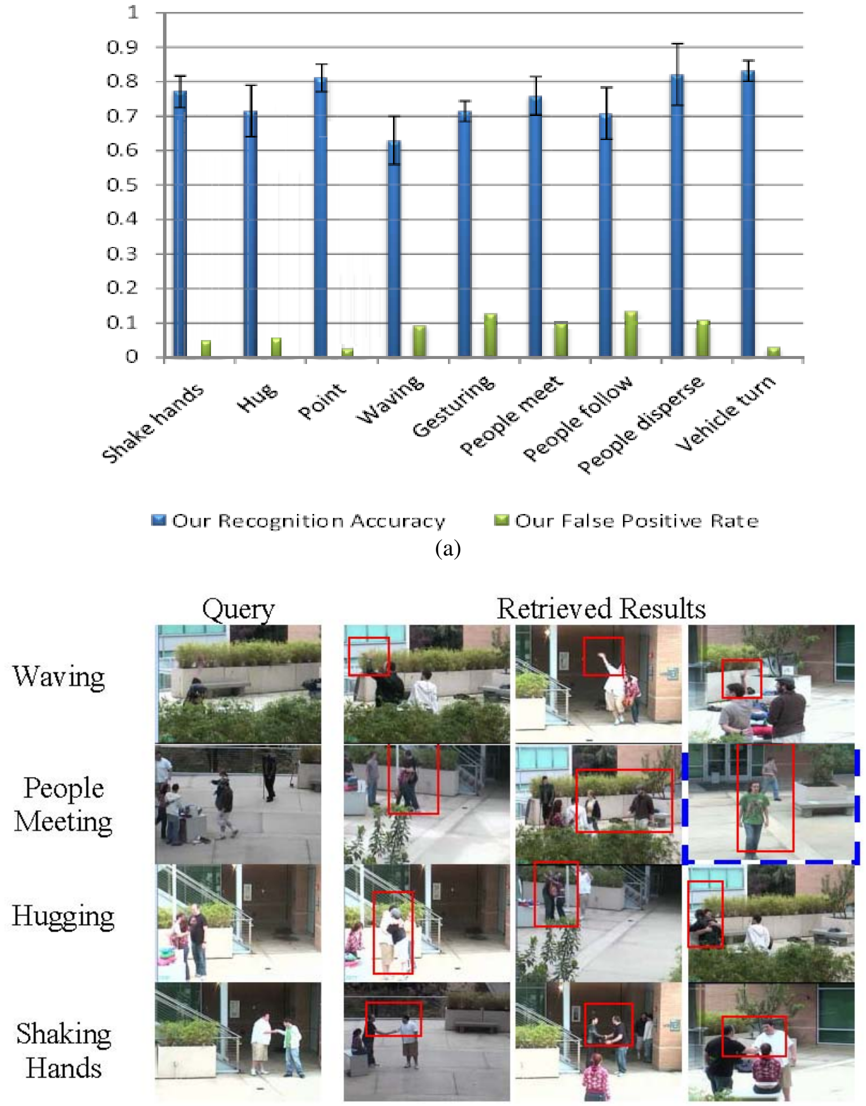
Finally, we test our system on activity retrieval using one query video on UT Interaction dataset. Some results are shown in Fig. 2.13. The query videos are shown on the left and the other three columns show the top three best matches.

2.5.3 Track-Based SFG Results

In this set of experiments, we work on VIRAT dataset (parking lot 04) to evaluate the performance of our track-based SFG scheme. Activities whose motion pattern can be determined by the underlying tracks are of interest here. The activities in interest include 25 single vehicle activities (6 vehicle-backup, 13 vehicle-turn, and 2 vehicle u-turn), 9 people interactions (5 people dispersion, 2 people walking together, and 2 people gathering), and 29 people-vehicle interactions (15 people approaching vehicle and 14 people leaving vehicle). For object detection and tracking, we applied the methodology we have developed in [106].

2.5.3.1 Performance Comparison

In order to assess the performance of different motion descriptors of single tracks in activity recognition - NDG, NMG and joint NDG-NMG which concatenates NDG and N-MG, we test our system on VIRAT Testing Dataset to classify single vehicle activities whose characteristics only depend on features of individual tracks - vehicle-turn, vehicle-u-turn, and vehicle-backup. There are twenty-five instances of the above activities in total in videos from parking lot 04. Each instance is applied as the query alternatively. We search across the whole dataset for activities of the same type. The results reported are obtained by averaging across all



(a)

(b)

Figure 2.12: (a): Recognition accuracy and false positives on 9 activities from the UCR VideoWeb dataset in a query-based retrieval framework. Standard deviation in performance (accuracy) for different queries is marked on the bars. (b): Retrieval results: The left column depicts the query videos and the other three columns are the best matches on UCR VideoWeb dataset. The bounding boxes of the sub-graphs that best match the feature graphs of the query video are shown. A blue dash box represents an incorrect match.

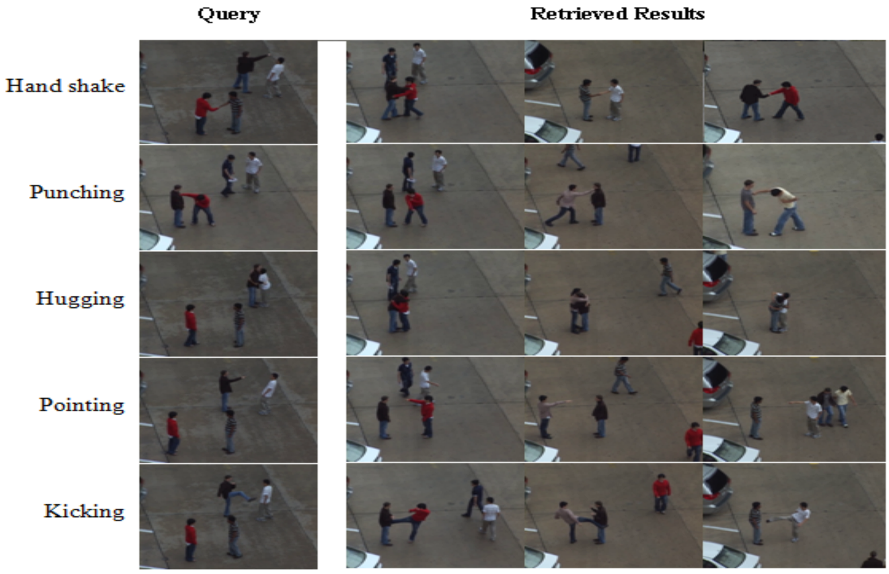


Figure 2.13: Retrieval results: The left column depicts the query videos and the other three columns are the best matches on UT-Interaction dataset.

the queries. From Fig. 2.14 (a) we can conclude that NMG descriptor outperforms other two single track descriptors.

2.5.3.2 Classification-Based results

In this part of experiments, we test the ability of the track-based SFG to classify both single object activities and interactions discussed in section 2.5.3 with and without object identification as shown in Fig. 2.14 (b). The object identifier can tell whether the underlying object associate with a given track is a vehicle or a person. Joint descriptors NDG-NMG and RD-SRD are applied.

2.5.3.3 Query-Based results

In order to demonstrate the effectiveness of our track-based SFG in retrieving activities, we search across videos of VIRAT dataset to find the tracks which match the query tracks. For each trial, underlying tracks of an interesting activity listed in 2.5.3 are the input, the algo-

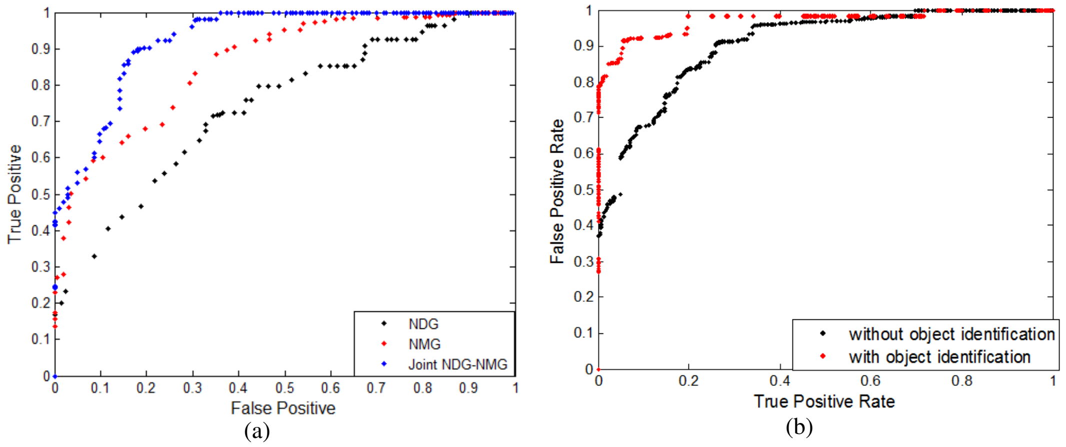
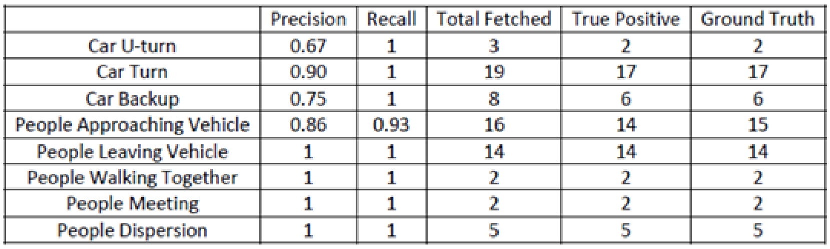


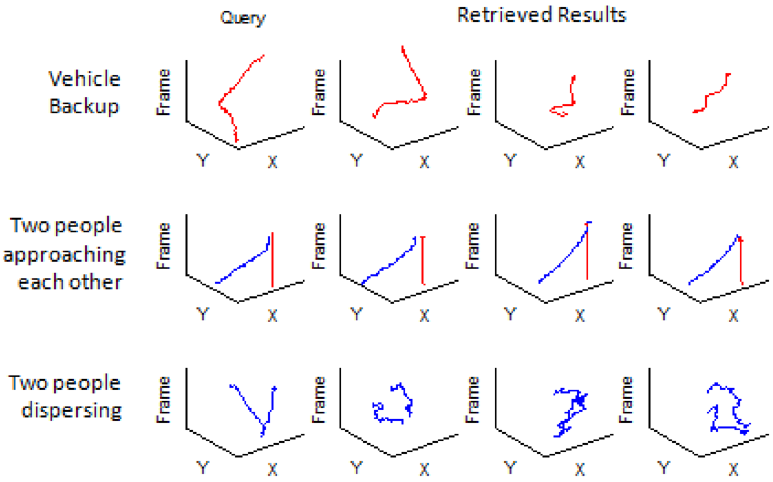
Figure 2.14: (a): Average performance of different kinds of motion descriptors on recognizing activities characterized by individual tracks. 7 car-backup, 16 car-turn, and 2 car-uturn. (Parameter of ROC: distance threshold, the same threshold is used for all kinds of activities, training can be applied to find the best distance threshold for each kind of activity). (b): Average performance of track-based SFG system on VIRAT Testing Dataset. For each run, only one training instance is used for each kind of activity, the rest are treated as testing instances. While the algorithm achieves high recognition performance, the object identifer further enhance the performance. Joint NDG-NMG and joint RD-SRD descriptors are used.

rithm exhaustively searches across the video dataset to find the sets of tracks of the same kind, which matches to the query tracks. The results are shown in Fig. 2.15 (a).



	Precision	Recall	Total Fetched	True Positive	Ground Truth
Car U-turn	0.67	1	3	2	2
Car Turn	0.90	1	19	17	17
Car Backup	0.75	1	8	6	6
People Approaching Vehicle	0.86	0.93	16	14	15
People Leaving Vehicle	1	1	14	14	14
People Walking Together	1	1	2	2	2
People Meeting	1	1	2	2	2
People Dispersion	1	1	5	5	5

(a)



(b)

Figure 2.15: (a): Average performance of track-based SFG system on VIRAT Testing Dataset with object identifier. Only one query instance is used for each query. (b): Examples of query results on VIRAT testing Dataset. The left column depicts the query tracks involved in the targeted activity and the other three columns are the best matches on part of VIRAT testing dataset.

Finally, we show samples of retrieved tracks in Fig. 2.15 (b). This demonstrates the capability of our track-based SFG system in locating the activities of interest in the spatial-temporal video volumes.

2.5.4 Adaptive Feature Selection

In this subsection, we implement the adaptive feature selection scheme on VIRAT Dataset, and compare the result with schemes without feature selection. Encouraging results are shown, demonstrating the efficacy of our adaptive feature selection to recognize complex activities with increased recognition accuracy and reduced computation complexity.

For the entire test video, we first compute the low resolution motion attributes. These attributes are used to detect the location of activities and to decide the mode of each activity. Whenever an activity is detected, the optimum feature type is selected based on the activity mode and a SFG is constructed on these features. The developed SFG is matched to the training SFGs using a voting scheme. Fig. 2.16 shows the switching scheme and recognized activities for one video.

Experimental analysis shows that a few simple heuristics can improve the performance of the method further. One is related to identifying regions where the track-based features do not perform very well, in spite of being chosen by the switching scheme. For this reason, our system identifies when the track-based recognition has low confidence and switches to the STIP-based mode. The track-based results are considered as unreliable when the similarity scores between the testing instance and *all* the training instances are low. It is based on the fact that STIP-based features can recognize both global and local activities, but track-based SFGs can recognize only global activities. A second case arises at the beginning and end of track segments. Experience suggests that local activities usually happen in these regions. Therefore, to minimize the chance of missing a local activity, we analyze the beginning and end of track segments that are not already identified by the switching module to detect if there are any local activities happening there.

Table 2.1 gives the recognition results for each type of activity using different types of features. From the results, we can see that features of different resolution can only recognize certain types of activities. Track-based low resolution features work better at recognizing global activities while STIP features work better at recognizing local activities. The proposed adaptive feature selection improves the recognition accuracy while reducing the overall computation complexity.

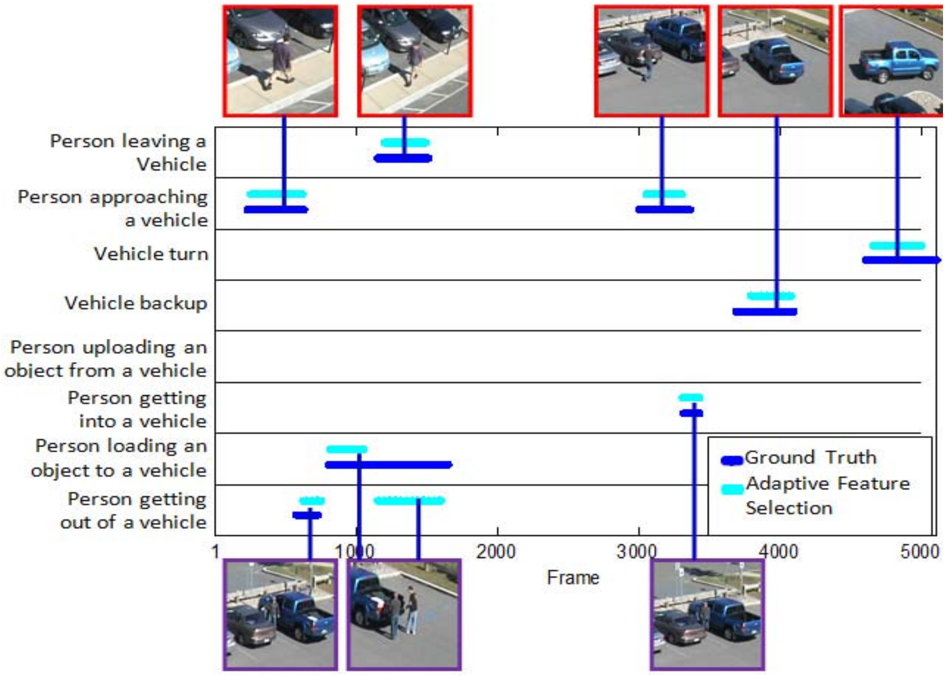
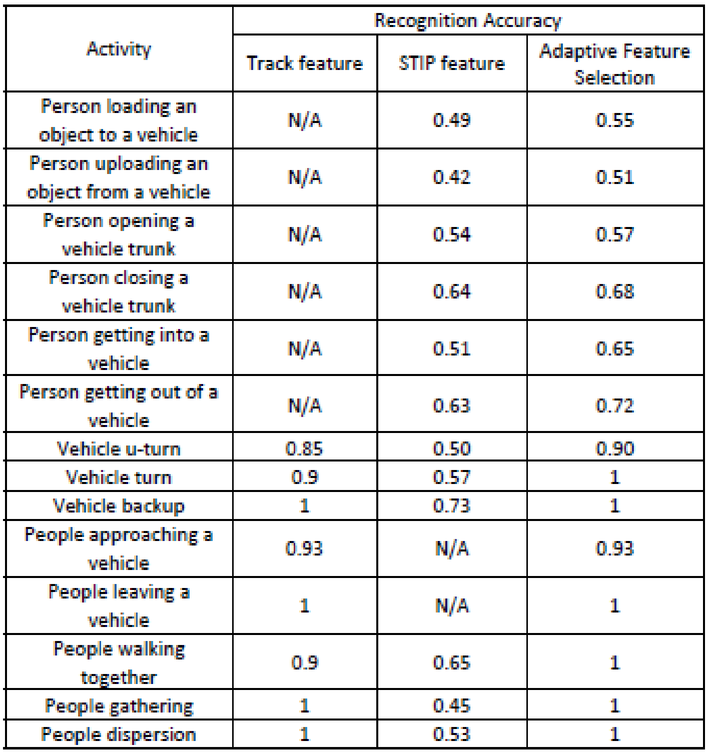
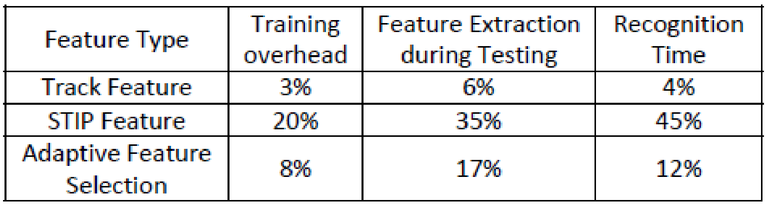


Figure 2.16: Switching results on an example video sequence are shown. One sample image for each activity is shown. Each cyan bar in the figure indicates the recognition result from the adaptive feature selection and compares it to the ground truth (blue bar). The length of the bar indicates the duration of the recognized activity. Red bounding box indicates track features are selected, and purple indicates STIP features are selected for the SFG model in adaptive feature selection. The results show that the system is able to automatically switch between different features.



Activity	Recognition Accuracy		
	Track feature	STIP feature	Adaptive Feature Selection
Person loading an object to a vehicle	N/A	0.49	0.55
Person uploading an object from a vehicle	N/A	0.42	0.51
Person opening a vehicle trunk	N/A	0.54	0.57
Person closing a vehicle trunk	N/A	0.64	0.68
Person getting into a vehicle	N/A	0.51	0.65
Person getting out of a vehicle	N/A	0.63	0.72
Vehicle u-turn	0.85	0.50	0.90
Vehicle turn	0.9	0.57	1
Vehicle backup	1	0.73	1
People approaching a vehicle	0.93	N/A	0.93
People leaving a vehicle	1	N/A	1
People walking together	0.9	0.65	1
People gathering	1	0.45	1
People dispersion	1	0.53	1

Table 2.1: Results of Adaptive Feature Selction. (Recognition accuracy on the VIRAT dataset by using fixed types of feature and adaptive feature selection. N/A means that the activity cannot be recognized by the corresponding feature)



Feature Type	Training overhead	Feature Extraction during Testing	Recognition Time
Track Feature	3%	6%	4%
STIP Feature	20%	35%	45%
Adaptive Feature Selection	8%	17%	12%

Table 2.2: Comparison of computation complexity. (Note that the computation time is given as as the approximate percentage of total computation time using STIP features only. Training overhead includes the time used to construct the training SFGs from the labeled and segmented video clips for track-based and stip-based algorithm, plus the attribute space construction time for adaptive feature selection algorithm)

Computation Complexity Table 2.2 shows the computation time of the whole activity recognition process including the training process. As discussed before, algorithms based on high resolution features are often time consuming. In our algorithm, the most time-consuming part is the graph matching. Assuming the number of nodes of the two graphs to be matched is n_Q and n_P, computational complexity of the graph matching in [65] is $O((n_Q n_P)^{\frac{2}{3}} + (max(n_Q, n_P) - \frac{1}{2}) \cdot min^2(n_Q, n_P))$ [65]. For a feature graph of the same time interval, the number of local features is of the order of tens of times the number of global features. Over a long period of time, this difference in computation can be large.

2.6 Conclusion

In this work, we argued that spatio-temporal relationships are critical to discriminate real-world activities. We proposed a feature model based on string representation of the video which respects the spatio-temporal dynamics of the complex activities. In order to quantize the similarity of two feature graphs, we leveraged a spectral matching technique to find correspondences between them. Finally, the string formed by the time-ordered set of local feature collections was matched with other strings in a dynamic programming framework to obtain the matching score. This matching score was used to classify a test video as being similar or nonsimilar to the template video. We show how the SFG can be constructed for high-resolution STIP features and low-resolution track features. To accelerate the matching process while enhancing the recognition accuracy, the proposed SFG algorithm is integrated into a scheme of adaptive feature selection which automatically chooses features for the recognition task based on the states of activities. Our experiments demonstrated the effectiveness of our approaches to successfully recognize and localize complex activities even with multiple interacting actors.

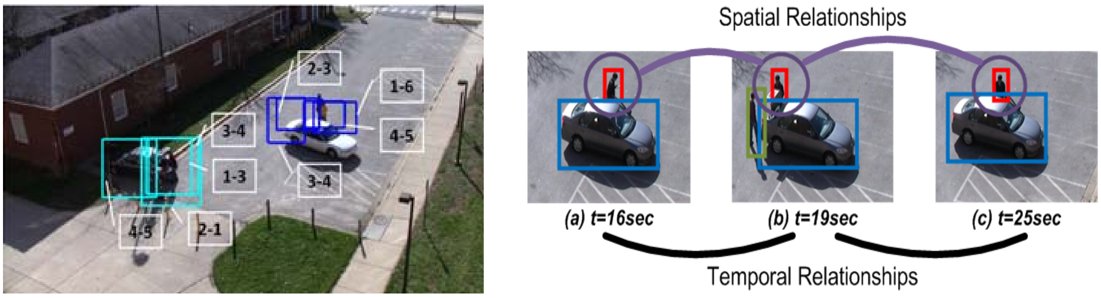
Chapter 3

Higher-Level Context Modeling - graphical models

3.1 Introduction

It has been demonstrated in [80] that context is significant in human visual systems. As there is no formal definition of context in computer vision, we consider all the detected objects and motion regions as providing contextual information about each other. Activities in natural scenes rarely happen independently as shown in Fig. 4.9(i). The spatial layout of activities and their sequential patterns provide useful cues for their understanding.

Consider the activities that happen in the same spatio-temporal region in Fig. 4.9: the existence of the nearby car gives information about what the person (bounded by red circle) is doing, and the relative position of the person of interest and the car says that activities (b) and (c) are very different from activity (a). Moreover, just focusing on the person, it may be hard to tell what the person is doing in (b) and (c) - "opening vehicle trunk" or "closing vehicle trunk". If we knew that these activities occurred around the same vehicle along time, it would be immediately



(i) (ii)

Figure 3.1: (i) Example images from a wide area video (interesting activities happening within about 2 minutes are shown). Activities in the same color in each video happen in the same local spatio-temporal region. Activity classes are listed in Fig. 1 in the supplementary material. (a). For the indices, the first index denotes the temporal order of the activity in the region, while the second number denotes the activity class, e.g., 1-6 means the activity belongs to class 6 and is the first activity that happens in this video volume. (ii) Example of context in activity recognition. A person of interest is located by red bounding box, surrounding objects are located by bounding boxes of other colors, and the circles in purple indicate the motion regions of the activities.

clear that in (b) the person is opening the vehicle trunk and in (c) the person is closing the vehicle trunk. This example shows the importance of spatial and temporal relationships for activity recognition.

3.1.1 Overview of the Framework

Many existing works on activity recognition assume that, the temporal locations of the activities are known [2, 78]. In practice, activity-based analysis of videos should involve reasoning about motion regions, objects involved in these motion regions, and spatio-temporal relationships between the motion regions. We focus on the problem of detecting activities of interest in *continuous* videos without prior information about the locations of the activities. In other words, our goal is to locate and label each activity of interest in videos. The main challenge is to develop a representation of the continuous video that respects the spatio-temporal relationships of the activities. To achieve this goal, we build upon existing well-known feature descriptors and spatio-temporal context representations that, when combined together, provide a powerful framework to model activities in continuous videos.

Given a continuous video, background substraction [143] is used to locate the moving objects. Moving persons are identified, and local trajectories of moving persons are generated (any existing tracking methods like [106] can be used). Spatio-temporal Interest Point (STIP) features [61] are generated only for these motion regions (note that any other local motion features can be used). Thus, STIPs generated by noise, such as slight tree shaking, camera jitter and motion of shadows, are avoided. Each motion region is segmented into action segments, which can be obtained by temporally dividing the motion regions into spatio-temporal volumes with a fixed length in time, or by using any motion segmentation algorithm such as nonlinear dynamic model (NDM) based approach with STIP histograms as the model observation as in [14].

These action segments are merged into candidate activities using preliminary activity detector that explores only motion features. Struct-SVM is modified for re-labeling these detected activities, integrating various context within and between the activities. However, this approach separates the activity localization and labeling, which may lead to loss in recognition accuracy due to ignoring the interaction between the two. An activity can be considered as a union of action segments or actions that are neighbors to each other closely in space and time. We provide an integrated framework that conducts multiple stages of video analysis, s-tarting with motion localization. The detected motion regions are divided into action segments, which are considered as the elements of activities. The goal then is to generate smoothed activity labels, which are optimum in a global sense, for the action segments; and thus obtaining semantically meaningful activity regions and corresponding activity labels. For this purpose, based on the modified Struct-SVM [138], we develop graphical models for the smoothly labeling of action segments, integrating various motion and context features. Three graphical models are described and compared in this chapter - structural model [140], higher-order CRF [142] and hierarchical-CRF [141]. The proposed models aim to smoothly label the action segments, resulting in meaningful activity regions, each is expected to contain one activity.

3.1.2 Contributions of Present Work

The main contribution of this chapter is three-fold.

(i) The parameter estimation of the modified Struct-SVM is formulated as a large-margin problem, which tries to maximize the margins around the decision plane which separates the negative and positive instances. We show how this problem can be modified to be an unconstrained convex optimization problem. Next, the modified bundle method in [112] is used to solve the optimization problem. This method iteratively searches for the increasingly tight upper and lower bounds of the objective function till convergence is reached.

(ii) We combine low-level motion segmentation with high-level activity model under one framework. With the detected individual action segments as the elements of activities, we design graphical models that jointly model the related activities in the scene.

(iii) We propose a weakly supervised approach that utilizes context within and between actions and activities that provide helpful cues for activity recognition. The proposed models integrates motion and various context features within and between actions and activities into a unified model. The proposed models can localize and label activities in continuous videos simultaneously, in the presence of multiple actors in the scene interacting with each other or acting independently.

(iv) With a task-oriented discriminative approach, the model learning problems are formulated as a max-margin problem and is solved based on the modified bundle method [113]. Greedy search algorithms, that are specifically designed for the proposed graphical models, are developed to infer the underlying activities efficiently without obvious loss in accuracy.

3.2 Struct-SVM

In this section, we modified the Struct-SVM [118] for activity recognition, which integrates motion features with various context features within and across activities to jointly model related activities in videos [118]. We show how to learn the model parameters via an unconstrained convex optimization problem and how to predict the correct labels for a testing instance consisting of multiple activities. To locate the activity regions, motion regions are first divided into action segments with fixed temporal length.

Sliding windows of different sizes are applied to the motion regions. In the experiment, Bag-of-words combined with multi-class support vector machine (BOW+SVM) [75] are used to label each window as one of the normal activity classes. Then, weighted average smoothing is applied to obtain the label of each temporal bin. Objects that occur in the images that overlap with motion regions are detected. These image features will be used for the development of the context features within activities.

3.2.1 Feature Descriptors

Assuming there are $M+1$ classes of activities in the scene, including a background class with label 0 and M classes of interest with labels $1,...,M$. We first define the concepts we use for the feature development. An activity is a 3D region consisting of one or multiple consecutive action segments. An agent is the underlying moving person along a trajectory. Motion region at frame n is a circular region surrounding the moving objects of interest in the n^{th} frame of the activity. Activity region is the smallest rectangular region that encapsulates the motion regions over all frames of the activity. Based on this, we can now encode motion and context information into feature descriptors.

Intra-Activity Motion Feature Descriptor Features of an activity that encode the motion information extracted from low-level motion features such as STIP features are defined as intra-activity motion features. We train a multi-SVM [12] classifier upon the detected action segments to generate the normalized confidence scores $s_{i,0},...,s_{i,M}$ of classifying the action segment i as activity classes $0,1,...,M$, such that $\sum_{j=0}^{M} s_{i,j} = 1$. We call the classifier as the baseline classifier. In general, any kind of classifier and low-level motion features can be used here. Given an activity, $x = [\max_{i\in\aleph} s_{i,0},...,\max_{i\in\aleph} s_{i,M}]$ is developed as the intra-activity motion feature descriptor, where $\aleph$ is a list of action segments in the activity.

Intra-Activity Context Feature Descriptor Features that capture the relationships between the agents, as well as other interacting objects, are defined as intra-activity context features. Objects including vehicles, opening/closing entrance/exit doors of facilities, boxes and bags that overlap with the motion regions, are detected. Persons and vehicles are detected using the publicly available software [26]. Opening/closing entrance/exit doors of facilities, boxes and bags are detected using method in [20] with Histogram of Gradient as the low-level feature and binary linear-SVM as the classifier. These high-level image features will be used for the development of the context features within activities.

We define a set G of attributes related to the scene and the involved objects in activities of interest. G consists of N_G subsets of attributes that are exclusively related to certain image-level features. The attribute subsets used for activity recognition are dataset/task specific, they are described in details in the section of experiments.

For a given activity, the above attributes are determined from image-level detection results. For frame n of an activity, we obtain $g_i(n) = I(G_i(n))$, where $I(\cdot)$ is the indicator function. $g_i(n)$ is then normalized so that its elements sum to 1. Fig. 3.11 shows an example of $g_i(n)$.

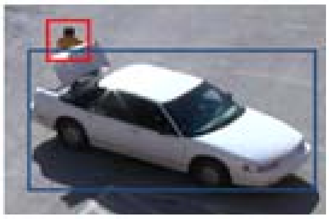


Figure 3.2: The image shows one frame of 'person unloading an object from a vehicle'. In the image, moving objects are the person and the vehicle, and the person is in the rear of the vehicle. So, for this frame, $g_1(n) = [1 \quad 0 \quad 0]$ and $g_2(n) = [0 \quad 1 \quad 0]$, where n is the frame number of this image in the activity.

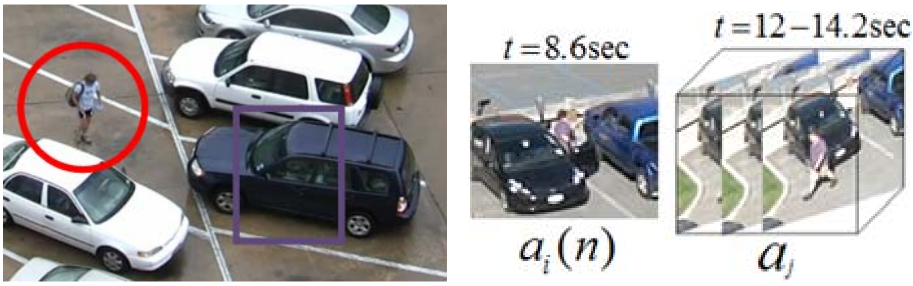
Let $g_i = \frac{1}{N_f}\sum_{n=1}^{N_f} g_i(n)$, where N_f is the total number of frames associated with the activity. The $\sum_{i=1}^{N_G} n_{G_i}$-bin histogram $g = \frac{1}{N_G}[g_1 \oplus \cdots \oplus g_{N_G}]$ is the intra-activity context feature vector of the activity, where $\oplus$ denotes the vector concatenation operator.

Inter-Activity Context Feature Descriptor Features that capture the relative spatial and temporal relationships of activities are defined as inter-activity context feature. Define the scaled distance between activity a_i and a_j at the n^{th} frame of a_i as

$$r_s(a_i(n), a_j) = \frac{d(O_{a_i}(n), O_{a_j})}{R_{a_i}(n) + R_{a_j}}, \tag{3.1}$$

where $O_{a_i}(n)$ and $R_{a_i}(n)$ denote the center and radius of the motion region of activity a_i at its n^{th} frame and O_{a_j} and R_{a_j} denote the center and radius of the activity region of activity a_j. $d(\cdot)$ denotes the Euclidean distance. Then, the spatial relationship of a_i and a_j at the n^{th} frame is modeled by $sc_{ij}(n) = bin(r_s(a_i(n), a_j))$ as in Fig. 3.3 (a). The normalized histogram $sc_{a_i,a_j} = \frac{1}{N_f}\sum_{n=1}^{N_f} sc_{ij}(n)$ is the inter-activity spatial feature of activity a_i and a_j.

Temporal context is defined by the following temporal relationships: n^{th} frame of a_i is before a_j, n^{th} frame of a_i is during a_j, and n^{th} frame of a_i is after a_j. $tc_{ij}(n)$ is the temporal relationship of a_i and a_j at the n^{th} frame of a_i as shown in Fig. 3.3 (b). The normalized histogram $tc = \frac{1}{N_f}\sum_{n=1}^{N_f} tc_{ij}(n)$ is the inter-activity temporal context feature of activity a_i with respect to activity a_j.



(a)

(b)

Figure 3.3: (a) The image shows an example of inter-activity spatial relationship. The red circle indicates the motion region of a_i at this frame while the purple rectangle indicates the activity region of a_j. Assume *SC* is defined by quantizing and grouping $r_s(n)$ into three bins: $r_s(n) \leq 0.5$ (a_i and a_j is at the same spatial position at the n^{th} frame of a_i), $0.5 < r_s(n) < 1.5$ (a_i is near a_j at the n^{th} frame of a_i), and $r_s(n) \geq 1.5$ (a_i is far away from a_j at the n^{th} frame of a_i). In the image, $r_s(n) > 1.5$, so, $sc_{ij}(n) = [0 \quad 0 \quad 1]$. (b) The image shows one example of inter-activity temporal relationship. The n^{th} frame of a_i occurs before a_j. So, $tc_{ij}(n) = [1 \quad 0 \quad 0]$.

3.2.2 Model Development

Suppose we are interested in M activity classes. Activity set $a = \{a_i : i = 1, ..., N\}$ is associated with a label vector $y = \{y_i : i = 1, ..., N\}$, where $y_i \in \{1, ..., M\}$ is the label of a_i. We model the activity set by the combination of motion features of individual activities and various context features discussed above. A potential function that measures the compatibility between features of a and label y is defined as $F(a, y)$:

$$\begin{aligned} \mathbf{F}(a,y) = & \sum_{i=1}^{N} \omega_{x,y_i}^T x_i + \sum_{i=1}^{N} \omega_{g,y_i}^T g_i \\ & + \sum_{i,j=1,i\neq j}^{N} \omega_{sc,(y_i,y_j)}^T sc_{ij} + \sum_{i,j=1,i\neq j}^{N} \omega_{tc,(y_i,y_j)}^T tc_{ij}, \end{aligned} \tag{3.2}$$

where $x_i \in R^{D_x}$ and $g_i \in R^{D_g}$ are the motion feature and intra-activity context feature of instance a_i, D_x and D_g are the dimension of x_i and g_i respectively. $\omega_{x,y_i} \in R^{D_x}$ and $\omega_{g,y_i} \in R^{D_g}$ are the weights that capture the valid motion and intra-activity context patterns of activity class y_i. $sc_{ij} \in R^{D_{sc}}$ and $tc_{ij} \in R^{D_{tc}}$ are the inter-activity context features associated a_i and a_j. D_{sc} and D_{tc} are the dimension of sc_{ij} and tc_{ij} respectively. $\omega_{sc,(y_i,y_j)} \in R^{D_{sc}}$ and $\omega_{tc,(y_i,y_j)} \in R^{D_{tc}}$ are the weights that capture the valid spatial and temporal relationships of activity classes y_i and y_j. In general, dimensions of the same kind of feature can be different for each activity class/class pairs.

In order to form a linear function with a single parameter, we rewrite (4.2) as:

$$\begin{aligned}\mathbf{F}(a,y) = \omega_x^T \sum_{i=1}^{N} \varphi(x_i,y_i) + \omega_g^T \sum_{i=1}^{N} \vartheta(g_i,y_i) \\ +\omega_{sc}^T \sum_{i,j=1,i\neq j}^{N} \psi(sc_{ij},y_i,y_j) + \omega_{tc}^T \sum_{i,j=1,i\neq j}^{N} \phi(tc_{ij},y_i,y_j),\end{aligned} \tag{3.3}$$

where ω_x, ω_g, ω_{sc} and ω_{tc} are weight vectors defined as

$$\begin{aligned}\omega_x &= \begin{bmatrix}\omega_{x,1}^T & \omega_{x,2}^T & \cdots & \omega_{x,M}^T\end{bmatrix}^T, \\ \omega_g &= \begin{bmatrix}\omega_{g,1}^T & \omega_{g,2}^T & \cdots & \omega_{g,M}^T\end{bmatrix}^T, \\ \omega_{sc} &= \begin{bmatrix}\omega_{sc,(1,1)}^T & \cdots & \omega_{sc,(1,M)}^T & \cdots & \omega_{sc,(M,M)}^T\end{bmatrix}^T, \\ \omega_{tc} &= \begin{bmatrix}\omega_{tc,(1,1)}^T & \cdots & \omega_{tc,(1,M)}^T & \cdots & \omega_{tc,(M,M)}^T\end{bmatrix}^T,\end{aligned}$$

and $\varphi(x_i,y_i)$ and $\vartheta(g_i,y_i)$ have non-zero entries at the position corresponding to class index y_i. $\psi(sc_{ij},y_i,y_j)$ and $\phi(st_{ij},y_i,y_j)$ have none-zero entries at the position corresponding to class pair (y_i,y_j).

Define the joint weight vector ω and joint feature vector $\Gamma(a,y)$ as

$$\omega = \begin{bmatrix}\omega_x \\ \omega_g \\ \omega_{sc} \\ \omega_{st}\end{bmatrix}, \Gamma(a,y) = \begin{bmatrix}\sum_i \varphi(x_i,y_i) \\ \sum_i \vartheta(g_i,y_i) \\ \sum_{i,j,i\neq j} \psi(sc_{ij},y_i,y_j) \\ \sum_{i,j,i\neq j} \phi(tc_{ij},y_i,y_j)\end{bmatrix},$$

where $i, j = 1, ..., N$. Then, the optimum label y^{opt} of x is obtained as

$$y^{opt} = \arg\max_{y}(\omega^T \Gamma(a,y)). \tag{3.4}$$

3.2.3 Model Learning and Inference

3.2.3.1 Learning Model Parameters

We will now describe our method for learning the model parameters from training sets. Suppose there are P collections of activities in the training videos. Let the training set be $(A_T, Y_T) = (a_T(1), y_T(1)), ..., (a_T(P), y_T(P))$, where each $a_T(i)$ is an activity set and $y_T(i)$is its label vector. Suppose there are $N_T(i)$ elements in $a_T(i)$. We use the following loss function to measure the correctness of labeling instance $a_T(i)$ with the candidate label $\widehat{y_T(i)}$:

$$\Delta\left(y_T(i), \widehat{y_T(i)}\right) = \sum_{j=1}^{N_T(i)} \Delta\left(y_T(i,j), \widehat{y_T(i,j)}\right),$$

$$where \quad \Delta\left(y_T(i,j), \widehat{y_T(i,j)}\right) = \begin{cases} 1 & y_T(i,j) \neq \widehat{y_T(i,j)} \\ 0 & y_T(i,j) = \widehat{y_T(i,j)} \end{cases}.$$

The model learning problem is formulated as an unconstrained convex optimization problem (derivation is shown in Sec. III in the supplementary material):

$$\omega^* = \arg\min_{\omega} f(\omega) = \arg\min_{\omega} \frac{1}{2}\omega^T\omega + \Lambda(\omega),$$
$$where \quad \Lambda(\omega) = C\sum_{i=1}^{P} \max(0, \Omega_\omega(i)),$$
$$\Omega_\omega(i) = \max_{\widehat{y_T(i)}} \left(\Delta\left(y_T(i), \widehat{y_T(i)}\right) + \omega^T\left(\Gamma\left(a_T(i), \widehat{y_T(i)}\right) - \Gamma(a_T(i), y_T(i))\right)\right). \tag{3.5}$$

Optimization Algorithm The problem in (3.5) can be solved by the modified bundle method in [112]. It iteratively searches for the increasingly tight quadratic upper and lower cutting planes of the objective function until the gap between the two bounds reaches a predefined threshold. A cutting plane of a convex function is defined by its first-order Taylor approximation and can be calculated as [112]

$$g_{\omega} = \omega^T \partial_{\omega} \Lambda(\omega) + b_{\omega}, \quad where \quad b_{\omega} = \Lambda(\omega) - \omega^T \partial_{\omega} \Lambda(\omega).$$

The algorithm is effective because of its very high convergence rate [112]. The bundle method specified for problem (3.5) is summarized in Algorithm 7:

Algorithm 2 Learning the Model Parameter Through Bundle Method

```
  Input:   $S = ((a_T(1), y_T(1)), \ldots, (a_T(P), y_T(P))), C, \varepsilon$
 Output:   Optimum model parameter $\omega$

Initialize $\omega$ as $\omega_0$ using empirical values, $\mathscr{G}$(cutting plane set) $\leftarrow \emptyset$.
for $t = 0$ to $\infty$ do
    for $i = 1, ..., P$ do
        find the most violated label vector for each training instance, if any, using $\omega_t$ (the value of $\omega$ at the $t^{th}$ iteration);
    end for
    find the cutting plane $g_{\omega_t}$ of $\Lambda(\omega)$ at $\omega_t$:
        $g_{\omega_t} = \omega^T \partial_{\omega} \Lambda(\omega_t) + b_{\omega_t}, \quad where \quad b_{\omega_t} = \Lambda(\omega_t) - {\omega_t}^T \partial_{\omega} \Lambda(\omega_t)$.
    $\mathscr{G} \leftarrow \mathscr{G} \cup g_{\omega_t}(\omega)$;
    update $\omega$:
        $\omega_{t+1} = \arg\min_{\omega} f_{\omega_t}(\omega),$
        $f_{t+1}(\omega) = f_{\omega_t}(\omega_{t+1}),$
        $where \quad f_{\omega_t}(\omega) = \frac{1}{2}\omega^T \omega + max(0, max_{j=1,\ldots,t} g_{\omega_j}(\omega)).$
    $gap_{t+1} = \min_{t' \le t} f_{\omega_{t'+1}}(\omega_{t'+1}) - f_{\omega_t}(\omega_{t+1})$;
    if $gap_{t+1} \le \varepsilon$, then return $\omega_{t+1}$;
end for
```

Efficient Implementation We call the label vector $\widehat{y_T(i)}$ of the i^{th} instance that maximizes $\Omega_{\omega_t}(i)$ the most-violated label of the i^{th} instance in the $(t+1)^{th}$ iteration, if $\Omega_{\omega_t}(i) > 0$. If $\Omega_{\omega_t}(i) \le 0$ for all $\widehat{y_T(i)}$, the constraints on i^{th} instance will not be violated in the $(t+1)^{th}$

iteration. In each iteration, we need to check and find the most-violated label for each instance. Finding the most violated label is NP hard (we need to enumerate all the possible label vectors). A greedy forward search is proposed in [22] to balance the computation efficiency and algorithm accuracy.

The computation of cutting planes requires knowledge of the most violated labels for all the training instances in each iteration. When the number of training instances is large, finding violated label for each instances in each iteration is inefficient. Like other online optimization techniques such as large scale SVM, we try to shrink the working space in order to improve efficiency. During the learning process, it is often revealed early that the constraints in (3.5) of certain instances are unlikely to be violated. Let us consider the history of violated instances over the last k iterations. If the constraints of an instance are not violated at each of the last k iterations, it is likely that they will not be violated before the optimum solution is reached. Considering that cutting planes do not depend on these instances in the subsequent iterations. Such instances are excluded from the working space and the solution space is stored. Since this heuristic can fail, the constraints for the eliminated instances are checked after convergence. If necessary, the optimization process is restarted from the solution stored previously. Also, to ensure the algorithm does not restart frequently, we maintain a minimum number P_{min} of training instances in the working space.

3.2.3.2 Inference

With the learned model parameter vector ω, we now describe how to identify the optimum label vector y_{test} for an input instance a_{test}. Suppose the testing instance has n activity-based segments $a_{test} = [a_{test}(1),...,a_{test}(n)]$. The greedy forward search [22] is used to find the optimum labels of the targeted activities. We greedily instantiate the segment that, when labeled as an activity class of interest, can increase the value of compatibility function F by the largest amount. The algorithm stops when all the regions are labeled or labeling any other segments decreases the value of compatibility function F. Algorithm 6 gives the overview of the inference process. While this greedy search algorithm cannot guarantee a globally optimum solution, in practice it works well to find good solutions as demonstrated in the experimental results. The papers [64][22] give theoretical explanation of the effectiveness of the method in finding the optimum solution to the problems of the kind.

Algorithm 3 Greedy Search Algorithm

```
Input:   Testing instance
Output:  Interested activities A and label vector Y

1. initialize $(A,Y) \leftarrow \{\emptyset,\emptyset\}$ and $F = 0$.

2. repeat
     $\Delta F(a_i,y_i)_{(a_i)\nsubseteq(A)} = F((A,Y)\cup(a_i,y_i)) - F((A,Y))$;
     $(a_i,y_i)^{opt} = \arg\max_{(a_i)\nsubseteq(A)} \Delta F(a_i,y_i)$;
     $(A,Y) \leftarrow (A,Y)\cup(a_i,y_i)^{opt}$;

3. end if all activities are labeled.
```

3.3 Structural Model

The above Struct-SVM-based approach separate activity segmentation and labeling. In [140], we proposed a structural model to explicitly model the durations, motion, intra-activity context and the spatio-temporal relationships between the activities. In this section, we described the structural model in [140]. With the obtained action segments in motion regions of interest, we learn a structural model that merges these segments into activities and generates the optimum activity labels for them.

Fig. 3.4 shows the framework of our approach. Given a video, we detect the motion regions using background subtraction. The segmentation algorithm aims to divide a continuous motion region into action segments, whose motion pattern is consistent and is different from its

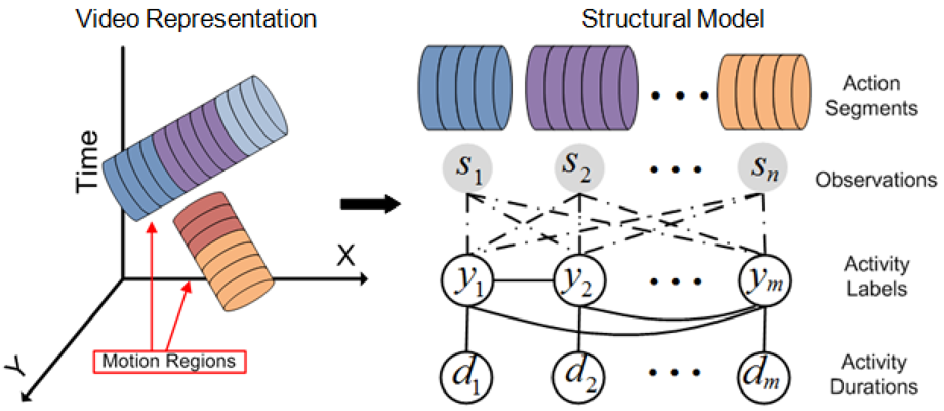


Figure 3.4: The left graph shows the video representation of an activity set with n motion segments and m activities (for testing, m needs to be determined by inference of the learned structural model). The right graph shows the graphical representation of our model. The gray nodes in the graph are the feature observations and the white nodes are the model variables. The dashed lines indicate that the connections between activity labels and the observations of action segments are not fixed, i.e., the structure of connections is different for different activity sets.

adjacent segments. The main challenge now is to develop a representation of the continuous video that respects the spatio-temporal relationships of the activities. To achieve this goal, we build upon existing well-known feature descriptors and spatio-temporal context representations that, when combined together, provide a powerful framework to model activities in continuous videos. Action segments that are related to each other in space and time are grouped together into activity sets. For each set, the underlying activities are jointly modeled and recognized by a structural model with the activity durations as the auxiliary variables. For the testing, the action segments, which are considered as the basic elements of activities, are merged together and assigned activity labels by inference on the structural model.

3.3.1 Feature Descriptors

Similar to the feature descriptors developed in section 3.4.1, we develop intra-activity motion feature x and intra-activity context feature g for each activity; inter-activity spatial context feature sc and inter-activity context feature tc for each pair of activities in the activity set under consideration.

3.3.2 Model Development

For an activity set a with n action segments, we assign an auxiliary duration vector $d = [d_1, \cdots, d_m]$ ($\sum_{i=1}^{m} d_i = n$) and a label vector $y = [y_1, \cdots, y_m]$. $y_i \in \{0, ..., M\}$ is the activity label of the i^{th} activity and d_i is its activity duration, for $i = 1, \cdots, m$. Thus, for $a = [a_1, \cdots, a_m]$, a_i is the i^{th} activity in the set. Assume $x_i \in R^{D_x}$ and $g_i \in R^{D_g}$ to be the motion feature and intra-activity context feature of instance a_i, and D_x and D_g to be the dimension of x_i and g_i respectively. $\omega_{d,y_i} \in R^{D_x}$, $\omega_{x,y_i} \in R^{D_x}$ and $\omega_{g,y_i} \in R^{D_g}$ are the weight vectors that capture the valid duration, motion and intra-activity context patterns of activity class y_i. $sc_{ij} \in R^{D_{sc}}$ and $tc_{ij} \in R^{D_{tc}}$ are the inter-activity context features associated with a_i and a_j. D_{sc} and D_{tc} are the dimensions of sc_{ij} and tc_{ij} respectively. $\omega_{sc,y_i,y_j} \in R^{D_{sc}}$ and $\omega_{tc,y_i,y_j} \in R^{D_{tc}}$ are the weight vectors that capture the valid spatial and temporal relationships of activity classes y_i and y_j. In general, dimensions of the same kind of feature can be different for each activity class/class pairs. Four potentials are developed to measure the compatibilities between the assigned variables (y, d) and the observed features of activity set a.

Activity-duration potential measures the compatibility between the activity label y_i and its duration d_i for activity a_i. It is defined as

$$F_d(y_i, d_i) = d_i \omega_{d,y_i}^T I(d_i). \tag{3.6}$$

If d_{max} is the maximum duration of an activity, $I(d_i)$ generates a $d_{max} \times 1$ vector with one for the $(d_i)^{th}$ element and zeros otherwise.

Intra-activity motion potential measures the compatibility between the activity label of a_i and the intra-activity motion feature x_i developed from the associated action segments as

$$F_x(y_i, d_i) = d_i \omega_{x,y_i}^T x_i. \tag{3.7}$$

Intra-activity context potential measures the compatibility between the activity label of a_i and its intra-activity context feature g_i as

$$F_g\left(y_i, d_i\right) = d_i \omega_{g,y_i}^T g_i. \tag{3.8}$$

Inter-activity context potential measures the compatibility between the activity labels of a_i and a_j and their spatial and temporal relationships sc_{ij} and tc_{ij} as

$$F_{sc,tc}\left(y_i, y_j, d_i, d_j\right) = d_i d_j (\omega_{sc,y_i,y_j}^T sc_{ij} + \omega_{tc,y_i,y_j}^T tc_{ij}). \tag{3.9}$$

Combined potential function $F(a,y,d)$ is defined to measure the compatibility between (y,d) of the activity set a and its features:

$$\begin{aligned} F\left(a,y,d\right) = & \sum_{i=1}^{m} F_d\left(y_i, d_i\right) + \sum_{i=1}^{m} F_x\left(y_i, d_i\right) \\ & + \sum_{i=1}^{m} F_g\left(y_i, d_i\right) + \sum_{i,j=1}^{m} F_{sc,tc}\left(y_i, y_j, d_i, d_j\right). \end{aligned} \tag{3.10}$$

The optimum assignment of (y,d) for a maximizes the potential function $F(a,y,d)$.

3.3.3 Model Learning and Inference

3.3.3.1 Learning Model Parameters

We define the weight vector ω as the concatenation of all the weight vectors defined above as

$$\omega = [\omega_d{}^T, \omega_x{}^T, \omega_g{}^T, \omega_{sc}{}^T, \omega_{tc}{}^T]^T, \tag{3.11}$$

where ω_d is obtained by concatenating the w_{d,y_i} for all the $M+1$ activity classes. ω_x, ω_g, ω_{sc} and ω_{tc} are developed similarly. Thus, the potential function $F(a,y,d)$ can be converted into a linear function with a single parameter ω,

$$F(a,y,d) = \omega^T \Gamma(a,y,d), \tag{3.12}$$

where $\Gamma(a,y,d)$, called the joint feature of activity set a, can be easily obtained from (3.10).

Suppose we have P activity sets for training. Let the training set be $(A,Y,H) = (a^1,y^1,d^1),...,(a^P,y^P,d^P)$, where a^i is the activity set, y^i is the label vector and d^i is the auxiliary vector. The loss function for assigning a^i with $(\widehat{y}^i,\widehat{d}^i)$, $\Delta(a^i,\widehat{y}^i,\widehat{d}^i)$, equals the number of action segments that associate with incorrect activity labels (an action segment is mislabeled if over half of the segment is mislabeled). The learning problem can now be written as

$$\begin{aligned}\omega^* = \arg\min_{\omega} \Bigg\{ \frac{1}{2}\omega^T \omega - C\sum_{i=1}^{P} \omega^T \Gamma\left(a^i,y^i,d^i\right) \\ + C\sum_{i=1}^{P} \max_{(\widehat{y}^i,\widehat{d}^i)} \left[\omega^T \Gamma\left(a^i,\widehat{y}^i,\widehat{d}^i\right) + \Delta(a^i,\widehat{y}^i,\widehat{d}^i)\right] \Bigg\},\end{aligned} \tag{3.13}$$

where where C controls the tradeoff between the errors in the training model and margin maximization [8]. The problem in (3.13) can be converted to an unconstrained convex optimization problem [22] and solved by the modified bundle method in [112]. It iteratively searches for the increasingly tight quadratic upper and lower cutting planes of the objective function until the gap between the two bounds reaches a predefined threshold. The algorithm is effective because of its high convergence rate [112]. We set all weights related to background activities to be zeros.

3.3.3.2 Inference

With the learned model parameter vector ω, we now describe how to identify the optimum label vector y_{test} and duration vector d_{test} for an input instance a_{test}. Suppose the testing instance has n action segments. Greedy forward search [22] is used to find the optimum labels and durations of the targeted activities. The potential function F is initialized as 0. We greedily instantiate d_i consecutive segments denoted as a_i that, when labeled as a specific activity class, can increase the weighted value of the compatibility function, F, by the largest amount. The algorithm stops when all the action segments are labeled. Algorithm 6 gives the overview of the inference process. The time complexity of the greedy search is $O(d_{max}Mn^2)$. While this greedy search algorithm cannot guarantee a globally optimum solution, in practice it works well to find good solutions for problems of our kinds [22].

Algorithm 4 Greedy Search Algorithm

```
  Input:            Testing instance with $n$ action segments
  Output:   Interested activities $A$, label vector $Y$ and the duration vector $D$

  1. initialize $(A,Y,D) \leftarrow \{\O,\O,\O\}$ and $F = 0$.

  2. repeat
         $\underset{a_i \not\subseteq A}{\Delta F(a_i,y_i,d_i)} = \frac{F((A,Y,D)\cup(a_i,y_i,d_i))-F(A,Y,D)}{d_i}$;
         $(a_i,y_i,d_i)^{opt} = \arg\max_{a_i \not\subseteq A} \Delta F(a_i,y_i,d_i)$;
         $(A,Y,D) \leftarrow (A,Y,D) \cup (a_i,y_i,d_i)^{opt}$;

  3. end if $\underset{\forall a_i \not\subseteq A}{\Delta F(a_i,y_i,d_i)} < 0$ or $\sum_i {d_i}^{opt} = n$.
```

3.4 Hierarchical-CRF

In [141], we proposed a higher-order CRF as well as a hierarchical-CRF model which represents the related activities in a hidden activity layer interacting with a lower-level action layer. Representing activities as hidden activity variables simplifies the inference problem, by associating each hidden activity with a small set of neighboring action segments, and enables efficient iterative learning and inference algorithms. Rather than modeling only the activity-level context, we also implicitly or explicitly model the contextual relationships between actions, as well as those between action and activity. Specifically, the modeling of more aspects of the activities of interest adds additional feature functions that measure both action and activity variables. Since more information about the activities to be recognized is modeled, the recognition accuracy is improved as demonstrated by the experiments. In this section, we describe the two graphical models.

After obtaining action segments, we perform an initial labeling to group adjacent action segments into semantically meaningful activities using a baseline activity detector. Any existing activity detection method, such as sliding window bag-of-words (BOW) with a support vector machine (SVM) [75] can be used in this step. We call the labeled groups of action segments as the candidate activities. Candidate activities that are related to each other in space and time are grouped together into activity sets. For each set, the underlying activities are jointly modeled and recognized with the proposed two-layer Conditional Random Field model, which models the hierarchical relationship between the action segments and activities. We refer this proposed two-layer Hierarchical-CRF as Hierarchical-CRF in short for simplicity of expression. First, the action layer is modeled as a linear-chain CRF model with the activity labels with the action segments as the random variables. Latent activity variables, which represent the detected activities, are then introduced in the hidden activity layer. Doing so, action-activity consistency and intra-activity potentials, as the higher-order smoothness potentials, can be introduced into the model to smooth the preliminary activity labels in the action layer. Finally, the activity layer variables, whose underlying activities are within the neighborhoods of each other in space and time, are connected to utilize the spatial and temporal relationships between activities. The resulting model is the action-based two-layer Hierarchical-CRF model.

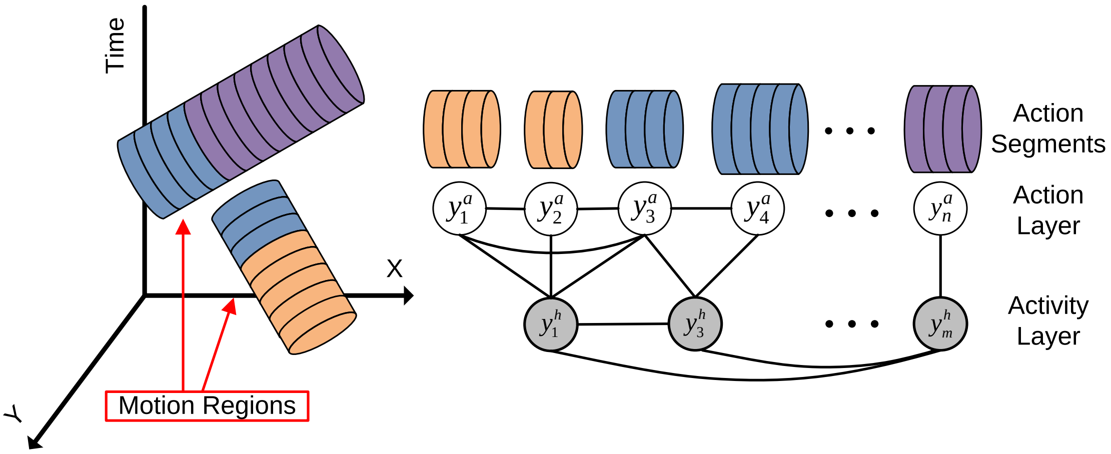


Figure 3.5: The left graph shows the video representation of an activity set with n motion segments and m candidate activities. The right graph shows the graphical representation of our Hierarchical-CRF model. The white nodes are the action variables and the gray nodes in the graph are the hidden activity variables. Note that observations associated with the model variables are not shown for clear representation.

Potentials in and between the action and activity layers are developed to represent the motion and context patterns of individual variables and groups of them in both action and activity levels, as well as action-activity consistency patterns between variables in the two layers. The action-activity potentials upon sets of action nodes and their corresponding activity nodes are introduced between action and activity layers. Such potentials, as smoothness potentials, are used to enforce label consistency of action segments within activity regions while allowing the label inconsistency for certain circumstances. This allows the rectification of the preliminary activity labels of action segments during the inference of the Hierarchical-CRF model according to the motion and context patterns in and between actions and activities.

Fig. 3.5 shows the framework of our approach. Given a video, we detect the motion regions using background subtraction. Then, the segmentation algorithm aims to divide a continuous motion region into action segments, whose motion pattern is consistent and is different from its adjacent segments. These action segments, as the nodes in the action layer, are modeled as a linear-chain CRF and the proposed Hierarchical-CRF model is built accordingly as described above.

The model parameters are learned automatically from weakly-labeled training data with the location and labels of activities of interest. Image-level features are detected and organized to form the context for activities. Common sense domain knowledge about the activities of interest is used to guide the formulation of these context features within activities from the weakly-labeled training data. We utilize a structural model in a max-margin framework, iteratively inferring the hidden activity variables and learning the parameters of different layers. For the testing, the action segments, which are merged together and assigned with activity labels by the preliminary activity detection method, are relabeled through inference on the learned Hierarchical-CRF model.

3.4.1 Feature Descriptors

We now define the concepts we use for the feature development. An activity is a 3D region consisting of one or multiple consecutive action segments. An agent is the underlying moving person(s) or a trajectory. Motion region at frame n is the region surrounding the moving objects of interest in the n^{th} frame of the activity. Activity region is the smallest rectangle region that encapsulates the motion regions over all frames of the activity. In general, same type of features for different class or class pair can be different. There are mainly three kinds of features in our model: action-layer features, action-activity features and activity-layer features, which can be further divided into five types of features. We now describe how to encode motion and context information into feature descriptors.

Intra-Action Feature: $\varphi_V(\mathbf{x}_i^a, y_i^a)$ encodes the motion information of the action segment i that is extracted from low-level motion features such as STIP features. Since in the action layer, we obtain action segments by utilizing their discriminative motion patterns, we use only motion features for the development of action-layer features. STIP histograms are generated for each action segment using bag-of-word method [75]. We train a kernel multi-SVM upon action segments to generate the normalized confidence scores, $s_{i,j}$, of classifying the action segment i as activity class j, where $j \in \{0,1,...,M\}$, such that $\sum_{j=0}^{M} s_{i,j} = 1$. In general, any kind of classifier and low-level motion features can be used here. Given an action segment i, $\varphi_V(\mathbf{x}_i^a, y_i^a) = [s_{i,0} \cdots s_{i,M}]^T$ is developed as the intra-action feature descriptor of action segment i.

Inter-Action Feature: $\varphi_\mathcal{E}(\mathbf{x}_i^a, \mathbf{x}_j^a, y_i^a, y_j^a)$ encodes the probabilities of coexistence of action segments i and j according to their features and activity labels. $\varphi_\mathcal{E}(\mathbf{x}_i^a, \mathbf{x}_j^a, y_i^a, y_j^a) = \mathbf{I}(y_i^a)\mathbf{I}(y_j^a)$, where $\mathbf{I}(y_k^a)$ is the Dirac measure that equals 1 if the true label of segment k is y_k^a and equals to 0 other wise, for $k = i, j$.

Action-Activity Consistency Feature: $\varphi_{c,l}(\mathbf{y}_c^a, y_c^h)$ encodes the labeling information within clique c as

$$\varphi_{c,l}(\mathbf{y}_c^a, y_c^h) = \begin{cases} 1 & y_c^h = l_f \\ \frac{\sum_{i \in c} I(y_i^a = y_c^h)}{N_c} & y_c^h \in \mathscr{L} \end{cases} .$$

where $I(\cdot)$ is the Dirac measure and N_c is the number of action segments in clique c.

Intra-Activity Feature: $\varphi_{c,f}(\mathbf{x}_c^a, x_c^h, \mathbf{y}_c^a, y_c^h)$ encodes the intra-activity motion and context information of activity c. To capture the motion pattern of an activity, we use the intra-action features of action segments which belong to the activity. Given an activity, $[\max_{i \in \aleph} s_{i,0}, ..., \max_{i \in \aleph} s_{i,M}]$ is developed as the intra-activity motion feature descriptor, where $\aleph$ is a list of action segments in activity c.

Intra-activity context feature captures the context information about the agents and relationships between the agents, as well as the the interacting objects (e.g. the object classes, interactions between agents and their surroundings). We define a set, G, of attributes that describes such context for activities of interest, using common-sense knowledge about the activities of interest (how to identify such attributes automatically is another research topic that we do not address in this dissertation). For a given activity, whether the defined attributes are true or not are determined from image-level detection results. The resulting feature descriptor is a normalized feature histogram. The attributes used and the development of intra-activity context features are different for different tasks (please refer to Section 3.6.1.2 for the details).

Finally, the weighted motion and context features are used as the input to a multi-SVM and the output confidence scores are used to develop the intra-activity feature as $\varphi_{c,f}(\mathbf{x}_c^a, y_c^h) = [s_{c,0}, ..., s_{c,M}]^T$.

Inter-Activity Spatial and Temporal Features: $\varphi_{sc}(\mathbf{x}_s^h, \mathbf{x}_d^h, y_s^h, y_d^h)$ and $\varphi_{tc}(\mathbf{x}_s^h, \mathbf{x}_d^h, y_s^h, y_d^h)$ capture the spatial and temporal relationships between activities s and d. Define the scaled distance between activities s and d at the n^{th} frame of s as

$$r_s(s(n), d) = \frac{D(O_s(n), O_d)}{R_s(n) + R_d}, \tag{3.14}$$

where $O_s(n)$ and $R_s(n)$ denote the center and radius of the motion region of activity s at its n^{th} frame and O_d and R_d denote the center and radius of the activity region of activity d. $D(\cdot)$ denotes the Euclidean distance. Then, the spatial relationship of s and d at the n^{th} frame is modeled by $sc_{sd}(n) = bin(r_s(s(n), d))$ as in Fig. 3.11 (a). The normalized histogram $sc_{s,d} = \frac{1}{N_f} \sum_{n=1}^{N_f} sc_{sd}(n)$ is the inter-activity spatial feature of activity s and d.

Let TC be defined by the following temporal relationships: n^{th} frame of s is before d, n^{th} frame of s is during d and n^{th} frame of s is after d. $tc_{sd}(n)$ is the temporal relationship of s and d at the n^{th} frame of s as shown in Fig. 3.11 (b). The normalized histogram $tc = \frac{1}{N_f} \sum_{n=1}^{N_f} tc_{sd}(n)$ is the inter-activity temporal context feature of activity s with respect to activity d.

3.4.2 Model Development

CRF is a discriminative model often used usually used for labeling problems of image and image objects. Essentially, CRF can be considered as a special version of Markov Random Field (MRF) where the variable potentials are conditioned on the observed data. Let $\mathbf{x}$ be the model observations and $\mathbf{y}$ be the label variables. The posterior distribution $p(\mathbf{y}|\mathbf{x},\omega)$ of the label variables over the CRF is a *Gibbs* distribution and is usually represented as

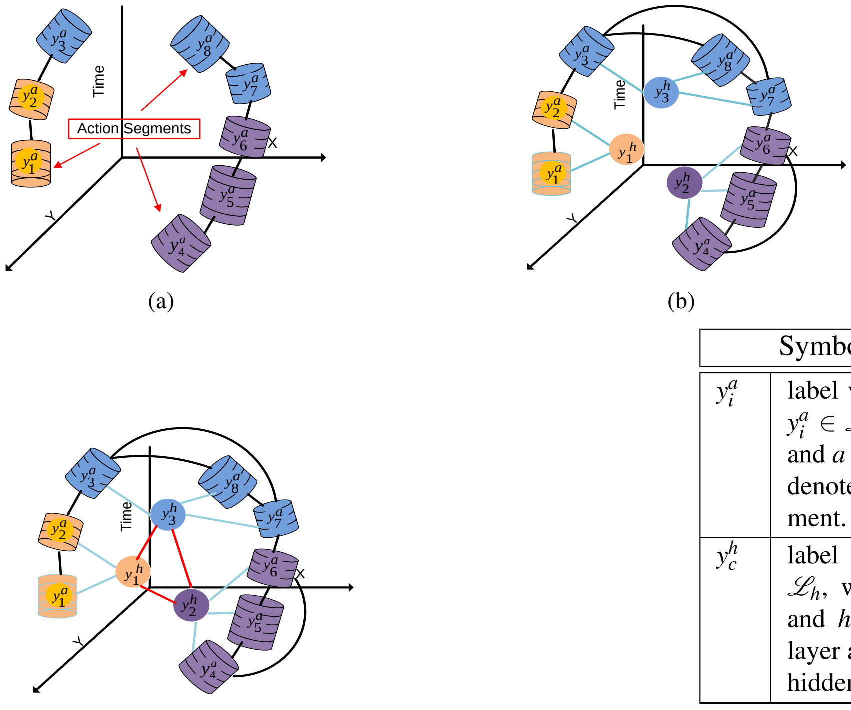
$$p(\mathbf{y}|\mathbf{x},\omega) = \frac{1}{Z(\mathbf{x},\omega)} \prod_{c \in C} exp({\omega_c}^T \varphi_c(\mathbf{x},\mathbf{y}_c)), \tag{3.15}$$

where ω_c is a model weight vector, which needs to be learned from training data. $Z(\mathbf{x},\omega)$ is a normalizing constant called the partition function. $\varphi_c(\mathbf{x},\mathbf{y}_c)$ is a feature vector derived from the observation $\mathbf{x}$ and the label vector, $\mathbf{y}_c$, in the clique c.

The potential function of the CRF model given the observations $\mathbf{x}$ and model weight vector ω is defined as

$$\psi(\mathbf{y}|\mathbf{x},\omega) = \sum_c {\omega_c}^T \varphi_c(\mathbf{x},\mathbf{y}_c). \tag{3.16}$$

For the development of the Hierarchical-CRF model, the action layer is first modeled as a linear-chain CRF. Activity layer variables which are associated with detected activities are then introduced for the smoothing of the action-layer variables. Finally, activity-layer variables are connected to represent the spatial and temporal relationships between activities. The evolution of the proposed two-layer Hierarchical-CRF model from the one-layer CRF model is shown in Fig. 3.6. Details on the development of these models will be described in the following sub-sections. The various feature vectors used for the calculation of the potentials are described in Section 3.4.1.



(a)

(b)

(c)

Symbols used in this figure.	
y_i^a	label variable for action segment i, $y_i^a \in \mathscr{L}$, where $\mathscr{L} = \{0, 1, ..., M\}$ and a denotes the action-layer and i denotes the index of the action segment.
y_c^h	label variable for activity c, $y_c^h \in \mathscr{L}_h$, where $\mathscr{L}_h = \{0, 1, ..., M\} \bigcup l_f$ and h denotes the hidden activity layer and c denotes the index of the hidden activity.

(d)

Figure 3.6: Illustration of CRF models for activity recognition. (a): Action-based Linear-Chain CRF; (b): Action-based higher-order CRF model (with latent activity variables); (c): Action-based two-layer HCRF. Note that all the observations for the random variables are omitted for compactness. One action segment denotes a random variable in the action layer, whose value is the activity label for the action segment. A colored circle denotes a random variable in the activity layer, whose value is the label for its connected clique. As shown in (a), in the action layer, action segments that belong to the same trajectory are modeled as a linear-chain CRF. Then, hidden activity-level variables with action-activity edges (in light blue) are added for each action cliques to form higher-order CRF as shown in (b). An activity nodes and its associated action nodes have a same color. Finally, pair-wise activity edges (in red) are added to form the proposed two-layer HCRF mdoel.

3.4.2.1 Action-Based Linear-Chain CRF

We first describe the linear-chain CRF model in Fig. 3.6(a). We first define the following items: *intra-action potential* $\psi_\mathcal{V}(y_i^a|\mathbf{x},\omega)$, which measures the compatibility of the observed feature of i and its label y_i^a; *inter-action potential* $\psi_\mathcal{E}(y_i^a, y_j^a|\mathbf{x},\omega)$, which measures the consistency between two connected action segments i and j. Let $\mathcal{V}^a$ be the set of vertices, each representing an action segment as the element in the action layer and $\mathcal{E}^a$ denotes the set of connected action pairs. The potential function of the action-layer linear-chain CRF is

$$\begin{aligned}\psi(\mathbf{y}^a|\mathbf{x},\omega) &= \sum_{i\in\mathcal{V}^a}\psi_\mathcal{V}(y_i^a|\mathbf{x},\omega) + \sum_{ij\in\mathcal{E}^a}\psi_\mathcal{E}(y_i^a,y_j^a|\mathbf{x},\omega) \qquad (3.17)\\ &= \sum_{i\in\mathcal{V}^a}{\omega^a_{\mathcal{V},y_i^a}}^T\varphi_\mathcal{V}(\mathbf{x}_i^a,y_i^a) + \sum_{ij\in\mathcal{E}^a}{\omega^a_{\mathcal{E},y_i^a,y_j^a}}^T\varphi_\mathcal{E}(\mathbf{x}_i^a,\mathbf{x}_j^a,y_i^a,y_j^a),\end{aligned}$$

where $\varphi_\mathcal{V}(\mathbf{x}_i^a,y_i^a)$ is the *intra-action feature vector* that describes action segment i. $\omega^a_{\mathcal{V},y_i^a}$ is the weight vector of the intra-action features for class y_i^a. $\varphi_\mathcal{E}(\mathbf{x}_i^a,\mathbf{x}_j^a,y_i^a,y_j^a)$ is the *inter-action feature*, which is derived from the labels y_i^a, y_j^a and intra-action feature vectors $\mathbf{x}_i^a$ and $\mathbf{x}_j^a$. $\omega^a_{\mathcal{E},y_i^a,y_j^a}$ is the weight vector of the inter-action features for class pair y_i^a, y_j^a.

3.4.2.2 Incorporating Higher Order Potentials

According to experimental observations, action segments in a candidate activity region, which are generated by activity detection methods [138], tend to have the same activity labels. However, consistent labeling is not guaranteed due to inaccurate detections. Let an action clique c^a denote the union of action segments in a candidate activity c. The linear-chain CRF can be converted to a higher-order CRF by adding a latent activity variable y_c^h, representing the label of c, for each action clique c^a. All action variables associated with the same activity variable are connected. Then, the associated higher-order potential $\psi_c(\mathbf{y}_c^a|\mathbf{x},\omega)$ is introduced to encourage action segments in the clique c^a to take the same label, while still allowing some of them to have different labels without additional penalty. The resulting CRF model is shown in 3.6 (b). The potential function ψ for the higher-order CRF model is represented as

$$\begin{aligned}\psi(\mathbf{y}^a, y_c^h|\mathbf{x}, \omega) = &\sum_{i \in \mathscr{V}^a} {\omega_{\nu,y_i^a}^a}^T \varphi_\nu(\mathbf{x}_i^a, y_i^a) \\ &+ \sum_{ij \in \{\mathscr{E}'^a\}} {\omega_{\varepsilon,y_i^a,y_j^a}^a}^T \varphi_\varepsilon(\mathbf{x}_i^a, \mathbf{x}_j^a, y_i^a, y_j^a) + \sum_{c \in C^{ah}} \psi_c(\mathbf{y}_c^a|\mathbf{x}, \omega),\end{aligned} \tag{3.18}$$

where $\mathscr{E}'^a$ denotes the set of connected action pairs in the new model. C^{ah} is the set of action-activity cliques and each action-activity clique c in C^{ah} corresponds to an action clique c^a in the action layer and its associated activity c in the activity layer. Let $\mathscr{L} = 0, 1, \cdots, M$ be the activity label set in the action layer, from which the action variables may take values. The activity variable y_c^h takes values from an extended label set $\mathscr{L}_h = \mathscr{L} \cup l_f$, where $\mathscr{L}$ is the set of variable values in the action layer. When an activity variable takes value l_f, it allows its child variables to take different labels in $\mathscr{L}$, without additional penalty upon label inconsistency.

We define $\varphi_{c,l}(\mathbf{y}_c^a, y_c^h)$ as the *action-activity consistency feature* of activity c, and $\omega_{c,l,y_c^h}^{ah}$ to be the weight vector of the action-activity consistency feature for class y_c^h. Define $\varphi_{c,f}(\mathbf{x}_c^a, y_c^h)$ as the *intra-activity feature* for activity c, and $\omega_{c,f,y_c^h}^{ah}$ to be the weight vector of intra-activity feature for class y_c^h. The corresponding action-activity higher-order potential can be defined as

$$\begin{aligned}\psi(\mathbf{y}_c^a|\mathbf{x}, \omega) &= \max_{y_c^h} {\omega_{c,y_c^h}^{ah}}^T \varphi_c(\mathbf{x}_c^a, \mathbf{x}_c^h, \mathbf{y}_c^a, y_c^h) \\ &= \max_{y_c^h} [{\omega_{c,l,\mathbf{y}_c^a,y_c^h}^{ah}}^T \varphi_{c,l}(\mathbf{y}_c^a, y_c^h) + {\omega_{c,f,y_c^h}^{ah}}^T \varphi_{c,f}(\mathbf{x}_c^a, y_c^h)],\end{aligned} \tag{3.19}$$

where ${\omega_{c,l,y_c^h}^{ah}}^T \varphi_{c,l}(\mathbf{y}_c^a, y_c^h)$ measures the labeling consistency within the activity c. Intuitively, the higher-order potentials are constructed such that a latent variable tends to take a label from $\mathscr{L}$ if majority of its child nodes take the same value, and take the label l_f if its child nodes take diversified values. ${\omega^{ah}_{c,f,y^h_c}}^T \varphi_{c,f}(\mathbf{x}^a_c, y^h_c)$ is the *intra-activity potential* that measures the compatibility between the activity label of clique c and its activity features.

3.4.2.3 Incorporating Inter-Activity Potentials

As stated before, it would be helpful to model the spatial and temporal relationships between activities. For this reason, we connect activity nodes in the higher-order CRF model. The resulting CRF is shown in Fig. 3.6(c). We define $\varphi_{sc}(\mathbf{x}^h_s, \mathbf{x}^h_d, y^h_s, y^h_d)$ as the *inter-activity spatial feature* that encodes the spatial relationship between activities s and d, and $\omega^h_{sc,y^h_s,y^h_d}$ to be the weight vector of inter-activity spatial feature for class pair (y^h_s, y^h_d). Define $\varphi_{tc}(\mathbf{x}^h_s, \mathbf{x}^h_d, y^h_s, y^h_d)$ as the *inter-activity temporal feature* that encodes the temporal relationship between activities s and d, and $\omega^h_{tc,y^h_s,y^h_d}$ to be the weight vector of inter-activity temporal feature for class pair (y^h_s, y^h_d).

The pairwise activity potential between clique s and d is defined as

$$\begin{aligned}\psi(\mathbf{y}^h|\mathbf{x},\omega) = \sum_{sd\in\mathscr{E}^h} [&{\omega^h_{sc,y^h_s,y^h_d}}^T \varphi_{sc}(\mathbf{x}^h_s, \mathbf{x}^h_d, y^h_s, y^h_d) \\ &+ {\omega^h_{tc,y^h_s,y^h_d}}^T \varphi_{tc}(\mathbf{x}^h_s, \mathbf{x}^h_d, y^h_s, y^h_d)], \qquad (3.20)\end{aligned}$$

where ${\omega^h_{sc,y^h_s,y^h_d}}^T \varphi_{sc}(\mathbf{x}^h_s, \mathbf{x}^h_d, y^h_s, y^h_d)$ is the pairwise spatial potential between activities s and d that measures the compatibility between the candidate labels of s and d and their spatial relationship. ${\omega^h_{tc,y^h_s,y^h_d}}^T \varphi_{tc}(\mathbf{x}^h_s, \mathbf{x}^h_d, y^h_s, y^h_d)$ is the pairwise temporal potential between activities s and d that measures the compatibility between the candidate labels of s and d and their temporal relationship.

3.4.3 Model Learning and Inference

The parameters of the overall potential function $\psi(\mathbf{y}|\mathbf{x},\omega)$ for the two-layer hierarchical CRF include ω^a_v, ω^a_ε, $\omega^{ah}_{c,l}$, $\omega^{ah}_{c,f}$, ω^h_{sc} and ω^h_{tc}. We define the weight vector as the concatenation of these parameters:

$$\omega = [\omega_v^a, \omega_\varepsilon^a, \omega_{c,l}^{ah}, \omega_{c,f}^{ah}, \omega_{sc}^h, \omega_{tc}^h]. \tag{3.21}$$

Thus, the potential function, $\psi(\mathbf{y}|\mathbf{x},\omega)$, can be converted into a linear function with a single parameter ω as

$$\psi(\mathbf{y}^a) = \max_{\mathbf{y}^h} \omega^T \Gamma(\mathbf{x}, \mathbf{y}^a, \mathbf{y}^h), \tag{3.22}$$

where $\Gamma(\mathbf{x}, \mathbf{y}^a, \mathbf{y}^h)$, called the joint feature of activity set $\mathbf{x}$, can be easily obtained by concatenating various feature vectors in (3.18),(3.19) and (3.20).

3.4.3.1 Learning Model Parameters

Suppose we have P activity sets for learning. Let the training set be $(X, Y^a, Y^h) = (\mathbf{x}^1, \mathbf{y}^{1,a}, \mathbf{y}^{1,h}), ..., (\mathbf{x}^P, \mathbf{y}^{P,a}, \mathbf{y}^{P,h})$, where $\mathbf{x}^i$ denotes the i^{th} activity set as well as the observed features of the set. $\mathbf{y}^{i,a}$ is the label vector in the action layer and $\mathbf{y}^{i,h}$ is the label vector in the hidden activity layer. While there are various ways of learning the model parameters, we choose a task-oriented discriminative approach. We would like to train the model in such a way that it increases the average precision scores on a training data and thus tend to produce the correct activity labels for each action segment.

A natural way to learn the model parameter ω is to adopt the latent structural SVM. The loss $\Delta(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,a})$ of labeling $\mathbf{x}^i$ with $\widehat{\mathbf{y}}^{i,a}$ in the action layer equals the number of action segments that associate with incorrect activity labels (an action segment is mislabeled if over half of the segment is mislabeled). From the construction of the higher-order potentials in section 3.4.2.2, it is observed that, in order to achieve the best labeling of the action segments, the optimum latent activity label of an action clique must be the dominant ground truth label l_c of its child nodes in the action layer; or the free label l_f if no dominant label exists for the action clique. Thus the loss $\Delta(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,h})$ of labeling the activity layer of $\mathbf{x}^i$ with $\widehat{\mathbf{y}}^{i,h}$ is

$$\Delta(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,h}) = \sum_{c \in \mathscr{V}^h} I(y_c^{i,h} \neq \{l_c^i, l_f\}), \tag{3.23}$$

where $I(\cdot)$ is the indicator function which equals 1 if the inside equation is satisfied and 0 otherwise. (3.23) counts the number of activity labels in $\widehat{\mathbf{y}}^{i,h}$ that are neither a free label nor the dominant label of its child nodes. Finally, the loss function of assigning $\mathbf{x}^i$ with $(\widehat{\mathbf{y}}^{i,a}, \widehat{\mathbf{y}}^{i,h})$ is defined as the summation of the two, that is

$$\Delta(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,a}, \widehat{\mathbf{y}}^{i,h}) = \Delta(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,a}) + \Delta(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,h}). \tag{3.24}$$

Next, we define a convex function $F(\omega)$ and a concave function $J(\omega)$ as

$$\begin{aligned} F(\omega) &= \frac{1}{2}\omega^T \omega \\ &+ C\sum_{i=1}^{P} \max_{(\widehat{\mathbf{y}}^{i,a}, \widehat{\mathbf{y}}^{i,h})} \left[\omega^T \Gamma\left(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,a}, \widehat{\mathbf{y}}^{i,h}\right) + \Delta\left(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,a}, \widehat{\mathbf{y}}^{i,h}\right)\right], \\ \text{and} \quad J(\omega) &= -C\sum_{i=1}^{P} \max_{\mathbf{y}^{i,h}} \omega^T \Gamma\left(\mathbf{x}^i, \mathbf{y}^{i,a}, \mathbf{y}^{i,h}\right). \end{aligned} \tag{3.25}$$

The model learning problem is given as:

$$\omega^* = \arg\min_{\omega} [F(\omega) + J(\omega)] \tag{3.26}$$

Although the objective function to be minimized in (3.26) is not convex, it is a combination of a convex function and a concave function [82]. Such kind of problems can be solved using the Concave-Convex Procedure (CCCP) [131, 132]. We describe an algorithm similar to the CCCP in [131] that iteratively infers the latent variables $\mathbf{y}^{i,h}$ for $i = 1,...,P$ and optimizes the weight vector ω. The inference and optimization procedures continue until convergence or a predefined maximum number of iterations is reached.

The limitation of all learning algorithms that involve gradient optimization is that they are susceptible to local extrema and saddle points [56]. Thus, the performance of the proposed latent structural model is sensitive to initialization. There have been many works dealing with the problem of learning the parameters of hierarchical models [27, 111]. We use a coarse to fine scheme that separately initializes the model parameters using piecewise training, and then refines the model parameters jointly in a globally optimum manner. Specifically, the separately learned model parameters are used as the initialization values for the proposed learning algorithm. Given the weakly labeled training data with activity labels for each action segment, the dominant label l_c for each action clique can be determined. We initialize the latent activity variable of c with the dominant label l_c of its action clique c^a, and with l_f if there is no dominant label for c^a.

In the "E step", we infer latent variables using the previously learned weight vector ω_t (or the initially assigned weight vector for the first iteration) leading to

$$\mathbf{y}_{t+1}^{i,h^*} = \arg\max_{\mathbf{y}^{i,h}} \omega_t{}^T \Gamma\left(\mathbf{x}^i, \mathbf{y}^{i,a}, \mathbf{y}^{i,h}\right). \tag{3.27}$$

Then, in the "M step", with the inferred latent variable $\mathbf{y}_{t+1}^{i,h^*}$, we solve a fully visible structural SVM (SSVM). Let us define the risk function at iteration $t+1$, $\Lambda(\omega)$, as

$$\begin{aligned}\Lambda_{t+1}(\omega) = C\sum_{i=1}^{P} \max_{(\widehat{\mathbf{y}}^{i,a},\widehat{\mathbf{y}}^{i,h})} &\left\{\Delta\left(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,a}, \widehat{\mathbf{y}}^{i,h}\right)\right. \\ &\left. + \omega^T \left[\Gamma\left(\mathbf{x}^i, \widehat{\mathbf{y}}^{i,a}, \widehat{\mathbf{y}}^{i,h}\right) - \Gamma\left(\mathbf{x}^i, \mathbf{y}^{i,a}, \mathbf{y}_{t+1}^{i,h^*}\right)\right] \right\}.\end{aligned} \tag{3.28}$$

Thus, the optimization problem in (3.26) is converted to a fully visible SSVM as

$$\omega_{t+1}^{*} = \arg\min_{\omega} \left\{ \frac{1}{2} \omega^{T} \omega + \Lambda_{t+1}(\omega) \right\}. \tag{3.29}$$

The problem in (3.29) can be converted to an unconstrained convex optimization problem [138] and solved by the modified bundle method in [114]. The algorithm iteratively searches for the increasingly tight quadratic upper and lower cutting planes of the objective function until the gap between the two bounds reaches a predefined threshold. The algorithm is effective because of its very high convergence rate [112]. The visible SSVM learning algorithm specified for our problem is summarized in Algorithm 7.

Algorithm 5 Learning the model parameter in (3.29) through bundle method

```
Input:   $S = ((a_T(1), y_T(1)), \ldots, (a_T(P), y_T(P))), \omega_t^*, \mathbf{y}_{t+1}^{i,h^*}, C, \varepsilon$
Output:  Optimum model parameter $\omega_{t+1}^*$

1.  initialize $\omega_{t+1}^0$ with $\omega_t^*$, $\mathscr{G}_{t+1}$(cutting plane set) $\leftarrow \emptyset$.
2.  for $k = 0$ to $\infty$ do
3.      for $i = 1, ..., P$ do
            find the most violated label vector for each training instance,
            if any, using $\omega_{t+1}^k$ (the value of $\omega_{t+1}$ at the $k^{th}$ iteration);
4.      end for
5.      find the cutting plane $g_{\omega_{t+1}^k}$ of $\Lambda(\omega)$ at $\omega_{t+1}^k$:
        $g_{\omega_{t+1}^k} = \omega^T \partial_\omega \Lambda_{t+1}(\omega_{t+1}^k) + b_{\omega_{t+1}^k}$,
        where   $b_{\omega_{t+1}^k} = \Lambda_{t+1}(\omega_{t+1}^k) - {\omega_{t+1}^k}^T \partial_\omega \Lambda(\omega_{t+1}^k)$.
6.      $\mathscr{G}_{t+1} \leftarrow \mathscr{G}_{t+1} \cup g_{\omega_{t+1}^k}(\omega)$;
7.      update $\omega_{t+1}$:   $\omega_{t+1}^{k+1} = \arg\min_\omega F_{\omega_{t+1}^k}(\omega)$,
        where   $F_{\omega_{t+1}^k}(\omega) = \frac{1}{2}\omega^T \omega + max(0, max_{j=1,...,k} g_{\omega_{t+1}^j}(\omega))$.
8.      $gap_{k+1} = \min_{k' \le k} F_{\omega_{t+1}^{k'+1}}(\omega_{t+1}^{k'+1}) - F_{\omega_{t+1}^k}(\omega_{t+1}^{k+1})$;
9.      if $gap_{k+1} \le \varepsilon$, then return $\omega_{t+1}^* = \omega_{t+1}^{k+1}$;
10. end for
```

3.4.3.2 Inference

Suppose the model parameter vector ω is given. We now describe how to identify the optimum label vector $\mathbf{y}^a$ for a test instance $\mathbf{x}$ that maximizes (3.22). The inference problem is generally NP hard for multi-class problems, thus MAP inference algorithms, such as loopy belief propagation [82], are slow to converge. We propose an approximation method that alternatively optimizes the hidden variable $\mathbf{y}^h$ and the label vector $\mathbf{y}^a$. Such an algorithm is guaranteed to increase the objective at every iteration [82]. Let us define the activity layer potential function as

$$\psi^h(\mathbf{y}^h) = \sum_{c \in C^a} \psi(\mathbf{y}^a_c|\mathbf{x}, \omega) + \psi(\mathbf{y}^h|\mathbf{x}, \omega). \tag{3.30}$$

For each iteration, with current predicted label vector $\mathbf{y}^a$ fixed, the inference sub-problem is to find the $\mathbf{y}^h$ that maximizes $\psi^h(\mathbf{y}^h)$. An efficient greedy search method is used to find the optimum $\mathbf{y}^h$ as described in Algorithm 6. In order to simplify the inference, we force the edge weights between non-adjacent actions to be zeros. With the inferred hidden variable $\mathbf{y}^h$, the model is reduced to a one-layer discriminative CRF. The inference sub-problem of finding the optimum $\mathbf{y}^a$ can now be solved by computing the exact mixed integer solution. We initialize the process by holding the hidden variable fixed using the values obtained from automatic activity detection. The process continues until convergence or a predefined maximum number of iterations is reached.

3.5 Computational Analysis

Since the computational complexity of the training and inference of CRF models depend mainly on the computational complexity of the inference procedure, we now discuss the

Algorithm 6 Greedy Search Algorithm for the sub-problem of finding optimum hidden variable $\mathbf{y}^h$

```
    Input:    Testing Instance with Action Layer Labels $\mathbf{y}^a$
    Output:   Hidden variable labels $\mathbf{y}^h$

    1. initialize $(\mathscr{V}^h, \mathbf{y}^h) \leftarrow \{\emptyset, \emptyset\}$ and $\psi^h = 0$.

    2. repeat
          $\Delta\psi^h(y_c^h)_{c \notin \mathscr{V}^h} = \psi(\mathbf{y}^h \cup y_c^h) - \psi(\mathbf{y}^h)$;
          $y_c^{h^{opt}} = \arg\max_{c \notin \mathscr{V}^h} \Delta\psi^h(y_c^h)$;
          $(\mathscr{V}^h, \mathbf{y}^h) \leftarrow (\mathscr{V}^h, \mathbf{y}^h) \cup (c, y_c^{h^{opt}})$;

    3. end if all activities are labeled.
```

computational complexity of inference for a particular activity set consists of n action segments and m activities. Assuming there are M activity classes in the problem. For the graphical model in [140], the time complexity of the inference as discussed in the paper is $O(d_{max} n^2 M)$, where d_{max} is the maximum number of action segments one activity may have. The inference on both the higher-order CRF and hierarchical-CRF is carried out layer-by-layer, and so the overall time complexity is linear in the number of layers used. Specifically, we use two-layer CRFs with an action layer and an activity layer. For the higher-order CRF model, inference on the activity layer takes $O(mM)$ computation to obtain the activity labels for each candidate activity. With the inferred activity labels, inference on the action layer takes $O(nM^2)$, since the model is reduced to a chain-CRF. For the hierarchical-CRF, the increase of computational complexity over the higher-order CRF lies in the inference on the activity layer, because the activities are connected with each other in this model. Using the proposed greedy search algorithm, the time complexity for inference on the activity layer is $O(m^2 M)$. Thus, the overall complexity of inference is $O[T \cdot ((mM) + O(nM^2))]$ for higher-order CRF and $O[T \cdot ((m^2 M) + O(nM^2))]$ for hierarchical-CRF, where T is the number of iterations. Furthermore, the number of action segments n is usually several times of the number of activities, that is $n = \alpha m$, where α is a small positive value larger that one. d_{max} and T are small positive value larger than one. Assuming n, m and M are in the same order, which is a reasonable assumption for our case, the asymptotic computational complexity of the model in [140] and the compared higher-order CRF and hierarchical-CRF models is of the same order.

3.6 Experiments

To assess the effectiveness of our graphical models in activity modeling and recognition, we perform experiments on the public UCLA Office Dataset [104] and VIRAT Ground Dataset [79].

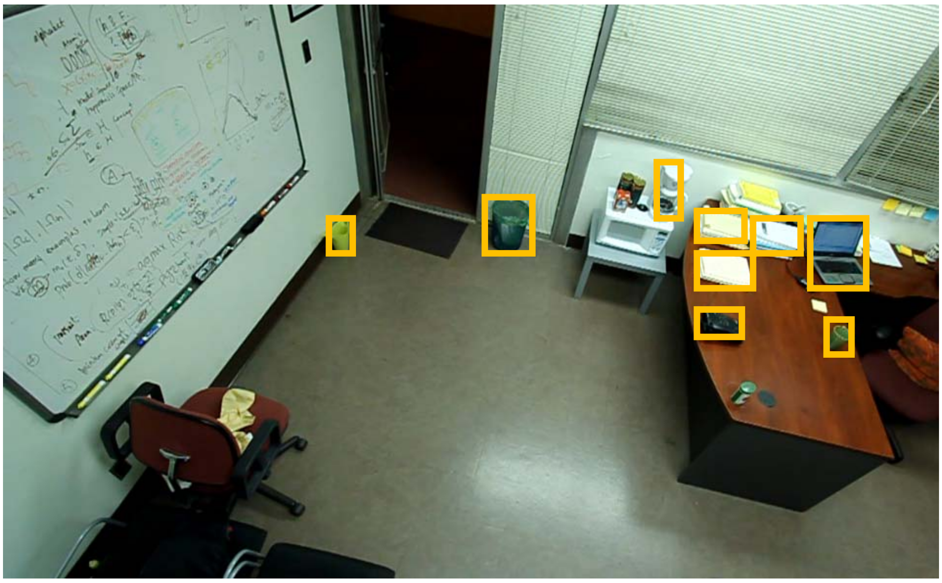
3.6.1 Datasets

In this subsection, we introduce the UCLA Office Dataset [104] and VIRAT Ground Dataset [79] and the preprocessing related to these datasets.

3.6.1.1 UCLA Dataset

The UCLA Office Dataset [104] consists of indoor and outdoor videos of single activities and person-person interactions. Here, we perform experiments on the videos of office scene containing about 35 minutes of activities in an office room that captured with a single fixed camera. We identify 10 frequent activities as the activities of interest: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. Each activity occurs 9 to 26 times in the dataset. Since the dataset contains only single person activities, it is natural to model activities in one sequence together. The dataset is divided into 8 sets, each set contains 2 sequences of activities and each sequence contains 2 to 19 activities of interest, as well as varying number of background activities. We use leave-one-set-out cross validation for the evaluation: use 7 sets for training and 1 set for testing.

Preprocessing Intra-activity context feature is based on interactions between the agent and the surroundings. In the office dataset, there are 7 classes of objects that are frequently involved in the activities of interest: laptop, garbage can, papers, phone, coffee maker and cup. Fig. 4.2 shows the detected objects of interest in the office room. Since the UCLA Dataset consists of single person activities, the intra-activity attributes considered include agent-object interactions and their relative locations. We identify $(N_G = 10)$ subsets of attributes for the development of intra-activity context features in the experiment as shown in Fig. 3.8. For a given activity, the above attributes are determined from image-level detection results. The locations of objects are automatically tracked. Similar to [104], if enough skin color is detected within the areas of laptop, paper and phone,the corresponding attributes are considered as true. Fig. 3.9 shows examples of detected agent-object interactions.



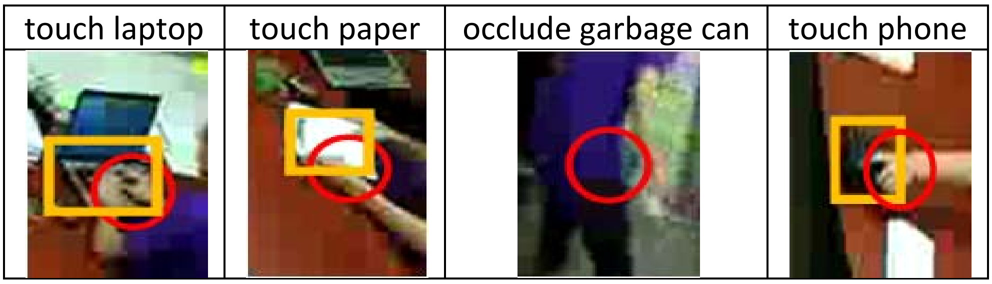
(a)

Figure 3.7: Detected objects of interest in the UCLA office scene.

Whether the agent is near or far away from an object is determined by the distance between the two based on normal distributions of the distances of the two scenarios. Probabilities

Attribute Subset	Associated Attributes
G_1 G_2 G_3	the agent is touching / not touching laptop^1, paper^2, phone^3.
G_4 G_5	the agent is occluding / not occluding the garbage can^4, coffee maker^5.
G_6 G_7 G_8	the agent is near / far away from the garbage can^6, coffee maker^7, door^8.
G_9	the agent disappears / not disappears at the door.
G_{10}	the agent appears / not appears at the door.

Figure 3.8: Subsets of context attributes used for the development of intra-activity context features for UCLA Dataset (the superscripts indicates the correspondence between the subsets and the objects).



(a)

Figure 3.9: Examples of agent-object interactions detected from image.

indicating how likely the agent is near or far away from an object are thus obtained. For frame n of an activity, we obtain $g_i(n) = I(G_i(n))$, where $I(\cdot)$ is the indicator function. $g_i(n)$ is then normalized so that its elements sum to 1.

Related candidate activities are connected. Whether two activities are related can be naturally determined by their temporal distances. One way to decide if the relationships between two candidate activities should be modeled is to see if they are in the α-neighborhood of each other in time. Two activities are said to be in the α-neighborhood of each other if there are less than α other activities occurring between the two.

3.6.1.2 VIRAT Ground Dataset

The VIRAT Ground Dataset is a state-of-the-art activity dataset with many challenging characteristics, such as wide variation in the activities and clutter in the scene. The dataset consists of surveillance videos of realistic scenes with different scales and resolution, each lasting 2 to 15 minutes and containing upto 30 events. The activities defined in Release 1 include 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. We work on the all the scenes in Release 1 except scene 0002 and use half of the data for training and the rest for testing. Five more activities are defined in VIRAT Release 2 as: 7 - person gesturing; 8 - person carrying an object; 9 - person running; 10 - person entering a facility; 11 - person exiting a facility. We work on the all the scenes in Release 2 except scene 0002 and 0102, and use two-third of the data for training and the rest for testing.

Preprocessing Motion regions that do not involve people are excluded from the experiments since we are only interested in person activities and person-vehicle interactions. For the development of STIP histograms, nearest neighbor soft-weighting scheme [75] is used.

Since we work on the VIRAT Dataset with individual person activities and person-object interactions, we use the following $N_G = 7$ subsets of attributes for the development of intra-activity context features in the experiments as shown in Fig. 3.10.

Subset	**Associated Attributes**
G_1	moving object is a person; moving object is a vehicle trunk; moving object is of other kind.
G_2	the agent is at the body of the interacting vehicle; the agent is at the rear/head of the interacting vehicle; the agent is far away from the vehicles.
G_3	the agent disappears at the body of the interacting vehicle; the agent appears at the body of the interacting vehicle; none of the two.
G_4	the agent disappears at the entrance of a facility; the agent appears at the exit of a facility; none of the two.
G_5	velocity of the agent (in pixel) is larger than a predefined threshold; velocity of object of interest is smaller than a predefine threshold.
G_6	the activity occurs at parking areas; the activity occurs at other areas.
G_7	an object (e.g. bag/box) is detected on the agent; no object is detected on the agent.

(a)

Figure 3.10: Subsets of context attributes used for the development of intra-activity context features.

Persons and vehicles are detected based on the part-based object detection method in [9]. Opening/closing entrance/exit doors of facilities, boxes and bags are detected using method in [6] with binary linear-SVM as the classifier. Using these high-level image features, we follow the description in Section 4.4.1.2 to develop the feature descriptors for each activity set. The first three sets of attributes in Fig. 3.10 are used for the experiments on Release 1, and all are used for the experiments on Release 2. Fig. 3.11 shows examples of $g_i(n)$ defined as in Section 4.4.1.2 for different activities in VIRAT. Since, in VIRAT, activities are naturally related to each other, the activity layer nodes are fully connected to utilize the spatio-temporal relationships of activities occurring in the same local space-time volume.

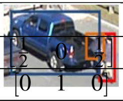
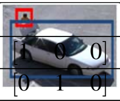
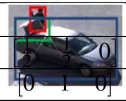
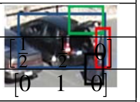
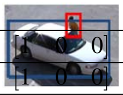
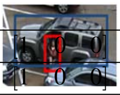
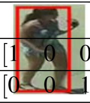
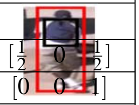
Activity	person loading	person unloading	opening trunk	closing trunk
Example Image				
$g_1(n)$	$[\frac{1}{2} \quad 0 \quad \frac{1}{2}]$	$[1 \quad 0 \quad 0]$	$[\frac{1}{2} \quad \frac{1}{2} \quad 0]$	$[\frac{1}{2} \quad \frac{1}{2} \quad 0]$
$g_2(n)$	$[0 \quad 1 \quad 0]$	$[0 \quad 1 \quad 0]$	$[0 \quad 1 \quad 0]$	$[0 \quad 1 \quad 0]$
$g_5(n)$	$[0 \quad 1]$	$[0 \quad 1]$	$[0 \quad 1]$	$[0 \quad 1]$
$g_6(n)$	$[1 \quad 0]$	$[1 \quad 0]$	$[1 \quad 0]$	$[1 \quad 0]$
$g_7(n)$	$[1 \quad 0]$	$[0 \quad 1]$	$[0 \quad 1]$	$[1 \quad 0]$
Activity	getting into vehicle	getting out of vehicle	gesturing	carrying object
Example Image				
$g_1(n)$	$[1 \quad 0 \quad 0]$	$[1 \quad 0 \quad 0]$	$[1 \quad 0 \quad 0]$	$[\frac{1}{2} \quad 0 \quad \frac{1}{2}]$
$g_2(n)$	$[1 \quad 0 \quad 0]$	$[1 \quad 0 \quad 0]$	$[0 \quad 0 \quad 1]$	$[0 \quad 0 \quad 1]$
$g_5(n)$	$[0 \quad 1]$	$[0 \quad 1]$	$[0 \quad 1]$	$[0 \quad 1]$
$g_6(n)$	$[1 \quad 0]$	$[1 \quad 0]$	$[1 \quad 0]$	$[1 \quad 0]$
$g_7(n)$	$[0 \quad 1]$	$[0 \quad 1]$	$[0 \quad 1]$	$[1 \quad 0]$

Figure 3.11: Examples of detected intra-activity context features. The example images are shown with detected high-level image features. Object in red bounding box is a moving person; object in blue bounding box is a static vehicle; object in orange bounding box is a moving object of other kind; object in black bounding box is a bag/box on the agent.

3.6.2 Preprocessing

We first develop an automatic motion segmentation algorithm by detecting boundaries where the statistics of motion features change dramatically, and thus obtain the action segments. Let two NDMs be denoted as M_1 and M_2, and d_s be the dimension of the hidden states. The distance between the models can be measured by the normalized geodesic distance

$dist(M_1, M_2) = \frac{4}{d_s \pi^2} \sum_{i=1}^{d_s} {\theta_i}^2$, where θ_i is the principal subspace angle (please refer to [14] for details on the distance computation).

A sliding window of size T_s, where T_s is the number of temporal bins in the window, is applied to each detected motion region along time. A NDM $M(t)$ is built for the time window centered at the t^{th} temporal bin. Since an action can be modeled as one dynamic model, the model distances between subsequences from the same action should be small, compared to those of subsequences from a different action. Suppose an activity starts from temporal bin k; the average model distance between temporal bin $j > k$ and k is defined as the weighted average distance between model j and neighboring models of k as

$$DE_k(j) = \sum_{i=0}^{T_d - 1} \gamma_i \cdot dist(M(k+i), M(j)), \tag{3.31}$$

where T_d is the number of neighboring bins used, and γ_i is the smoothing weight for model $k+i$ that decreases along time. When the average model distance grows above a predefined threshold d_{th}, an action boundary is detected. Action segments along tracks are thus obtained.

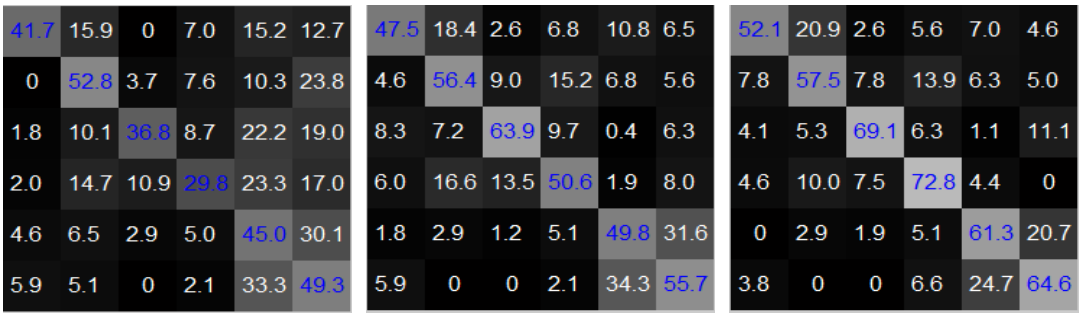
A multi-class SVM is trained upon the intra-activity features (as described in Section 3.4.1) of activities of different classes. After obtaining the action segments, we use the sliding window method with the trained multi-class SVM to group adjacent action segments into candidate activities. To speed up, we only work on candidate activities with confidence scores larger than a predefined threshold, indicating they are likely to be of activity classes of interest.

3.6.3 Structural Model

To assess the effectiveness of our structural model in activity modeling and recognition, we perform experiments on the public VIRAT Ground Dataset [79]. We use the NDM method in [14] with the SVM classifier as the baseline (referred to as NDM + SVM) and integrate our context model with it. The attribute subsets in Fig. 3.11 are used for the development of intra-activity context features. We compare our results with the popular activity recognition method, BOW+SVM [75], and recently developed methods - string of feature graphs (SFG) [36] and sum-product networks (SPN) [2].

3.6.3.1 Recognition Results on VIRAT Release 1

Fig. 3.12 shows the confusion matrix for the baseline classifier and our model with different kinds of features. As an example of the importance of context features, the baseline classifier often confuses "open a vehicle trunk" and "close a vehicle trunk" with each other. However, if the two activities happen closely in time in the same place, the first activity in time is probably "open a vehicle trunk". This kind of contextual information within and across activity classes are captured by our model and used to improve the recognition performance.



41.7	15.9	0	7.0	15.2	12.7
0	52.8	3.7	7.6	10.3	23.8
1.8	10.1	36.8	8.7	22.2	19.0
2.0	14.7	10.9	29.8	23.3	17.0
4.6	6.5	2.9	5.0	45.0	30.1
5.9	5.1	0	2.1	33.3	49.3

(a)

47.5	18.4	2.6	6.8	10.8	6.5
4.6	56.4	9.0	15.2	6.8	5.6
8.3	7.2	63.9	9.7	0.4	6.3
6.0	16.6	13.5	50.6	1.9	8.0
1.8	2.9	1.2	5.1	49.8	31.6
5.9	0	0	2.1	34.3	55.7

(b)

52.1	20.9	2.6	5.6	7.0	4.6
7.8	57.5	7.8	13.9	6.3	5.0
4.1	5.3	69.1	6.3	1.1	11.1
4.6	10.0	7.5	72.8	4.4	0
0	2.9	1.9	5.1	61.3	20.7
3.8	0	0	6.6	24.7	64.6

(c)

Figure 3.12: Recognition Results for VIRAT Release 1. (a): Confusion matrix for the baseline classifier; (b): Confusion matrix for our approach using motion and intra-activity context features; (c): (b): Confusion matrix for our approach using motion and intra- and inter- activity context features.

We show the results on VIRAT Release 1 using precision and recall in Fig. 3.13. We have compared our results with the popular BOW+SVM approach, the more recently proposed String-of-Feature-Graphs approach [36] and the baseline classifier. Our approach outperforms

Activity Class	BOW[75]	SFG [36]	Baseline	Our Method (1)	Our Method (2)
loading-object	44.2(42.8)	50.7(52.3)	43.6(41.7)	42.1(47.5)	**51.6**(**52.1**)
unloading-object	51.1(57.2)	57.1(55.4)	34.9(52.8)	61.3(56.4)	**62.7**(**57.5**)
opening-trunk	58.5(39.3)	38.4(50.3)	59.7(36.8)	64.2(63.9)	**68.5**(**69.1**)
closing-trunk	47.2(33.4)	60.0(61.2)	40.6(29.8)	44.4(50.6)	**55.2**(**72.8**)
getting-into-vehicle	40.4(48.2)	61.8(59.2)	32.7(45.0)	53.0(49.8)	**67.5**(**61.3**)
getting-out-of-vehicle	42.2(53.8)	41.6(68.0)	32.1(49.3)	49.6(55.7)	**65.2**(**64.6**)
Average	47.2(45.8)	51.6(57.8)	40.6(42.5)	52.4(53.8)	**61.7**(**62.9**)

Figure 3.13: Precision and recall (in parenthesis) for the six activities defined in VIRAT Release 1. Baseline: NDM+SVM; Our method (1): the proposed structural model with motion feature and intra-activity context feature; our method (2): the proposed structural model with motion feature, intra-activity and inter-activity context features. Note that SVM+BOW works on video clips; while other methods work on continuous videos.

the other methods. The results are expected since the intra-activity and inter-activity context give the model additional information about the activities beyond the motion information encoded in low-level features. SFG approach models the spatial and temporal relationships between the low-level features and thus takes into account the local structure of the scene. However, it does not consider the relationships between various activities and thus our method outperforms the SFGs. Fig. 3.14 shows examples that demonstrate the significance of context in activity recognition.

3.6.3.2 Recognition Results on VIRAT Release 2

We work on VIRAT Release 2 to further evaluate the effectiveness of the proposed approach. We follow the method defined above to get the recognition results on this dataset. Fig. 3.15 compares the recognition accuracy using precision and recall for different methods. We can see that the performance of our method is comparable to that in [2]. In [2], an SPN on BOW is learned to explore the context among motion features. However, [2] works on video clips, each containing an activity of interest with additional 10 seconds occurring randomly

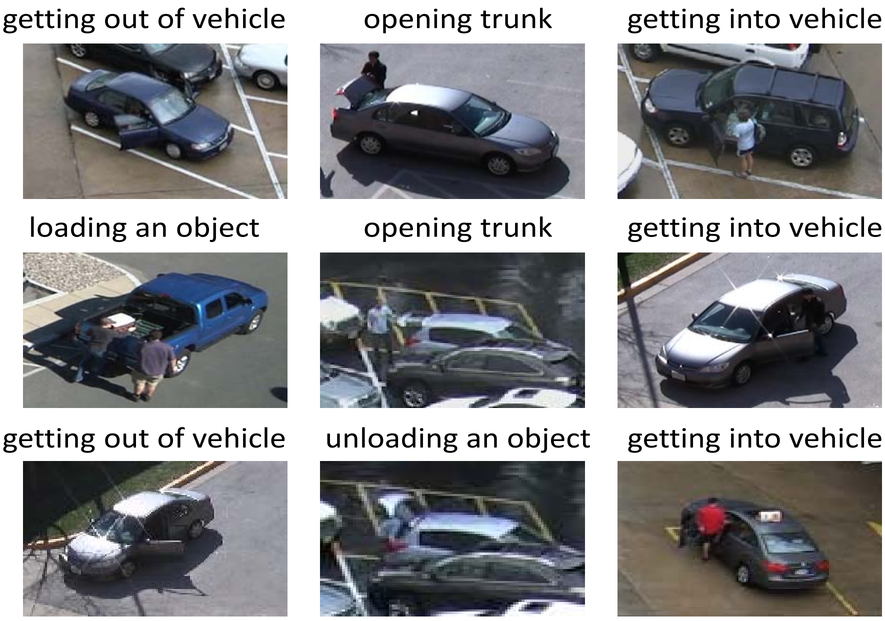


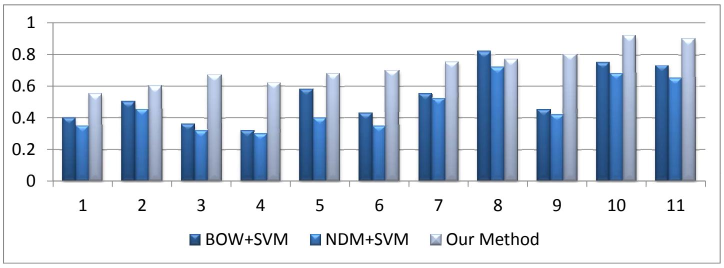
Figure 3.14: Example activities (from VIRAT Release 1) correctly recognized by baseline classifier (top), incorrectly by baseline classifier but corrected using intra-activity context (middle), and incorrectly recognized by baseline classifier and intra-activity context, but rectified using inter-activity context (bottom).

	BOW+SVM[75]	SPN[2]	Our Method
Precision	52.3	72	71.8
Recall	55.4	70	73.5

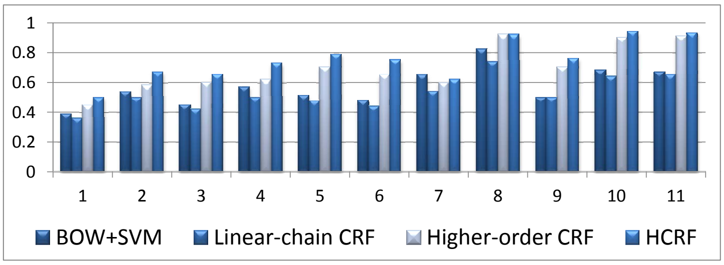
Figure 3.15: Precision and recall (in parenthesis) for different methods (averaged across activities).

before or after the target activity instance, while we work on continuous video.

Fig. 5.5 compares the precision and recall for the eleven activities defined in VIRAT Release 2 for BOW+SVM method, the baseline classifier, and our method. We see that by modeling the relationships between activities, those with strong context patterns, such as "person closing a vehicle trunk"(4) and "person running"(9), achieve larger performance gain compared to activities with weak context patterns such as "person gesturing"(7). Fig. 3.17 shows example results on activities in Release 2.



(a)



(b)

Figure 3.16: Precision (a) and recall (b) for the eleven activities defined in VIRAT Release 2.

3.6.4 Hierarchical-CRF

The goal of our framework is to locate and recognize activities of interest in continuous videos using both motion and context information about the activities; therefore, datasets with segmented video clips or independent activities like Weizmann [35], KTH [100], UT-Interaction Dataset [91] and Collective Activity Dataset [17] do not fit our evaluation goal.

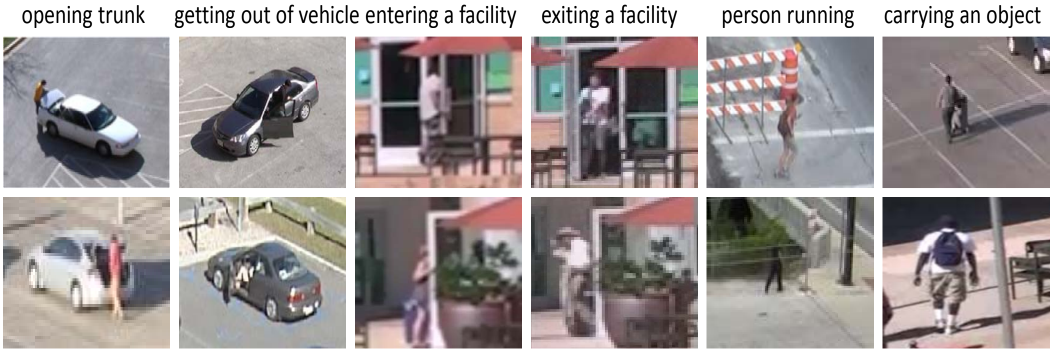
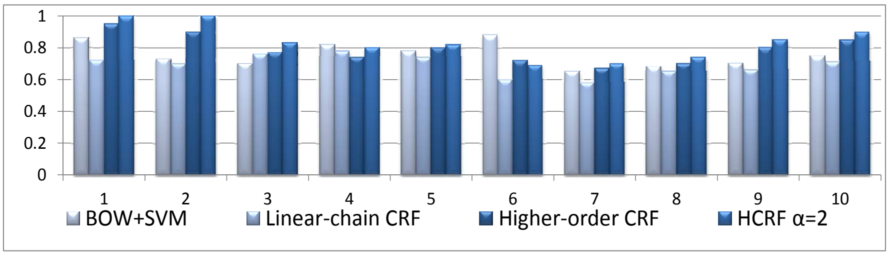


Figure 3.17: Examples (from VIRAT Release 2) in the bottom row show the effect of context features in correctly recognizing activities that were incorrectly recognized by the baseline classifier, while other examples of the same activities correctly recognized by the baseline classifier are shown in the top row.

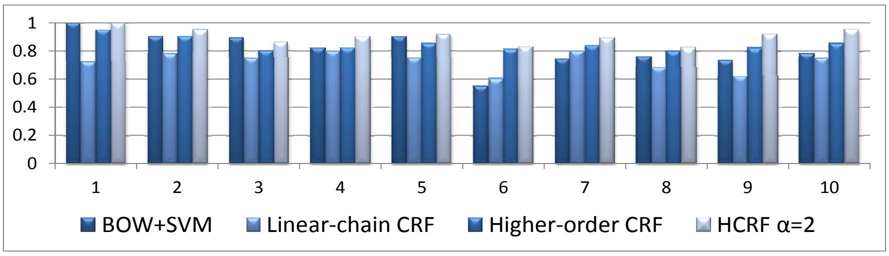
To assess the effectiveness of our framework in activity modeling and recognition, we perform experiments on two challenging datasets containing long duration videos: the UCLA office Dataset [104] and VIRAT Ground Dataset [79].

3.6.4.1 Recognition Results on UCLA Dataset

Although UCLA Dataset has been used in [104], the recognition accuracy for the office dataset has not been provided in the paper. We compare the performance of the popular BOW+SVM classifier and our model. The experiment results in precision and recall as shown in Fig. 3.18. In order to show the affects of incorporating different kinds of motion and context features, we also show results of using the action-based linear-chain CRF approach and the action-based higher-order CRF approach (Fig. 3.6 (a) and 3.6 (b)). It can be seen that the use of intra-activity context increases the recognition accuracy of activities with obvious context patterns. For example, "enter room" is characterized by the context that the agent appears at the door. The increased recognition accuracy of "enter room" by using intra-activity context features indicates that our model successfully captures this characteristics. From the performance of higher-order CRF approach and Hierarchical-CRF approach, we can see that for activities with strong spatio-temporal patterns, such as "pick phone" and "place phone down", modeling the inter-activity spatio-temporal relationships increases the recognition accuracy significantly. Next, we change the value of α to see how it influences the recognition accuracy of the Hierarchical-CRF approach. Fig. 3.19 compares the overall accuracy of different methods and the Hierarchical-CRF approach with different α values. From the results, we can see that Hierarchical-CRF approach with $\alpha = 2$ outperforms other models. This is expected. When α is too small, the spatio-temporal relationships of related activities are not fully utilized, while Hierarchical-CRF with fully connected activity layer models the spatio-temporal relationships



(a)



(b)

Figure 3.18: Precision (a) and recall (b) for the ten activities in UCLA Office Dataset. The activities are defined in Section 3.6.1.1. HCRF is the short of Hierarchical-CRF.

of unrelated activities. For instance, in the UCLA office Dataset, one typical temporal pattern of activities is a person sits down to work on the laptop, then, the same person stands up to do other things, and then sits down to work on the laptop. All these activities are conducted sequentially. Thus, Hierarchical-CRF model with fully connected activity layer captures the false temporal pattern of "stand up" followed by "work on the laptop". The optimum value of α can be obtained using cross validation on the training data.

Method	Overall	Average per-class
BOW+SVM	82.2	80.7
Linear-chain CRF	73.6	72.5
Higher-order CRF	83.9	84.3
HCRF ($\alpha = 1$)	87.9	87.1
HCRF ($\alpha = 2$)	89.9	90.8
HCRF (fully connected)	73.5	74.2

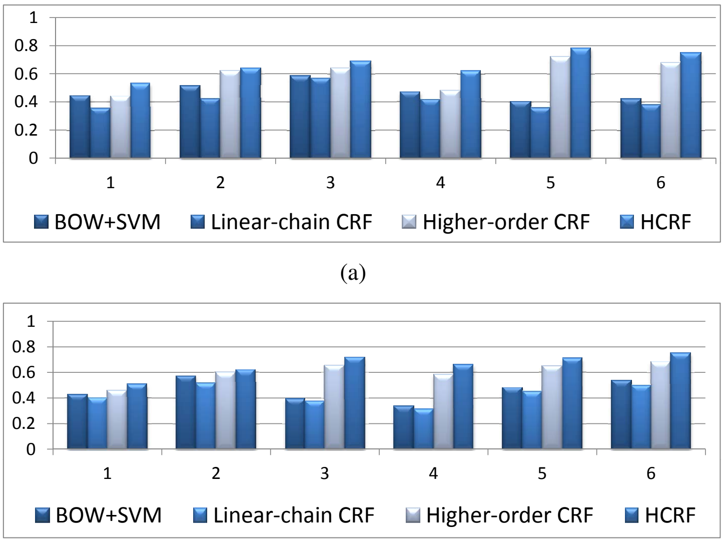
Figure 3.19: Overall and average per-class accuracy for different methods on UCLA Office Dataset. The BOW+SVM method is tested on video clips, while other results are in the framework of our proposed action-based CRF models upon automatically detected action segments. HCRF is the short of Hierarchical-CRF.

3.6.5 Recognition Results on VIRAT Release 1

Fig. 3.20 compares the precision and recall for the six activities defined in VIRAT Release 1 using BOW+SVM method and our approach with different kinds of features. The results show, as expected, the recognition accuracy increases by encoding the various context features. For instance, the higher-order CRF approach encodes intra-activity context patterns of activities of interest. Thus, activities with strong intra-activity context pattern, such as "person getting into vehicle", are better recognized by the higher-order CRF approach than by the linea-chain CRF approach, which does not model intra-activity context of activities. The Hierarchical-CRF approach further encodes inter-activity context patterns of activities. Thus, activities with strong spatio-temporal relationships with each other are better recognized by the Hierarchical-CRF approach. For instance, the higher-order CRF approach often confuses "open a vehicle trunk" and "close a vehicle trunk" with each other. However, if the two activities happen closely in time in the same place, the first activity in time is probably "open a vehicle trunk". This kind of contextual information within and across activity classes are captured by the Hierarchical-CRF approach and used to improve the recognition performance. Fig. 3.21 shows examples that demonstrate the significance of context in activity recognition.

We also show the results on VIRAT Release 1 for different methods using overall and average accuracy in Fig. 3.22. We have compared our results with the popular BOW+SVM approach, the more recently proposed String-of-Feature-Graphs approach [36] and structural model in [140].

The Hierarchical-CRF approach outperforms the other methods. The results are expected since the intra-activity and inter-activity context within and between action and activities give the model additional information about the activities of interest beyond the motion infor-



(a)

(b)

Figure 3.20: Precision (a) and recall (b) for the six activities defined in VIRAT Release 1.

mation encoded in low-level features. SFG approach models the spatial and temporal relationships between the low-level features and thus takes into account the local structure of the scene; However, it does not consider the relationships between various activities and thus our method outperforms the SFGs. Structural model in [140] models the intra and inter context within and between activities, however, it does not model the action layer and the interactions between action and activities.

3.6.6 Recognition Results on VIRAT Release 2

VIRAT Release 2 defines additional activities of interest. We work on VIRAT Release 2 to further evaluate the effectiveness of the proposed approach. We follow the method defined above to get the recognition results on this dataset. Fig. 5.5 compares the precision and recall for the eleven activities defined in VIRAT Release 2 for BOW+SVM method, the structural model in [140], and our method. We see that by modeling the relationships between activities, those with strong context patterns, such as “person closing a vehicle trunk”(4) and “person

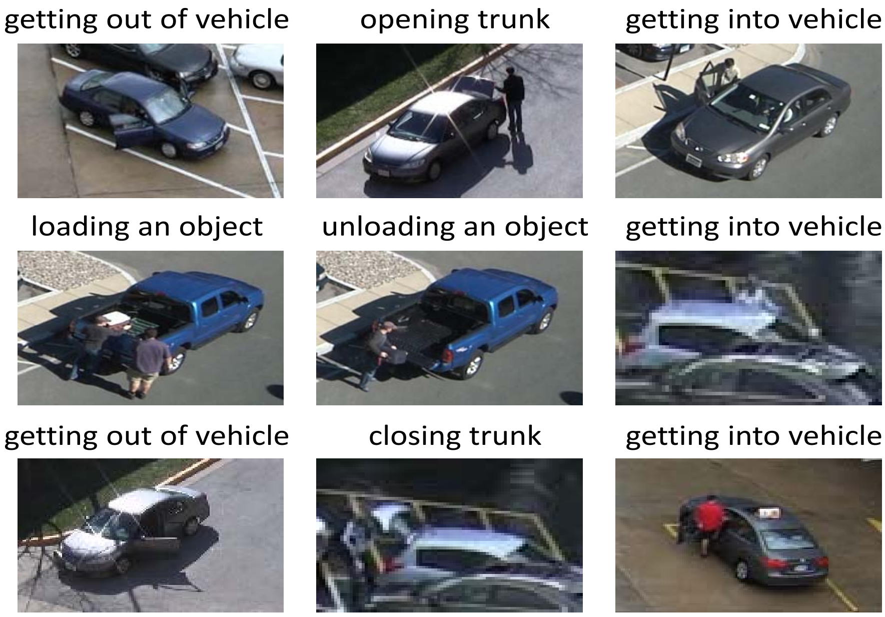


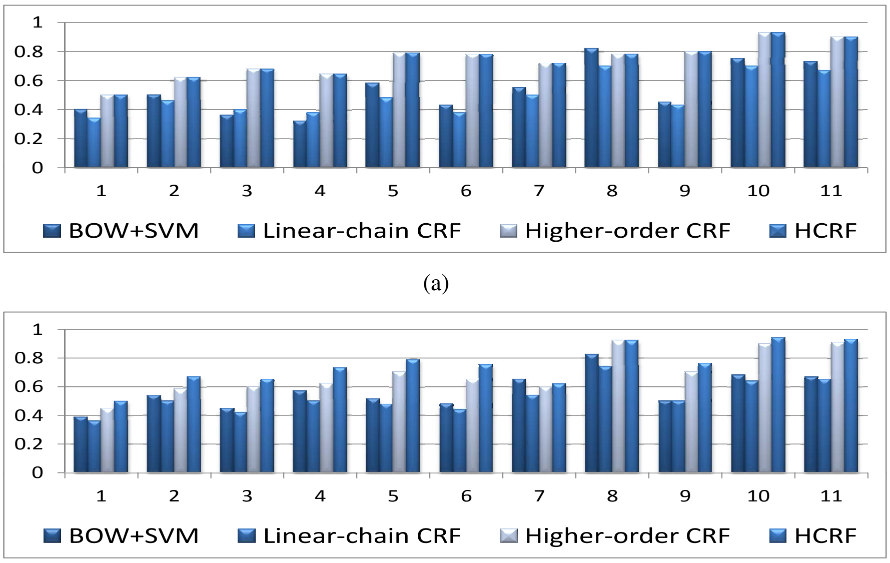
Figure 3.21: Example activities (defined in VIRAT Release 1) correctly recognized by action-based linear-chain CRF (top), incorrectly by linear-chain CRF but corrected using higher-order CRF with intra-activity context (middle), and incorrectly recognized by higher-order CRF, but rectified using action-based hierarchical CRF with inter-activity context (bottom).

running"(9), achieve larger performance gain compared to activities with weak context patterns such as "person gesturing"(7).

Fig. 3.24 compares the recognition accuracy using recall for different methods. We can see that the performance of our Hierarchical-CRF approach is comparable to the recently proposed method in [2]. In [2], a SPN on BOW is learned to explore the context among motion

Method	average accuracy
BOW+SVM [75]	45.8
SFG [138]	57.6
Structural Model [140]	62.9
Linear-chain CRF	42.6
Higher-order CRF	60.4
Hierarchical-CRF	66.2

Figure 3.22: Average accuracy for the six activities defined in VIRAT Release 1. Note that SVM+BOW works on video clips; while other methods work on continuous videos. Note that BOW+SVM works on video clip while others work on continuous video.



(a)

(b)

Figure 3.23: Precision (a) and recall (b) for the eleven activities defined in VIRAT Release 2.

Method	average accuracy
BOW+SVM [75]	55.4
SPN [2]	70
Structural Model [140]	73.5
Linear-chain CRF	52.5
Higher-order CRF	69.4
Hierarchical-CRF	75.1

Figure 3.24: Average accuracy (in recall) for different methods.

features. However, [2] works on video clips, each containing an activity of interest with additional 10 seconds occurring randomly before or after the target activity instance, while we work on continuous video.

3.7 Conclusion

In this chapter, we present novel graphical models to jointly model a variable number of activities in continuous videos. We have addressed the problem of automatic motion segmentation based on low-level motion features and the problem of high-level representations of activities in the scene. Upon the detected activity elements, we can build high-level graphical models that integrates various features within and between activities. The models explicitly learns the activity durations and motion patterns for each activity class as well as the context patterns within and across action and activities of different classes from training activity sets. It has been demonstrated that joint modeling of activities by encapsulating object interactions and spatial and temporal relationships of activity classes can significantly improve the recognition accuracy. Our experiments have shown our superior performance over other competing methods. The proposed graphical models can utilize more sophisticated baseline classifier can be used to improve the activity recognition accuracy.

It is worth noticing that more complex activities can be modeled by adding additional layers to the hierarchical-CRF model. However, the additional layers increase the learning and inference complexity by increases the tree width. Balance between the representation power of the hierarchical model and the computational complexity of the model should be achieved.

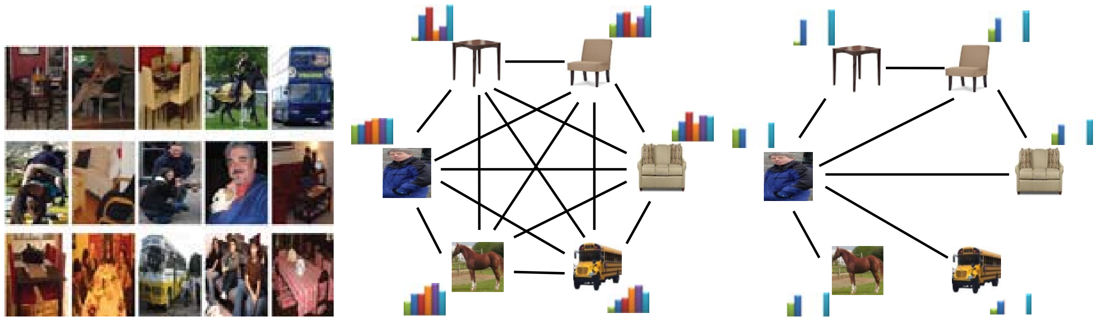
Chapter 4

Learning Sparse Graphical Representation

In this chapter, we develop a framework of sparse modeling for the joint recognition of inter-dependent visual objects (such as activities and objects in images/videos) based on context-aware graphical models and $l1$-group regularization. We evaluate our approach on two important visual recognition tasks: object recognition and activity recognition in natural scenes. The experimental results demonstrate the benefits of using the proposed sparse modeling approach for the two visual recognition tasks over the state-of-the-art methods.

4.1 Introduction

Sparse feature selection has been addressed in many works using graphical models for visual recognition. While sparse features are considered in context-aware graphical models, sparse graphical structure should also be preferred. Considering object recognition in images and videos of natural scenes, certain classes of visual objects may not be related to each other in an informative way. For instance, in images of natural scenes, a dinning table often coexists with chairs, but the coexistence relationships between bottle and boat seem rare and random. In natural videos, an activity of “a person sitting down” is often closely related to the same person “standing up” in both space and time, but is not clearly related to the person “kicking a ball”. Thus, relationships between closely related visual objects should be modeled. In terms of the graph structure, this preference enforces a sparse graphical structure as in Fig. 4.1, in which unrelated object classes are not connected.



(a) Dataset Images (b) Dense Graphical Model (c) Sparse Graphical Model

Figure 4.1: Illustration on the proposed sparse modeling for context-aware visual recognition. Starting with candidate contextual features represented by a dense graph, our goal is to automatically learn an optimum sparse graphical model, with both sparse contextual features and as well as a sparse graphical structure, capturing the informative contextual patterns within and between object classes. Node that the two object classes may be connected by multi-edges, representing different types of relationships.

Advantages of sparse modeling for context-aware graphical models are obvious. When the model features are sparse, it would be more efficient and effective to estimate the parameters, and provide higher accuracy by concentrating on informative features and avoiding noises induced by correlated features [51]. Furthermore, inter-relationships between different objects are naturally sparse, especially when the number of involved objects is large. By enforcing a sparse graphical structure, we may be able to learn the intrinsic structure of relationships between different objects to be recognized and exclude the relationships of occasionally related objects from the learned graphical model. This would potentially improve recognition accuracy by focusing on the unary features of such objects when they occur together in the scene.

However, few existing works on context-aware graphical models for visual pattern recognition systematically address the problem of sparse modeling.

4.1.1 Sparse Modeling Approach

l1-regularization techniques have been widely used to enforce sparsity in the solutions of estimation problems [68, 123, 128]. When the penalty on the l1-regularization is strong enough, many of the parameters in the optimal solution will be forced to be zero. It has been demonstrated by many researches that l1-regularized linear regression can outperform l2-regularized regression, especially when a large amount of redundant information exists in the features [24, 51, 115].

Most existing works on sparse modeling for context-aware graphical models in computer vision use element-wise l1 regularized graphical models [22, 137] or a graphical model with a predefined sparsity degree [58]. However, element-wise sparsity may not be the most optimal representation of the features or graph structure. At the feature level, if a set of intra-object contextual features related to the same contextual object is redundant or not discriminative, we would rather not use this group of features for the labeling and thus prefer the group of associated model parameters to be zeroes. As an example, when an candidate contextual objects do not help distinguish the activities to be recognized, it would be better to remove all features related to the contextual objects from the learned model. Doing this also decreases the pre-processing computation by eliminating the feature extraction for the contextual object as a whole. Furthermore, features encoding the relationships between objects to be recognized can be grouped according to the related object pair. Enforcing sparsity on parameter groups of these relationship features results in a graphical model with a sparse graphical structure. In multi-edge graphical models like those in [22, 140], two nodes may be connected by multiple edges associated with different pairwise parameters. For instance, in activity recognition, pairwise parameters represent the spatial and temporal relationships between activities. Two related activities usually are connected by both spatial and temporal relationships. Element-wise l1-regularization does not necessarily enforce a sparse graphical structure, even if the intrinsic graphical structure is sparse.

Group lasso, as an extension of lasso, does group-level feature selection on predefined parameter groups. It has been applied successfully in linear regression and logistic regression [30, 70, 105] to enforce the sparsity of model parameters at a group level. Group $l1$- regularization extends group lasso, by using $l1$ penalty at the group level and different norms ($l1$, $l2$ and ∞) for element-wise penalty within parameter groups (within-group penalty), to enforce group sparsity of the model parameters, as well as the desirable property on element-wise parameters.

In this work, based on the existing graphical models for context-aware recognition of visual objects, we utilize *group $l1$-regularization* to enforce the model sparsity. Without loss of generality, we choose the popular CRF model [22, 58, 140] as our baseline model for its simplicity while concentrating on showing how our sparse modeling approach can be incorporated.

Candidate regions of visual objects are firstly identified using existing object detection methods upon low-level features. We call the results as the preliminary detection results. Various candidate attributes and inter-relationships between visual objects are identified and used for the development of contextual feature vectors of the candidate visual objects. Then, group $l1$-regularized model learning is proposed to automatically select the most informative contextual features within and between object classes. In the experiment, we work on two challenging tasks of visual pattern recognition: activity recognition in continuous videos and object recognition in natural scene, and show state-of-art performance of the proposed model.

4.1.2 Contributions of The Present Work

The main contribution of this work is two-fold.

1) We propose a framework of sparse modeling for the recognition of potentially inter-dependent visual objects, by introducing group $l1$-regularization for feature selection on existing CRF models. This framework selects the optimal set of contextual features, *as well as* the optimal sparse graph structure, for the discriminative modeling of the inter-dependent visual objects. The proposed sparse modeling approach can be easily adapted to work with graphical models with hidden variables such as HCRF.

2) We formulate the learning procedure as a non-smooth convex optimization problem. Based on modified bundle methods [113], we propose a two-stage cutting-plane-based algorithm that iteratively searches for the increasingly tight lower bounds of the objective function until convergence.

4.2 Formulation of Sparse Models

In this section, we describe the standard context-aware graphical modeling and recognition of inter-dependent visual objects. Then, the problem of sparse modeling is formulated based on the discriminative training approach of the graphical model [140] and the group $l1$-regularization.

4.2.1 Model Features

The context-aware visual recognition models usually involves three kinds of features: Intra-object Intrinsic Features (Intra-IF), Intra-object Contextual Features (Intra-CF) and Inter-object Contextual Features (Inter-CF). We define these kinds of features used in our work as below. How these features are developed for specific recognition tasks are described in the experiment section.

Intra-IF of a visual object describes the object's intrinsic characteristics. These features are closely related to low-level features of visual objects that are used to generate candidate regions, each expected to contain one object of interest. Both supervised features such as STIP features for motion in video, as well as features learned using unsupervised techniques such as Topographic ICA [62], may be used as the low-level features. With candidate regions, any probabilistic multi-classifier is used to generate the classification scores for each candidate region. Then, a normalized score vector is developed as the Intra-IF for each candidate region of visual object or object part.

Intra-CF of a visual object is developed from attributes that define object's context. This kind of features may not be the intrinsic characteristics of its object class but probably can help improve object recognition performance. All intra-object features other than those used for preliminary detection and classification are considered as Intra-CF. Examples of such features include features encoding scene label information and object attributes in activity recognition. We collect all candidate contextual attributes which are potentially helpful, categorize them and develop an attribute vector for each contextual category.

Inter-CF are developed from attributes that capture the relations between visual objects. These features can encode the object layout patterns in images and temporal and spatial relations between activities. Similar to [22, 140], we develop a normalized feature vector that encodes the interactions between a visual object and its surrounding objects to be recognized as its Inter-CF feature.

4.2.2 The Context-Aware CRF Model

The problem of visual object recognition in natural scenes requires two main tasks: to detect candidate regions and to label these detected regions. The detection and labeling problems can be solved simultaneously as proposed in [76] or separately as proposed in [22, 140]. For the latter, candidate regions are usually detected before the labeling task (how to detect the candidate regions are task specific and will be described in the experiment section). The problem of object recognition is then converted to a problem of labeling, that is, to assign each candidate region with an optimum class label. We propose a context-aware CRF model for the labeling of inter-dependent visual objects.

CRF model is frequently used for the problem of labeling in computer vision. We now describe the context-aware CRF model for the labeling of inter-dependent objects based on [22, 138], which jointly models related instances through integrating features of individual instances with features representing inter-relationships between instances. Let **a** be the visual objects to be labeled as well as the model observations. Let **y** be the label variables. The posterior distribution $p(\mathbf{y}|\mathbf{a},\omega)$ of the label variables over the CRF is a *Gibbs* distribution and is usually represented as

$$p(\mathbf{y}|\mathbf{a},\omega) = \frac{1}{Z(\mathbf{a},\omega)} \prod_{c \in C} exp({\omega_c}^T \varphi_c(\mathbf{a},\mathbf{y}_c)), \tag{4.1}$$

where ω_c is the model parameter called weight vector, which needs to be learned from the training data. $Z(\mathbf{a},\omega)$ is a normalizing constant called the partition function. $\varphi_c(\mathbf{a},\mathbf{y}_c)$ is a feature vector derived from the observation **a** and the label vector $\mathbf{y}_c$ of clique c.

Suppose we are interested in M instance classes (a background class that does not belong to any classes of interest may be introduced). An instance set $\mathbf{a} = \{a_i : i = 1,...,N\}$ is associated with a label vector $\mathbf{y} = \{y_i : i = 1,...,N\}$, where $y_i \in \{1,...,M\}$ is the label of a_i. We model the instance set by the combination of features of individual instances and context features within and between instances. Let $\mathscr{V}$ be the set of vertices, each representing a candidate region to be labeled. Let $\mathscr{E}$ denote the set of connected object pairs. For pattern recognition tasks, given the observations **a** and model weight vector ω, the CRF in (4.1) is usually represented by a potential function defined as

$$\begin{aligned}\psi(\mathbf{a},\mathbf{y}) &= \sum_{i\in\mathscr{V}} \psi_x(\mathbf{a},y_i) + \sum_{i\in\mathscr{V}} \psi_g(\mathbf{a},y_i) + \sum_{ij\in\mathscr{E}} \psi_d(\mathbf{a},y_i,y_j)\\ &= \sum_{i=1}^{N} \omega_{x,y_i}^T \varphi(\mathbf{x}_i,y_i) + \sum_{i=1}^{N} \omega_{g,y_i}^T \vartheta(\mathbf{g}_i,y_i)\\ &+ \sum_{i,j=1,i\neq j}^{N} \omega_{d,(y_i,y_j)}^T \phi(\mathbf{d}_{ij},y_i,y_j),\end{aligned} \tag{4.2}$$

where $\psi_x(\mathbf{a},y_i)$ is the Intra-IF potential that measures the compatibility between Intra-IF of a_i and its label y_i. $\psi_g(\mathbf{a},y_i)$ is the Intra-CF potential that measures the compatibility between Intra-CF of a_i and its label y_i. $\psi_d(\mathbf{a},y_i,y_j)$ is the Inter-CF potential that measures the consistency between two connected visual objects i and j and their labels. These potential functions are developed as a linear function of related features. $\mathbf{x}_i \in R^{D_x}$ and $\mathbf{g}_i \in R^{D_g}$ are the intra-instance feature and intra-object context feature of a_i, D_x and D_g are the dimension of $\mathbf{x}_i$ and $\mathbf{g}_i$ respectively. $\omega_{x,y_i} \in R^{D_x}$ and $\omega_{g,y_i} \in R^{D_g}$ are the weights that capture the valid intra-object pattern and intra-activity context patterns of instance class y_i. $\mathbf{d}_{ij} \in R^{D_d}$ is the inter-object context features associated a_i and a_j. D_d is the dimension of $\mathbf{d}_{ij}$. $\omega_{d,(y_i,y_j)} \in R^{D_d}$ are the weights that capture the valid inter-relationships between object classes y_i and y_j. In general, dimensions of the same kind of feature can be different for each object class/class pairs.

In order to form a linear function with a single parameter, we rewrite (4.2) as:

$$\begin{aligned}\psi(\mathbf{a},\mathbf{y}) &= \omega_x^T \sum_{i=1}^{N} \varphi(\mathbf{x}_i,y_i) + \omega_g^T \sum_{i=1}^{N} \vartheta(\mathbf{g}_i,y_i) \qquad (4.3)\\ &+ \omega_d^T \sum_{i,j=1,i\neq j}^{N} \phi(\mathbf{d}_{ij},y_i,y_j),\end{aligned}$$

where ω_x, ω_g and ω_d are weight vectors defined as

$$\omega_x = \begin{bmatrix} \omega_{x,1}^T & \omega_{x,2}^T & \cdots & \omega_{x,M}^T \end{bmatrix}^T,$$

$$\omega_g = \begin{bmatrix} \omega_{g,1}^T & \omega_{g,2}^T & \cdots & \omega_{g,M}^T \end{bmatrix}^T,$$

$$\omega_d = \begin{bmatrix} \omega_{d,(1,1)}^T & \cdots & \omega_{d,(1,M)}^T & \omega_{d,(2,2)}^T & \cdots & \omega_{d,(M,M)}^T \end{bmatrix}^T,$$

and $\varphi(\mathbf{x}_i, y_i)$ and $\vartheta(\mathbf{g}_i, y_i)$ have non-zero entries at the positions corresponding to class index y_i. $\psi(\mathbf{d}_{ij}, y_i, y_j)$ has none-zero entries at the positions corresponding to class pair (y_i, y_j).

Define the joint weight vector ω and joint feature vector $\Gamma(a,y)$ as

$$\omega = \begin{bmatrix} \omega_x \\ \omega_g \\ \omega_d \end{bmatrix}, \Gamma(a,y) = \begin{bmatrix} \sum_i \varphi(\mathbf{x}_i, y_i) \\ \sum_i \vartheta(\mathbf{g}_i, y_i) \\ \sum_{i,j,i \neq j} \phi(\mathbf{d}_{ij}, y_i, y_j) \end{bmatrix}, \tag{4.4}$$

where $i, j = 1, ..., N$. Then, the optimum label $\mathbf{y}^{opt}$ of $\mathbf{x}$ is obtained as

$$\mathbf{y}^{opt} = \arg\max_{\mathbf{y}} \psi(\mathbf{y}|\mathbf{a}, \omega) = \arg\max_{\mathbf{y}} (\omega^T \Gamma(\mathbf{a}, \mathbf{y})). \tag{4.5}$$

4.2.3 Sparse Modeling

Structural-SVM training schemes are proposed to learn the model parameters of MRF and CRF models [22, 58]. These schemes directly maximize the recognition accuracy through an objective function that minimizes the upper bound of the classification error on training data. We choose such a task-oriented discriminative approach to train the model in such a way that it increases the average precision scores on the training data and thus tend to produce the correct activity labels for each detected visual object. This goal can be achieved by finding the parameter vector ω that minimizes the sum of empirical risks on the training set. The empirical risk $\Delta(\mathbf{y}(i), \widehat{\mathbf{y}}(i))$ of labeling the instance set $\mathbf{a}(i)$ with $\widehat{\mathbf{y}}(i)$ is defined as the sum of 0-1 loss of labeling each instance in $\mathbf{a}(i)$ (detections overlap true positives more than 50% are not penalized). Also, we want the potential function ψ to score higher for true label $\mathbf{y}(i)$ of $\mathbf{a}(i)$ than for all other hypothesized label $\widehat{\mathbf{y}}(i)$, i.e., $\psi(\mathbf{y}(i)|\mathbf{a}(i), \omega) \geq \psi(\widehat{\mathbf{y}}(i)|\mathbf{a}(i), \omega)$, where ψ is the potential function defined in (4.5).

Furthermore, sparsity of unary parameters in the model directly correspond to sparsity of Intra-IF and Intra-CF, and the sparsity of the edge (pairwise) parameters directly correspond to the sparsity of Inter-CF as well as the graph structure. As explained in the introduction section, we prefer parameter sparsity at the group level, as well as particular element-wise parameter properties within parameter groups. Thus the sparse modeling is achieved by solving the group $l1$-regularized optimization problem as

$$
\begin{aligned}
\omega^* &= \arg\min_{\omega} f(\omega) = \arg\min_{\omega} R(\omega) + \Omega(\omega), \qquad (4.6)\\
\textit{where} \quad \Omega(\omega) &= \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} \max\left(0, \xi_{\omega}(i)\right),\\
R(\omega) &= \lambda_x \|\omega_x\|_{p_x} + \lambda_g \sum_{l=1}^{N_g} \|\omega_{g_l}\|_{p_g} + \lambda_d \sum_{i=1}^{M} \sum_{j=1}^{M} \|\omega_{d_{ij}}\|_{p_d},\\
\xi_{\omega}(i) &= \max_{\widehat{\mathbf{y}}(i)} \left(\Delta\left(\mathbf{y}(i), \widehat{\mathbf{y}}(i)\right)\right.\\
&\quad \left. + \omega^T \left(\Gamma\left(\mathbf{a}(i), \widehat{\mathbf{y}}(i)\right) - \Gamma\left(\mathbf{a}(i), \mathbf{y}(i)\right)\right)\right),
\end{aligned}
$$

where N_{train} is the number of training sets and N_g is the number of contextual attribute groups. p_x, p_g and p_d are the p-norms used for Intra-IF, Intra-CF and Inter-CF, respectively. $\lambda_{\cdot}$ is the penalty regularization parameter of corresponding parameter group. If $\lambda_{\cdot}$ is large, many groups of parameterswill be forced to be zero, achieving feature sparsity at the group level. (4.6) is group $l1$-regularized in the sense that $l1$ norm is used to sum up the regularizers of grouped parameters.

p_x, p_g and p_d are the norms used for within-group penalties. Generally, p_x, p_g and p_d can be any of $l1$, $l2$ and ∞-norms. $l1$-norm prefers sparsity within the groups, and can be used for parameter groups for which sparsity is expected to be beneficial. For instance, for a large parameter group, the associated features is liekly to be correlated and redundant, element-wise sparsity and thus $l1$-penalty is preferred for this group. $l2$-norm does not place bias on the direction and usually leads to dense parameters within the group with l2-norm. For instance, element-wise sparsity is not likely to be beneficial for a compact parameter group associated with un-correlated features, and $l2$-penalty should be preferred. ∞ norm tends to force all parameters having the same weights and is not proper for our problem. When within-group norms are all $l2$-norm, feature sparsity at group level is guaranteed. And the regularization format is the same as group lasso [70]. When within-group norms are all $l1$-norm, feature sparsity at both group and individual levels is guaranteed. And the effect on parameter sparsity is the same as sparse group lasso in [30, 105]. Other cases are between the two. In the experiments, we evaluate the performance of different within group penalties in different applications.

4.3 Sparse Model Learning and Inference

In this section, we first show how to solve the sparse modeling problem in (4.6) using the modified bundle method. Then, top-k greedy search method used for model inference is described.

4.3.1 Optimization Algorithm

The non-smooth regularized risk minimization problems can be solved efficiently by bundle methods [113] if the regularizers in the objective function are of the same type. However, the regularizer $R(\omega)$ in (4.6) can be the summation of $l1$ and $l2$ norms. In this section, we extend the bundle method to solve (4.6). The convergence of the proposed approach is guaranteed by the convergence of the bundle method [113].

Let $\partial_\omega \Omega(\omega)$ denote the sub-gradient of $\Omega(\omega)$. The cutting plane of $\Omega(\omega)$ at ω, denoted as g_ω, is defined by its first-order Taylor approximation with its sub-gradient as

$$
\begin{aligned}
g_\omega = \omega^T \partial_\omega \Omega(\omega) + b_\omega; \quad b_\omega = \Omega(\omega) - \omega^T \partial_\omega \Omega(\omega),\\
\partial_\omega \Omega(\omega) = \sum_{i=1}^{N_{train}} \delta(i) \left[\Gamma(\mathbf{a}(i), \mathbf{y}^*(i)) - \Gamma(\mathbf{a}(i), \mathbf{y}(i)) \right],
\end{aligned}
$$

where $\delta(i)$ is an indicator function whose value is 1 if $\xi_\omega(i) > 0$ and 0 otherwise. $\mathbf{y}^*(i)$ is the most violated label of $\mathbf{a}(i)$, which maximizes $\xi_\omega(i)$.

Similar to [113], the proposed algorithm approximates $\Omega(\omega)$ with an increasingly tight piecewise linear function, which consists of a set of m iteratively found cutting planes of $\Omega(\omega)$. m is a predefined constant and can be decreased as the learning algorithm approaching convergence. The overall approach for solving problem (4.6) is summarized in Algorithm 7.

4.3.1.1 Iterative Soft-Thresholding Sub-Routine

In Algorithm 7, at t^{th} iteration, we need to solve the sub-routine problem

$$
\begin{aligned}
\omega_{t+1} = arg \min_{\omega} \psi_{\omega_t+1}(\omega) = arg \min_{\omega} R(\omega) + G_{\omega_t}(\omega), \qquad (4.7)\\
where \quad G_{\omega_t}(\omega) = max\left(0, max_{j=t-m+1,\ldots,t} g_{\omega_j}(\omega)\right).
\end{aligned}
$$

Algorithm 7 Solve (4.6) using bundle method [113].

```
Input:   $S = ((a(1),y(1)),\dots,(a(N_{train}),y(N_{train}))),\lambda,\varepsilon$
Output:  Optimum weight vector $\omega$

1. Initialize $\omega$ as $\omega_0$ using empirical values, $\mathscr{G}$(cutting plane set) $\leftarrow \emptyset$.
2. for $t = 0$ to $\infty$ do
3.     find the cutting plane $g_{\omega_t}$ of $\Omega(\omega)$ at $\omega$.
4.     $\mathscr{G} \leftarrow \mathscr{G} \cup g_{\omega_t}(\omega)$;
5.     update $\omega$:
           $\omega_{t+1} = \arg\min_{\omega} \psi_{\omega_t}(\omega)$;
           $\psi_{\omega_t}(\omega) = R(\omega) + max(0, max_{j=t-m+1,\dots,t} g_{\omega_j}(\omega))$;    $R(\omega)$ is the model regularizer;
6.     $gap_{t+1} = \min_{(t-t')<m} \psi(\omega_{t'}) - \psi_{\omega_t}(\omega_{t+1})$;
7.     if $gap_{t+1} \le \varepsilon$, then return $\omega_{t+1}$;
8. end for
```

$R(\omega)$ has components of $l1$-regularizer, which is not differentiable when all the parameters in the group equal to zeros. Two-metric Projection is usually used to cast the elements of $l1$-norm to positive and negative parts and the resulting problem can be solved as a smooth constrained convex problem [9]. The obvious drawback of this approach is that it increases the size of model parameters. To overcome this drawback, we refer to iterative soft-thresholding [127], which solves the non-smooth optimization problem directly using a projective-like operator very efficiently. Since $G_{\omega_{t-1}}(\omega_t{}^k)$ is a piecewise linear function of model parameter $\omega_t{}^k$, we can use its first-order approximation in the optimization procedure. By regularizing the distance between consecutive solutions, (4.7) can be tailored to solve a subproblem iteratively, generating a sequence of iterates $\{\omega_t^k, k = 0, 1, \cdots\}$. The subproblem is defined as

$$\begin{aligned}\omega_{\mathbf{t}}{}^{k+1} =& arg\min_{\mathbf{z}} G_{\omega_{t-1}}(\omega_t{}^k) + (z - \omega_t{}^k)^T \partial_\omega G_{\omega_{t-1}}(\omega_t{}^k)\\ &+ \frac{1}{2a_t^k}\|z - \omega_t{}^k\|_2^2 + R(\mathbf{z}).\end{aligned} \tag{4.8}$$

The above equation is equivalent to solve the problem

$$\omega_{\mathbf{t}}{}^{k+1} = arg\min_{\mathbf{z}} \frac{1}{2}\|\mathbf{z} - \mathbf{u}_t^k\|_2^2 + a_t^k R(\mathbf{z}), \tag{4.9}$$

$$where \quad \mathbf{u}_t^k = \omega_t{}^k - a_t^k \partial_\omega G_{\omega_{t-1}}(\omega_t{}^k).$$

The solution to (4.9) is the soft-thresholding operator $S_q(\mathbf{u}_t^k, a_t^k)$, where q denotes the dual norm of the norm regularizer used. The soft-threshold function $S_q(u,a)$ is defined as

$$S_q(u,a) = \frac{max\{\|u\|_q - a, 0\}}{max\{\|u\|_q - a, 0\} + a} u. \tag{4.10}$$

The dual norm of $l1$-norm and $l2$-norm are the ∞-norm and $l2$-norm, respectively. Thus, in our case, ω is obtained as

$$\omega_{\pi,t}^{k+1} = S_{q_\pi}(\mathbf{u}_{\pi,t}^k, a_t^k \lambda_\pi), \quad \pi \in \{x, g_l, d_{ij}\}, \tag{4.11}$$

where $i, j = 1, ..., M; l = 1, ..., N_g$, and a is the step size. The sub-routine stops when the relative change in the objective function (4.7) smaller than a predefined threshold.

Efficient Implementation: It has been demonstrated that the non-monotonic version of Armijo condition can provide a large improvement in the convergence rate of projected gradient method [7]. To expedite the convergence, we use a similar criteria to the non-monotonic version of Armijo condition that accepts a solution of (4.9) if the objective value of (4.7) is at least slightly smaller than the largest values over the past m iterations. Thus, for each subproblem in the form of (4.9), we start a_t^k with the initial value, and increase a_t^k by a factor of γ iteratively, where $\gamma > 1$, until the solution ω_t^{k+1} satisfies

$$\psi_{\omega_t}(\omega_t^{k+1}) \leq \max_{i=k-m+1:k} \psi_{\omega_t}(\omega_t^i) + \upsilon \frac{1}{2a_{\pi,t}^k} \|\triangle\omega_t^{k+1}\|_2^2,$$

where $\upsilon \in (0,1)$ is a very small constant.

4.3.2 Speeding Up the Optimization Procedure

To find the optimum model parameter in (4.6), we need to solve the problem for multiple values of the regularization factors λs. Active-set methods are usually used for regularized optimization problems to greatly speed up the computation [9]. Such methods start the optimization problem with a small set of model parameters called as active set and add the profitable parameters into the active set iteratively using certain parameter evaluation criteria. Furthermore, with closely related λs, warm-start using the previous solution as a "warm start" for the new value of λ is usually applied to the regularized optimization problem to speed up the computation [44].

Specifically, for the problem of (4.6), the optimality condition that $\mathbf{0} \in \partial f(\omega)$ for a parameter group with $\omega_\pi = 0$ is

$$||\partial_{\omega_\pi}\Omega(\omega)||_p \leq \lambda_\pi, \tag{4.12}$$

where $\pi \in \{x, g_l, d_{i,j}\}$. From (4.12), we can conclude that a parameter group with zero norm and large partial derivative $||\partial_{\omega_\pi}\Omega(\omega)||_p > \lambda_\pi$ can locally improve the objective function by moving it away from zero.

When λ is large enough, all model parameters will be forced to be zeros. Thus, we can start solving the problem with a small set of parameters. Then, by decreasing λ sequentially with a multiplication ratio of $\gamma < 1$, we can add the parameter groups with a zero norm that are not conforming to (4.12) to the active set and solve the optimization problem iteratively.

Following the above analysis, to further expedite our optimization procedure in the previous section, we introduce active-set with warm-start method to our optimization problem in Algorithm 8, by starting to solve the problem with large λ and sequentially decreasing the value until the set of model parameters to optimize does not change between iterations of the optimization procedure. The method iterates between parameter selection and problem solving using the proposed optimization algorithm in Section 4.3.1.

Algorithm 8 Speeding Up Training using Active-Set with Warm-Start

```
Input:   $S = ((a(1), y(1)), \dots, (a(N_{train}), y(N_{train})))$
Output:  Optimum model parameter $\omega$

Initialize $\lambda_\pi$ with large values, $\pi \in \{x, g_i, \varepsilon_{i,j}\}$.
for $l = 1, 2 ...$
    Find groups $\omega_\pi$ such that
        $\omega_\pi \neq \mathbf{0}$, or $\omega_\pi = \mathbf{0}$ and $||\partial_{\omega_\pi}\Omega(\omega)||_q > \lambda_\pi(l-1)$;
    if the selected parameter groups do not change between iterations
        return $\omega(l-1)$;
    else $\lambda_\pi(l) = \gamma\lambda_\pi(l-1)$
        solve for $\omega(l)$ with respect to the selected parameter groups
        with $\omega(l-1)$ as the initialization solution, using the
        optimization algorithm in section 4.3.1;
end for
```

4.3.3 Model Inference with Top-K Greedy Search

With the learned weight vector ω^*, model inference is carried out to find the best label vector for test data, as well as to find the most violated labels in each iteration [22]. We now describe how to identify the optimum label vector $\mathbf{y}$ for an object set $\mathbf{a}$ under the current model parameter ω. The difference in potential scores of the best label and the second best one for each instance can be small. Thus the greedy forward search which greedily instantiates the most discriminative object based on previously instantiated ones may fail. Inspired by top-k matching in [117], we apply the top-k greedy search for the inference procedure.

At each iteration, the proposed approach instantiates top k objects which increase the potential ψ most when being instantiated for each labeling path, based on the previously instantiated ones. And the best k labeling paths which have the highest potential scores are kept at the end of each iteration. The algorithm stops when all the regions are labeled or labeling any other segments decreases the value of compatibility function ψ. In practice this greedy search algorithm works well to find good solutions and outperforms the best greedy search algorithm [22] by about 1% when $k = 2$. The computation complexity is at the same level as the best greedy search approach, since at each iteration, the computation differs in a multiplication factor k^2, where k is a small integer.

4.4 Experiments

Our goal is to evaluate the effectiveness of the proposed sparse modeling in the selection of a powerful set of contextual features from a candidate pool as well as finding the optimum sparse graphical structure for the recognition tasks. We work on two dominant tasks in visual recognition: activity recognition in continuous videos and object recognition in natural scene, where modeling the context and inter-relationships between objects are potentially beneficial.

While different initial λ in Algorithm 8 can be used for different parameter groups, we use the same value for each group in order to simplify the problem of parameter selection. $\lambda = 2$ and $\gamma = 0.6$ are used for all experiments. Top-2 greedy search is used for model inference and $m = 10$ previous cutting planes as a typical value is used.

4.4.1 Activity Recognition

4.4.1.1 UCLA Dataset

Please refer to Section 3.6.1.1 for the UCLA Office Dataset [104]. The dataset is divided into 14 sets, each set containing 2 to 19 activities of interest, as well as varying number of background activities. We use 12 sets for training and the rest for testing via cross validation for the evaluation.

4.4.1.2 Preprocessing

Given a video, we detect the motion regions using background subtraction. Then, we divide the motion regions into spatio-temporal segments with a fixed temporal length t_{win} in frame. $t_{win} = 10$ is used in the experiment. Then, using the multi-SVM upon histogram of STIP features, we develop a 11x1 normalized score vector for each 3D motion segment. It is observed that the baseline detector tends to generate the same activity labels for segments from the same activity. Thus, labels generated by multi-SVM are smoothed and consecutive 3D motion segments with the same labels are grouped together to form the candidate activity regions. The true label of the candidate activity is the mode of the true labels of its segments. This method decreases the number of variables in the graph model and thus the size of the optimization problem over approaches using individual segments as the graph variables [140, 130]. We call the obtained activities with labels as the preliminary results. To penalize the randomly large classification score of segments of minority classes, the mean of normalized score vectors of segments is developed as the Intra-IF of the candidate activity (experimental results also show that the mean score vector outperforms using max pooling of score vectors of segments in our problem).

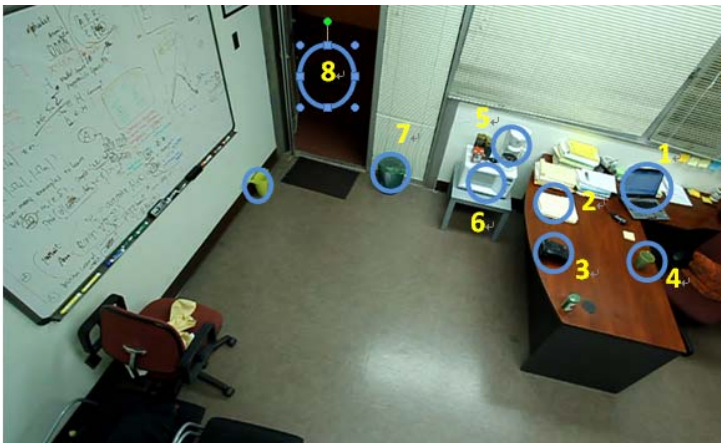


Figure 4.2: Sample image of UCLA office scene with objects of interest marked on the image.

	Attributes
g_1 g_2 g_3 g_4	A is touching / not touching laptop^1, paper^2, phone^3, cup^4 .
g_5 g_6 g_7	A is occluding / not occluding the coffee maker^{5}microwave^6 garbage can^7.
g_8 g_9 g_{10} g_{11}	A is near / far away from the coffee maker^8, microwave^9, garbage can^{10}, door^{11}.
g_{12}	A disappears / not disappears at the door.
g_{13}	A appears / not appears at the door.

Figure 4.3: Subsets of context attributes used for the development of Intra-CF for UCLA Dataset. “A” denotes the agent.

We identify 8 classes of frequent objects that are involved in actions: laptop, papers, phone, cup, coffee maker, microwave, garbage can and door. Fig. 4.2 shows a sample image with these objects marked. A 26-bin histograms of attributes are developed as the Intra-CF with 13 groups of attributes. Fig. 4.3 shows the grouped contextual attributes. Whether two activities are related are decided by their temporal distance in terms of intermediate candidate activities between them. Temporal attributes include “before”, “during” and “after”, and spatial attributes include “overlapping”, “near” and “faraway”, are considered for each activity pair. Inter-CFs are developed as attribute histograms encoding the inter-relationships between candidate activities.

4.4.1.3 Experimental Results

In order to evaluate the performance of different norms within contextual parameter groups in the proposed group sparse modeling (GSM), we fix the norm for Intra-IF parameter

Intra-CF PGs	Inter-CF PGs	mAP
L1	L1	91.2
L1	L2	90.5
L2	L1	90.3
L2	L2	88.4

Table 4.1: Comparing combinations of different norms used for parameter groups of contextual features. PG is short for parameter group.

groups, and change the norms used for different types of contextual parameter groups. Table 4.1 shows the results of activity recognition on UCLA office scene.

We can see that using $l1$-norm within each parameter group provides the best recognition accuracy. This is because the number of training instances (around one hundred) are limited compared to the dimension of model features (around one thousand). The results are similar to the conclusion in element-wise regularization that $l1$-regularization is more likely to outperform $l2$ when training instances are limited compared to the feature dimension. Since $l1$-norm works best as the within group norm, in the following experiments on UCLA dataset, we use $l1$-norm for within group penalty.

In order to evaluate the contribution of different kinds of contextual features to the recognition accuracy, we show the preliminary labeling results, GSM^1 (the proposed GSM approach using Intra-CF as the context) and GSM^2 (the proposed GSM approach using both Intra-CF and Inter-CF as the context). The results are shown in Fig. 4.4. It can be seen that the proposed GSM (group-wise regularized CRF provides better recognition accuracy using richer context, as has been demonstrated for element-wise regularized CRF in [140].

To see the benefits of the proposed sparse modeling approach over element-wise regularized graphical models, we compare our performance with And-Or graph [86], structural model [140] in Tab. 4.2. For [86, 140], we have used the average figures reported in their papers or obtained from the authors. To demonstrate the benefit of grouped sparsity in contextual

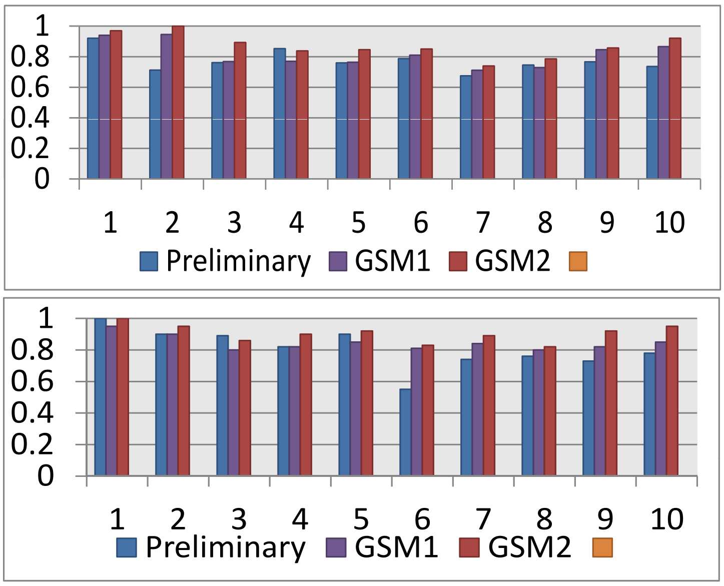
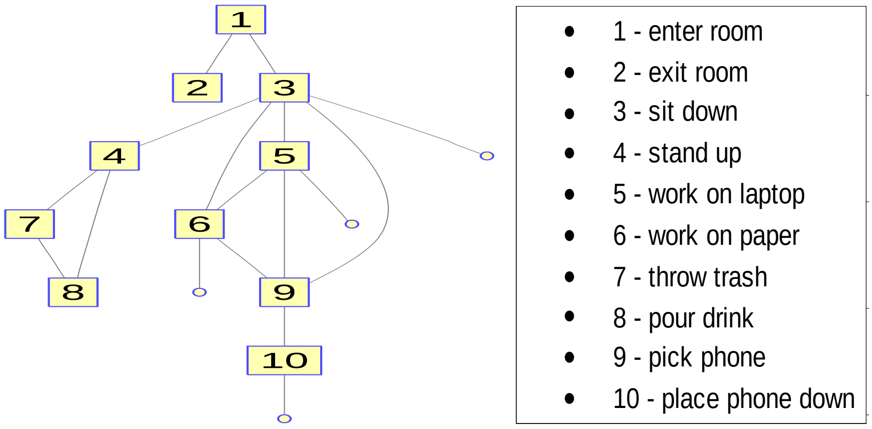


Figure 4.4: Precision (top) and recall (bottom) for the ten activities in UCLA Office Dataset.

Method	Pei [86]	zhu [140]	Preliminary
Accuracy	90.6	90.8	82.6
Method	$l2$-CRF	$l1$-CRF	GSM^2
Accuracy	87.4	89.7	**91.2**

Table 4.2: Comparison of different methods in recognition accuracy for the UCLA Office dataset.

modeling, we also compare with elementwise $l2$-regularized CRF ($l2$-CRF) and $l1$-regularized CRF ($l1$-CRF), solved using the modified bundle method [113]. Note that $l1$-CRF is different from our sparse modeling using l1 regularizer on each parameter group, since in our model, the l1 regularizer is upon each parameter group while in $l1$-CRF l1 regularizer is upon each model parameter. It can be seen that incorporating group $l1$-regularization for feature selection as well as enforcing a sparse graphical structure of the underlying relationship graph provides best performance among the compared approaches. The performance gain is partly due to the parameter grouping, which provides an informative prior information about the model parameters.



(a) (b)

Figure 4.5: (a): The learned graphical structure using the proposed GSM.

The experimental results also show that group $l1$- regularization successfully enforces the group-level sparsity of model parameters, resulting in an optimal graphical model with a s-parse structure. 6 out of 13 Intra-CF parameter groups are excluded from the learned graphical model. About 75% of node connections are eliminated from the learned graph. And because $l1$-penalty is used for each contextual parameter group, sparsity in the feature level is also achieved. Only 23.6% of model parameters are non-zero. This would effectively expedite the inference procedure significantly, since the inference computation dominated with product of feature vec-

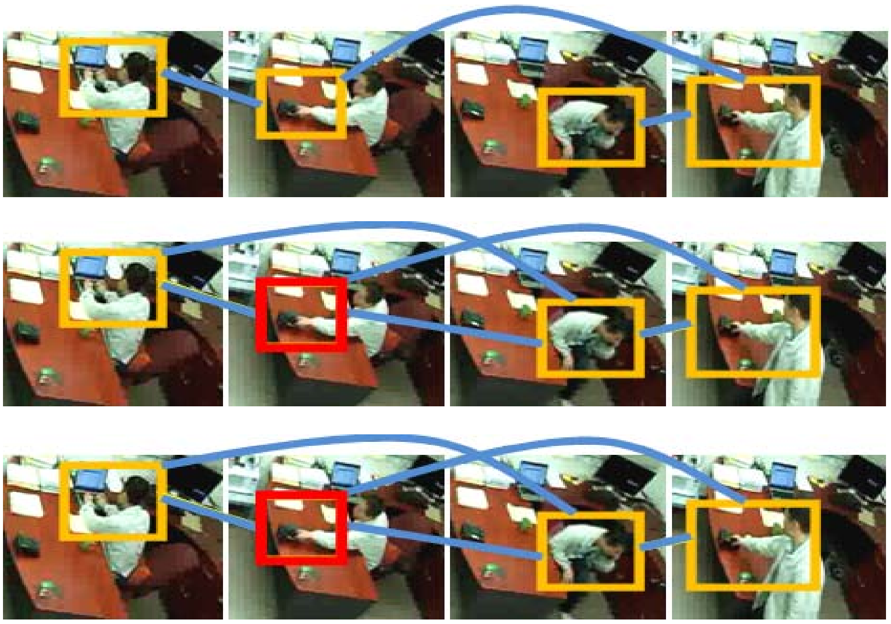


Figure 4.6: Sample results of activity localization and labeling. Top: results of using GSM^2; Bottom: results of using $l1$-CRF and $l2$-CRF. The labeling results using $l1$-CRF and $l2$-CRF are the same in this case. Note the different structures of inter-relationships among the detected activities. Red bounding box indicates the mislabeled activity.

tors and their associated parameters. Fig. 4.5(b) shows the learned sparse graph representing the inter-dependence between activity classes. The connection between a rectangle node and a circle node denotes self connections, which indicates the inter-dependence of activities of the same classes. The retained relationships between activity classes, such as 1 - enter room and 2 - sit down, are more logical than those of the eliminated ones, such as 8 - pour drink and 9 - pick phone.

Fig. 4.4.1.3 shows sample results of using the proposed sparse modeling approach with $l2$-CRF and $l1$-CRF. It can be seen that $l1$-CRF does not enforce sparse graphical structure in this case.

4.4.2 Object Recognition

In this set of experiments, we work on PASCAL VOC 2007 for object recognition in natural scenes.

Intra-CF PGs	Inter-CF PGs	mAP
L1	L1	37.5
L1	L2	38.2
L2	L1	40.0
L2	L2	40.5

Table 4.3: Comparing combinations of different norms used for parameter groups of contextual features. PG is short for parameter group.

4.4.2.1 PASCAL VOC Dataset

PASCAL VOC 2007 is a standard image database for object detection, classification and image segmentation. For the task of object detection, there are 20 objects of interest, including plane, bike, bird, boat, bottle, bus, car , cat, chair, cow, table, dog, horse, motor, person, plant, sheep, sofa, train, tv. Each image has one or multiple objects to be recognized.

4.4.2.2 Preprocessing

We use part-based object detector [34] for the detection of candidate objects, for the detector outperforms many previous ones. After non-maxima suppression (NMS) procedure, an optimum confidence score is trained for each object class which maximizes the F-score of the detector on the train-val dataset. The detection thresholds for each object class are herein obtained. Only detected regions that have higher confidence scores than the corresponding detection thresholds are retained for further process.

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mAP
DMMOL[22]	28.8	56.2	3.2	14.2	29.4	38.7	48.7	12.4	16.0	17.7	24.0	11.7	45.0	39.4	35.5	15.2	16.1	20.1	34.2	35.4	27.2
CDXH[16]	41.0	**64.5**	15.1	19.5	33.0	57.9	63.2	27.8	23.2	28.2	29.1	16.9	**63.7**	53.8	47.1	18.3	28.1	42.2	53.1	49.3	38.7
voc5[34]	36.6	62.2	12.1	17.6	28.7	54.6	60.4	25.5	21.1	25.6	26.6	14.6	60.9	50.7	44.7	14.3	21.5	38.2	49.3	43.6	35.4
Preliminary	42.1	59.6	10.7	23.7	27.9	53.9	61.6	25.0	22.3	27.8	26.5	14.2	57.9	50.4	46.1	18.5	33.6	41.4	49.6	42.4	36.8
$l1$-CRF	45.0	62.9	8.6	25.9	29.2	54.5	58.6	28.2	21.7	30.5	33.4	13.5	53.8	51.1	45.7	22.3	36.9	38.5	51.7	43.3	37.9
$l2$-CRF	44.1	63.5	10.6	26.6	30.1	57.2	63.4	28.8	25.2	31.4	34.6	14.8	58.8	53.5	48.3	**23.4**	38.6	41.0	53.8	43.8	39.7
GSM^2	**46.3**	64.0	11.0	**27.5**	**34.0**	**58.0**	**64.2**	29.2	**25.5**	31.8	35.3	15.6	59.0	**54.2**	**50.8**	23.0	38.0	**43.5**	**54.0**	44.0	**40.5**

Figure 4.7: Comparison of our approach with the state-of-the-art object detectors on VOC 2007. GSM^2: the proposed group sparse modeling approach with $l2$-norms within all parameter groups.

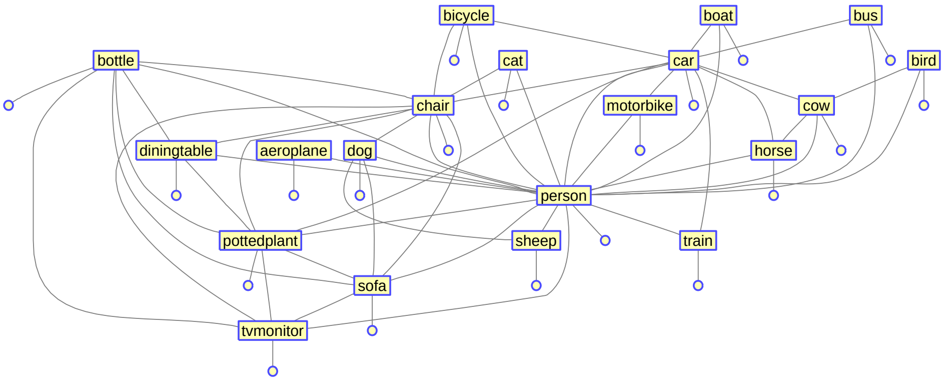


Figure 4.8: Learned sparse graph for the context-aware object recognition task.

In order to utilize correlations between object classes in low-level features, we prefer multi-classification scores. Spatial Pyramid Matching (SPM) using sparse coding has been reported to have best performance in several popular object recognition databases [129]. Thus, multi-SVM with SPM kernel [129] upon (de-noised) dense SIFT descriptors is trained, and a 21x1 normalized classification score vector (one background class and 20 object classes of interest) is developed as the Intra-IF for each image region. Codebook of size 1024 is used for K-SVD SIFT. We call the resulting object classification as the preliminary results.

Color Name descriptor [50] of each candidate region is generated as an Intra-CF descriptor for each candidate region. Besides, 20 scene categories are generated by kernel-kmeans using a combination of Color Name and dense SIFT descriptors of the scene. A scene labeling vector of size 20x1, which contains the scene labeling scores is developed as the Intra-CF features of the containing candidate objects. Six spatial layout attributes are used for the development of Inter-CF features, including Ontop, Overlap (less than 25%, greater than 75% and between), Nearby (which indicates two objects are close to each other but do not ontop of one another or overlap with each other), and Faraway.

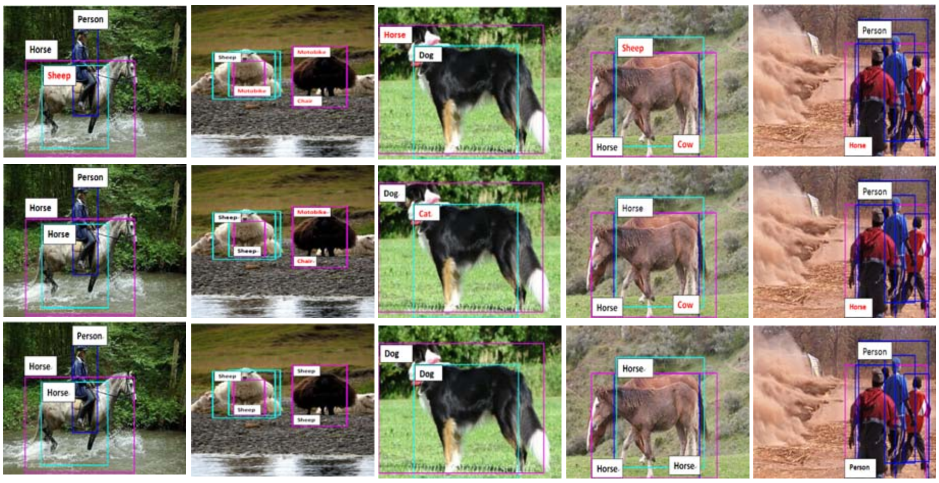


Figure 4.9: Example test images. For each column, the top image shows the detection results of $l1$-CRF; the middle one shows the detection results of $l2$-CRF; and the bottom one shows the detection results of our group sparse modeling using l2 norms for within group parameter regularization. In each image, objects with red labels are incorrectly labeled.

4.4.2.3 Experimental Results

Table 4.3 shows the performance of object recognition on VOC 07. We can see that using $l2$-norm within each parameter group provides best recognition accuracy. Intuitively, parameters associated with Intra-CF and Inter-CF are not intrinsically sparse within their parameter groups. For instance, chairs often coexist with tables. Their relative locations (overlap, on-top, nearby and faraway, etc.) can be diverse in different images. In other words, the pairwise parameter group associated with chair and table should not be sparse within the group. Thus, enforcing sparsity within the parameter groups of contextual features in turn offsets the recognition accuracy. However, VOC 07 consists of a large number images containing only one object class. Enforcing sparsity within groups of pair-wise parameters does not offset the recognition performance much as demonstrated by the results of case 3 and case 4. Since $l2$-norm works best as the within group norm, in the following experiments on VOC 2007, we use $l2$-norm for within group penalty.

In Fig. 4.7, we compare our results to $l1$-CRF and $l2$-CRF as well as several recent works exploring context in object recognition. We use mAP as the evaluation criteria, which is the mean of the average precision of each object category. It can be seen that our sparse model (GSM^2) obtains the best mAP for 10 object classes and the highest mAP 40.8% among all the compared approaches. This demonstrates the benefit of group sparsity over element-wise sparsity of graphical model for the recognition problem. Note that [22] used $l2$-CRF to integrate spatial layouts of objects. The improvement of our $l2$-CRF implementation over [22] mainly comes from better baseline (preliminary) detection and the strategic selection of candidate detections. In [22], an older version of [34] was used.

Fig. 4.8 shows the learned sparse graph structure. We can see that about 70% of the class connections in a fully connected graph are enforced to be zeros. In the learned sparse graph, while significantly related objects such as person and bus are still connected weakly related objects such as TV-monitor and airplane are not connected. Since the majority groups of contextual parameters are zeroed out from the learned model, the learned model has sparse features at both group and individual level. This also expedite the inference procedure because we do not need to calculate the potentials associated with parameters which have a zero value.

Fig. 4.9 shows examples of detection results from our model (GSM^2) as compared to element-wise regularized CRF models. Our model appears to produce better detections by better understanding the intrinsic interactions between objects classes. For instance, it learns how to correctly enforce mutual exclusion between dogs and cats, allowing sheep appear together but not sheep and motorbike.

4.5 Conclusion

In this chapter, we present a novel approach of sparse modeling for the context-aware visual recognition tasks in computer vision. Based on the popular CRF model for context-aware recognition of inter-dependent visual objects, group $l1$-regularization is integrated to select an optimum sets of features from a candidate pool, resulting in feature sparsity as well as graphical sparsity of the learned model. Our experiments on object and activity recognition demonstrate the benefits of the proposed group sparse modeling approach, even with strong baseline detectors. In Section 4.2.3, we describe the role of the choice of within-group penalties in ensuring feature and group level sparsity. Our experiments show that, for the datasets chosen, both feature and group level sparsity provide the best results in activity recognition. However, for object recognition, the best results are obtained when only group level sparsity is considered. These conclusions could be different for another dataset. The strength of our proposed approach lies in providing a general framework that could consider the choice of different within-group norm penalties, and consequently the level of sparsity, for the application in hand. This is not hand-engineered, but automatically determined by the proposed framework.

Chapter 5

Anomalous Activity Detection

5.1 Introduction

Video surveillance systems monitor people's activities and generate alerts when anomalous activities are detected. Usually, samples of anomalous activities are rare. Given a set of normal samples, the system is trained to learn frequent patterns of normal activities using methods of activity recognition. Activities whose patterns deviate from the learned frequent patterns are detected as anomalies.

Most methods developed in the literature on anomalous activity recognition have concentrated on analyzing individual motion patterns of activities. These methods model activities individually and aim to learn discriminative patterns for each activity class. Activities with abnormal patterns are considered as anomalous activities. However, activities in natural scenes rarely happen independently. The interdependence between activity classes provides important cues for activity recognition, as well as the detection of anomalous activities. Jointly modeling and recognizing related activities in space and time can improve the accuracy of activity recognition. This, in turn, will help detect anomalous activities better.

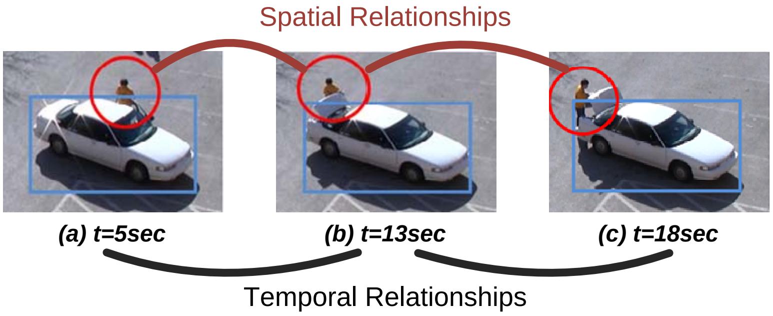


Figure 5.1: An example that demonstrates the importance of context in activity recognition. Motion region surrounding the person of interest is located by red circle, interacting vehicle is located by blue bounding box.

5.1.1 Overview of the Framework

It has been demonstrated in Chapter 3 that context is significant for activity recognition. Human-object interaction has been frequently used as context in many past works [71][130]. Consider the activities in Fig. 5.1. The existence of the nearby car gives information about what the person (bounded by red circle) is doing, and the relative position of the person of interest and the car says that activities (b) and (c) are very different from activity (a). Moreover, just focusing on the person, it may be hard to tell what the person is doing in (b) and (c) - "opening vehicle trunk" or "closing vehicle trunk". If we knew that these activities occurred around the same vehicle along time, it would be immediately clear that in (b) the person is opening the vehicle trunk and in (c) the person is closing the vehicle trunk. This example shows the importance of spatial and temporal relationships for activity recognition. This example illustrates a pattern among different activities. Harnessing such spatial and temporal relationships could be very beneficial for activity recognition.

Motivated by the above, we use the modified Struct-SVM in Section 3.2.2 to explicitly models the motion patterns for normal activities, as well as spatial and temporal relationships of activities and captures useful spatio-temporal patterns for each pair of normal/known activity classes during the learning process. These learned motion and context patterns are used for classifying between normal activities. Activities whose motion and context patterns deviate from the normal motion and context patterns are considered as anomalies. With the learned pattern parameters, distributions of feature distances for different normal activity classes are estimated from training examples. Normality factors are introduced to measure the normalcy of activities based on their motion and context features. p-test is applied to determine if a detected activity is normal or not. Specifically, activities with one or more normality factors lower than the predefined thresholds (which can be learned a priori) are considered as anomalies.

5.1.2 Contributions of The Present Work

The main contribution of this work is to show how context can be exploited for anomaly detection in video. We focus on the joint modeling and recognition of normal activities in videos of a wide scene, using both motion and context information, and how the learned model can be used for the detection of anomalous activities.

(i) Based on types of abnormal attribute an anomalous activity has, we define three kinds of anomalous activities - point anomaly, contextual anomaly and collective anomaly for the detection of anomalous activities. These definitions make the task of anomaly detection more clear.

(ii) We propose a novel framework for anomalous activity detection based on context-aware activity models. Rather than only detecting activities with abnormal motion patterns as in previous works, our approach also detects activities with abnormal contextual attribute and/or abnormal relationships with other activities.

5.2 The Algorithm

For anomaly detection, we assume that we have instances of all the normal activities. Test instances whose patterns deviate from the learned model are anomalies. Once the activity label is assigned to a test instance, we focus on the analysis of whether this activity is anomalous.

For discriminative models such as the modified Struct-SVM 3.2.2, the separating hyperplane between two classes can be obtained by subtracting the associated weight vectors as discussed in [8]. For normal instances belonging to a certain class, the distances to their associated separating hyperplanes are expected to follow certain distributions [8], which can be estimated through kernel density estimation from the training data. An instance with infrequent distances can be considered as anomaly. For this reason, four kinds of distances, which can be used to evaluate the normality of an activity and pair of activities, are developed based on the weight vectors learned for the proposed structural model.

With weight vectors $\omega_{x,i}$, $\omega_{x,j}$, $\omega_{g_k,i}$ and $\omega_{g_k,j}$ for $k \in \{1,...,N_G\}$, $i,j \in \{1,...,M\}$ and $i \neq j$, we define the unbiased motion hyperplane $HP_x(i,j)$ and intra-context hyperplane $HP_{g_k}(i,j)$ by their normal vectors as

$$\begin{aligned} HP_x(i,j) &= \omega_{x,i} - \omega_{x,j}, \\ HP_{g_k}(i,j) &= \omega_{g_k,i} - \omega_{g_k,j}, \end{aligned} \tag{5.1}$$

where an unbiased hyperplane means a hyperplane that passes through the origin. Thus, hyperplanes $HP_x(i,j)$ and $HP_{g_k}(i,j)$, translated along the directions of their normal vectors by a constant, can separate classes i and j based on motion and intra-context features respectively. With weight vectors $\omega_{sc,(i,j)}$, $\omega_{sc,(i',j')}$, $\omega_{tc,(i,j)}$ and $\omega_{tc,(i',j')}$ for $i,j,i',j' \in \{1,...,M\}$ and $(i,j) \neq (i',j')$, define unbiased hyperplanes $HP_{sc}((i,j),(i',j'))$ and $HP_{tc}((i,j),(i',j'))$ by its normal vectors as

$$HP_{sc}\left((i,j),(i',j')\right) = \omega_{sc,(i,j)} - \omega_{sc,(i',j')},$$
$$HP_{tc}\left((i,j),(i',j')\right) = \omega_{tc,(i,j)} - \omega_{tc,(i',j')}. \tag{5.2}$$

Similarly, $HP_{sc}((i,j),(i',j'))$ and $HP_{tc}((i,j),(i',j'))$, translated along the directions of their normal vectors by a constant, can separate class pairs (i,j) and (i',j') based on inter-context spatial and temporal features respectively.

Consider an activity a with motion feature x_a, intra-activity context feature g_a and class label y_a^{opt} generated by the structural model. Define the distance of motion feature x_a to hyperplane $HP_x(y_a^{opt},j)$, as $d_x^a(y_a^{opt},j)$ and distance of intra-activity context feature $g_{a,k}$, for $k = 1,...,N_G$ to hyperplane $HP_{g_k}(y_a^{opt},j)$ as $d_{g_k}^a(y_a^{opt},j)$, where $j \neq y_a^{opt}, \quad j \in 1,...,M$. These distances can be calculated as

$$d_x^a\left(y_a^{opt},j\right) = \frac{HP_x\left(y_a^{opt},j\right)^T \cdot x_a}{norm\left(HP_x\left(y_a^{opt},j\right)\right)},$$
$$d_{g_k}^a\left(y_a^{opt},j\right) = \frac{HP_{g_k}\left(y_a^{opt},j\right)^T \cdot g_{a,k}}{norm\left(HP_{g_k}\left(y_a^{opt},j\right)\right)},$$

where $norm()$ is the Euclidean norm. Assume activity collection A with member activities $a_1,a_2,...,a_N$ related to each other in space and time with class labels $Y^{opt} = [y_1^{opt},y_2^{opt},...,y_N^{opt}]$. $sc_{a_1,a_2},...,sc_{a_{N-1},a_N}$ and $tc_{a_1,a_2},...,\ tc_{a_{N-1},a_N}$ are their inter-activity context features. The distance of inter-activity context feature sc_{a_i,a_j} to hyperplane $HP_{sc}((y_i^{opt},y_j^{opt}),(i',j'))$ is defined as $d_{sc}^{a_i,a_j}((y_i^{opt},y_j^{opt}),(i',j'))$ (denoted as $d_{sc}^{a_i,a_j}(i'j')$ for simplicity), and distance of tc_{a_i,a_j} to hyperplane $HP_{tc}((y_i^{opt},y_j^{opt}),(i',j'))$ is defined as $d_{tc}^{a_i,a_j}((y_i^{opt},y_j^{opt}),(i',j'))$ (denoted as $d_{tc}^{a_i,a_j}(i'j')$), where $(i',j') \neq (y_i^{opt}, y_j^{opt})$, $i',j' \in 1,...,M$. These distances can be calculated as

$$d_{sc}^{a_i,a_j}(i'j') = \frac{HP_{sc}\left(\left(y_i^{opt}, y_j^{opt}\right),(i',j')\right)^T \cdot sc_{a_i,a_j}}{norm\left(HP_{sc}\left(\left(y_i^{opt}, y_j^{opt}\right),(i',j')\right)\right)},$$
$$d_{tc}^{a_i,a_j}(i'j') = \frac{HP_{tc}\left(\left(y_i^{opt}, y_j^{opt}\right),(i',j')\right)^T \cdot tc_{a_i,a_j}}{norm\left(HP_{tc}\left(\left(y_i^{opt}, y_j^{opt}\right),(i',j')\right)\right)}.$$

The probability density distributions of distances $d_x^a(y^{opt},j)$, $d_{sc}^{a_i,a_j}((y_i^{opt},y_j^{opt}),(i',j'))$, $d_g^a(y^{opt},j)$ and $d_{tc}^{a_i,a_j}((y_i^{opt},y_j^{opt}),(i',j'))$ can be estimated from training instances using kernel density estimation [8] for $j,i',j' \in \{1,...,M\}$, $j \neq y^{opt}$ and $(i',j') \neq (y_i^{opt},y_j^{opt})$. Abnormal activities are expected to have one or more infrequent potential distance scores.

5.2.1 Anomaly Definitions

Analogous to outlier detection in data mining [43], we introduce the concepts of point anomaly, contextual anomaly, and collective anomaly, whose definitions are given in the subsections below.

5.2.1.1 Point Anomaly

Point anomalies are detected without any contextual information [47]. Typically, for an atomic event in a video, the motion information captured from its local motion features follow certain patterns, which have been demonstrated by the popular activity classification method - BOW+SVM [60] upon STIP features. In our case, motion pattern of each activity class is reflected in the distributions of their distances to the hyperplanes HP_x. Denote the learned structural model as M. Given a test activity a_t with motion score histogram x_{a_t} and class label $y_{a_t}^{opt}$, define probability $p_x^{a_t}(j)$ as

$$p_x^{a_t}(j) = p\left(d_x^a\left(y_{a_t}^{opt}, j\right) < d_x^{a_t}\left(y_{a_t}^{opt}, j\right) | M\right), \tag{5.3}$$

for $j \in \{1, ..., M\}$ and $j \neq y_{a_t}^{opt}$. We use the p-test to determine whether an anomaly exists or not [29]. The (one-sided) p-value $P_x^{a_t}(j) = min(p_x^{a_t}(j), 1 - p_x^{a_t}(j))$ measures the probability that the normal distribution of $d_x^a(y_{a_t}^{opt}, j)$ generates a value at least as extreme as $d_x^{a_t}(y_{a_t}^{opt}, j)$. The lower the p-value is, the more safe to say that the observed value does not belong to the normal distribution. So, we define the Motion-based Normality Factor MNF of a_t as the geometric mean of the associated p-values as

$$MNF(a_t) = \left(\prod_{j=1, j \neq y_{a_t}^{opt}}^{M} P_x^{a_t}(j)\right)^{\frac{1}{M-1}}. \tag{5.4}$$

Geometric mean is used here to measure the typical value of the set of p-values. As normal activities, which are known to us follow certain motion patterns captured by the distances, anomalous activities whose motion patterns deviate from the learned motion patterns significantly will have infrequent distances and thus a lower MNF than a threshold TH_{MNF}. Fig 5.2 shows an example of distances of point anomaly.

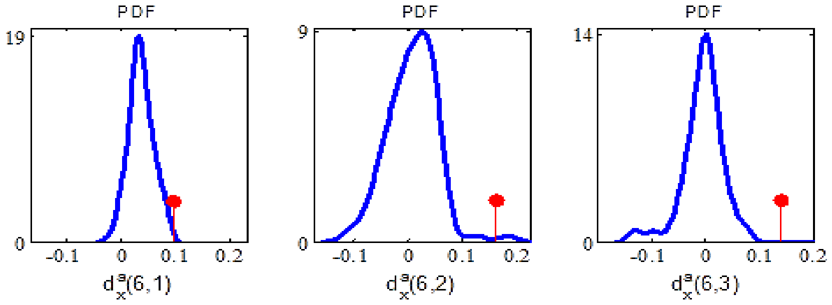


Figure 5.2: Example of probability density functions of $d_x^a(y^{opt}, j)$, $\quad j = 1, 2$ and 3 for normal activities and the corresponding distances of a point anomaly (indicated by red circle). The first ten activities in Fig. 1 in the supplementary material are considered as normal activities, and used to train the structural model. For the point anomaly detected $y^{opt} = 6$.

5.2.1.2 Contextual Anomaly

Two kinds of attributes are generally involved with events: contextual attributes of the event define context, such as the location and surrounding objects; behavioral attributes define the motion of objects involved in the event. Contextual anomaly has normal behavioral attributes but abnormal contextual attributes. Given the intra-activity context feature g_{a_t} of the test activity with class label $y_{a_t}^{opt}$, define $p_{g_k}^{a_t}(j)$ as

$$p_{g_k}^{a_t}(j) = p\left(d_{g_k}^{a}\left(y_{a_t}^{opt}, j\right) < d_{g_k}^{a_t}\left(y_{a_t}^{opt}, j\right) | M\right), \tag{5.5}$$

for $k \in \{1, ..., N_G\}$, $j \in \{1, ..., M\}$ and $j \neq y_{a_t}^{opt}$. The probability that a_t belongs to class $y_{a_t}^{opt}$ based on g_k of a_t and $HP_{g_k}(y_a^{opt}, j)$ is $P_{g_k}^{a_t}(j) = min(p_{g_k}^{a_t}(j), 1 - p_{g_k}^{a_t}(j))$. We define the Context-based Normality Factor CNF_k of a_t as the geometric mean of the associated p-values as

$$CNF_k(a_t) = \left(\prod_{j=1, j\neq y_{a_t}^{opt}}^{M} P_{g_k}^{a_t}(j)\right)^{\frac{1}{M-1}}. \tag{5.6}$$

If a_t has a high $MNF(a_t)$ and any of $CNF_k(a_t)$ for $k = 1, ..., N_G$, where N_G is the number of intra-activity context subsets, is lower than a threshold TH_{CNF}, a_t is considered as contextual anomaly. Fig 5.3 shows an example of distances for contextual anomaly.

5.2.1.3 Collective Anomaly

A collection of activities forms a collective anomaly if the events as a whole deviate significantly from the entire training set. The collective anomaly can be further divided into sequential anomaly and co-occurrence anomaly. In our case, collective anomaly can also be considered as contextual anomaly since the detection of it utilizes the inter-activity context features - spatial and temporal relationships of activities. Assume activity collection A_t with member

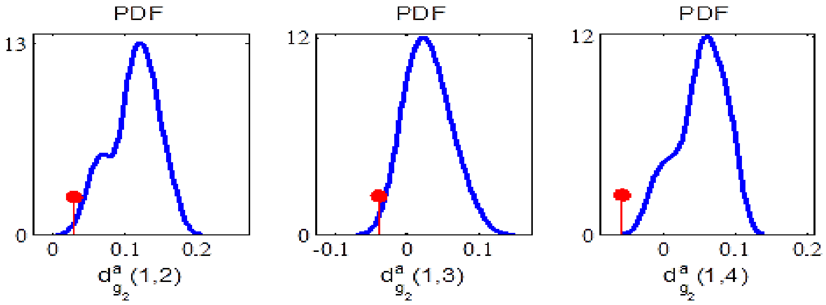


Figure 5.3: Example of probability density functions of $d^a_{g_k}(y^{opt}, j)$, $k = 2$, $j = 2,3$ and 4, for normal activities and the corresponding distances of a contextual anomaly (indicated by red circle). The first ten activities in Fig. 1 in the supplementary material are considered as normal activities and used to train the structural model. For the contextual anomaly detected $y^{opt} = 1$. Intra-activity context subset G_2 is defined in Sec. 3.4.1.

activities $a^t_1, a^t_2, ..., a^t_N$ related to each other in space and time are represented with class labels $Y^{opt}_t = [y^{opt}_1, y^{opt}_2, ..., y^{opt}_N]$, and $sc_{a^t_1,a^t_2}, ..., sc_{a^t_{N-1},a^t_N}$ and $tc_{a^t_1,a^t_2}, ..., tc_{a^t_{N-1},a^t_N}$ are their inter-activity context features. Define

$$p^{(a^t_i,a^t_j)}_{sc}(i',j') = p\left(d^{a_i,a_j}_{sc}(i' j') < d^{(a^t_i,a^t_j)}_{sc}(i',j')\,|M\right),$$
$$p^{(a^t_i,a^t_j)}_{tc}(i',j') = p\left(d^{a_i,a_j}_{tc}(i' j') < d^{(a^t_i,a^t_j)}_{tc}(i',j')\,|M\right).$$

Let

$$P^{(a^t_i,a^t_j)}_{sc}(i',j') = min(p^{(a^t_i,a^t_j)}_{sc}(i',j'), 1 - p^{(a^t_i,a^t_j)}_{sc}(i',j')),$$
$$P^{(a^t_i,a^t_j)}_{tc}(i',j') = min(p^{(a^t_i,a^t_j)}_{tc}(i',j'), 1 - p^{(a^t_i,a^t_j)}_{tc}(i',j')).$$

Define the Spatial Normality Factor SNF and Temporal Normality Factor TNF of

(a_i^t, a_j^t) as

$$SNF\left(a_i^t, a_j^t\right) = \left(\prod_{i',j'=1}^{M} P_{sc}^{\left(a_i^t, a_j^t\right)}\left(i', j'\right)\right)^{\frac{1}{M^2-1}}, \tag{5.7}$$

$$TNF\left(a_i^t, a_j^t\right) = \left(\prod_{i',j'=1}^{M} P_{tc}^{\left(a_i^t, a_j^t\right)}\left(i', j'\right)\right)^{\frac{1}{M^2-1}}. \tag{5.8}$$

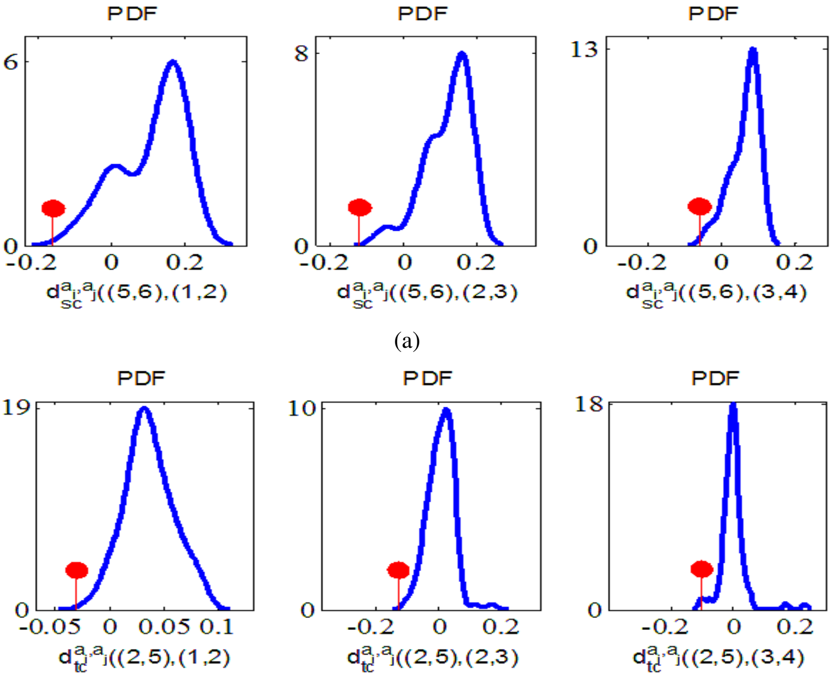
In (5.7) and (5.8), the condition - $(i', j') \neq (y_i^{opt}, y_j^{opt})$ - for the product is omitted for compactness of expression. If all member activities in A_{test} have high MNF and CNF values, but at least one of $SNF(a_i^t, a_j^t)$ is lower than a threshold TH_{SNF}, it is considered as a collective spatial anomaly. If at least one of $TNF(a_i^t, a_j^t)$ is lower than a threshold TH_{TNF} it is considered as a collective temporal anomaly. Fig 5.4(a) shows an example of distances of collective spatial anomaly and 5.4(b) shows an example of distances of collective temporal anomaly.

5.3 Experiments

To assess the effectiveness of our structural model in activity-based anomaly detection, we work on Release 2 for anomaly detection using BOW+SVM as the baseline classifier.

5.3.1 Preprocessing

Motion regions that involve only vehicles moving are excluded from the experiments since we are only interested in person related normal and anomalous activities. For the BOW+SVM classifier, $k = 1000$ visual words and a 9-nearest neighbor soft-weighting scheme are used. For the SFG-based classifier, the size of each temporal bin used is 5 frames while other settings are the same as in [36]. For the SFG method [36], activity localization is implicitly included in the recognition process. The method generates similarity scores $s \in \{0, 1\}$ that quantize the similarity between two activities. In the training process of the baseline classifier, we calculate the



(a)

(b)

Figure 5.4: (a): Example of probability density functions of $d_{sc}^{a}((y_i{}^{opt}, y_j{}^{opt}), (i', j'))$ for normal activities and the corresponding distances of a collective spatial anomaly (indicated by red circle). For the detected collective anomaly a_i and a_j $y_i{}^{opt} = 2$, $y_j{}^{opt} = 5$. (b): Example of probability density functions of $d_{tc}^{a}((y_i{}^{opt}, y_j{}^{opt}), (i', j'))$ for normal activities and the corresponding distances of a collective temporal anomaly (indicated by red circle). For the detected collective anomaly a_i and a_j $y_i{}^{opt} = 5$, $y_j{}^{opt} = 6$ (In both cases, the first ten activities in Fig. 1 in the supplementary material are considered as normal activities and used to train the structural model).

intra-class similarity (the average similarity score between a given activity instance and other instances of the same class) and inter-class similarity (the average similarity score between a given instance and other instances of different classes). We define the representative instances as the ones with maximum difference in their intra-class similarity and inter-class similarity. A fixed number of representative instances for each activity class are selected during the training of the baseline classifier (5 are used in the experiments).

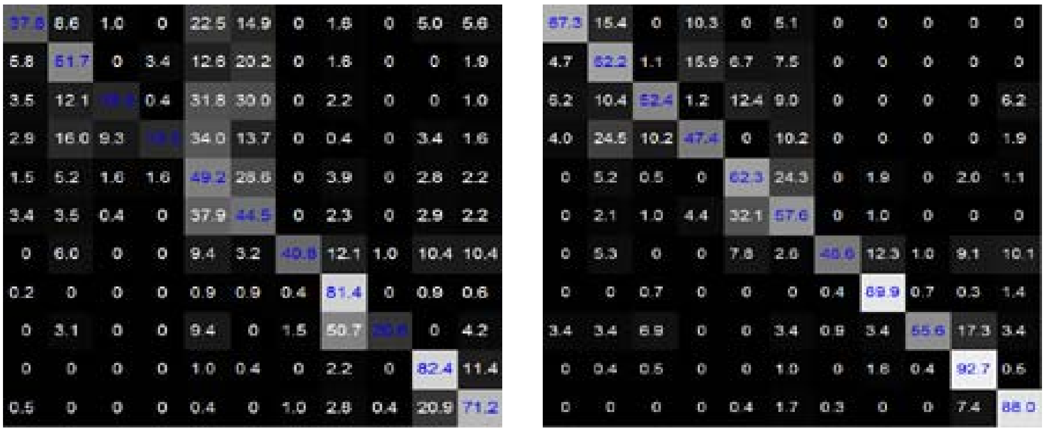
Persons and vehicles are detected using the publicly available software [26]. Opening/closing of doors of facilities, boxes and bags are detected using method in [20] with Histogram of Gradient as the low-level feature and binary linear-SVM as the classifier. Motion score histograms described in Sec. 3.4.1 are generated for each activity. The score histogram of an activity contains the average similarity scores between the activity and the representative examples. For experiment 1, the intra-activity context features are built based on first two cues in Fig. 3.10, and all cues are used for experiment 2.

5.3.2 Results

In order to access the performance of the proposed model on anomaly detection, we work on VIRAT Release 2, in which eleven activity classes are defined as shown in Fig. 1 in the supplementary material. We follow the method defined above to get the recognition results on this dataset. Fig. 5.5 shows the confusion matrix for VIRAT Release 2.

5.3.2.1 Point Anomaly

For the detection of point anomaly, we randomly select one of the eleven activity classes as abnormal, and treated other activities as normal. Cross-validation is used to assess the performance of anomaly detection. For each run, we assume that we do not have training



(a)

37.8	8.6	1.0	0	22.5	14.9	0	1.8	0	5.0	5.6
5.8	61.7	0	3.4	12.6	20.2	0	1.8	0	0	1.9
3.5	12.1	[illegible]	0.4	31.8	30.0	0	2.2	0	0	1.0
2.9	16.0	9.3	[illegible]	34.0	13.7	0	0.4	0	3.4	1.6
1.5	5.2	1.6	1.6	49.2	28.6	0	3.9	0	2.8	2.2
3.4	3.5	0.4	0	37.9	44.5	0	2.3	0	2.9	2.2
0	6.0	0	0	9.4	3.2	40.8	12.1	1.0	10.4	10.4
0.2	0	0	0	0.9	0.9	0.4	81.4	0	0.9	0.6
0	3.1	0	0	9.4	0	1.5	50.7	[illegible]	0	4.2
0	0	0	0	1.0	0.4	0	2.2	0	82.4	11.4
0.5	0	0	0	0.4	0	1.0	2.8	0.4	20.9	71.2

(b)

57.3	15.4	0	10.3	0	5.1	0	0	0	0	0
4.7	62.2	1.1	15.9	6.7	7.5	0	0	0	0	0
6.2	10.4	52.4	1.2	12.4	9.0	0	0	0	0	6.2
4.0	24.5	10.2	47.4	0	10.2	0	0	0	0	1.9
0	5.2	0.5	0	62.3	24.3	0	1.9	0	2.0	1.1
0	2.1	1.0	4.4	32.1	57.6	0	1.0	0	0	0
0	5.3	0	0	7.8	2.8	48.6	12.3	1.0	9.1	10.1
0	0	0.7	0	0	0	0.4	89.9	0.7	0.3	1.4
3.4	3.4	6.9	0	0	3.4	0.9	3.4	55.6	17.3	3.4
0	0.4	0.5	0	0	1.0	0	1.6	0.4	92.7	0.6
0	0	0	0	0.4	1.7	0.3	0	0	7.4	88.0

Figure 5.5: Recognition Results for VIRAT Release 2. (a): Confusion matrix for BOW+SVM baseline classifier; (b): Confusion matrix for our approach using BOW+SVM as the baseline classifier. The activities considered are listed in Fig. 1 in the supplementary material

instances for abnormal activities, so, the activities of abnormal class are excluded from the learning process. We use BOW+SVM as the baseline classifier.

One-class SVM is often used for point anomaly detection [99]. To access the effectiveness of our model in detecting point anomalies, we compare our results with those using one-class SVM. For fair comparison, we also apply the proposed framework on video clips, each containing one activity of the eleven classes. Fig. 5.6 shows the ROC curves of BOW+SVM and our method. The areas under curve are 79.8% for our method on video clips, 72% for one-class svm on video clips and 68.5% for our method working on continuous videos.

5.3.2.2 Contextual Anomaly

For the detection of contextual anomalies, we consider activities that are normal in terms of motion features but with abnormal or infrequent intra-activity context features as discussed in Sec. 5.2.1.2. The normality threshold $TH_{CNF} = 0.05$ in the experiment. Fig. 5.7 shows examples of detected contextual anomalies. In the first example, person getting into a vehicle usually occurs in the parking area, and the anomaly is detected when it happens in an

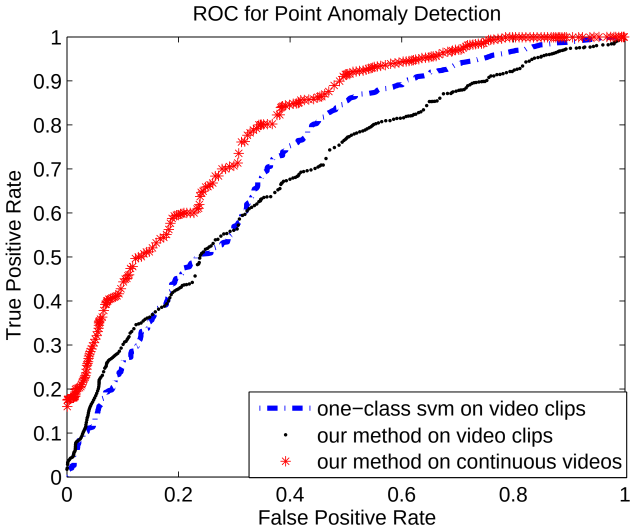
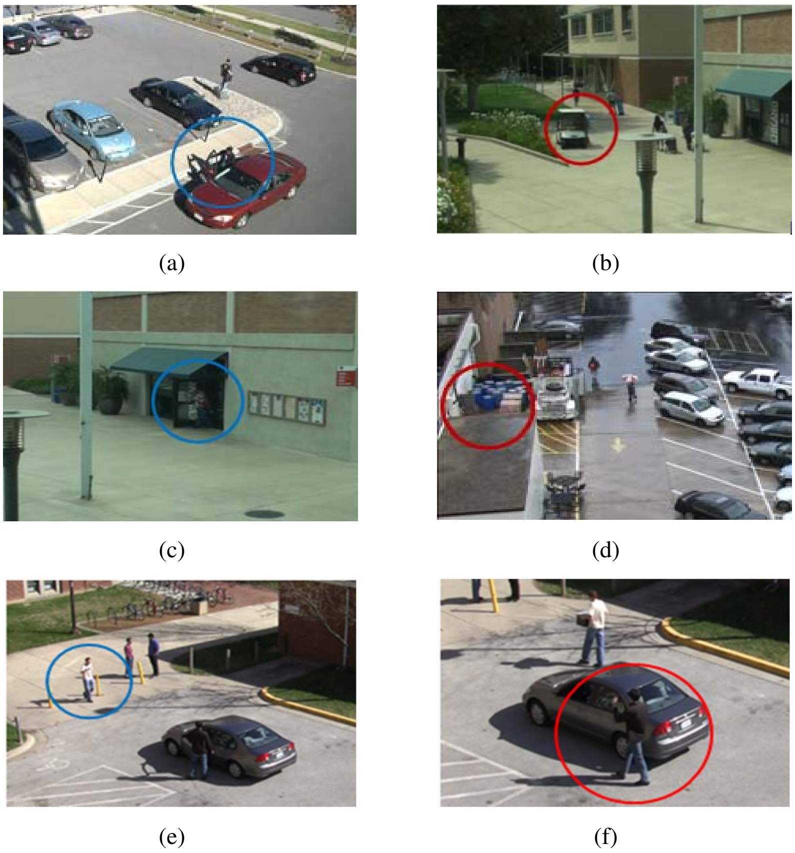


Figure 5.6: ROC curves for point anomaly detection

area not for parking. In the second example, the anomaly of ‘person exiting a facility’ is detected when the person exits from a door of a facility that is rarely used. In the third example, the anomaly of ‘gesturing’ occurs near the trunk of the vehicle while others in our dataset usually occur faraway from the vehicle. None of these could have been detected without the modeling of the intra-activity context feature.

5.3.2.3 Collective anomaly

Collective anomaly can be detected based on the learned inter-activity context patterns and the inter-activity contextual features of the test instances. The normality thresholds $TH_{SNF} = 0.05$ and $TH_{TNF} = 0.05$ are used. For the first example, we consider two activities - ‘person getting into a vehicle’ and ‘person unloading an object from a vehicle’. For most of the examples in the dataset, when these two activities happen together, the unloading happens from the trunk of the car while the person enters through the driver’s door. Thus, as shown in Fig. 5.8(a),(b), a collective spatial anomaly is detected when the unloading and entering happen near the same part of the vehicle. An example of a collective temporal anomaly is shown in Fig.



(a)

(b)

(c)

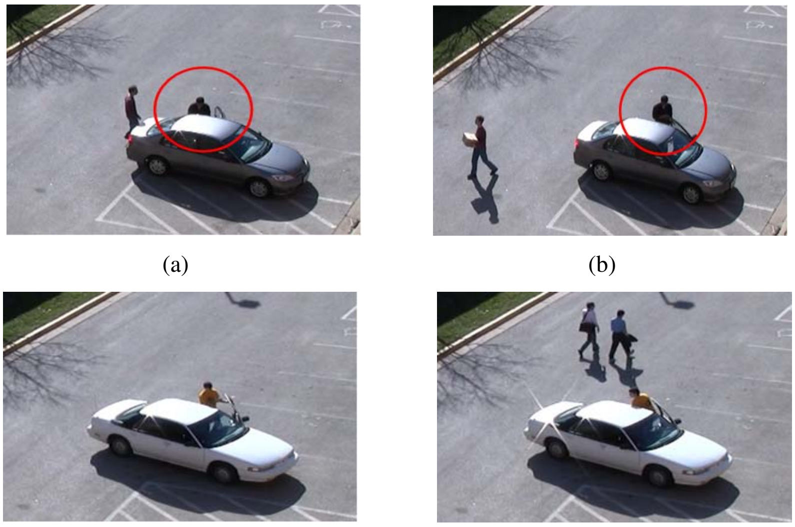
(d)

(e)

(f)

Figure 5.7: (a, c, e): Examples of normal activities; (b, d, f): Examples of detected contextual anomalies. First row: Person getting into a vehicle usually occurs in the parking area (a), and the anomaly is detected when it happen in an area not for parking (b). Second row: Person exiting a facility happens at a normal exit (c), whereas an anomaly is detected when the person exits from a door that is rarely used (d). Third row: A person gesturing far from a vehicle is normal in our dataset (e), whereas in (f) the 'gesturing' occurs near the trunk of the vehicle, which is identified as a contextual anomaly.

5.8(c),(d). The example of a 'person getting out of a vehicle' usually occurs before 'person getting into a vehicle', however, in the detected anomaly, 'person getting out of a vehicle' occurs after 'person getting into a vehicle'.



(a)

(b)

(c)

(d)

Figure 5.8: Example of collective spatial anomaly and collective temporal anomalies. (a, b): we consider two activities - 'person getting into a vehicle' and 'person unloading an object from a vehicle'. For most of the examples in the dataset, when these two activities happen together, the unloading happens from the trunk of the car while the person enters the driver's door. Thus, a collective spatial anomaly is detected when the unloading and entering happen near the same part of the vehicle. (c, d): The example of a 'person getting out of a vehicle' usually occurs before 'person getting into a vehicle', however, in the detected anomaly, 'person getting out of a vehicle' occurs after 'person getting into a vehicle'.

5.4 Conclusion

In this chapter, we present a novel approach to jointly model a variable number of activities in videos, that can also be used to detect abnormal activities. We represent a video of a wide area by sets of activities that are spatially and temporally related. A structural model is proposed to learn the motion patterns and context patterns within and across activity classes from training sets of activities. The inference process tries to generate the correct labels for testing instances using the learned parameters through a greedy search method. Our experiments have shown that encapsulating object interactions and spatial and temporal relationships of activity classes can be used to significantly improve the recognition accuracy. The proposed model can detect point anomalies, contextual anomalies, and collective anomalies based on the motion and various context features.

Chapter 6

Discussion

In this chapter, we discuss several issues that have not been explored in this work, as well as possible directions of future work.

The "String-of-feature" model for activity recognition use STIP features for the recognition of complex activities, which involve obvious non-rigid body motion. Similarly to Bag-of-Word framework using the distances between STIP clusters as the node distances is expected to improve the recognition accuracy over using STIP distances. Furthermore, the computational complexity of graph matching algorithm depends on the number of nodes in the graph. The number of local motion features (STIPs) in an event is usually high, reducing the number of salient motion points/cubics is obviously beneficial. One solution can be using the location-s of body joints, whose appearance and spatial-temporal relationships has been demonstrated to carry distinguishing motion for different human activities. Finally, the computation can be expedited by using matrix factorization in the spectral technique for the calculation of graph similarity. As in [135],graph matching using matrix factorization decomposes the problem of similarity computation into several sub-problems of small sizes, .

The developed graphical models are built with a set of predefined contextual attributes, which are manually designed for a particular human activity dataset at work. In the future, we would like to explore how to learn such contextual attributes automatically from the raw pixels and/or features for a large number of activities and scenes, and conduct the model learning as well as the feature selection in an online fashion.

Furthermore, complex graphical models have a larger number of features than that of the one-layer CRF used in Chapter 4. Intuitively, these features are more likely to be redundant. Thus, applying the sparse modeling on more complex graphical models such as the hierarchical-CRF for the selection of features as well as the graph structure is expected to have a better recognition performance.

Finally, In the proposed anomaly detection approach, a simple p-test is applied to test the abnormality of an event or a pair of events based on the feature observations and the learned model that summarizes intra- and inter- properties of all the training/observed events. However, the criticism of p-test are mainly two-fold. Firstly, it is based on an arbitrary choice of significance level. It is known to be incompatible with the likelihood principle. Secondly, p-value depends on the statistic of the testing example in question. It would be beneficial to evaluate the abnormality of a testing example by weighting p-value together with all other evidence about the abnormality of the example such as the prior evidence. This framework of context-aware anomaly detection can be use for the detection of different kinds of anomalous instances that are potentially inter-dependent with each other, such as the detection of anomalous image objects on images.

Bibliography

[1] J. K. Aggarwal and S. Park. Human mtion: modeling and recognition of actions and iteractions. In *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission*, 2004.

[2] Mohamed R. Amer and Sinisa Todorovic. Sum-product networks for modeling activities with stochastic structure. In *CVPR*, 2012.

[3] Mohamed R. Amer, Dan Xie, Mingtian Zhao, Sinisa Todorovic, and Song-Chun Zhu. Cost-sensitive top-down/bottom-up inference for multiscale activity recognition. In *ECCV*, 2012.

[4] P. A. Anderson. *Nonverbal Communication: Forms and Functions*. Waveland Press, 2nd edition, 2008.

[5] O. Ayad, M. Sayed-Mouchweh, and P. Bllaudel. Switched hybrid dynamic systems identification based on pattern recognition approach. In *IEEE Intl. Conf. on Fuzzy Systems*, 2010.

[6] Yannick Benezeth, Pierre-Marc Jodoin, and Venkatesh Saligrama. Abnormality detection using low-level co-occuring events. In *Pattern Recognition Letters*, 2011.

[7] Ernesto G. Birgin, Jose Mario Martinez, and Marcos Raydan. Nonmonotone spectral projected gradient methods on convex sets. In *SIAM Journal on Optimization*, 2000.

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

[9] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[10] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. In *ICCV*, 2011.

[11] Alexey Castrodad and Sapiro Guillermo. Sparse modeling of human actions from motion imagery. In *International Journal of Computer Vision*, pages 1–15, 2012.

[12] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[13] R. Chaudhry, 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 *IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[14] Rizwan Chaudhry, Avinash Ravichandran, Gregory Hager, and Rene Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *CVPR*, 2009.

[15] F. Chen and W. Wang. Activity recognition through multiscale dynamic Bayesian network. In *16th International Conference on Virtual Systems and Multimedia*, 2010.

[16] G. Chen, Y. Ding, J. Xiao, and T. X. Han. Detection evolution with multi-order contextual co-occurrence. In *CVPR*, 2013.

[17] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. In *CVPR*, 2011.

[18] Wongun Choi and Silvio Savarese. A unified framework for multi-target tracking and collective activity recognition. In *ECCV*, 2012.

[19] Sparse Reconstruction cost for abnormal events detection. Venkatesh saligrama and zhu chen. In *CVPR*, 2011.

[20] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[21] G. Denina, B. Bhanu, H. Nguyen, C. Ding, A. Kamal, C. Ravishankar, A. Roy-Chowdhury, A. Ivers, and B. Varda. Videoweb dataset for multi-camera activities and non-verbal communication. In *Distributed Video Sensor Networks*. Springer, 2011.

[22] Chaitanya Desai, Deva Ramanan, and Charless C. Fowlkes. Discriminative models for multi-class object layout. In *International Journal of Computer Vision*, 2011.

[23] Piotr Dollar, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005.

[24] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. In *Ann. Statist.*, vol. 32,no. 2,pp. 407-499,2004.

[25] C. Fanti, L. Z. Manor, and P. Perona. Human motion: modeling and recognition of actions and iteractions. In *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission*, 2005.

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

[27] Pedro F Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminative trained part based models. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2010.

[28] Jiashi Feng, Xiaotong Yuan, Zilei Wang, Huan Xu, and Shuicheng Yan. Auto-grouped sparse representation for visual analysis. In *ECCV*, 2012.

[29] David Freedman, Robert Pisani, and Roger Purves. *Statistics*. W. W. Norton Company, 4th edition, 2007.

[30] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A note on the group lasso and a sparse group lasso. In *arXiv preprint arXiv:1001.0736*, 2010.

[31] Aphrodite Galata, Anthony Cohn, Derek Magee, and David Hogg. Modeling interaction using learnt qualitative spatio-temporal relations and variable length markov models. In *Proceedings of the European Conference on Artificial Intelligence*, 2002.

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

[33] U. Gaur, B. Song, and Amit K. Roy-Chowdhury. Query-based retrieval of complex activities using "strings of motion-words. In *IEEE Workshop on Motion and Video Computing*, 2009.

[34] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models,release 5. http://people.cs.uchicago.edu/ rbg/latent-release5/.

[35] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Dec. 2007.

[36] U. Guar, Y. Zhu, B. Song, and A. K. Roy-Chowdhury. A "string of feature graphs" model for recognition of complex activities in natural videos. In *IEEE Intl. Conf. on Computer Vision*, 2011.

[37] Tanaya Guha and Rabab K. Ward. Learning sparse representations for human action recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, pages 1576–1588, 2012.

[38] A. Gupta and L. S. Davis. Objects in action: An approach for combining action understanding and object perception. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[39] Abhinav Gupta, Praveen Srinivasan, Jianbo Shi, and Larry S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *CVPR*, 2009.

[40] Raffay Hamid, Amos Johnson, Samir Batta, Aaron Bobick, Charles Isbell, and Graham Coleman. Detection and explanation of anomalous activities: Representing activities as bagsof event n-grams. In *cvpr*, 2005.

[41] Raffay Hamid, Siddhartha Maddi, Aaron Bobick, and Irfan Essa. Structure from statistics - unsupervised activity analysis using suffix trees. In *ICCV*, 2007.

[42] Dong Han, Liefeng Bo, and Cristian Sminchisescu. Selection and context for action recognition. In *ICCV*, 2009.

[43] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining Concepts and Techniques*. Elsevier, 2011.

[44] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining,Inference and Prediction*. Springer, 2nd edition, 2009.

[45] Y. A. Ivanov and A. F. Bobick. Recognition of visual activities and interactions by s-tochastic parsing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2000.

[46] F. Jiang, Y. Wu, and A. K. Katsaggelos. A dynamic hierarchical clustering method for trajectory-based unusual video event detection. In *IEEE Transaction on Image Processing*, 2009.

[47] Fan Jiang, Junsong Yuan, Sotirios A. Tsaftaris, and Aggelos K. Katsaggelos. Anomalous video event detection using spatiotemporal context. In *Computer Vision and Image Understanding*, 2011.

[48] Seong-Wook Joo and Rama Chellappa. Attribute grammar-based event recognition and anomaly detection. In *cvpr*, 2006.

[49] Sameh Khamis, Vlad I. Morariu, and Larry S. Davis. Combining per-frame and per-track cues for multi-person action recognition. In *ECCV*, 2012.

[50] Fahad Shahbaz Khan, Rao Muhammad Anwer, Joost van de Weijer, Andrew D. Bagdanov, Maria Vanrell, and Antonio M. Lopez. Color attributes for object detection. In *CVPR*, 2012.

[51] Seung-Jean Kim, K. Koh, M. Lustig, Stephen Boyd, and Dimitry Gorinevsky. An interior-point method for large-scale l1-regularized least squares. In *IEEE Journal of Selected Topics in Signal Processing*, vol. 1,no. 4,December 2007.

[52] J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1998.

[53] Pushmeet Kohli, Lubor Ladicky, Philip H.S., and Torr. Robust higher order potentials for enforcing label consistency. In *IJCV*, 2010.

[54] Nikos Komodakis. Learning to cluster using high order graphical models with latent variables. In *ICCV*, 2011.

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

[56] L. Ladicky, C. Russell, P. Kohli, and P. H Torr. Associative hierarchical crfs for object class image segmentation. In *ICCV*, 2009.

[57] Lubor Ladicky, Paul Sturgess, Karteek Alahari, Chris Russell, and Philip H.S. Torr. What, where & how many? combining object detectors and crfs. In *ECCV*, 2010.

[58] Tian Lan, Yang Wang, Stephen N. Robinovitch, and Greg Mori. Discriminative latent models for recognizing contextual group activities. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2012.

[59] Tian Lan, Yang Wang, Weilong Yang, and Greg Mori. Beyond actions: Discriminative models for contextual group activities. In *The Neural Information Processing Systems*, 2010.

[60] I. Laptev and T. Lindeberg. Space-time interest points. In *IEEE Intl. Conf. on Computer Vision*, pages 432–439, 2003.

[61] Ivan Laptev. On space-time interest points. In *International Journal of Computer Vision (IJCV)*, 2005.

[62] Quoc V. Le, Jiquan Ngiam, Zhenghao Chen, Daniel Chia, Pang Wei Kob, and Andrew Y. Ng. Tiled convolutional neural networks. In *NIPS*, 2010.

[63] M.W. Lee and R. Nevatia. Human pose tracking in monocular sequence using multilevel structured models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[64] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 17–32, 2004.

[65] Marius Leordeanu and Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. In *IEEE Intl. Conf. on Computer Vision*, 2005.

[66] Xiaoxing Li, Tao Jia, and Hao Zhang. Expression-insensitive 3d face recognition using sparse representation. In *CVPR*, 2009.

[67] J. Liu, S. Ali, and M. Shah. Recognizing human actions using multiple features. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[68] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *ICCV*, 2009.

[69] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[70] L Meier, S Van De Geer, and P Bhlmann. The group lasso for logistic regression. In *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2008.

[71] Vlad I. Morariu and Larry S. Davis. Multi-agent event recognition in structured scenarios. In *CVPR*, 2011.

[72] Meinard Müller. *Information Retrieval for Music and Motion*. Springer Verlag, 2007.

[73] J. C. Nascimento, M. A. Figueiredo, and J. Marques. Recognition of human activities using space dependent switched dynamical models. In *Intl. Conf. on Image Processing*, 2005.

[74] N. Nayak, R. Sethi, B. Song, and A. Roy-Chowdhury. Motion pattern analysis for modeling and recognition of complex human activities. In *Guide to Video Analysis of Humans: Looking at People*. Springer, 2011.

[75] Y.-G. Jiang G.-W. Ngo and J. Yang. Towards optimal bag of word for object categorization and semantic video retrieval. *ACM Intl. Conf. on Image and Video Retrieval*, 2007.

[76] M. Nguyen, Z. Lan, and F. DellaTorre. Joint segmentation and classification of human actions in video. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.

[77] J. C. Niebles, H. Wang, and L. Fei Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision (IJCV)*, Sep. 2008.

[78] J. C. Niebles, H. Wang, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010.

[79] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, J.K. Aggarwal, Hyungtae Lee, Larry Davis, Eran Swears, Xiaoyang Wang, Qiang Ji, Kishore Reddy, Mubarak Shah, Carl Vondrick, Hamed Pirsiavash, Deva Ramanan, Jenny Yuen, Antonio Torralba, Bi Song, Anesco Fong, Amit Roy-Chowdhury, and Mita Desai. A large-scale benchmark dataset for event recognition in surveillance video. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.

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

[81] N. Oliver, B. Rosario, and A. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):831–843, August 2000.

[82] Kevin p. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[83] S. Park and J. K. Aggarwal. Recognition of two-person interactions using a hierarchical Bayesian network. *ACM Journal of Multimedia Systems, Special Issue on Video Surveillance*, 2004.

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

[85] V. Pavlovic, J. Rehg, and J. MacCormick. Learning switching linear models of human motion. *Neural Information Processing Systems Foundation*, 2000.

[86] Mingtao Pei, Yunde Jia, and Song-Chun Zhu. Parsing video events with goal inference and intent prediction. In *ICCV*, 2011.

[87] S. Prajna and A. Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *Hybrid Systems: Computation and Control, volume 2993 of Lecture Notes in Computer Science*, pages 271–274. Springer Berlin / Heidelberg, 2004.

[88] R.Hamid, Siddhartha Maddi, Aaron Bobick, and Irfan Essa. Unsupervised analysis of activity sequences using event-motifs. In *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, 2006.

[89] M. S. Ryoo and J. K. Aggarwal. Recognition of composite human activities through context-free grammar based representation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[90] M. S. Ryoo and J. K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *IEEE Intl. Conf. on Computer Vision*, 2009.

[91] M. S. Ryoo and J. K. Aggarwal. UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html, 2010.

[92] M. S. Ryoo, C-C Chen, J. K. Aggarwal, and A Roy-Chowdhury. An overview of contest on semantic description of human activities (SDHA). In *Intl. Conf. on Pattern Recognition*, 2010.

[93] 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.

[94] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. on Acoustics, Speech and Signal Processing*, Feb. 1978.

[95] Imran Saleemi, Khurram Shafique, and Mubarak Shah. Probabilistic modeling of scene dynamics for applications in visual surveillance. In *PAMI*, 2009.

[96] Venkatesh Saligrama and Zhu Chen. Video anomaly detection based on local statistical aggregates. In *CVPR*, 2012.

[97] Venkatesh Saligrama, Janusz Konrad, and Pierre-Marc Jodoin. Video anomaly identification. In *IEEE Signal Processing Magazine*, 2010.

[98] 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.

[99] Bernhard Scholkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. In *Neural Computation, 13*, pages 1443–1471, 2001.

[100] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *Intl. Conf. on Pattern Recognition*, 2004.

[101] H. J. Seo and P. Milanfar. Detection of human actions from a single example. In *IEEE Intl. Conf. on Computer Vision*, 2009.

[102] Ricky J. Sethi and Amit K. Roy-Chowdhury. Modeling and recognition of complex multi-person interactions in video. In *Multimodal Pervasive Video Analysis*, 2010.

[103] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[104] Z. Si, M. Pei, B. Yao, and S.C. Zhu. Unsupervised learning of event and-or grammar and semantics from video. In *ICCV*, 2011.

[105] N Simon, J Friedman, and T Hastie. A sparse-group lasso. In *Journal of Computational and Graphical Statistics*, 2013.

[106] B. Song, T. Jeng, E. Staudt, and A. Roy-Chowdury. A stochastic graph evolution framework for robust multi-target tracking. In *Euro. Conference on Computer Vision*, 2010.

[107] Xi Song, Tianfu Wu, Yunde Jia, and Song-Chun Zhu. Discriminatively trained and-or tree models for object detection. In *CVPR*, 2013.

[108] Ju Sun, Xiao Wu, Shuicheng Yan, Loong-Fah Cheong, Tat-Seng Chua, and Jintao Li. Hierarchical spatio-temporal context modeling for action recognition. In *CVPR*, 2009.

[109] Zhendong Sun and Shuzhi Sam Ge. *Stability Theory of Switched Dynamical Systems*. Springer, 1st edition, 2011.

[110] Kevin Tang, Li Fei-Fei, and Daphne Koller. Learning latent temporal stucture for complex event detection. In *CVPR*, 2012.

[111] Ben Taskar, Vassil Chatalbashev, and Daphne Koller. Learning associative markov networks. In *ICML*, 2004.

[112] Choon Hui Teo, Quoc Le, Alex Smola, and S. V. N. Vishwanathan. A scalable modular convex solver for regularized risk minimization. In *SIGKDD*, pages 727–736, 2007.

[113] Choon Hui Teo, S.V.N. Vishwanathan, Alex Smola, and Quoc V. Le. Bundle methods for regularized risk minimization. In *The Journal of Machine Learning Research*, 2010.

[114] Choon Hui Teo, S.V.N. Vusgwanathan, Alex Smola, and Quoc V. Le. Bundle methods for regularized risk minimization. In *Journal of Machine Learning Research*, 2010.

[115] R. Tibshirani. Regression shrinkage and selection via the lasso. In *Journal of the Royal Statistical Society*, ser. b,vol. 58,no. 3,pp. 549-571,1996.

[116] S. Tran and L. S. Davis. Visual event modeling and recognition using Markov logic networks. In *Euro. Conference on Computer Vision*, 2008.

[117] Thanh Tran, Haofen Wang, Sebastian Rudolph, and Philipp Cimiano. Top-k exploration of query candidates for efficient keyword search on graph-shaped (RDF) data. In *ICDE*, 2009.

[118] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. In *Journal of Machine Learning Research 6*, pages 1453–1484, 2005.

[119] Kewei Tu, Maria Pavlovskaia, and Song-Chun Zhu. Unsupervised stucture learning of stochastic and-or grammars. In *NIPS*, 2013.

[120] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. In *IEEE Trans. on Circuits and Systems for Video Technology*, 2008.

[121] P.K. Turaga, R. Chellappa, V.S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Trans. on Circuits and Systems for Video Technology*, 18(11):1473–1488, November 2008.

[122] Douglas L. Vail and Manuela M. Veloso. Feature selection for activity recognition in multi-robot domains. In *AAAI*, 2008.

[123] C. Wang, S. Yan, L. Zhang, and H. Zhang. Multi-label sparse coding for automatic image annotation. In *CVPR*, 2009.

[124] Jiang Wang, Zhuoyuan Chen, and Ying Wu. Action recognition with multiscale spatio-temporal contexts. In *CVPR*, 2011.

[125] Xiaogang Wang, Xiaoxu Ma, and W. Eric L. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. In *PAMI*, 2009.

[126] Yang Wang and Greg Mori. Max-margin hidden conditional random fields for human action recognition. In *CVPR*, 2009.

[127] Stephen J. Wright, Robert D. Nowak, and Mario A. T. Figueiredo. Sparse reconstruction by separable approximation. In *IEEE Transactions on Signal Processing*, 2009.

[128] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[129] Jianchao Yangy, Kai Yuz, Yihong Gongz, and Thomas Huangy. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[130] Bangpeng Yao and Li Fei-Fei. Modeling mutual context of object and human pose in human object interaction activities. In *CVPR*, 2010.

[131] Chun-Nam John Yu and Thorsten Joachims. Learning structural svms with latent variables. In *International Conference on Machine Learning*, 2009.

[132] A. L. Yuille and Anand Rangarajan. The concave-convex procedure (CCCP). In *Neural Computation*, volume 15, April 2003.

[133] Shaoting Zhang, Junzhou Huang, Yuchi Huang, Yang Yu, Hongsheng Li, and Dimitris N. Metaxas. Automatic image annotation using graph sparsity. In *CVPR*, 2010.

[134] Elena Zheleva, Lise Getoor, and Sunita Sarawagi. Higher-order graphical models for classification in social and affiliation networks. In *NIPS*, 2010.

[135] F. Zhou and F. De la Torre. Auto-grouped sparse representation for visual analysis. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, under review.

[136] Z. Zhou, A. Wagner, H. Mobahi, and J. Wright. Face recognition with contiguous occlusion using markov random fields. In *ICCV*, 2009.

[137] Jun Zhu, Eric P. Xing, and Bo Zhang. Primal sparse max-margin markov networks. In *KDD*, 2009.

[138] Y. Zhu, N. Nayak, and A. Roy-Chowdhury. Context-aware activity recognition and anomaly detection in video. In *IEEE Journal of Selected Topics in Signal Processing*, 2013.

[139] Y. Zhu, N. M. Nayak, and A. K. Roy-Chowdhury. Modeling multi-object interactions using ”string of feature graphs”. In *Journal of Computer Vision and Image Understanding*, 2012, DOI.

[140] Y. Zhu, Nandita M. Nayak, and Amit K. Roy-chowdhury. Context-aware modeling and recognition of activities in video. In *CVPR*, 2013.

[141] Y. Zhu, Nandita M. Nayak, and Amit K. Roy-Chowdhury. Context-aware activity modeling using hierarchical conditional random fields. In *PAMI*, DOI 10.1109/TPAMI.2014.2369044.

[142] Y. Zhu and Amit K. Roy-Chowdhury. Graphical models for context-aware analysis of continuous videos. In *GlobalSIP*, 2013.

[143] Z.Zivkovic. Improved adaptive Gaussian mixture model for background subtraction. In *ICPR*, 2004.

Appendix A

Publication List