VIDEO ANALYTICS WITH SPATIO-TEMPORAL
CHARACTERISTICS OF ACTIVITIES
Guangchun Cheng

Dissertation Prepared for the Degree of
DOCTOR OF PHILOSOPHY

UNIVERSITY OF NORTH TEXAS
May 2015

APPROVED:

Bill P. Buckles, Major Professor
Yan Huang, Committee Member
Kamesh Namuduri, Committee Member
Paul Tarau, Committee Member
Barrett Bryant, Chair of the Department of
Computer Science & Engineering
Costas Tsatsoulis, Dean of the College of
Engineering
Costas Tsatsoulis, Interim Dean of the
Toulouse Graduate School

As video capturing devices become more ubiquitous from surveillance cameras to smartphones, the demand of automated video analysis is increasing as never before. One obstacle in this process is to efficiently locate where a human operator’s attention should be, and another is to determine the specific types of activities or actions without ambiguity. It is the special interest of this dissertation to locate spatial and temporal regions of interest in videos and to develop a better action representation for video-based activity analysis.

This dissertation follows the scheme of “locating then recognizing” activities of interest in videos, i.e., locations of potentially interesting activities are estimated before performing in-depth analysis. Theoretical properties of regions of interest in videos are first exploited, based on which a unifying framework is proposed to locate both spatial and temporal regions of interest with the same settings of parameters. The approach estimates the distribution of motion based on 3D structure tensors, and locates regions of interest according to persistent occurrences of low probability.

Two contributions are further made to better represent the actions. The first is to construct a unifying model of spatio-temporal relationships between reusable mid-level actions which bridge low-level pixels and high-level activities. Dense trajectories are clustered to construct mid-level actionlets, and the temporal relationships between actionlets are modeled as Action Graphs based on Allen interval predicates. The second is an effort for a novel and efficient representation of action graphs based on a sparse coding framework. Action graphs are first represented using Laplacian matrices and then decomposed as a linear combination of
primitive dictionary items following sparse coding scheme. The optimization is eventually formulated and solved as a determinant maximization problem, and 1-nearest neighbor is used for action classification. The experiments have shown better results than existing approaches for regions-of-interest detection and action recognition.
Copyright 2015
by
Guangchun Cheng
ACKNOWLEDGMENTS

At this moment when this dissertation has been written, I wish to express my gratitude to people without whom this dissertation could not be in its present form. Over the past five years, they have given me enormous support and encouragement.

First and foremost I am grateful to my advisor, Dr. Bill Buckles, for his visionary support and unwavering guidance throughout the course of my Ph.D. research. He accepted me into his wonderful group and directed me to a career of computer science researcher and scholar. Though he is extremely knowledgeable and provides immediate advising when needed, Dr. Buckles encourages independent research and thinking, which was demonstrated in projects I worked on. Without him this dissertation research would not have been completed.

My sincere gratitude is reserved to the members of my Ph.D. committee, Dr. Yan Huang, Dr. Kamesh Namuduri and Dr. Paul Tarau. Each of them provided me help in various ways, from constructive suggestions and discussions to generous financial support.

I also extend my thanks to my friends and the department’s staff for their various forms of help. They made this a joyful and rewarding journey. Of them I especially want to acknowledge Yiwen Wan, Wasana Santiteerakul, Shijun Tang, Yassine Belkhouche, Qiang Guan, Abdullah Naseer, Stephanie Deacon, Kwang Buckles and Kevin & Lisa Evans. Their care and belief in me helped me overcome setbacks, and I appreciate their friendship from my heart.

My thanks also go to Toulouse Graduate School, which supported my research through Master’s and Doctoral Fellowship (MDF) and Thesis and Dissertation Fellowship (TDF). These financial support allowed me to focus on the research.

Most importantly, I am greatly indebted to my family for their unconditional love, encouragement and comforts. I feel grateful to my brother and sister for being with my father. Finally, this journey would not start without my wife Yanan, who has been motivating and supporting my research and life. Thank you for being always caring and encouraging, and bringing to our life the joy and beautiful Aaron our son.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGMENTS iii</td>
</tr>
<tr>
<td>LIST OF TABLES vii</td>
</tr>
<tr>
<td>LIST OF FIGURES viii</td>
</tr>
</tbody>
</table>

## CHAPTER 1 INTRODUCTION 1

1.1. Motivation 2

1.1.1. Detecting Regions of Interest in Videos 2

1.1.2. Exploiting Representation of Temporal Relationships Between Features 2

1.2. Problem Definition and Objectives 3

1.3. Contributions 4

1.4. Dissertation Organization 7

## CHAPTER 2 BACKGROUND AND RELATED RESEARCH 9

2.1. Detection of Regions of Interest 9

2.2. Video Representations 11

2.2.1. Local Features 11

2.2.2. Trajectory as a Feature 13

2.2.3. Mid-level Representation Using Dense Trajectories 14

2.2.4. Temporal Description of Features 15

2.2.5. Action Description Models 16

2.3. Sparse Coding for Spatio-Temporal Information 17

2.3.1. Spatio-Temporal Volumes 17

2.3.2. Spatio-Temporal Trajectories 18

2.3.3. Representations for Spatio-Temporal Relationships 20
2.3.4. Sparse Coding for Visual Computing 21

2.4. Machine Learning for Video Activity Analysis 22
   2.4.1. Learning Based on BoW Model 22
   2.4.2. Learning Based on Graph Models 22

CHAPTER 3 DETECTING SPATIO-TEMPORAL REGIONS OF INTEREST 24
   3.1. Introduction 24
   3.2. Pixel-level Wide-View ROI Detection 26
      3.2.1. Feature: A 3D Structure Tensor 26
      3.2.2. Representation: Probability Distribution of Structure Tensors 28
      3.2.3. Interest Detection as Low Probability Occurrence 31
   3.3. Experimental Results 32
      3.3.1. Datasets 33
      3.3.2. Parameter Sensitivity Experiments 34
      3.3.3. Experiment Design and Results 36
   3.4. Relationship between Eigenvectors and Motion Direction 40
   3.5. Summary 42

CHAPTER 4 TEMPORAL STRUCTURE FOR ACTION RECOGNITION 43
   4.1. Introduction 43
   4.2. Structure of Trajectory Groups 45
      4.2.1. Dense Trajectories 45
      4.2.2. Trajectory Descriptors 48
      4.2.3. Grouping Dense Trajectories 49
      4.2.4. Bag of Components 50
      4.2.5. Temporal Structure 51
   4.3. Learning and Recognition 54
   4.4. Experimental Results 55
      4.4.1. KTH Dataset 57
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4.2. Weizmann Dataset</td>
<td>58</td>
</tr>
<tr>
<td>4.4.3. Comparison with Logic-Based Approach</td>
<td>59</td>
</tr>
<tr>
<td>4.5. Summary</td>
<td>62</td>
</tr>
<tr>
<td>CHAPTER 5 ACTION GRAPH DECOMPOSITION BASED ON SPARSE CODING</td>
<td>65</td>
</tr>
<tr>
<td>5.1. Introduction</td>
<td>65</td>
</tr>
<tr>
<td>5.2. Action Graph from Dense Trajectories</td>
<td>65</td>
</tr>
<tr>
<td>5.2.1. Grouping Dense Trajectories</td>
<td>66</td>
</tr>
<tr>
<td>5.2.2. Action Graphs</td>
<td>67</td>
</tr>
<tr>
<td>5.3. Action Graph Representation with Sparse Coding</td>
<td>68</td>
</tr>
<tr>
<td>5.3.1. Laplacian Matrix of Action Graphs</td>
<td>68</td>
</tr>
<tr>
<td>5.3.2. Sparse Coding for Action Graphs</td>
<td>69</td>
</tr>
<tr>
<td>5.3.3. Distance between Action Graphs</td>
<td>70</td>
</tr>
<tr>
<td>5.4. Experimental Results</td>
<td>71</td>
</tr>
<tr>
<td>5.5. Summary</td>
<td>73</td>
</tr>
<tr>
<td>CHAPTER 6 CONCLUSIONS AND FUTURE WORK</td>
<td>74</td>
</tr>
<tr>
<td>6.1. A Nonparametric Approach for ROI Detection</td>
<td>74</td>
</tr>
<tr>
<td>6.2. Combining Temporal Relationships with BoW for Action Recognition</td>
<td>75</td>
</tr>
<tr>
<td>6.3. Novel Temporal Relationship Representation</td>
<td>75</td>
</tr>
<tr>
<td>6.4. Discussions on Future Work</td>
<td>76</td>
</tr>
<tr>
<td>APPENDIX RELATED PUBLICATIONS</td>
<td>77</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>79</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 3.1.</td>
<td>Areas under curve (AUC) for different Gaussian derivative filter sizes</td>
<td>36</td>
</tr>
<tr>
<td>Table 4.1.</td>
<td>Temporal matrix construction based on Allen predicates.</td>
<td>55</td>
</tr>
<tr>
<td>Table 4.2.</td>
<td>Accuracy for KTH dataset</td>
<td>57</td>
</tr>
<tr>
<td>Table 4.3.</td>
<td>Accuracy for Weizmann dataset</td>
<td>59</td>
</tr>
<tr>
<td>Table 4.4.</td>
<td>Example predicates and rules to define actions.</td>
<td>62</td>
</tr>
<tr>
<td>Table 4.5.</td>
<td>Accuracy comparison between proposed approach and logic-based approach</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>(8-actions).</td>
<td></td>
</tr>
<tr>
<td>Table 4.6.</td>
<td>Comparing proposed approach with logic-based approaches.</td>
<td>63</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

| Figure 1.1. | Research topics of the dissertation: Regions of interest are located first; mid-level actionlets are shared among domains of applications whose temporal relationships are described using Allen’s predicates, and the sparse coding is perform on the action graphs. | 5 |
| Figure 2.1. | Results of STIP detection for a synthetic sequence with impulses having varying extents in space and time (left) and a realistic motion of the legs of a walking person (right) [72] (©2005 Springer-Verlag). | 12 |
| Figure 2.2. | Class diagram showing examples of feature detectors and descriptors as implemented in OpenCV 2.4 [1]. | 13 |
| Figure 2.1. | Illustration of dense trajectory description from [139] (©2011 IEEE) Left: Feature points are sampled densely for multiple spatial scales. Middle: Tracking is performed in the corresponding spatial scale over L frames. Right: Trajectory descriptors of HOG, HOF and MBH. | 19 |
| Figure 3.1. | Flow chart of the proposed method. Training and testing share feature extraction and computation of feature distance. The probability density function (PDF) is learned for sampled sites in the video. | 27 |
| Figure 3.2. | Example of activity representation by the distribution of the structure tensors at sampled sites. | 31 |
| Figure 3.3. | Anomaly detection results in a video with multiple types of motion. (a) A scene with a water fountain. No object moves behind during training. (b) Activity distribution is learned using 3D structure tensors. Several distributions are shown on their locations respectively. (c) Anomaly map shows ROI during the test phase. | 32 |
| Figure 3.1. | Detection performance under different temporal sliding window $W_T$. | 35 |
Figure 3.2. Detection performance under different Gaussian derivative filter windows $w_f$. 36

Figure 3.3. Abandoned bag detected as low-probability occurrence. The distribution is $\hat{f}(d)$ where the site is marked with a red $x$ on the top-left figure. Occurrence probability and the detection result are shown in the bottom two figures, respectively. 37

Figure 3.4. ROIs detected for both spatial and temporal pattern changes. Regions within red circles are regarded as ROIs. (a)-(b): abandoned bag is of interest but the usual pedestrian bypassing is not; (c)-(d) suspicious pedestrian lounging is but usual normal traffic is not; (e)-(f): car moving behind is but usual water fountain is regarded as background motion; (g)-(h): boat is of interest but waving water is regarded as background motion. 38

Figure 3.5. Receiver operating characteristic curve of hits for the video from BOSTON and CAVIAR datasets. (3.5(a)) shows the detection ROC of unusual motion (car moving behind fountain) from the background motion (the water spray). (3.5(b)) gives a comparison of the results from our method and a semi-supervised trajectory-based method [119], a context-based approach [142] and Gaussian mixture [46]. 39

Figure 3.6. At each sampled threshold, the average localization precision and recall are shown for the entire video of “car passing behind water fountain”. 40

Figure 4.1. Examples of trajectories from object-based tracking (first row) and dense optical flow-based feature tracking (second row). The dense trajectories are grouped based on their spatio-temporal proximity. 46

Figure 4.2. Dense trajectory extraction. Feature points are extracted at different scales on a grid basis. Red points are new detected feature points, and green points are the predicted location of feature points based on optical flow. 47

Figure 4.3. Illustration on component assignment of trajectory groups. Trajectory groups C1 and C2 are mapped to the same component W1. 51

Figure 4.4. Allen’s temporal relationships between two intervals. 52
Figure 4.5. Histograms of temporal relation *meets* for five different actions in KTH dataset. The X and Y axes are the types of group codes, and the Z values are the frequency before normalization. Among them, histograms of jogging and walking are relatively close to each other. So are boxing and handclapping.  

Figure 4.1. Flowchart of learning and classification. (Better viewed in color.)  

Figure 4.1. Sample frames from KTH dataset (row (a)) and Weizmann dataset (row (b) and (c)).  

Figure 4.2. Confusion matrix for KTH dataset. Blue means 0, and red means 1. (Better viewed in color.)  

Figure 4.3. Confusion matrix for Weizmann dataset. Blue means 0, and red means 1. (Better viewed in color.)  

Figure 4.4. Segmentation of trajectory of a body part. The top left is the original trajectory for right hand, the bottom right is the direction assignment, and the others are the segmentation results.  

Figure 5.1. Illustration on trajectory grouping based on spatio-temporal proximity.  

Figure 5.2. Laplacian matrix of action graphs for *overlaps* of five different actions in KTH dataset. The X and Y axes are different types of actionlets.  

Figure 5.1. Plot of the optimal sparse coding solutions. Notice the the sparseness of the coefficients.  

Figure 5.2. Average sparse coding coefficients $s_i$ for each category of videos  

Figure 5.3. The maximum coefficients from the sparse coding.
CHAPTER 1

INTRODUCTION

Human has an extraordinary capacity for perceiving and processing action from visual information. Most vision-related tasks can be fulfilled with very little time, such as object localization and tracking, human action recognition and detection, and human activity and even intents inference to name a few. Of all the sensory information relayed to the brain, four-fifths is visual in origin according to the American Optometric Association. Unsurprisingly, we have seen an exponential increasing use of videos and corresponding automated processing system in many areas for recent years. The most important yet challenging problem for many artificial intelligent video systems is to understand the actions or activities contained in videos. Action understanding can be used for applications such as video indexing and search, sports video analysis, personalized advertising, and video surveillance among many others [100]. Research on video action/activity understanding has evolved together with the increasing complexity and variety of different video analysis tasks, from simple gesture recognition in constrained environment to human activity detection in constraint free settings. Many efforts and breakthroughs haven been done for automated action and activity understanding from videos (as surveyed in [42, 90, 3, 105, 2, 141]), however it remains as an unsolved problem – new tasks are emerging and the solutions to most existing tasks are far from perfect.

A video can be thought of as a visual document which may be represented from different dimensions such as frames, objects and other different levels of features. Since the proposal of corner points in 1980s, local features have been designed and applied in image and video analysis with great success for many tasks. For video activity analysis in real scenarios, it is crucial to explore beyond the simple use of local features. Two trends for action representation are becoming evident: (1) instead of using signal-level features for recognition, higher-level features [70, 107, 154] and/or semantics and attributes [80, 150] become common choice of features; (2) the temporal/spatial relationship between features are attracting increasing attention and efforts [30, 88, 124, 129, 69, 154]. This dissertation falls into the second category, and studies the
spatio-temporal characteristics for video activity analysis.

1.1. Motivation

As video capturing devices become more ubiquitous from surveillance cameras to smartphones, the demand of analyzing videos is increasing as never before. Pushing the boundary of efficient video activity analysis would enable more applications. It is the special interest of this dissertation to locate regions of interest (ROI) in videos and to develop a better representation for the actions in videos.

1.1.1. Detecting Regions of Interest in Videos

Nowadays cameras have recorded sufficient visual information for different purposes, and it is impossible for human operators or automated systems to analyze each pixel in detail. It is ideal that only interesting portion or actions of videos are analyzed. From a bottom-up perspective, interest is usually related to anomalies. Most anomaly detection systems are based on trajectories associated with the object’s speed and other properties. Then the extracted features are used to develop a model to represent the different cases. By defining and comparing the similarity between trajectories, actions of interest can be detected. Though video activity analysis is a computationally intensive task, a commonly-used framework for the above process requires modeling the entire video to obtain the actions or activities of interest. This often involves computation whose results are not of final interest. We refer to this pattern as “recognize to locate” because it recognizes everything to locate the interest.

Therefore, decoupling of interesting actions and non-interesting actions allows a better understanding of the actions and a more reasonable allocation of computing resources. We proposed a different paradigm to locate potential regions of interest prior to higher-level processing such as object tracking or action recognition. Moreover, we intend to have a single model that can locate both spatial and temporal regions of interest in videos.

1.1.2. Exploiting Representation of Temporal Relationships Between Features

Traditional approaches extract local features from video frames, then a vector representation of the features is obtained by quantizing and pooling the features. Such feature vectors are fed
into classifying or clustering algorithms to analyze the activities in videos. One commonly-used pooling policy is so-called bag of features which represents actions by a histogram based on a dictionary. Though simple and efficient in constrained environment, such pooling abandons the temporal/spatial relationships between the features, which are important motion characteristics for actions or activities.

Just like bag of words is not regarded as a good language model in natural language processing because of its loss of word ordering and ignorance of semantics [88], representation in computer vision needs a better model to describe the interaction between “features” as well. One of most important factors to understand activities is the temporal ordering of short-term actions we call actionlets. The temporal relationships are typically represented by graphical models such as hidden Markov models [98, 155] and dynamic Bayesian networks [35, 43]. Unfortunately, these approaches generally assume a fixed number of actions, and require large training sets in order to learn the model structure and their parameters.

Another popular representation is logic-based models [92, 130], where the relationships are represented by logic rules. It usually requires a large amount of effort from videos such as feature extraction, predicate and rule definition and feature-predicate mapping among others.

We need a method that can make use of the large volume of temporal information in videos and combine it with traditional method for activity analysis. A data-driven approach is desirable because it will free human operators from time-consuming model construction. In addition, the variation of temporal properties is huge even within the same category of actions. This is expected for real-world videos where it is hard to find two instances of the same action exactly the same even from the same performer. This draws us to the exploration of the unique units of the temporal relationship representations.

1.2. Problem Definition and Objectives

The existing automated activity analysis systems are far from perfect. The ultimate goal of this dissertation is to understand the activities in videos in an efficient way. To achieve that goal, we exploit the spatio-temporal relationships of mid-level action representations from pixels. We encountered three problems and proposed corresponding solutions to them.
Problem 1: Is there a general model that can locate both spatial and temporal regions of interest in videos with the same set of parameters? Usually only a few parts, both spatially and temporally, of a video are of our interest, so it is beneficial to detect them for its own sake or for other processing such as action recognition.

Problem 2: When we get started to analyze the actions/activities in a video, how do temporal relationships help the analysis and how the they can be characterized and used?

Problem 3: Further, what structures do the temporal relationships share in common inter- and intra-classes of actions? What is the best representation for those temporal relationships?

We investigate each of these problems and propose methods for them as shown in Section 1.3 and following chapters in detail. Figure 1.1 provides a visual overview of the problems and relationship between them.

1.3. Contributions

This dissertation contributes to each of the three components described above in the following way:

[ROI] Detecting both spatial and temporal regions of interest in videos under a unifying framework. The proposed approach uses the same set of parameters and their settings. This enables detection of interesting part of videos prior to detailed analysis such as action recognition.

[STR] Event spatio-temporal relationship representation. Mid-level actions and their spatio-temporal relationships are exploited quantitatively using action graphs. Mid-level actions are designed to be application-independent and may be deployed as a basis for describing high-level events in many domains.

[SCR] Recognition framework based on sparse coding\(^1\), which also mimics human vision system to represent and infer knowledge. We put forward “action graphs” to represent the temporal relationships. To our best knowledge, we are the first using sparse graph coding for event analysis.

\(^1\text{Sparse coding, to be brief, is the representation of data with few (many only one) exemplary cases.}\)
Figure 1.1. Research topics of the dissertation: Regions of interest are located first; mid-level actionlets are shared among domains of applications whose temporal relationships are described using Allen’s predicates, and the sparse coding is perform on the action graphs.
The proposed research follows the scheme of “locating” ([ROI]) and then “recognizing” ([STR] and [SCR]). That is locations of potentially interesting activities are identified before performing in-depth analysis (i.e., recognition in this context). This is especially important for event analysis due to the massive amount of videos. For each component, we propose new methods that either unify different scenarios or extend state-of-the-art approaches. The research tackles major issues with video-based event analytics, and promotes its advancement with peer recognition.

[ROI]: Following the proposed scheme, we improve the computation efficiency of video event analysis by first locating potential regions of interest. With the ubiquitous availability of camera sensors, video analysis is usually large-scale with many cameras or camera arrays. The data volume of videos is high especially for high-definition recordings. Therefore, it becomes essential to allocate computing resources to the analysis of meaningful actions and objects. We take a “locating and recognizing” approach which allocates computing resources to the analysis of activities within regions of interest, not only spatially in one frame but also temporally in one video. This innovation increases the reach of real-time video event analysis in many fields, and has both intellectual and practical benefits.

[STR]: The investigator has made progress by being among the first to construct a unifying model of spatio-temporal relationships between mid-level actions. Video-based event analysis has a special demand on high-level understanding of the depicted scenes. I recognize that there is a huge gap between numeric pixel input and categorical events to interpret videos. To bridge the gap, we have conceived reusable mid-level actions which with their relationships are represented by spatio-temporal graphs. In general, STR ensures cross-domain application of the new model. Thus, this research is well-conceived and innovative for its generic action representation.

[SCR]: Sparse graph coding has been created by us and is a new use of the sparse coding framework. Its application is applicable but not limited to event recognition. I found a limitation of sparse coding, although it enjoys success in many fields. There are fundamental problems when we extend sparse coding from one dimensional to two dimensional, such as the requirements for the graphs and the mathematical optimization techniques. I have made efforts in this dissertation and proposed a novel spatio-temporal relationship representation based on sparse coding. While
built on existing work [120, 157], our approach is clearly distinct.

Events in different applications vary according to the goal, yet they share mid-level actions. Our framework extracts mid-level action representation with spatio-temporal relationships. These mid-level actions are building blocks of more complex events in specific applications. Improvement on sparse coding has been pursued to exploit its representation power for the spatio-temporal graphs.

To sum up, this research benefits many sectors using video-based monitoring systems by relieving the need for constant attention by human operators. For the purpose of high performance and quick deployment, we propose a new framework, upon which events across different applications can be recognized over shared mid-level actions efficiently when preceded by the detection of regions of interest.

1.4. Dissertation Organization

The remaining of this dissertation is organized as follows.

Chapter 2 reviews the related work on action and activity analytics for videos. I first discuss the exiting studies on detection of regions of interest and anomalies in videos. Then, related research on the modeling of spatio-temporal relationship in video action recognition is presented. In the end, I survey sparse coding and discuss its use for image and video understanding.

Chapter 3 presents our work on video-based region-of-interest detection, and the key outcome is a general method for both spatial and temporal region-of-interest detection. It is an unsupervised approach base on use of 3D structure tensors.

Chapter 4 discusses the use of temporal relationships between trajectory groups in videos for action recognition. First, we show the clustering of trajectories to obtain our mid-level representation for short-term actions referred to as actionlets. Then, the temporal relationship between actionlets are modelled using Allen’s temporal predicates. Finally, an extended bag-of-features model combined with temporal relationship is used for action representation and classification.

Chapter 5 is our efforts to explore the underlying structure for the temporal relationships. We propose a new temporal relationship representation, called action graphs based on Laplacian
matrices and Allen’s temporal relationships. It is our intention for action graphs as a general representation to incorporate spatial relationships in future.

Chapter 6 concludes the dissertation with a summary of this work and remarks on the direction of future research.
CHAPTER 2

BACKGROUND AND RELATED RESEARCH

Activity analysis from videos or images is one of the most important and studied research areas in machine vision community, which has attracted much research from neuroscience and computer science. Many breakthroughs have been made since 1980s, including new local features, action representations, machine learning algorithms and related tools for efficient computation. There exists an emerging trend that spatial/temporal characteristics are exploited for activity analysis in videos, which is the focus of this study. We divide our research into two phases: detection of interesting regions in videos and recognition of actions within them. Below we provide a review on related research for both tasks.

2.1. Detection of Regions of Interest

There has been much research on anomaly detection in computer vision and other fields. ROIs in videos consist of two categories: spatio-temporal outliers and task-driven occurrences. Task-driven ROIs are determined by not only the low-level features but also the observer’s tasks and context information [143]. Although task-driven approaches are appealing, the current focus is on the detection of spatio-temporal outliers. In this dissertation, we focus on spatio-temporal outlier detection with the aim of pixel-level anomalous detection in videos, especially those with wide-angle views.

The representation of events basically includes object trajectories and functional descriptors. In a trajectory-based representation, objects are detected, tracked, and the trajectories are used to model events. Those trajectories deviating greatly from the more frequent ones are deemed outliers/anomalies [61, 119]. In a functional descriptor-based representation, static background and background motion are modeled by functions using methods such as mixture of Gaussians [46] and kernel density estimation [89].

There exist two different paradigms for video anomaly analysis, namely (A) event recognition followed by anomaly determination and (B) (potential) anomaly detection followed by event analysis. In paradigm A, most anomaly detection systems are based on tracked trajectories associ-
ated with the object’s speed and other properties [151, 117]. Then the extracted features are used to develop the models in a supervised manner. [104] used a single-class support vector machine (SVM) to cluster the trajectories for anomalous event detection. [95] used a hidden Markov model to encode the spatiotemporal motion characteristics, and abnormal trajectories were detected. By defining and comparing the similarity between trajectories, [160] proposed a framework for anomaly detection in different scenes. A general weakness of paradigm A is that the anomaly detection accuracy depends on the results of tracking or event recognition. For wide-angle views such as surveillance videos, anomalies are rare, so tracking each object is not necessarily needed.

In the second paradigm, a potential ROI is first detected. [113] proposed a strategy that detects ROIs prior to higher-level processing such as object tracking, tagging and classification. It avoids unnecessary processing caused by assessing normal recurring activities by constructing a behavior image, a.k.a background activity map. [103] compared explicit event recognition and anomaly detection, then combined both for surveillance anomaly detection. [44] recognized scene events without tracking with the aid of pixel change history (PCH). [158] borrowed an idea from document-keyword clustering and used a co-occurrence matrix of spatial motion histograms to detect unusual activities in videos. Other research on anomaly detection include [147] who modeled background activity with moving blobs. [65] who developed HMMs with spatiotemporal motion patterns, and [62] who used MRFs to detect both local and global abnormalities, among others.

Some of the methods above are parametric while the others are nonparametric. Parametric methods such as MRF [62] and HMM [66] have been used by many. Although they are concise and more precise if the assumptions are correct, parametric approaches generally cannot be robustly applied to different scenes without modification. It involves model selection, estimation of the model order, and the model parameters. [44] provide an excellent example of estimating both the model order and the parameters. There are also many nonparametric approaches, most of which use local information and detect anomalies in a bottom-up manner. Based on the framework of image visual saliency detection [53], [52] developed a model that computes the degree of low-level surprise at each location, which allows more sophisticated analysis within the most “surprising” subsets. [40] also extended the computational framework of [53] to identify inconsistent regions in
video streams. As another typical nonparametric method, kernel density estimation [14] was used to detect network anomalies subject to a tolerance [4]. [74] compared the Gaussian mixture model (GMM) with a kernel density estimator (KDE) using sea traffic data. They concluded that KDE more accurately characterizes features in the data but the detection accuracy of the two models is similar. Surveillance videos exhibit many variations in both the actions involved and the scene conditions, we base our work on nonparametric methods.

2.2. Video Representations

The research on action recognition was on its sub-topics such as gesture recognition for human-computer interfaces, facial expression recognition, and movement behavior recognition for video surveillance. For the past decade, many action recognition models, algorithms and datasets have been proposed. On one side, simple actions such as those in KTH or Weizmann datasets already have near-perfect solution; on the other side, more challenging datasets are created to advance the automated action recognition for real scenarios. We refer the reader to Poppe [105], Aggarwal and Ryoo [2] and Jiang et al. [56] for a complete survey of the topic. Below we divide the discussion on the related research into local feature representation, mid-level feature representation, temporal relationships, and action description models.

2.2.1. Local Features

Features play a critical role in video action recognition, and feature engineering has been one of the most active topics in computer vision since the beginning. Good features can lead to appealing performance even an average machine learning technique is used. This discovery leads to a popular research area recently called representation learning, also known as deep learning [11]. Many local features have been “designed,” and we discuss the most significant ones below.

Because of its robustness to partial occlusion, local features have a long history in image and video representation, and they usually refer to 2D or 3D interest points. The interest points, a.k.a corner points in some cases, are firstly detected/located by algorithms such as Moravec operator [94], Harris and Stephens detector [49], and tracking-friendly feature detector [118]. Another popular and widely used one is Lowe’s difference of Gaussian (DoG) [81] which detects points
that differs from the surrounding area in a scale-independent fashion. It applies Gaussian filtering to one frame of the video at different scales, and the difference between blurred images of two consecutive scales is obtained as an approximation to Laplacian of Gaussian (LoG). For comparisons of different local feature detectors, reader is referred to [87] for a comprehensive review of several local patch detectors. For video representation, Laptev [72] extended spatial interest points to spatio-temporal interest points (STIP) and used them for video event recognition. Figure 2.1 illustrates the STIP detection. Most 2D feature detectors and descriptors have been extended to accommodate 3D cases, such as 3D scale-invariant feature transform (SIFT3D) [115] and 3D histogram of oriented gradients (HOG3D) [64] among others.

Figure 2.1. Results of STIP detection for a synthetic sequence with impulses having varying extents in space and time (left) and a realistic motion of the legs of a walking person (right) [72] (©2005 Springer-Verlag).

Many descriptors have been designed for detected local features. For some detectors, corresponding descriptors were designed when they were proposed, such as SIFT [81]. Figure 2.2 shows 2D local feature detectors and descriptors as implemented in Opencv 2.4.1 [1]. As we can see, some methods, such as FAST [111] and good features to track (GFTT Detector) [118], are only for feature detection, while others include a combination of detectors and descriptors, such as Speeded-Up Robust Features (SURF) [10] and SIFT [81]. Unsurprisingly, more efficient feature detectors and descriptors are being designed. A most recent work combined to OpenCV is KAZE
features [5].

![Class diagram showing examples of feature detectors and descriptors as implemented in OpenCV 2.4](image)

**Figure 2.2.** Class diagram showing examples of feature detectors and descriptors as implemented in OpenCV 2.4 [1].

To organize all local features and have a compact representation of the video content, Bag-of-words (BoW) is a popular representation in computer vision [115, 97, 122]. Bag-of-words is a global descriptor which characterizes the occurrence counts of each distinctive feature and put them into a vector. BoW is simple and has efficient learning methods existing. While it achieved success for some action recognition tasks, BoW model discards the spatial or temporal relationships between local features and has difficulty in analysing complex realistic videos. Emphasis on spatial and/or temporal properties in videos is shared within computer vision research community, and is demonstrated by recent publications.

### 2.2.2. Trajectory as a Feature

Trajectories contain spatio-temporal information captured by tracking local features or detected objects. Theoretically, trajectories are superior to descriptors mentioned above such as HOG or SURF because trajectories require the detection of a discriminative point or region over a sustained period of time, unlike the latter that computes pixel-based statistics subjected to a predefined spatio-temporal neighborhood [56]. Yilmaz *et al.* [152] and recently Li *et al.* [77] provide comprehensive surveys on object tracking.
In this dissertation research, the foundation is on dense trajectories. Dense trajectories have been extracted using optical flow methods [138]. A number of feature detectors such as Harris3D, Gabor Filters, Hessian3D, and SIFT [124] are used to detect interest points that are tracked over time. Even though, ideally, long-duration trajectories are preferred, it has been shown that short-duration dense trajectories significantly outperform tracking of sparse key-points on several human action recognition benchmarks [139]. Other efforts, including this dissertation (see Section 4.2.1), have been made to compensate the camera motion to more descriptive dense trajectories. Jiang et al. [57] proposed to use local and global reference to model the motion of dense trajectories, especially the motion patterns of object relationships. They model the mean motion displacement of a trajectory cluster and then amending each trajectory. The method also uses visual codebook of dense trajectory cluster centers as local reference points and introduces a pairwise motion representation. Very competitive results were observed on several human action recognition benchmarks. Part of the work in Chapter 4 of this dissertation is independently in a similar spirit.

The description of a trajectory can be various such as shape [139, 57], velocity histories [85], histogram of optical flow (HOF), histogram of oriented gradients (HOG) and motion boundary histogram (MBH) [138]. Among them, MBH is based on the derivatives of optical flow field, which is able to suppress constant motion caused by camera movement for example. It has been shown very effective for realistic action recognition in videos. Information is lost when switching to short-duration trajectories causing dense trajectories to be preferred in some studies [139]. With short-duration trajectories, the semantic information is not collected because identifying from which objects or which body parts the trajectories originate is difficult. In order to overcome the semantic information loss, clustering of short-duration trajectories has been used to infer actions [107]. We follow this principle. Individual trajectories (or trajectory clusters) are treated as a sequence of features which indicate atomic or primitive activities.

2.2.3. Mid-level Representation Using Dense Trajectories

The large number of dense trajectories however makes it possible to perform statistical learning of the meaningful clusters. Lan et al. [70] and Raptis et al. [107] proposed to use action parts for action recognition and localization. Both models utilized latent variables and trained
the models discriminatively. Lan et al. [70] constructed a chain-structured graph to represent the relations between features which are action bounding boxes. Spatial relations and temporal smoothness was used to construct the model, and the recognition was achieved by measuring the compatibility between a given video and the configurations of bounding boxes of actions with known labels. Raptis et al. [107] extracted mid-level action parts to express salient spatio-temporal structures of actions in videos, and constructed a graphical model to incorporate appearance and motion constraints for action parts and their dependencies. The action parts in [107] were obtained by forming clusters of trajectories, which are similar to those in this paper. However, Raptis et al. [107] didn’t explore the temporal relations among the action parts. This dissertation develops a method to explore their dependencies and temporal constraints of action parts.

2.2.4. Temporal Description of Features

Most actions, especially high-level actions, are recognized based on two components: meaningful short-term subactions (referred to as actionlets hereafter) and the spatial/temporal arrangement of them. The actionlets can be raw trajectories of tracked points [86], or a cluster of spatio-temporally similar trajectories [107] as stated before. The bag-of-words representation models the actionlets without explicit treatment for spatial/temporal relations. The spatial/temporal relations of actionlets are described by probabilistic models such as hidden Markov models [98, 155] and dynamic Bayesian networks [35, 43]. Unfortunately, these approaches generally assume a fixed number of actions, and require large training sets in order to learn the model structure and their parameters. Bobick and Davis [12] described motion energy image and motion history image to represent the space-time volume of a specific action, and applied template matching for recognition. Description-based models incorporate expert domain knowledge into the definition of actions, and simplify the recognition in structured scenarios [2]. In order to express the temporal relationships, Allen [6] described 13 predicates to describe the temporal relations between any two time intervals. Many approaches are proposed using Allen’s temporal predicates to express temporal relationships between actionlets [91, 116, 112]. Most of such approaches are based on a logic description of the actions. Though versatile, logic-based approaches require a large amount of effort to define and ground predicates from videos, which usually involves feature extraction,
predicate and rule definition and feature-predicate mapping among others.

2.2.5. Action Description Models

Most action recognition models can be categorized into local feature-based, part-based, and global template-based in terms of action representation [154]. Local feature-based models, e.g. bag-of-words models, and global template-based models, e.g. cuboid models, are widely used. They achieve impressive results in certain settings. However, they have limitations in representing complicated actions in the real world. Local features only capture limited spatio-temporal information, and the global templates are not sufficiently flexible to recognize the variations within a class of actions. Part-based models bridge the local features and global actions by modeling the representative motion of object parts [39]. Trajectories are commonly used for motion modeling. Part-based models, however, are not widely employed because of the impedance between low-level signals and high-level action understanding.

Action recognition requires a discriminative description of the videos. Features such as trajectories and local descriptors are commonly used characteristics and are often obtained by encoding frequencies of spatial and/or temporal features. While descriptors from each frame, such as gradients, are appropriate for some scenarios [64], trajectories extracted through tracking are widely used as observations to construct the codebook of “visual words” [146, 125]. Many approaches encode the trajectories using a series of interest point based descriptors including 3D-SIFT [115], 3D-HOG [64], histogram of optical flow (HoF), large displacement optical flow (LDOF) [126], motion boundary histograms (MBH) [34, 138], cuboids or a, combination. The trajectories can be formed by tracking interest points using a tracker such as Kanade-Lucas-Tomasi (KLT) feature tracker [82]. As pointed out by Wang [137], sparse interest points performed worse than dense sampling of tracking points for both image classification and action recognition. Based on this observation, Wang et al. [138] proposed an approach to describe videos by dense trajectories, and designed a descriptor to encode the dense trajectories for action recognition. While dense trajectories provide comprehensive information about the motion in the video, they are both a redundant and low-level representation for forming meaningful codewords. As Liu et al. [79] stated, meaningful grouping of vision features within the original bag-of-words assists the classification. This
notion inspires the methods described here.

2.3. Sparse Coding for Spatio-Temporal Information

For best understanding of the actions and activities in videos, inclusion of the spatial and temporal relationship between objects is crucial, though previous video retrieval systems are largely dependent on statistical moments, shape, color or texture from low-level pixel content. The spatio-temporal relationships in bag-of-features are local. Recently, spatial and temporal information has attracted increasing awareness and changed the way modern research is conducted for video action analysis [108]. Ren et al. [108] provide a review on the use of spatio-temporal information in video retrieval.

This study aims to advance the research on spatio-temporal relationships by exploiting a novel way to represent and characterize the relationships. We base my research on sparse coding, which is an approach for effective feature representation. Below we first summarize popular methods for representations of spatio-temporal information in Section 2.3.1 and Section 2.3.2, followed by a research review on sparse coding in visual computing.

2.3.1. Spatio-Temporal Volumes

Spatio-temporal volumes are intuitive representation of spatio-temporal relationships. It uses a 3D \((x,y,t)\) volume to represent objects, their spatial and temporal relationships. This representation explicitly provides object’s spatial and temporal continuity. The temporal information is contained in the stacked consecutive images of the third dimension, and the activities can be represented by the analysis of this 3D space based on trajectories, shape and motion analysis.

Based on Bobick and Davis’s [13] work on movement, various approaches have been explored to extend it for action recognition. Hu et al. [51] proposed to combine both motion history image (MHI) and appearance information for better characterization of human actions. Two kinds of appearance-based features were proposed. The first appearance-based feature is the foreground image, obtained by background subtraction. The second is the histogram of oriented gradients feature (HOG), which characterizes the directions and magnitudes of edges and corners.

Qian et al. [106] combined global features and local features to classify and recognize
human activities. The global feature was based on binary motion energy image (MEI), and its contour coding of the motion energy image was used instead of MEI as a better global feature because it overcomes the limitation of MEI where hollows exist for parts of human blob are undetected. For local features, an object’s bounding box was used. The feature points were classified using multi-class support vector machines. Roh et al. [110] also extended Bobick and Davis’s [13] MHI from 2-D to 3-D space, and proposed volume motion template for view-independent human action recognition using stereo videos.

Similarly, motivated by a gait energy image [48], Kim et al. [63] proposed an accumulated motion image (AMI) to represent spatiotemporal features of occurring actions. The AMI was the average of image differences. A rank matrix was obtained using ordinal measurement of AMI pixels. The distance between rank matrices of query video and candidate video was computed using $L_1$-norms, and the best match, spatially and temporally, was the candidate with the minimum distance.

Guo [47] viewed an action as a temporal sequence of local shape-deformations of centroid-centered object silhouettes. Each action was represented by the empirical covariance matrix of a set of 13-dimensional normalized geometric feature vectors that captured the shape of the silhouette tunnel. The similarity of two actions was measured in terms of a Riemannian metric between their covariance matrices. The silhouette tunnel of a test video is broken into short overlapping segments and each segment was classified using a dictionary of labeled action covariance matrices and the nearest neighbor rule.

2.3.2. Spatio-Temporal Trajectories

Trajectory-based approaches are based on the observation that the tracking of joint positions is sufficient for humans to recognize actions [58]. Trajectories are usually constructed by tracking joint points or other interest points on human body. Various representations and corresponding algorithms match the trajectories for action recognition.

Messing et al. [86] extracted feature trajectories by tracking Harris3D interest points using a KLT tracker [82], and the trajectories were represented as sequences of log-polar quantized velocities. It used a generative mixture model to learn a velocity-history language and classified
A weighted mixture of bags of augmented trajectory sequences was modeled for action classes. These mixture components can be thought of as velocity history words, with each velocity history feature being generated by one mixture component, and each activity class has a distribution over these mixture components. Further, they showed how the velocity history feature can be extended, both with a more sophisticated latent velocity model, and by combining the velocity history feature with other useful information, like appearance, position, and high level semantic information.

Wang et al. [139] proposed an approach to describe videos by dense trajectories. They sampled dense points from each frame and tracked them based on displacement information from a dense optical flow field. Local descriptors of HOG, HOF and MBH (motion boundary histogram) around interest points were computed. This is shown in Fig. 2.1. We improved Wang’s dense trajectories by removing trajectories from camera motion and used them as low-level features for our action recognition.

In addition to volumes and trajectories, spatio-temporal local features usually capture short-term relationships or context. To overcome the limitations of local features, efforts have been made to obtain holistic representation from many local features, such as from clouds [16] or clusters [59].
of local features. Our representation of spatio-temporal information is based on dense trajectories, and interested reader can refer to survey [2].

2.3.3. Representations for Spatio-Temporal Relationships

For complex video activity analysis, most existing methods build models to represent the relationship between simple actions, especially the temporal relationships. Two popular categories of methods are probabilistic graphical models and logic inductive models.

As a probabilistic approach, hidden Markov model (HMM) and its variants [112, 50] have been popular and obtained success in recognizing gestures and actions for more than two decades [148, 15, 76, 54]. As a probabilistic graphical model, hidden Markov models usually require a large data set for training.

One influential group of researchers have adapted the event calculus (EC) of Kowalski and Sergot [7][8][9][123]. Time is represented by a totally ordered set of scalars so both ordering and cardinality constraints are used. Events, $E$, are instantaneous state changes and fluents, $F$, are actions that occur over time.

Since mid-1990s, researchers have exploited the combination of probabilistic models and logic-based ones, which is now coined as statistical relational learning. As a representative of this effort, Markov logic networks (MLN) [109] model the knowledge using first-order logic (FOL) and construct a Markov network from the FOL rules and data set for inference. The first-order logic is used to describe the knowledge and each rule has a weight to represent the confidence. It has been applied for video action analysis with success in some tasks in [96, 131, 91] among others.

Existing work in these categories mostly use graphs as inference engine. Recent work also uses graphs for action representation, and the recognition is accomplished with graph matching such as permutation and random walk [18, 140]. For example, [18] builds graphs to capture hierarchical, temporal and spatial relationship between action tubes. Cheng et al. [30] proposed a data-driven temporal dependency model for joint video segmentation and classification, which is an extension of the 1st-order Markov models. They break a visual sequence into segments of varied lengths and label them with events of interest or a null or background event. The temporal struc-
ture is modeled by Sequence Memoizer (SM) which is an unbounded-depth, hierarchical, Bayesian nonparametric model of discrete sequences. To effectively represent a sequence, SM uses a prefix trie that can be constructed from an input string with linear time and space complexity.

2.3.4. Sparse Coding for Visual Computing

Sparse coding and dictionary learning have attracted interests during the last decade, as reviewed in [144]. Originated from computational neuroscience, sparse coding is a class of algorithms for finding a small set of basis functions that capture higher-level features in the data, given only unlabeled data [75]. Since its introduction and promotion by Olshausen and Field [99], sparse coding has been applied into many fields such as image/video/audio classification, image annotation, object/speech recognition and many others.

Zhu et al. encode local 3D spatial-temporal gradient features with sparse codes for human action recognition [161]. [156] uses sparse coding for unusual events analysis in video by learning the dictionary and the codes without supervision. It is worth noting that all of these approaches use vectorized features as input without consideration on the structure information among the features. [157] combines the geometrical structure of the data into sparse coding framework, and achieves better performance in image classification and clustering. Further, [120] proposes tensor sparse coding for positive definite matrices as input features. This motivates this work by combining graph representation of actions [28] with sparse coding.

Differing from most existing research, the elementary objects of dictionary learning and sparse coding operations are graphs in our approach. More specifically, it is the graphs that describe the temporal relationships that comprise our mid-level features. Graphs have been used in the activity analysis in literature. Gaur et al. [41] proposed a “string of feature graphs” model to recognize complex activities in videos. The string of feature graphs (SFGs) describe the temporally ordered local feature points such as spatio-temporal interest points (STIPs) within a time window. Ta et al. [127] provide a similar idea but using hyper-graphs to represent the spatio-temporal relationship of more than two STIPs. The recognition in both works is fulfilled by graph matching. Using individual STIPs to construct the nodes can result in unstable graphs and performance. A study similar to ours is that of Brendel and Todorovic [18], who built a spatio-temporal graph based
2.4. Machine Learning for Video Activity Analysis

2.4.1. Learning Based on BoW Model

Researchers have developed learning methods for BoW model, which can be categorized into generative and discriminative models. A generative model specifies the joint distribution over observation-label pairs. Bayes models are simple yet popular as a generative model in natural language processing and computer vision. Naïve Bayes [32] and other variants (such as [121]) have had success in object recognition and action recognition. Niebles et al. [97] represented a video sequence as a collection of extracted space-time interest points and exploited probabilistic latent semantic analysis (pLSA) and latent Dirichlet allocation (LDA) for human action categorization.

One of the most popular discriminative classification methods with BoW model is support vector machine (SVM) with a $\chi^2$-kernel. Schuldt et al. [114] first proposed to use SVM with local features for human action recognition. Later, SVM became a standard method for this task in works such as [73] and [84]. In an evaluation of spatio-temporal features for action recognition, Wang et al. [137] also used SVM for the learning and classification. May other discriminative methods however, such as used k-Nearest Neighbors [36], are also used with BoW model for action recognition.

For part of work in Chapter 4, we combined temporal relationships between spatio-temporal features into BoW model, and exploited the use of a discriminative approach to model the probabilistic of features associated with each category of action.

2.4.2. Learning Based on Graph Models

Graphs are treated as an ideal representation to capture the structure of an action while ignoring some details caused by rotation, scales or illumination changes. Graph matching have been used in computer vision for a variety of problems such as object categorization [38] and feature tracking [55]. It is typically formulated as the quadratic assignment problem (QAP). Recently graph matching has attracted more and more attention for motion and action analysis, especially
since 2000s. The nodes represent local features and the edges model different spatial/temporal connections (i.e., relationships), i.e., attributed graphs are usually used.

To overcome the limits of local space-time interest points, Ta et al. [127] constructed proximity hyper graphs based on the extracted local interest points, and formulate the activity recognition as (sub)graph matching between a model graph and a (potentially larger) scene graph. It models both the temporal and spatial aspects. The experiments show it achieves a comparable performance with some previous methods and outperforms others. Çeliktutan et al. [20] also used hyper graphs to represent the spatio-temporal relationships but proposed an exact, instead of approximate, graph matching algorithm for real-time human action recognition.

In [18], Brendel and Todorovic built spatio-temporal graphs to represent different structures of activities in videos, and used graph matching to find the optimal label of the action. The nodes are blocks of homogeneous pixels in a video and the edges describe temporal, spatial and hierarchical relationships.

Cho and Lee [31] described a graph progression algorithm to speed up the construction of graphs while matching graphs. Though it is for image matching, the method could be extended to video or action matching. Zhou and Torre [159] proposed a new factorization method for graph matching called Deformable Graph Matching and apply it to various object and shape comparison data sets. Both works can be potentially applied to action recognition problems.

Many efforts on graph matching’s application in computer vision are made and published, such as [149, 19, 41, 60]. Image and video analysis methodologies are experiencing “from bags of words to graphs” for better characterization of the spatial/temporal structure of activities. In this study, we propose to match two graphs by decomposing each into a linear combination of primitive graph representations and classifying the coefficients.

http://liris.cnrs.fr/christian.wolf/
3.1. Introduction

Efficient region of interest identification has been an active topic in computer vision fields such as visual attention in images and anomaly detection in video sequences. Currently the cameras record sufficient visual information for monitoring purposes. In fact, in most instances it is impractical for either human observers or automated systems to analyze each pixel in detail. Therefore selective operations are needed at different sites in a visual scene. Through region-of-interest (ROI) detection, non-interesting (e.g. normal) events can be excluded and further explicit event recognition methods can be applied to the remainder. This mimics the primates’ visual system. However, it is not a trivial problem for man-made systems to understand the scenes and perform such selective processing.

The localization of ROIs becomes more urgent when given wide-angle camera views with background motion clutter. Most surveillance videos are produced in this manner to obtain a large and efficient coverage of a monitored area. The challenge we address is that the greatest portion, both spatially and temporally, of the video is not of interest. Traditional approaches such as tracking-based methods are not efficient, especially in clustered scenarios. Tracking-based approaches perform well in narrow-angle views or sparse scenarios with limited objects. In wide-angle views much more information must be processed which lowers the efficiency. More importantly, because it is not goal-driven, much computation is needed to establish which are the “normal” trajectories or other representations. Therefore, some researchers have begun applying methods that first localize the regions of interest, followed by operations such as tracking and anomaly recognition. This work is also motivated by this framework. The focus is on identifying potential ROIs in videos for further analysis.

Region of interest detection is basically a classification problem for which visual informa-

Parts of this chapter have been previously published, either in part or in full, from Guangchun Cheng and Bill P Buckles, A nonparametric approach to region-of-interest detection in wide-angle views, Pattern Recognition Letters 49 (2014), 24-32. Reproduced with permission from Elsevier.
tion is assigned labels of “interesting” and “non-interesting”. For local feature representation, the description can be descriptive (such as common trajectory) or probabilistic (such as histogram). Correspondingly, the identification of local interest is based on the distance or probability of an observation compared with the canonical description. The relationships among information of different sites are also exploited in some probabilistic graphical models (e.g. conditional random fields).

In order to model the information and detect the regions of interest, existing studies mainly use local information to model the activities [62, 113, 66]. Usually it is assumed that the statistical information follows a basic distribution (e.g. Gaussian) or a mixture of them. The training phase is designed to compute the parameters according to optimization criteria. It is not always straightforward to estimate the parameters and it is difficult to determine the form of the distribution or the number of the mixed models that should be applied to arbitrary videos. The innovations of the described method are given below.

- 3D structure tensors are used as the basis to extract tracking-free features to characterize the motion and spatial dimensions concurrently; bypassing object tracking avoids the computational expense and the errors it may induce.
- A nonparametric approach models the distribution of tensor instances, treating observations with large deviation from the norm as statistical outliers to localize the regions of interest. This approach avoids the estimation of parameters as is required in parametric models.
- Characteristics of abnormal (or normal) spatial and motion patterns need not be explicitly specified; unsupervised training is applied to detect the norms and then the regions of interest.

Our first assumption is that the underlying processes that produce the motion change distribution are stationary and ergodic. That is, the mean of an observed sequence is an approximation of the mean of the population. While it is not difficult to exhibit non-stationary examples, we observe the motion changes at a specific site are most likely stationary and ergodic for extended periods. Switching to a new context, e.g., daytime activity vs nighttime activity, is a simple matter
of reconstructing the motion pattern. For some types of videos such as movies, the interests are often defined by the story and intent, which fall outside the scope of this dissertation. From a bottom-up perspective or view, which seeks interests from features instead of goals, we also assume that interesting events are rare although the converse may not be valid. Our approach is to mark interesting events and allow for further video analytics to classify.

The rest of the chapter is structured as follows. In Section 2.1, a brief review on anomaly detection in videos is given with the focus on non-tracking methods. Section 3.2 then explains in detail our approach to detect the regions of interest. Section 3.3 verifies our approach through experiments. The conclusion is in Section 3.5 with a discussion of limitations and future work.

3.2. Pixel-level Wide-View ROI Detection

Video is commonly considered a sequence of frames $I^{(t)}, t = 1, 2, ..., T$. Examiners usually can find objects or regions of interest from the sequence without any knowledge beyond the video. It is our hypothesis that outliers to the statistics within frames mark the ROIs. There are assumptions. First, normal activities outnumber anomalies. That is, statistical outliers correspond to regions of interest. Second, normal activities are sufficiently repetitive to form majority patterns which are the basis for statistical methods. Third, normal patterns have a finite lifespan. Changes in normal patterns, i.e. “context switches”, are not covered by this work. These assumptions are common in cases such as traffic, crowds, and security zone surveillance.

The framework is illustrated in Fig. 3.1. We first compute a 3D structure tensor to capture the motion at the sampled site $\vec{x} = (x, y)$ for each frame $I^{(t)}$. Next the probability distribution of structure tensor is estimated in the online training phase using the structure tensor’s eigenvalues. This is followed by the interest point detection as occurrences with low probability. ROIs are obtained using filtered interest points.

3.2.1. Feature: A 3D Structure Tensor

A structure tensor is a matrix derived from the gradient of an image to measure the uncertainty of a multidimensional signal. It is more robust than measures such as intensity because of the local simplicity hypothesis, i.e. in the spatial realm, x and y, variation of the gradient is less
FIGURE 3.1. Flow chart of the proposed method. Training and testing share feature extraction and computation of feature distance. The probability density function (PDF) is learned for sampled sites in the video.

than variation of the image itself [45]. It has been widely used in 2D image processing [82, 132], and has been extended to 3D cases for motion analysis [135, 136, 145, 37, 25].

A 3D structure tensor at \((x, y, t)\) is defined as follows ((\(x, y, t\) is omitted hereafter for simplicity):

\[
S_t = w_w \times \nabla I \nabla I^T = w_w \times \begin{bmatrix}
I_x^{(t)} I_x^{(t)} & I_x^{(t)} I_y^{(t)} & I_x^{(t)} I_t^{(t)}
I_y^{(t)} I_x^{(t)} & I_y^{(t)} I_y^{(t)} & I_y^{(t)} I_t^{(t)}
I_t^{(t)} I_x^{(t)} & I_t^{(t)} I_y^{(t)} & I_t^{(t)} I_t^{(t)}
\end{bmatrix}
\]  

where \(w_w\) is a weighting function, and \(\nabla I \nabla I^T\) is the outer product of gradient \(\nabla I(\vec{x}, t) = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial t})^T \triangleq (I_x, I_y, I_t)^T\). In order to reduce the influence of noise, the video is filtered prior to computing the gradient \(\nabla I\). In the remaining discussion, a 3D Gaussian filter \(G\) is used with variance \(\sigma_f^2\) and window size \(w_f\) (subscript \(f\) indicates filtering). Equivalently, the gradient is obtained by convolving the video with a 3D Gaussian derivative kernel:

\[
\nabla I = \left( \frac{\partial (I \ast G)}{\partial x}, \frac{\partial (I \ast G)}{\partial y}, \frac{\partial (I \ast G)}{\partial t} \right)^T
\]

\[
= I \ast \left( \frac{\partial G}{\partial x}, \frac{\partial G}{\partial y}, \frac{\partial G}{\partial t} \right)^T
\]
It was shown in [25] that the structure tensor $S_t$ is an unbiased estimator of the covariance matrix of $\nabla I^T$ if the weight function is an averaging filter. Therefore, methods for covariance matrix analysis can potentially be used with the structure tensor. Also different weight functions can be used. A 3D average and 3D Gaussian weighting functions were examined. The latter one gave slightly better results. Therefore, a 3D Gaussian with variance $\sigma_w^2$ and size $w_w$ ($w_w > w_f$) is used in this study. The choice for the window sizes are demonstrated in experiment section.

The structure tensor maps 1D intensity at $(x, y)$ into $3 \times 3$ dimensional space, which contains more information regarding intensity changes. Structure tensor based motion analysis mainly depends on the eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of $S$ ($\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$). Note that $S$ is a positive semidefinite matrix. (1) If $\text{trace}(S) = I_{xx} + I_{yy} + I_{tt} \leq Th$ (a threshold), all three eigenvalues should be small, i.e., there is no intensity variation and no motion in any direction. (2) If $\lambda_1 > 0, \lambda_2 = \lambda_3 = 0$, the change is in one direction. (3) If $\lambda_1 > 0, \lambda_2 > 0, \lambda_3 = 0$, there is no change in one direction. (4) However, if $\lambda_1 > 0, \lambda_2 > 0, \lambda_3 > 0$, the changes due to motion cannot be estimated. The eigenvalues have strong correspondence with the occurrence of activities. $\lambda_3$ contains the most information about the temporal changes, i.e. motion, and $\lambda_1$ and $\lambda_2$ describe the spatial changes. For the purpose of region of interest localization, both temporal and spatial factors should be considered. Therefore, eigenvalues of a structure tensor are used in this study as features from which the tensor’s likelihood is estimated. By constructing the distribution of distances between structure tensors, we avoid the necessity of selecting values for $Th$.

3.2.2. Representation: Probability Distribution of Structure Tensors

In a wide-angle view, it is preferable to allocate attention (and computing resources) to potential ROIs first rather than track/analyze every object in the field. In this work, probability distributions of structure tensors at particular sites are obtained using kernel density estimation. Since the eigenvalues of the structure tensor have a wide range and sparse distribution, a distance metric of eigenvalues is used to model its probability distribution. Both training and detection is based on the probabilities collected within a sliding window $W_T$.

After obtaining $S_{t:t+W_T}$, $t = 1, 2, ..., M - W_T$ where $M$ is the number of training frames and $W_T$ is the window size for batch processing, eigendecomposition is performed to get its eigenvalue
representation $\Lambda_{t} = (\lambda_{1}, \lambda_{2}, \lambda_{3})_{t:t+W_{T}}$ within each window $\{ t : t + W_{T} \}$. The distance of each $\Lambda_{t_{i}} = (\lambda_{1}, \lambda_{2}, \lambda_{3})_{t_{i}}$ to the mean $\mu_{t}$ of $\Lambda_{t:t+W_{T}}$ is computed using Mahalanobis distance as

$$d(\Lambda_{t_{i}}) = \sqrt{(\Lambda_{t_{i}} - \mu_{t})^{T} \Sigma_{t}^{-1} (\Lambda_{t_{i}} - \mu_{t})}, t_{i} \in \{1 : M\}$$

where $\Sigma_{t}$ is the covariance matrix of $\Lambda_{t_{i}}$'s within the sliding window ($t_{i} = t, t+1, ..., t+W_{T}$). The statistics are updated by a linear strategy as

$$\mu_{t+1} = \alpha \mu_{t} + (1 - \alpha) \times \text{mean}(\Lambda_{t})$$

$$\Sigma_{t+1} = \alpha \Sigma_{t} + (1 - \alpha) \times \text{cov}(\Lambda_{t})$$

The choice of Mahalanobis distance is due to its statistical meaning when measuring distance between $\Lambda_{t_{i}}$ for instances having large scale differences among the three eigenvalues. Experimentally, the Mahalanobis distance metric also gave better detection performance.

To construct the distribution of $d(\Lambda_{t_{i}})$, $t_{i} \in \{1 : M\}$, a kernel density estimation (KDE) method is used [14]. Although a histogram is a simple estimate of the distribution, there are several drawbacks. First, a histogram is only based on the frequencies which may not be continuous. Second, it is not easy to update the histogram online especially when given previously-unseen data values. Third, there are probably missing values in the training for which we need to determine their probabilities. This happens frequently when there are several modes of motion in the video. KDE overcomes these problems with the use of a continuous kernel function. In this work, a standard normal kernel $\Phi(x)$ is used for its mathematical properties and practicality. Based on $d(\Lambda_{t_{i}})$, $t_{i} \in \{1 : M\}$, the kernel density estimator is

$$\hat{f}(d) = \frac{1}{M} \sum_{t=1}^{M} \Phi(d - d(\Lambda_{t})) = \frac{1}{hM} \sum_{t=1}^{M} \Phi_{h} \left( \frac{d - d(\Lambda_{t})}{h} \right)$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} \frac{1}{hW_{T}} \sum_{t_{i}=nW_{T}}^{(n+1)W_{T}-1} \Phi_{h} \left( \frac{d - d(\Lambda_{t_{i}})}{h} \right)$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} \hat{f}_{n}(d)$$
where $\Phi_h(x)$ is the scaled kernel with smoothing parameter $h$ called \textit{bandwidth}, $N$ is the number of sampling windows, and $\hat{f}_n(d)$ is the kernel density estimate using the sample from the $n$th window. The variance of $\Phi_h(x)$ should be sufficiently small to avoid over-smoothing the distribution.

Using KDE, the distribution estimation is possible in a progressive manner, and this is shown as the last equation in (6). The estimate for the whole training data can obtained by averaging the estimate of each window. It is an advantage to use histograms as both previously seen and unseen data are now treated identically.

Depending on the activity in a video, such as its frequency and distinctiveness, the training may terminate at different values of $N$. At some point, a stable estimation of the distribution is obtained. For the termination criteria, there are many choices, such as setting a maximum $N$ or $M$ and using the distance measure between two consecutive estimates [68]. The latter gives an estimation independent of specific scenarios, while the former applies to scenarios where the $N$ or $M$ is available to obtain a stable distribution. In the experiments below, we used a predefined $N$, but other termination methods can be easily employed.

For each sampled site $\vec{\mathbf{x}}$, one distribution $\hat{f}(d|\vec{\mathbf{x}})$ is learned. To improve the efficiency by avoiding computing formula (6) each time a query $d$ is given, the estimate of the distribution is first quantized to $\hat{F}_B(d|\vec{\mathbf{x}})$ and saved together with the distances $d(\Lambda)$ when at the end of training. The distances $d(\Lambda)$ at different sites probably have different ranges, and thus the probability estimates are represented as a five-tuple array as follows:

\begin{equation}
\hat{F}_{\vec{\mathbf{x}}} = (\mu, \Sigma, d_{\text{min}}, d_{\text{max}}, \hat{F}_B(d))
\end{equation}

where

\begin{equation}
d_{\text{min}} = \arg\min_d \{\hat{f}(d|\vec{\mathbf{x}}) \geq T_d\}
\end{equation}

\begin{equation}
d_{\text{max}} = \arg\max_d \{\hat{f}(d|\vec{\mathbf{x}}) \geq T_d\}
\end{equation}

$T_d$ is a quantization threshold for $\hat{F}_B(d|\vec{\mathbf{x}})$ to cover at least 95\% percent of the probabilities of the distribution $\hat{f}(d|\vec{\mathbf{x}})$. In this work, $T_d$ is obtained via binary search by trying values from 1.0
to 0.0. For each value $T_d$, the cumulative probability in $[d_{\min}, d_{\max}]$ is computed. If it is less than 0.95, we set $T_a = T_d/2$ and the procedure is repeated.

Finally, the distribution of activities in the video is characterized by a two dimensional structure $[\hat{F}_{\vec{x}}]$. $[\hat{F}_{\vec{x}}]$ captures the background motion/activity in a compact manner. Those sites experiencing similar motion have similar description $\hat{F}_{\vec{x}}$, while it is different for regions with different motion patterns. This is shown in Fig. 3.2.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3_2.png}
\caption{Example of activity representation by the distribution of the structure tensors at sampled sites.}
\end{figure}

3.2.3. Interest Detection as Low Probability Occurrence

During the operational phase, a structure tensor $S_i$ is extracted for each sampled site $\vec{x}$. Its distance to the center of training data $d_i$ is computed using (3), where the statistical mean and covariance matrix are retrieved from $\hat{F}_{\vec{x}}$.

Abnormal occurrence detection can be treated as the problem of deciding if the occurrence follows a normal distribution, which is approximated using $\hat{F}_B(d_{\vec{x}})$. There are different strategies. Here we use the average log-probability of consecutive frames, $W_T$ frames. That is

\begin{equation}
\ell_{\vec{x}}(t) = \frac{1}{W_T} \sum_{t' \in \mathcal{W}_T} \log \left( \hat{F}_B(d_{t'}|\vec{x}) \right)
\end{equation}

where the temporal window $\mathcal{W}_T = \{t - W_T + 1 : t\}$. Note that a sliding window may be used during the testing phase, which speeds up the process by avoiding the computation of structure tensors, eigen-decomposition and probability for temporally overlapped sites. This average log-probability is computed for each sampled site to obtain the occurrence probabilities of different
sites at time $t$, denoted as $[\ell_{\mathcal{F}}(t)]$. Correspondingly, an averaged log-probability $L_{\mathcal{F}}$ is computed for $\hat{F}_B(d)_{\mathcal{F}}$ in the same way as computing $\ell_{\mathcal{F}}(t)$.

An anomaly map $A(t)$ is generated by thresholding $[\ell_{\mathcal{F}}(t)]$:

$$
\alpha_{\mathcal{F}}(t) = \begin{cases} 
1 & \text{if } \ell_{\mathcal{F}}(t) - L_{\mathcal{F}} \leq \theta \\
0 & \text{otherwise}
\end{cases}
$$

Next we obtain a low dimensional description of the motion in the video stream, i.e. one binary value for each site. This anomaly map filters out the normal activities (moving or static events) in the stream, and gives the regions of interest as blobs. Fig. 3.3 shows the power of this model to detect the motion with different patterns. Dependent on the application, operations (such as morphological opening) may be needed to remove small noisy blobs. This operation should be applied with caution because the objects are probably small in a wide angle view. Moreover, the wide-angle analysis is usually followed by other processing involving a logical “zoom-in.” That is, ROIs may be subject to further analyses that include tracking, recognition, and so on. This is beyond the scope of this dissertation and is one direction of future work.

![Figure 3.3](image.png)

**Figure 3.3.** Anomaly detection results in a video with multiple types of motion. (a) A scene with a water fountain. No object moves behind during training. (b) Activity distribution is learned using 3D structure tensors. Several distributions are shown on their locations respectively. (c) Anomaly map shows ROI during the test phase.

### 3.3. Experimental Results

In this section we describe the experiments performed to evaluate our method. Both qualitative and quantitative results on several datasets are included. Unless otherwise specified, the
parameters were set as: $w_f = 5$ and $\sigma_f = 2.5$ for computing derivatives, $w_w = w_f + 2$ for size of the weighting function, $\alpha = 0.9$ for updating the mean and covariance matrix, $\theta = -3$, and the temporal window size $W_T = 23$. The validation of these parameters is presented in the following section of Parameter Sensitivity Experiments. During testing, a sliding window was used for the computation of structure tensors and their eigenvalues.

3.3.1. Datasets

Recently many video datasets for computer vision research have been created. Most have been collected for action recognition purposes [2]. While our work may assist action recognition, the purpose is quite different. We need relatively long videos to train and test our method because of its unsupervised property. From a data requirement standpoint, this is similar to the case for tracking or surveillance. The videos we tested were collected from several publicly available datasets.

The BOSTON dataset was used for anomaly detection from a statistical perspective [113]. The dataset includes videos from different scenes such as traffic and nature. In these videos, the background behaviors (i.e., normal motion) may occur simultaneously with anomalous activities. This category of video is the target application of the proposed approach. These videos are not distributed as formal datasets for experiments, and the ground truth of anomalies for them are not available. We obtained the videos from a website $^3$, and used them for qualitative validation.

The CAVIAR dataset $^4$ consists of videos and footage for a set of behaviors performed by actors in the entrance lobby of INRIA Labs. They contain around 60 complete tracks. Since the training and the testing can be performed using different segments of the same video, we selected a subset of videos from the dataset. We developed a separate program for semi-automatic labeling of anomalous regions in each frame of the videos $^5$. Qualitative and quantitative analyses are given in the next subsection.

The UCSD dataset has been used with increasing frequency for anomalous activity detection. It consists of two sets of videos in uncontrolled environments. The anomaly patterns involve non-pedestrian or anomalous pedestrian motion. In the work by [83], it was employed to evaluate an approach based on mixtures of dynamic textures. Our approach was not designed directly for
pedestrian anomaly detection, therefore, this dataset is not ideal for evaluation, yet we do provide experimental results and analysis for it.

3.3.2. Parameter Sensitivity Experiments

Though the proposed approach itself is nonparametric, several parameters need to be tuned to obtain input and output for the purpose of experiments. For the parameters above, experiments were conducted to determine the optimal or suboptimal values for them. We recognize the Gaussian derivative filter window size \( w_f \) and the temporal sliding window size \( W_T \) as variables for analysis. The variance of the Gaussian filter, \( \sigma_f^2 \), is determined from the \( w_f \) to cover more than 95% of the energy, i.e. \( 2\sigma_f \geq w_f / 2 \). In experiments, we set \( \sigma_f = w_f / 2 \).

We defined measurements to evaluate the results. After the anomaly map \( A(t) \) was obtained for each frame in the test videos, it was compared with the ground truth data. We use the accuracy of anomaly detection alarms. If more than 40% of anomalous pixels are contained in the anomaly map, then it is registered as a hit, which is a measurement of temporal localization. By varying the threshold \( \theta \), it was possible to obtain the receiver operating characteristic (ROC) curves for hits. For spatial localization evaluation under different thresholds \( \theta \), precision and recall are used, which are defined below in (12) and (13), where \( G(t) \) is the ground truth of \( t \)th frame, and \( T \) is the total number of frames in the test video.

\[
\text{precision} = \frac{1}{T} \sum_{t=1}^{T} \frac{\sum A(t) \cap G(t)}{\sum A(t)}
\]

(12)

\[
\text{recall} = \frac{1}{T} \sum_{t=1}^{T} \frac{\sum A(t) \cap G(t)}{\sum G(t)}
\]

(13)

Figure 3.1 shows the performance of the system under different temporal sliding window sizes \( W_T \). We obtained the performance for \( W_T = 11, W_T = 23 \) and \( W_T = 31 \). These sliding windows move forward one frame each time, and results were collected for \( \theta \in [-18.4, 0.4] \) with each

3http://vip.bu.edu/projects/vsns/behavior-subtraction/
4http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/
5http://students.csci.unt.edu/ gc0115/groundtruthgenerator.html
window size configuration respectively. Figure 3.1(a) shows the anomaly hit rate, Figure 3.1(b) and Figure 3.1(c) present the spatial precision and recall for anomaly detection. As shown in the figures, the approach is not very sensitive to the sliding window size $W_T$ given a reasonable threshold $\theta$. For the remaining experiments, we set $W_T = 23$.

![Graphs showing detection performance](image)

(a) Hit rate

(b) Precision

(c) Recall

**Figure 3.1.** Detection performance under different temporal sliding window $W_T$.

Figure 3.2 presents the precision and ROC curves of hits for different Gaussian derivative filter sizes $w_f$. The protocol is similar to the one used for $W_T$, but we keep $W_T = 23$ and change $w_f$. From Figure 3.2(a), the performance does not differentiate among $w_f$ of 3, 5 and 7. As $w_f$ increases, the performance decreases. The areas under the ROC curves, shown in Table 3.1, come
with the same observation. This is consistent with the usual filtering size used in image processing systems. $w_f = 5$ is chosen for the experiments below.

**Figure 3.2.** Detection performance under different Gaussian derivative filter windows $w_f$.

**Table 3.1.** Areas under curve (AUC) for different Gaussian derivative filter sizes

<table>
<thead>
<tr>
<th>Window size</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>3x3</td>
<td>0.9307</td>
</tr>
<tr>
<td>5x5</td>
<td>0.9510</td>
</tr>
<tr>
<td>7x7</td>
<td>0.9394</td>
</tr>
<tr>
<td>9x9</td>
<td>0.8736</td>
</tr>
<tr>
<td>11x11</td>
<td>0.8216</td>
</tr>
</tbody>
</table>

3.3.3. Experiment Design and Results

We tested the proposed method using sites sampled from an imposed rectangular grid. A site $\vec{x}$ is a small region centered at one pixel, such as each grid crossing in Fig. 3.2. The size of the region is related to the filter size computing the gradients and the length of the video itself. In the video shown in Fig. 3.3, one site at the ATM machine was chosen for observation. The corresponding distribution of the structure tensors is shown at top left. In the testing phase, one person abandoned a bag, which was detected as a low-probability occurrence (the bottom). The middle graph is the average log-probability $l_{\vec{x}}(t)$ (10) over time.

Spatial anomaly: From the perspective of motion at a single site, we recognize two sorts of regions of interest. One is detected via the change of spatial information, namely the gradient.
This sort is called spatial anomaly. This category includes events such as object left behind, object moved away and scene change, among many others. In Fig. 3.4(a)-(b), the bag left behind was identified as of interest while the person walking was not. It may be useful to correlate the bag with the person who dropped it. That, however, is the goal of post-ROI detection processing. (One may consult the activity recognition literature [17, 93] for approaches for doing so.) In Fig. 3.4(c)-(d), a person lounging on the bridge during the testing phase was detected. Note that the car was not regarded as an anomaly because there was normal background traffic flow during training. The method’s capacity for recognizing spatial anomalies lies in the fact that the eigenvalues also contain information of spatial changes.

**Figure 3.3.** Abandoned bag detected as low-probability occurrence. The distribution is \( \hat{f}(d) \) where the site is marked with a red x on the top-left figure. Occurrence probability and the detection result are shown in the bottom two figures, respectively.

Temporal anomaly: The other sort of region of interest arises from temporal behavior. As we saw in Fig. 3.4(a)-(d), although the spatial information at some sites changes over time (e.g.
places through which persons or cars pass), that still may be normal thus not of interest. To filter out random motion, temporal information must be considered. The distribution $f(d | \vec{x})$ captures the intrinsic patterns of temporal changes. The method was tested using videos with background motion clutter. In Fig. 3.4(e)-(f), the motion caused by motion of water was high-probability thus is not of interest. In this experiment, the probabilities were not thresholded but converted to the range $[0, 255]$ to show the level of anomaly. Fig. 3.4(g)-(h) depict a second example. Notice the discontinuity in the detection result. This is due to the sluggish motion and the uniform intensities within the sail boundary. It is a false positive.

**Figure 3.4.** ROIs detected for both spatial and temporal pattern changes. Regions within red circles are regarded as ROIs. (a)-(b): abandoned bag is of interest but the usual pedestrian bypassing is not; (c)-(d) suspicious pedestrian lounging is but usual normal traffic is not; (e)-(f): car moving behind is but usual water fountain is regarded as background motion; (g)-(h): boat is of interest but waving water is regarded as background motion.

Quantitative results: Fig. 3.5 shows the ROC for videos from the BOSTON and CAVIAR datasets. With FPR=3.8%, the system achieved TPR=100% for the detection of the anomalous “car moving behind water fountain” for a video from the BOSTON dataset as shown in Fig. 3.5(a). Fig. 3.5(b) shows the ROC for a video from the CAVIAR dataset to detect the anomalous activity together with a comparison to the results from existing studies, including a semi-supervised approach [119], a context-based approach [142] and Gaussian mixture [46]. [119] proposes a
trajectory-based semi-supervised anomaly detection approach, where anomaly is detected by a
one-class classifier over the cubic B-spline coefficients of a trajectory. The approach requires ob-
ject/feature detection and tracking, and labelling of training trajectories as well. [142] extracts
contextual features, i.e. types of behaviors and their commonality levels, by clustering trajectories
(types) and measuring the fraction of each cluster’s instances (commonality). A behavior type
with low commonality is treated as an anomaly. While it does not require labelling, [142] is also
based on an expanded period of object tracking as [119] requires. Following the procedure in
[46], we obtain the results from Gaussian mixture models. Similar to the proposed approach, it
is applied to sampled individual sites, but it assumes Gaussian distribution while ours does not.
As demonstrated in Figure 3.5(b), our approach outperforms the latter two [142, 46], and achieves
comparable performance with [119] which used a semi-supervised method.

![ROC curves for anomaly hits](image)

(a) ROC for a video from BOSTON (car behind fountain)       (b) ROC for a video from CAVIAR (bag left behind)

**Figure 3.5.** Receiver operating characteristic curve of hits for the video from
BOSTON and CAVIAR datasets. (3.5(a)) shows the detection ROC of unusual
motion (car moving behind fountain) from the background motion (the water spray).
(3.5(b)) gives a comparison of the results from our method and a semi-supervised
trajectory-based method [119], a context-based approach [142] and Gaussian mix-
ture [46].

The precision and recall are reported in Fig. 3.6. Given a high alarm detect rate as shown in
Fig. 3.5, reasonable spatial localization is achieved. We calculate the F-score of ROI localization for videos from the BOSTON and CAVIAR datasets, and the average F-scores for them are 0.67 and 0.72, respectively.

\[
F = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}
\]

\(14\)

**Figure 3.6.** At each sampled threshold, the average localization precision and recall are shown for the entire video of “car passing behind water fountain”.

3.4. Relationship between Eigenvectors and Motion Direction

We do not cover the relationship between eigenvectors and motion direction in this dissertation, however, we notice that there is a relationship between them. Below we show an example that demonstrates how a structure tensor changes when the direction of the motion is reversed. Let \(\nabla I(x, y, t) = (I_x, I_y, I_t)^T\) be the gradient at location \((x, y, t)\) for a un-reversed video. For simplicity, define

\[
I_x = I(x + 1, y, t) - I(x, y, t)
\]

\[
I_y = I(x, y + 1, t) - I(x, y, t)
\]
\[ I_t = I(x, y, t + 1) - I(x, y, t) \]

When the motion direction is reversed temporally, we have \( I_x \rightarrow I_x, I_y \rightarrow I_y \) and \( I_t \rightarrow -I_t \). Thus, according to the definition of structure tensor 1, the new structure tensor at \((x, y, t)\) becomes

\[
ST = w \star \begin{bmatrix}
I_x^{(t)} I_x^{(t)} & I_x^{(t)} I_y^{(t)} & -I_x^{(t)} I_t^{(t)} \\
I_y^{(t)} I_x^{(t)} & I_y^{(t)} I_y^{(t)} & -I_y^{(t)} I_t^{(t)} \\
-I_t^{(t)} I_x^{(t)} & -I_t^{(t)} I_y^{(t)} & I_t^{(t)} I_t^{(t)}
\end{bmatrix} \triangleq A
\]

Suppose the weight function is \( w = [1]_{w \times w} \). Let \( \lambda \) and \( x \) be one eigenvalue and corresponding eigenvector. So

\[ Ax = \lambda x \]

or equivalently

\[
\begin{bmatrix}
I_x^2 - \lambda & I_x I_y & -I_x I_t \\
I_x I_y & I_y^2 - \lambda & -I_x I_t \\
-I_x I_t & -I_y I_t & I_t^2 - \lambda
\end{bmatrix} x = 0
\]

\[
\Leftrightarrow \begin{bmatrix}
I_x^2 - \lambda & I_x I_y & -I_x I_t \\
I_x I_y & I_y^2 - \lambda & -I_x I_t \\
I_x I_t & I_y I_t & -(I_t^2 - \lambda)
\end{bmatrix} x = 0
\]

In order to obtain a non-zero solution, set

\[
det \begin{bmatrix}
I_x^2 & I_x I_y & -I_x I_t \\
I_x I_y & I_y^2 & -I_x I_t \\
I_x I_t & I_y I_t & -(I_t^2)
\end{bmatrix} = 0
\]

\[
= -det \begin{bmatrix}
I_x^2 & I_x I_y & I_x I_t \\
I_x I_y & I_y^2 & I_x I_t \\
I_x I_t & I_y I_t & (I_t^2)
\end{bmatrix} = 0
\]

Obviously, the solution of \( \lambda \) is identical to that of the structure tensor in the un-reversed video. Therefore, when the motion direction reverses, the eigenvalues at the corresponding \((x, y, t)\)
site remain the same. In other words, eigenvalue-based structure tensor analysis is insensitive to motion direction changes. This can be desirable in some cases and problematic in others.

Go back to equation (16) and examine how the eigenvector changes for one specific eigenvalue $\lambda_i$. Suppose that the corresponding eigenvector is $(x_1, x_2, x_3)$. Then we have

$$
\begin{bmatrix}
I_x^2 - \lambda & I_x I_y & -I_x I_t \\
I_x I_y & I_y^2 - \lambda & -I_y I_t \\
I_x I_t & I_y I_t & -(I_t^2 - \lambda)
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix}
= 0
$$

$$
\Rightarrow
\begin{bmatrix}
I_x^2 - \lambda_i & I_x I_y & I_x I_t \\
I_x I_y & I_y^2 - \lambda_i & I_y I_t \\
I_x I_t & I_y I_t & I_t^2 - \lambda_i
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix}
= 0
$$

Note that the coefficient matrix is the one used to compute the eigenvectors for the structure tensor of un-reversed video. Therefore, although the eigenvalues remain the same, the direction of the eigenvectors changes in the eigen space. We will make efforts to explore this discovery in future work.

3.5. Summary

In this chapter, we describe an unsupervised approach to detect regions of interest in a video. 3D structure tensors are applied as a compact basis to model the distribution of locally specific motion. The motion at a location is determined by both spatial and temporal changes. Statistical outlier sites constitute markers for the regions of interest. The experimental results indicate that it is a promising method for detection of anomalies and the corresponding regions of interest.

Wide-angle scenes typically encompass dozens to hundreds of objects simultaneously in motion. It is not feasible to analyze such scenes using action recognition techniques such as those described in [17, 93]. The methods described here can be used to narrow the focus area to a subimage amenable to event recognition.
CHAPTER 4

TEMPORAL STRUCTURE FOR ACTION RECOGNITION

4.1. Introduction

The public is becoming accustomed to the ubiquitous presence of camera sensors in private, public, and corporate spaces. Surveillance serves the purposes of security, monitoring, safety, and even provides a natural user interface for human machine interaction. A publication in the popular press estimates that, in the U.S. alone, nearly four billion hours of video are produced each week [134]. Cameras may be employed to facilitate data collection, to serve as a data source for controlling actuators, or to monitor the status of a process which includes tracking. Thus, the need for video analytic systems is increasing. This chapter of dissertation concentrates on action recognition in such systems.

Each video-based action recognition system is constructed from first principles. Signal processing techniques are used to extract mid-level information that is processed to extract entities (e.g., objects) which are then analyzed using deep model semantics to infer the activities in the scene. The major task and challenge for these applications is to recognize action or motion patterns from noisy and redundant visual information. This is partly because actions in a video are the most meaningful and natural expression of its content. The key issues involving action recognition include background modeling, object/human detection and description, object tracking, action description and classification, and others. Depending on specific domains, very different methods can be employed to fulfill each of these aspects. Here our goal is to reduce the processing steps in the gap between video signals and activity inference. This is accomplished by establishing mid-level categorical features over which pattern recognition is possible.

Existing methods for vision-based action recognition can be classified into two main categories: feature-based bag-of-words and state-based model matching. The latter is distinguished by the use of spatio-temporal relationships. “Bag-of-words” has been successfully extended from

Parts of this chapter have been previously published, either in part or in full, from Guangchun Cheng, Yan Huang, Yiwen Wan, and Bill P Buckles, Exploring temporal structure of trajectory components for action recognition, *International Journal of Intelligent Systems* 30 (2015), no. 2, 99-119. Reproduced with permission from John Wiley and Sons.
text processing to many activity recognition tasks [33, 71]. Features in the bag-of-words are local descriptors which usually capture local orientations. However, the spatio-temporal relations between the descriptors are not used in most bag-of-words-based methods. State-based matching methods establish a model to describe the temporal ordering of motion segments, which can discriminate between activities, even for those with the same features but different temporal ordering. Methods in this category typically use hidden Markov models (HMMs) [153] or spatio-temporal templates [78] among others. Difficulties with model matching methods include the determination of the model’s structure and the parameters.

In this chapter, a mixture model of temporal structure between features is proposed to explore the temporal relationships among the features for action recognition in videos. This work moves us towards a generic model that can be extended to different applications. Dense trajectories obtained by an optical-flow based tracker [138] are employed as observations, and then these trajectories are divided into meaningful groups by a graph-cut based aggregation method. Following the same strategy as bag-of-words, a dictionary for these groups is learned from the training videos. In this study, we further explore the temporal relations between these “visual words” (i.e. trajectory groups). Thus, each video is characterized as a bag-of-words and the temporal relationships among the words. We evaluate our model on public available datasets, and the experiments show that the performance is improved by combining temporal relationships with bag-of-words.

The contributions of this work follow.

• In order to extend to different applications, our model uses groups of dense trajectories as its basis to represent actions in videos. Dense trajectories provide an effective treatment for cross-domain adaptivity. We extended the research on dense trajectories in [139] [107] by clustering them to form “visual words”. These “visual words” constitute a dictionary to describe different kinds of actions.

• The statistical temporal relationships among “visual words” is explored to improve the classification performance. The temporal relationships are intrinsic characteristics of actions and the connection between detected low-level action parts. The effectiveness of this study is shown in the experiment section.
• We evaluate the proposed approach on publicly available datasets, and compare it with bag-of-words-based and logic-based approaches. The proposed approach requires less preprocessing yet yields better performance in terms of accuracy.

There has been many studies of action recognition as reviewed in Chapter 2, especially those based on trajectories. As observed from the aforementioned research, action recognition has attracted study from those investigating both feature-based and description-based approaches. The former is usually used as the basis for the latter, and the latter is closer to a human’s understanding of an action. This study recognizes actions by extracting mid-level actionlets we call components which are represented by trajectory groups and exploring their temporal relations quantitatively using Allen’s interval relations. These components and their temporal relations are more expressive and can be integrated into other higher-level inference systems.

The remainder of this chapter is organized as follows. We describe the trajectory component extraction and their temporal structure in Section 4.2, and present how the learning is performed in Section 4.3. Section 4.4 gives experimental analysis by comparing with existing approaches. We conclude the work in Section 4.5.

4.2. Structure of Trajectory Groups

In order to develop an application-independent approach for action recognition, we extract features to express meaningful components based on dense trajectories. For raw trajectory descriptors, we employ the form that Wang et al. proposed [139] but we remove object motion caused by camera movement. There exists a mismatch between raw trajectories and the description of actions as commonly understood. Actions are categorical phenomena. In this chapter, we therefore cluster these dense trajectories into meaningful mid-level components, and construct a bag-of-components representation to describe them.

4.2.1. Dense Trajectories

Trajectories based on feature point descriptors such as SIFT are usually insufficient to describe the motion, especially when consistent tracking of the feature points is problematic because of occlusion and noise. This leads to incomplete description of motion. In addition, these sparse
trajectories are probably not evenly distributed on the entire moving object but cluttered around some portions of it. We extract dense trajectories from each video to describe the motion of different parts of a moving object. Different from sparse feature-point approaches, the dense trajectories are extracted using the sampled feature points on a grid basis. Figure 4.1 illustrates the difference between them. To detect scale-invariant feature points, we constructed an image pyramid for each frame of a video, and the feature points are detected at different scales of the frame, as shown in Figure 4.2.

![Figure 4.1](image)

**Figure 4.1.** Examples of trajectories from object-based tracking (first row) and dense optical flow-based feature tracking (second row). The dense trajectories are grouped based on their spatio-temporal proximity.

For each pyramid image $I$ of a frame, it is divided into $W \times W$ blocks. We use $W = 5$ as suggested in [139] to assure a dense coverage of the video. For the pixel $p$ at the center of each block, we obtain the covariance matrix of intensity derivatives (a.k.a. structure tensor) over its neighborhood $S(p)$ of size $N_S$ as

\[
M = \left[ \begin{array}{cc} \sum_{S(p)} \left( \frac{df}{dx} \right)^2 & \sum_{S(p)} \frac{df}{dx} \cdot \frac{df}{dy} \\ \sum_{S(p)} \frac{df}{dx} \cdot \frac{df}{dy} & \sum_{S(p)} \left( \frac{df}{dy} \right)^2 \end{array} \right]
\]

(18)
where the derivatives are computed using Sobel operators. If the smallest eigenvalue, $\lambda^*$, of $M$ is greater than a threshold $\theta_{\lambda^*}$, $p$ is the location for a new feature point, and is added to be tracked. We repeat detecting new feature points at each and every frame as long as there are no existing points within a $W \times W$ neighborhood, and track these feature points. The detection and tracking of feature points are performed at different scales separately.

![Figure 4.2. Dense trajectory extraction. Feature points are extracted at different scales on a grid basis. Red points are new detected feature points, and green points are the predicted location of feature points based on optical flow.](image)

Tracking feature points is fulfilled based on optical flow. We use the Gunnar Farneback’s implementation in the OpenCV library to compute the dense optical flow. It finds optical flow, $f(y, x)$, of each pixel $(y, x)$ between two frames $I_t$ and $I_{t+1}$ in both y and x directions, so that

$$I_t(y, x) \approx I_{t+1}(y + f(y, x) \cdot y + f(y, x) \cdot x)$$

Once dense optical flow is available, each detected feature point $P^i_t$ at frame $I_t$ is tracked to $P^i_{t+1}$ at frame $I_{t+1}$ according to the majority flow of all the pixels within $P^i_t$’s $W \times W$ neighborhood. $(P^i_t, P^i_{t+1})$ is then added as one segment to the trajectory.

For videos containing camera motion, for example, those in KTH dataset, the basic optical flow-based tracking adds a large number of trajectories of the background objects. This is due to the fact that optical flow is the absolute motion which inevitably incorporates camera motion. Here
we use a simple method to check whether \((P_i^t, P_i^{t+1})\) should be added to the trajectory. Instead of treating each \((P_i^t, P_i^{t+1})\) pair separately, for each frame \(I_t\), we compute the majority displacement, \(\overrightarrow{P_t P_{t+1}}^*\), of vectors \(\overrightarrow{P_i^t P_{i+1}^t}\) for all the feature points \(P_i^t, i = 1, 2, \ldots, N_t\) where \(N_t\) is total number of feature point at frame \(I_t\). Then each candidate trajectory segment \((P_j^t, P_j^{t+1})\) is compared with \(\overrightarrow{P_t P_{t+1}}^*\). Those segments are treated as background movement when their directions are within \(\theta_{bgota}\) degrees of \(\overrightarrow{P_t P_{t+1}}^*\)’s and their magnitudes are within \(\theta_{bgom}\) times of \(\overrightarrow{P_t P_{t+1}}^*\)’s magnitude.

We found \(\theta_{bgota} = 15\) and \(\theta_{bgom} = 0.3\) give a good trade-off between removing background motion and keeping foreground motion. It is worth noticing that more complicated and comprehensive methods can be used to remove the camera motion, e.g., feature point-based homography or dynamic background modeling.

To overcome common problems with tracking due to occlusion such as lost or wrongly tracked feature points, we limit the length of trajectories to \(L + 1\) frames \((L = 15\) in experiments). After a feature point is tracked for consecutive \(L + 1\) frames, it is removed from the tracking list and the resulting trajectory is saved. If a feature point is lost prior to \(L + 1\) frames of tracking, the resulting trajectory is discarded. Because stationary objects are usually not of interest in action analysis, we rule out trajectories whose variances in both \(x\) and \(y\) directions are small, even though they achieve \(L + 1\) frames of length. These short dense trajectories will be expanded both spatially and temporally by trajectory grouping described in Section 4.2.3.

4.2.2. Trajectory Descriptors

To describe the dense trajectories, we largely follow prior work on tracklets\([107, 138]\). For each trajectory, we combine three different types of information together with space-time data, i.e. location-independent trajectory shape \((S)\), appearance of the objects being tracked (histograms of oriented gradients, HoG), and motion (histogram of optical flow, HoF, and motion boundary histograms, MBH). Therefore, the feature vector for a single trajectory is in the form of

\[
T = (t_s, t_e, \bar{x}, \bar{y}, S, HoG, HoF, MBH_x, MBH_y)
\]

where \((t_s, t_e)\) is the start and end time, and \((\bar{x}, \bar{y})\) is the mean coordinate of the trajectory, respectively. We briefly introduce the descriptors we use here.
The shape $S$ of a trajectory describes the motion of the tracked feature point itself. Because the same type of trajectories can be at different locations in different videos (scenarios), we use the displacement of points, $S = (\delta P_i^t, \delta P_i^{t+1}, ..., \delta P_i^{t+L-1})$, as shape descriptor, where $\delta P_i^t = P_i^{t+1} - P_i^t$. In our experiments, $S$ is normalized with respect to its length to make it length-invariant, i.e. $S \leftarrow \frac{S}{\|S\|_L}$ where $\| \cdot \|_L$ is a length operator that gives the length of a trajectory. Histogram of oriented gradients (HoG) has been widely used to describe the appearance of objects such as pedestrians. HoG is an 8-bin magnitude-weighted histogram of gradient orientations within a neighborhood for which each bin occupies 45 degrees. Here we divide each trajectory into $n_t = 3$ segments, and calculate HoGs within each segment’s neighborhood. The $n_t$ HoGs are averaged to form the appearance descriptor. HoF and MBH encode the motion of objects and its gradients, respectively. The same segmentation is performed as HoG to obtain average HoF and average MBH for each trajectory. MBH describes the gradients of optical flow in $x$ and $y$ directions, thus it is represented by $MBH_x$ and $MBH_y$ histograms. MBH is 8-dimensional while HoF is 9-dimensional because the last element represents optical flows with a small magnitude. It is worth mentioning that all the three descriptors can be efficiently calculated for different trajectories using the idea of integral images provided in OpenCV [1]. A routine to extract trajectory descriptors, including background motion removal, is available at http://students.csci.unt.edu/~gc0115/trajectory/.

4.2.3. Grouping Dense Trajectories

The trajectories are clustered into groups based on their descriptors, and each trajectory group consists of spatio-temporally similar trajectories which characterize the motion of a particular object or its part. The raw dense trajectories encode local motion, and the trajectory groups are mid-level representation of actions, each of which corresponds to a longer term of motion of an object part. To cluster the dense trajectories, we develop a distance metric between trajectories with the consideration of trajectories’ spatial and temporal relationships. Given two trajectories $\tau^1$ and $\tau^2$, the distance between them is

\begin{equation}
    d(\tau^1, \tau^2) = \frac{1}{L} d_S(\tau^1, \tau^2) \cdot d_{spatial}(\tau^1, \tau^2) \cdot d_t(\tau^1, \tau^2)
\end{equation}
where $d_S$ is the Euclidean distance between the shape vectors $S^1$ and $S^2$, $d_{\text{spatial}}(\tau_1, \tau_2)$ is the spatial distance between corresponding trajectory points, and $d_t(\tau_1, \tau_2)$ indicates the temporal distance. We choose the following in our experiments. By these definitions, spatio-temporally close trajectories with similar shapes have small distance.

\begin{equation}
\begin{align*}
  d_S(\tau_1, \tau_2) &= \sqrt{\sum_{i=1}^{L} (S^1_i - S^2_i)^2} \\
  d_{\text{spatial}}(\tau_1, \tau_2) &= \sqrt{(x^1 - x^2)^2 + (y^1 - y^2)^2} \\
  d_t(\tau_1, \tau_2) &= \begin{cases} 
    1 & |t^1_s - t^2_s| < L \\
    \infty & |t^1_s - t^2_s| \geq L
  \end{cases}
\end{align*}
\end{equation}

Trajectory clustering is based on a graph clustering algorithm GANC [128]. The reason for using a graph clustering algorithm is its capacity to generate clusters over different temporal spans. Though they are robust to unreliable tracking, dense trajectories describe short motion of fixed length. GANC enable temporally separated trajectories to be grouped using the distance metric above. As input to GANC, we compute an $n \times n$ affinity matrix $A$ of the trajectories, with each element $a(\tau_i, \tau_j) = \exp^{-d(\tau_i, \tau_j)}$, where $n$ is the number of trajectories in a video. GANC produces clusters minimizing the normalized cut criterion in a greedy agglomerative hierarchical manner. The second row of Figure 4.1 shows examples of trajectory groups for several video samples. Trajectories in different colors are from different types of groups, which are referred to as components below.

4.2.4. Bag of Components

The trajectories and their groups shown in Fig. 4.1 provide low level description to the action content in a video. Some separated trajectory groups from the previous step can have the same motion characterization. We proposed to use components to represent different types of trajectory groups. A mean feature vector, $\overline{T}$, is obtained for all the trajectories in the same trajectory group. Because of the large motion variation in even the same type of actions, our model constructs a trajectory component codebook, and assigns each trajectory group to its closest component in the codebook. The size of the codebook, $D$, is determined based on the experiments, and is set to 1000. K-means clustering is used over the $\overline{T}$’s ($S, HoG, HoF, MBH_x, MBH_y$) to generate the
components. We use Euclidean distance for each of the descriptors, and combine them using

\[ d(\tau^1, \tau^2) = e^{\sum_k \frac{d_k(\tau^1, \tau^2)}{\pi_k}} \]

where \( \tau^1, \tau^2 \) are two trajectory groups, \( k \in \{S, HoG, HoF, MBH_x, MBH_y\} \), and \( \pi_k = \max\{d_k(\tau^1, \tau^2)\} \) is the maximum distance between the descriptors \( k \) of two groups in the training datasets. The codebook construction and component assignment is illustrated in Figure 4.3. Figure 4.3(a) shows the results from GANC clustering, and Figure 4.3(b) illustrates the assignment of each group \( C_i \) to the closest component \( W_i \). For instance, both groups \( C_1 \) and \( C_2 \) correspond to the same component \( W_1 \). In the following, \( f : g \rightarrow w \) is used to indicate the mapping from a trajectory group to a component.

![Trajectory components and assignment to words](image)

**Figure 4.3.** Illustration on component assignment of trajectory groups. Trajectory groups \( C_1 \) and \( C_2 \) are mapped to the same component \( W_1 \).

Given the codebook, the trajectory groups of a video are assigned their closest component, and the video can have a bag-of-components representation as follows, where \( d_i \) is the frequency of component \( w_i \) in the video.

(22) \[ BoC = \{d_1, d_2, ..., d_D\} \]

4.2.5. Temporal Structure

To characterize the temporal relationships among actions, our model develops the statistical temporal relationships between the “components”, and combines them with bag-of-components representation. According to the conclusions of Allen [6], there exist 13 temporal relations between two actions based on the actions’ durance intervals. See Figure 4.4 for a summary on the
temporal relations. before_i means before inversely (i.e. after), and the same for other relations on the right column. As also noticed by Patridis et al. [102], symmetric geometry exists in these relations. To reduce the redundancy, seven temporal structures are used in our model to represent these temporal relationships, i.e. before(B), meets(M), overlaps(O), starts(S), during(D), finishes(F) and equals(E). Each of them is a two-dimensional matrix, and characterizes one temporal relationship and its inverse. This is achieved by putting each pair of the relationships above and below the diagonal of the matrix respectively.

![Temporal Relationships Diagram]

**Figure 4.4.** Allen’s temporal relationships between two intervals.

For each type of action, the temporal relationships between a pair of components are modeled by the seven two-dimensional histograms. Each histogram shows the frequencies with which the relationship is true between a pair of components. That is, for a temporal relation $R_i \in \{B, M, O, S, D, F, E\}$, $R_i(x, y)$ is the frequency of $x \ R_i \ y$ between two components $x$ and $y$. In our model, we construct the temporal relations for each type of action in a supervised manner, i.e. we learn discriminatively $p(R_i | \alpha)$ for each action type $\alpha$. Figure 4.5 shows an example of meets for different actions in one evaluation dataset. It can be observed that different actions exhibit different histograms, and similar actions have similar histograms. Examining each of the histograms shows which temporal relationship (such as meets for boxing) has a stronger response for some pairs of components than the others. This implies the major relationship between compo-
Suppose we have two components \( x \) and \( y \) in one video \( v \) of action \( \alpha \) after dense trajectory clustering, and their time intervals are \((t^x_s, t^x_e)\) and \((t^y_s, t^y_e)\), respectively. The \( R_i \)'s are constructed according to the relative position and size of \((t^x_s, t^x_e)\) and \((t^y_s, t^y_e)\). Table 4.1 lists the update scheme of relationship model according to the relative position of \((t^x_s, t^x_e)\) and \((t^y_s, t^y_e)\).

![Figure 4.5](image.png)

**Figure 4.5.** Histograms of temporal relation *meets* for five different actions in KTH dataset. The X and Y axes are the types of group codes, and the Z values are the frequency before normalization. Among them, histograms of jogging and walking are relatively close to each other. So are boxing and handclapping.

This process is performed for all pairs of trajectory groups in all the videos of action type \( \alpha \). We obtain the signature for action \( \alpha \) by combining the bag-of-components and the temporal relations: \( A = \{ BoC^\alpha, \{ R_i^\alpha \}_{i=1}^7 \} \), and this is used as the feature of our model.

During recognition a similar process is followed to extract the feature for the target video. Suppose it is \( F: \{ boc, \{ R_i \}_{i=1}^7 \} \). We seek an action \( \alpha^* \) which maximizes the likelihood:

\[
\alpha^* = \arg \max_{\alpha} \mathcal{L}(F|\alpha) = \arg \max_{\alpha} \prod \mathcal{L}(w_j|\alpha) \prod \mathcal{L}(R_i|\alpha)
\]

(23)

based on the assumption that different groups and temporal relations are independent.

\( \mathcal{L}(w_j|\alpha) \) can be directly retrieved from the signature of action \( \alpha \), denoted as \( p(w_j|\alpha) \) (see next section), and here we discuss how to obtain the likelihood of \( \{ R_i \}_{i=1}^7 \). We make use of the distance between \( R_i^\alpha \) and \( R_i \) to define the likelihood. Both \( R_i^\alpha \) and \( R_i \) are matrices, and their distance is defined according to equation (24) \[101\] where \( \lambda_j \)'s are eigenvalues of matrix \( R_i^\alpha R_i^{-\frac{1}{2}} R_i^{-\frac{1}{2}} R_i \). In case when \( R_i^\alpha \) is singular, its pseudo-inverse is used as its inverse to calculate
\[ R_i - \frac{1}{2}. \]

(24) \[ d(R_i^\alpha, R_i) = \sqrt{\sum_{j=1}^{L=2} (\log \lambda_j)^2} \]

The likelihood of \( R_i \) being action \( \alpha \) is defined as

\[ L(R_i|\alpha) = \frac{e^{-d(R_i^\alpha, R_i)}}{\int_{\mathbb{R}_i} e^{-d(R_i^\alpha, R_i)} dR_i} \]

If we disregard the normalizing constant in the denominator, and substitute into equation (23) we get

(25) \[ \alpha^* = \arg \max_{\alpha} \exp \left\{ -\sum_{i=1}^{7} d(R_i^\alpha, R_i) \right\} \prod_{j=1}^{|C|} p(w_j|\alpha) \]

where \(|C|\) is the number of trajectory components in the video. This problem can be solved effectively when the signatures of known actions and the features of the target video are available. The solution is described in the next section.

4.3. Learning and Recognition

To construct the signatures of actions, a supervised discriminative learning approach is applied to obtain the probability of every component given the action \( p(w_i|\alpha) \) and the seven histograms for temporal relations. We are able to learn the \( p(w_i|\alpha) \) and the temporal histograms for each type of action.

For a specific dataset, we assume that the labels of the actions, \( \alpha \)'s, are known, and the codebook of components is first learned from the dataset. To obtain the codebook for the bag-of-components representation, we cluster the trajectory groups from all the videos in each training dataset as described in Section 4.2.4. This codebook is also used for the test videos for component assignment.

We apply simple methods to learn the conditional probability and the temporal histograms. Following a Bayesian training procedure, we count the occurrence \( (T_{w_i}) \) of each component in all the videos of the same action, and then compute the conditional probability \( p(w_i|\alpha) \) using each component’s frequency. The temporal histograms are computed for each video and are then
averaged over all videos of an action. For each trajectory component in a video of action \( \alpha \), we compute its temporal distances to all of the other components in that video, determine the Allen temporal relationships between them, and count the frequency of each relationship. The seven temporal histograms are updated correspondingly.

**Table 4.1.** Temporal matrix construction based on Allen predicates.

<table>
<thead>
<tr>
<th>If</th>
<th>Then</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t^e_s &lt; t^y_s )</td>
<td>( B(f(x),f(y)) \leftarrow B(f(x),f(y))+1 )</td>
</tr>
<tr>
<td>( t^y_s &lt; t^e_s )</td>
<td>( B(f(y),f(x)) \leftarrow B(f(y),f(x))+1 )</td>
</tr>
<tr>
<td>( t^e_s = t^y_s )</td>
<td>( M(f(x),f(y)) \leftarrow M(f(x),f(y))+1 )</td>
</tr>
<tr>
<td>( t^e_s = t^y_s )</td>
<td>( M(f(y),f(x)) \leftarrow M(f(y),f(x))+1 )</td>
</tr>
<tr>
<td>( t^e_s &lt; t^y_s, t^e_s &lt; t^y_s, t^e_s &gt; t^y_s )</td>
<td>( O(f(x),f(y)) \leftarrow O(f(x),f(y))+1 )</td>
</tr>
<tr>
<td>( t^e_s &lt; t^y_s, t^e_s &lt; t^y_s, t^e_s &gt; t^y_s )</td>
<td>( O(f(y),f(x)) \leftarrow O(f(y),f(x))+1 )</td>
</tr>
<tr>
<td>( t^e_s &gt; t^y_s, t^e_s &lt; t^y_s )</td>
<td>( D(f(x),f(y)) \leftarrow D(f(x),f(y))+1 )</td>
</tr>
<tr>
<td>( t^e_s &gt; t^y_s, t^e_s &lt; t^y_s )</td>
<td>( D(f(y),f(x)) \leftarrow D(f(y),f(x))+1 )</td>
</tr>
<tr>
<td>( t^e_s = t^y_s, t^e_s &lt; t^y_s )</td>
<td>( S(f(x),f(y)) \leftarrow S(f(x),f(y))+1 )</td>
</tr>
<tr>
<td>( t^e_s = t^y_s, t^e_s &lt; t^y_s )</td>
<td>( S(f(y),f(x)) \leftarrow S(f(y),f(x))+1 )</td>
</tr>
<tr>
<td>( t^e_s &lt; t^y_s, t^e_s = t^y_s )</td>
<td>( F(f(x),f(y)) \leftarrow F(f(x),f(y))+1 )</td>
</tr>
<tr>
<td>( t^e_s &lt; t^y_s, t^e_s = t^y_s )</td>
<td>( F(f(y),f(x)) \leftarrow F(f(y),f(x))+1 )</td>
</tr>
<tr>
<td>( t^e_s = t^y_s, t^e_s = t^y_s )</td>
<td>( E(f(x),f(y)) \leftarrow E(f(x),f(y))+1 )</td>
</tr>
</tbody>
</table>

For recognition, the bag-of-components and temporal histograms are extracted from each test video, and compared with learned action signatures based on the distance metric discussed in Section 4.2.5. The final decision is made using equation (25). See Figure 4.1 for the flowchart of this process.

4.4. Experimental Results

Here I describe experiments to evaluate our approach using the KTH human motion dataset and Weizmann action dataset. The actions in both datasets were recorded in constrained settings.
Fig. 4.1 shows some sample frames from both datasets. Comparison experiments using bag-of-components representation were performed for both datasets, and a logic-based action recognition approach with temporal relationships was compared with our approach quantitatively and qualitatively. For a given test video, the classifier finds its action type based on one-against-all using (25).
The results are reported using recognition accuracy. For each type of action $i$, its accuracy is defined as

\[
acc_i = \frac{\text{#corrected recognized videos}}{\text{#total videos of action } i}
\]

For comparison experiments with bag-of-components representation, we discard the first term in (25), and keep the other steps the same. The experimental results show that the recognition accuracy improves by combining temporal information. We also conducted experiments on the Weizmann dataset using a logic-based approach, since Allen interval relationships are commonly used in formal logic systems. We follow the ideas in [131] to define actions based on Markov logic networks, and use the same inference engine as in [131] to perform action recognition.

4.4.1. KTH Dataset

The KTH dataset contains six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) performed several times by 25 subjects in four different scenarios, including outdoors, outdoors with scale variation, outdoors with different clothes, and indoors. All video sequences have static and homogeneous backgrounds at 25fps frame rate and $160 \times 120$ resolution. Altogether there are 2391 sequences.

<table>
<thead>
<tr>
<th></th>
<th>BoC</th>
<th>BoC+Temporal</th>
</tr>
</thead>
<tbody>
<tr>
<td>boxing</td>
<td>96.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>handclapping</td>
<td>78.0%</td>
<td>84.0%</td>
</tr>
<tr>
<td>handwaving</td>
<td>88.0%</td>
<td>92.0%</td>
</tr>
<tr>
<td>jogging</td>
<td>70.0%</td>
<td>76.0%</td>
</tr>
<tr>
<td>running</td>
<td>98.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>walking</td>
<td>82.0%</td>
<td>86.0%</td>
</tr>
<tr>
<td>Mean Accuracy</td>
<td>85.3%</td>
<td>89.7%</td>
</tr>
<tr>
<td>F-test</td>
<td>$p$-value=0.096</td>
<td></td>
</tr>
</tbody>
</table>
Long video sequences containing motion clutters were segmented into clips of around 20 seconds. This pre-processing reduces the number of the trajectories in a video for analysis, and does not affect the application of online action detection. For each category, we have 50 videos for training and 50 videos for testing. The average per-class classification accuracy is summarized in Table 4.2. In Table 4.2, the result for BoC is from using only bag-of-components based on our implementation using a naive Bayesian classifier. Our model achieves 89.7% of accuracy by combining bag-of-components and temporal relations. This verifies the performance improvement compared with the result of bag-of-components. The $p$-value from $F$-test is 0.096. Figure 4.2 shows the confusion matrix of recognition results for the KTH dataset.

![Confusion Matrix for KTH Dataset](image)

Figure 4.2. Confusion matrix for KTH dataset. Blue means 0, and red means 1. (Better viewed in color.)

### 4.4.2. Weizmann Dataset

The Weizmann dataset consists of 90 low-resolution (144x180 pixels) video sequences showing nine different persons, each performing 10 natural actions: bending, jumping, jumping-in-place (pjump), jacking, running, gallop sideways (side), skipping, walking, waving one hand (wave1) and waving two hands (wave2), as shown in Figure 4.1. Nine actions (not including
skipping) were also used for experiments. The recognition results for both 10-action and 9-action 
are shown in Table 4.3. We achieve 94.1% accuracy for 9-class actions, and 87.8% for 10-class 
actions. Both are better than their pure bag-of-components counterparts. The confusion matrix for 
9-action classification results is illustrated in Figure 4.3. We notice that confusion exists between 
gallop sideways and jumping. This is probably due to the fact that both have similar movement, 
and the only difference is that the participants are facing different directions, which our approach 
has not considered in modeling.

<table>
<thead>
<tr>
<th>Table 4.3. Accuracy for Weizmann dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>9-class Weizmann dataset</td>
</tr>
<tr>
<td>BoC</td>
</tr>
<tr>
<td>Mean Accuracy</td>
</tr>
<tr>
<td>10-class Weizmann dataset</td>
</tr>
<tr>
<td>BoC</td>
</tr>
<tr>
<td>Mean Accuracy</td>
</tr>
</tbody>
</table>

4.4.3. Comparison with Logic-Based Approach

Besides the comparisons with the bag-of-components approach, we compare our approach 
with a logic representation-based approach which only describes the relationships among compo-
nents. The temporal relationships are often employed in logic systems for action recognition, as in 
[91][131]. They are frequently expressed using Allen temporal predicates. To our best knowledge, 
no experimental results have been shown on widely used public dataset such as KTH and Weiz-
mann. Since there is no source code available for those existing approaches, we implemented our 
logic rules for actions in Weizmann dataset according to the guideline described in [91]. We used 
Alchemy⁶ as our inference engine as used in [131].

A large amount of preprocessing is needed to describe actions using logic rules, including 
feature (i.e., trajectory) extraction, predicates design and grounding, and action rule formalization.

⁶http://alchemy.cs.washington.edu/
We semi-automatically extracted trajectories of human body parts from the Weizmann dataset. The actions analyzed include bending, jacking, jumping, jumping in place, skipping, walking, single-hand waving and two-hand waving. We excluded running because the tracker could not give reasonable trajectories. Two videos from each action category were selected. Each video may contain a different number of action occurrences. For each video, we first manually marked the body parts (head, hands and feet) to track, and then ran the TLD tracker\(^7\) to generate trajectories for them. The first row of Figure 4.1 shows examples of the trajectories. Depending on the chosen object trackers, the extracted trajectories may not always be accurate. In Figure 4.1(d), for example, the TLD tracker confuses the two hands when they are close to each other. The trajectories were then segmented according to the moving directions, as illustrated in Figure 4.4 for the right-hand trajectory in skipping.

Based on these trajectory segments, we generated candidate time intervals during which an action happens. The relationships between two intervals were described by the same Allen interval.

\(^{7}\)http://personal.ee.surrey.ac.uk/Personal/Z.Kalal/tld.html
predicates as shown in Figure 4.4. To describe the spatio-temporal change of a trajectory, we defined predicate \( \text{Move}(P, BP, I, D) \) and \( \text{MoveInSX}(P, BP, I) \) where SX is a quantized movement scale. All the rules for actions were in first-order logic forms, as shown in Table 4.4 for several examples. For a complete list of predicates and rules, refer to appendix at http://students.cse.unt.edu/~gc0115/ar-append.pdf.

![Graph showing trajectory segmentation](image)

**Figure 4.4.** Segmentation of trajectory of a body part. The top left is the original trajectory for right hand, the bottom right is the direction assignment, and the others are the segmentation results.

Table 4.5 shows the recognition accuracy of the logic-based approach, with comparison with the proposed approach. The average accuracy for the logic-based system is 76.7%. The accuracy depends on the rules to express the actions, but the obtained accuracy is similar to the results reported in [91] though different test videos were used.

Besides the difference in performance between the approaches, we compared them in terms
of the applicability. Though logic-based approaches have better extensibility and capability to
describe multi-level actions, they require, without assisting tools, great efforts to pre-process the
data and describe the actions in first-order logic. Table 4.6 summarize some of the comparisons.

Table 4.4. Example predicates and rules to define actions.

<table>
<thead>
<tr>
<th>I. Predicates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move((P, BP, I, D))</td>
</tr>
<tr>
<td>MoveInS1((P, BP, I))</td>
</tr>
<tr>
<td>Bend((P, I))</td>
</tr>
<tr>
<td>(\ldots)</td>
</tr>
<tr>
<td>Jump((P, I))</td>
</tr>
<tr>
<td>Meet((I1.12),\text{Before}(I1.12),)</td>
</tr>
<tr>
<td>Finish((I1, I2),) Start((I1.12),) Allen temporal predicates</td>
</tr>
<tr>
<td>During((I1.12),\ldots)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>II. Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move((P, \text{Head}, I1, D6)) (\land) MoveInS1((P, \text{Head}, I1)) (\land) Move((P, \text{Head}, I2, D2)) (\land)</td>
</tr>
<tr>
<td>MoveInS1((P, \text{Head}, I2)) (\land) Start((I1.13)) (\land) Finish((I2, I3)) (\land) Meet((I1, I2))</td>
</tr>
<tr>
<td>(\Rightarrow) Bend((P, I3))</td>
</tr>
<tr>
<td>(\ldots)</td>
</tr>
</tbody>
</table>

4.5. Summary

We proposed an algorithm to explore the temporal relations between trajectory groups in
videos, and applied it to action recognition and intelligent human-machine interaction systems.
The trajectory components are application-independent features, and function well as mid-level
descriptors of actions in videos. The experiments demonstrated performance improvements com-
pared with a pure bag-of-features method. The success of this semantics-free recognition method
provides the potential to define high-level actions using low-level components and temporal the
relationships between them. This is similar to the way humans perceive and recognize actions.
TABLE 4.5. Accuracy comparison between proposed approach and logic-based approach (8-actions).

<table>
<thead>
<tr>
<th></th>
<th>Logic-based</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>bending</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>jacking</td>
<td>80%</td>
<td>100%</td>
</tr>
<tr>
<td>jumping</td>
<td>80%</td>
<td>67%</td>
</tr>
<tr>
<td>pjumping</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>running</td>
<td>-</td>
<td>89%</td>
</tr>
<tr>
<td>skipping</td>
<td>86%</td>
<td>-</td>
</tr>
<tr>
<td>walking</td>
<td>40%</td>
<td>78%</td>
</tr>
<tr>
<td>waving-1</td>
<td>75%</td>
<td>89%</td>
</tr>
<tr>
<td>waving-2</td>
<td>83%</td>
<td>100%</td>
</tr>
<tr>
<td>Mean Accuracy</td>
<td>76.7%</td>
<td>90.4%</td>
</tr>
</tbody>
</table>

TABLE 4.6. Comparing proposed approach with logic-based approaches.

<table>
<thead>
<tr>
<th>Type</th>
<th>Proposed</th>
<th>Description-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic unit</td>
<td>trajectory components</td>
<td>individual trajectories</td>
</tr>
<tr>
<td>Tracking entity</td>
<td>feature points</td>
<td>known objects</td>
</tr>
<tr>
<td>Temporal</td>
<td>quantitative</td>
<td>qualitative</td>
</tr>
<tr>
<td>relationships</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Level of hierarchy</td>
<td>limited (2-levels)</td>
<td>unlimited,</td>
</tr>
<tr>
<td></td>
<td>bottom-up</td>
<td>top-down</td>
</tr>
<tr>
<td>Handle imperfect</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>low-level input</td>
<td>(feature-level)</td>
<td>(inference-level)</td>
</tr>
<tr>
<td>Deployment efforts</td>
<td>few</td>
<td>a lot (days)</td>
</tr>
</tbody>
</table>

In this study, we used a simple inference/recognition method, the information extracted from the temporal relation between trajectory groups can be input to other inference engines.
The extracted components can be used as input for logic-based systems to construct the predicates, which demands semantic assignment to components. Compared to existing logic-based approaches, our approach requires less computationally intensive preprocessing and yet achieves better results. More relationships other than temporal ones can be explored in a similar way to describe more complex actions, as planned for future work.

From the experiments we have done with both approaches, we conclude that our approach can be more easily applied to different scenarios. It is worth mentioning that the trajectory segments in the analysis can be replaced by trajectory components. Aggregation of trajectories into components lead to more robust mid-level representation of actions. While one individual trajectory could be incomplete or erroneous, the collective descriptors of similar trajectories are mostly stable across different scenarios.

In this chapter, the temporal structures of trajectory components are measured by matrices which are incorporated into a bag-of-features framework directly. We will explore more effective ways to use such temporal relationships by exploring the characteristics of these temporal matrices. Chapter 5 shows how the temporal matrices are decomposed/interpreted with a sparse coding framework.
CHAPTER 5

ACTION GRAPH DECOMPOSITION BASED ON SPARSE CODING

5.1. Introduction

With the prevalence of video-related applications across different domains such as surveillance, human machine interaction and movie narration, automatically analysing video content has attracted attention from both research and industry. Action recognition is usually one of the most important and popular tasks, and requires the understanding of temporal and spatial cues in videos.

Efforts have been taken to build models for representation and inference for action recognition. Models at the early stage used local features, and achieved success under some specific conditions. To represent the actions in a video, many of them applied the bag-of-features scheme and neglected the spatial or temporal relationships between the local features. Since the same set of local features can represent different actions, this scheme is hard to handle complex scenarios.

Action recognition has put special emphasis on the representation of the spatial and temporal relationships between low-level or mid-level features. Graphical models are the choice for most work. Such models build a graph for higher-level or global action recognition based on lower-level features. Examples of such models include hidden Markov models, dynamic Bayesian networks among others. Recently, deep learning approaches construct a multi-layer graphical model to learn the representation from videos automatically, and achieve state-of-the-art performance.

Motivated by graphical models to preserve the structure of the signals, we propose an action graph-based sparse coding framework. Most models of sparse coding are based on a linear combination of dictionary signals to approximate the input, and the signals are usually \( n \)-dimensional vectors. Differing from traditional use, we apply sparse coding to action graphs, which are represented by \( n \times n \) dimensional matrices. Such an extension keeps the spatio-temporal structure at the signal level, and the sparse coding problem still remains tractable and has effective solvers.

5.2. Action Graph from Dense Trajectories

In this section, we describe the construction of action graphs. Dense trajectories [139] are employed as low-level features from which we extract meaningful local action descriptors (referred
to as actionlets hereafter). The action graphs describe the temporal relations between actionlets, and are used as features in the sparse coding framework in Section 5.3.

![Figure 5.1](image)

**Figure 5.1.** Illustration on trajectory grouping based on spatio-temporal proximity.

5.2.1. Grouping Dense Trajectories

The dense trajectories in [139] are extracted from multiple spatial scales based on dense optical field. Abrupt change and stationary trajectories are removed from the final results. For each trajectory, the descriptor combines trajectory shape, appearance (HoG), and motion (HoF and MBH) information. Therefore, the feature vector for a single trajectory is in the form of

\[
T = (S, HoG, HoF, MBH_x, MBH_y)
\]

where \( S = \frac{\sum_{j=t}^{L} \Delta P_j}{\| \Delta P \|} \) is the normalized shape vector, and its dimension \( L \) is the length of the trajectory. MBH is divided into \( MBH_x \) and \( MBH_y \) to describe the motion in \( x \) and \( y \) direction respectively.

The trajectories are clustered into groups based on their descriptors, and each group consists of spatio-temporally similar trajectories which characterize the motion of a particular object or its part. Given two trajectories \( t_1 \) and \( t_2 \), the distance between them is

\[
d(t_1, t_2) = \frac{1}{L} d_S(t_1, t_2) \cdot \bar{d}_{spatial}(t_1, t_2) \cdot d_t(t_1, t_2)
\]

where \( d_S \) is the Euclidean distance between the shape vectors of \( t_1 \) and \( t_2 \), \( \bar{d}_{spatial}(t_1, t_2) \) is the mean spatial distance between corresponding trajectory points, and \( d_t(t_1, t_2) \) indicates the temporal distance. Trajectories are grouped based on a graph clustering algorithm [128]. Figure 5.1 shows examples of grouped trajectories with background motion removed for some sample videos.
The trajectories provide low level description to the action content in a video. A mean feature vector, $x_i \in \mathbb{R}^d$, is obtained for all the trajectories in the same group. Because of the large motion variation even in the same type of actions, our model clusters these trajectory groups using K-means over $x \in \mathbb{R}^d$’s to generate a set of prototypes of trajectory clusters, which describes different types of local actions.

5.2.2. Action Graphs

Based on the bag-of-groups representation, our model develops the statistical temporal relations between the “groups”. we categorize Allen’s temporal relationships into two classes: overlaps($O$) and separates($S$), and construct two types of action graphs. It is also possible too use the original thirteen relations to construct action graphs. Because the procedure is the same, we use the two categorized relations for simplicity.

For each type of action, the temporal relationship between pairs of group words is modelled by an action graph, which is a two-dimensional histogram. Each histogram shows the frequencies with which the relation is true between a pair of group words. That is, a temporal relation $\mathcal{R}_i \in \{O, S\}$, $\mathcal{R}_i(x, y)$ is the frequency of $x \mathcal{R}_i y$ between two group words $x$ and $y$. In our model, we construct the temporal relations for each type of action in a supervised manner. Figure 5.2 shows an example of overlaps for different actions in one testing dataset. It can be observed that different actions exhibit different histograms, and similar actions have similar histograms. Examining each of the histograms shows which temporal relation (such as overlaps for boxing) has a stronger response for some pairs of group words than the others. This implies the major relation between...
5.3. Action Graph Representation with Sparse Coding

Given actionlets and their temporal relationship, we precede here to present a sparse coding approach which is based on the temporal relationship graphs, and apply it for video action recognition. Let \( X = [x_1, ..., x_n] \in \mathbb{R}^{d \times n} \) denote the data matrix of a video clip, where \( x_i \in \mathbb{R}^d \) denotes each actionlet descriptor. For the temporal relationships separate (\( S \)) and overlap (\( O \)), each is represented by an undirected action graph for this study. Therefore \( K = 2 \) action graphs \( G_k \) are employed to cover all the cases using a 1-of-K coding scheme. If actionlets \( a_i \) and \( a_j \) has a temporal relationship \( R_k \), then edge \((a_i, a_j)\) exists in graph \( G_k \). For each type of graph, sparse coding analysis is performed separately, and then the codes are combined to form the feature representation of a video clip for tasks such as classification.

In this section, we describe the Laplacian matrix of action graphs in Section 5.3.1, followed by discussion on sparse coding framework in Section 5.3.2 and Section 5.3.3.

5.3.1. Laplacian Matrix of Action Graphs

As representation of action graphs, the adjacency matrices are not an ideal choice to be adapted in a sparse coding framework. As shown in the following sections, symmetric positive definite matrices are desirable to compare action graphs and reduce the problem to a classic form. In this work, we use the Laplacian matrix, \( L \), to represent the action graphs. This is mainly because the Laplacian matrix of a graph is always symmetric positive semi-definite (SPSD), i.e. \( \forall x, x^T L x \geq 0 \).

There exists an easy conversion between the Laplacian matrix of a graph and its adjacency or incidence matrix. For adjacency matrix \( A \) representation of action graphs, its Laplacian matrix \( L = D - A \) where \( D \) is diagonal degree matrix. However, construction of Laplacian matrix from adjacency matrices only apply for simple graphs which are undirected without loops or multiple edges between two actionlets. Another way to obtain Laplacian matrix of a graph is through incidence matrices. Suppose \( M_{|V| \times |E|} \) is the incidence matrix, then \( L = MM^T \). For an undirected graph, we can use its oriented incidence matrix by arbitrarily defining an order of the actionlets;
it is straightforward to get $M$ and thus $L$ for a directed graph. We use the incidence matrix of a graph to obtain its Laplacian matrix for further extension although we use undirected graphs in this work.

To make the matrices of action graphs strictly positive definite (SPD), we regularize the Laplacian matrices by adding a small multiple of the identity matrix. Without further explanation, all the action graphs below are represented by regularized Laplacian matrices, including the dictionary and the action graph to be approximated.

5.3.2. Sparse Coding for Action Graphs

Action graphs describe the temporal relationship among the actionlets, and each is represented by a Laplacian matrix. For each of the two relationships, we collect several action graphs from different videos of the same type. For example, graph $O$ describes the “overlap” relationship between any two actionlets. If there exists a pair of actionlets $a_i$ and $a_j$ in a video whose time intervals overlap, then there is an edge between nodes $a_i$ and $a_j$ in the graph $O$, and the weight is the normalized frequency of $a_iOa_j$.

Given a set of video clips, an action graph $A_i$ is constructed for each of them. For localization and precise detection purpose, $A_i$’s are constructed from short clips or results after shot detection on an entire video. Let $D = [A_1, A_2, ..., A_p] \in \mathbb{R}^{(n \times n) \times p}$ be the dictionary of the action graphs, and $A_i$ be an $n \times n$ basis relationship, where $n$ is the total number of actionlet types across different actions. For given videos, let $G = [G_1, ..., G_m] \in \mathbb{R}^{(n \times n) \times m}$ be the action graphs extracted from them. Based on the dictionary, we decompose each graph $G_i$ into the linear combination of the basis relationships

\begin{equation}
G_i \approx \hat{G}_i = s_{i1}A_1 + s_{i2}A_2 + ... + s_{ip}A_p \triangleq s_iD
\end{equation}

where $s_i$ is the coefficient vector for action graph $G_i$. Let $S = [s_1, ..., s_m]$ be the coefficient matrix for $G$.

The empirical loss function $\ell(G, S) = \sum_{i=1}^{m} d(G_i, s_iD)$ evaluates the decomposition error by representing $G$ using $S$ based on dictionary $D$. $d(\cdot, \cdot)$ measures the distortion of the approximation $\hat{G}_i$ to its original action graph $G_i$, which can be evaluated by the distance between two
matrices. The objective function can then be formulated as in (30).

\[
\min_S \sum_{i=1}^{m} d(G_i, s_iD) + \alpha \| s_i \|_1
\]

where \( \| \cdot \|_1 \) denotes \( \ell_1 \) norm. \( \| s_i \|_1 \) is the sparsity term, and \( \alpha \) is a parameter which trades off between the empirical loss and sparsity.

5.3.3. Distance between Action Graphs

To evaluate the empirical loss, different distance metrics between action graphs, \( d(\cdot, \cdot) \), could be used. Let \( S_{++}^m \) denote the set of symmetric positive definite (SPD) matrices.

Given \( A, B \in S_{++}^m \), in this paper we use the Logdet divergence [67] as the distortion measurement because it results in a tractable convex optimization problem. The Logdet divergence between \( A, B \in S_{++}^m \) is defined by

\[
D_{ld}(A, B) = tr \left( AB^{-1} \right) - \logdet \left( AB^{-1} \right) - n.
\]

The Logdet is convex in \( A \), and therefore \( A \) can be \( G_i \) which is the true action graph we need to estimate based on a sparse combination of the basis action graphs in the dictionary \( D \). Following a similar procedure as in [120], we transform \( D_{ld}(A, B) \) to convert (30) into a known determinant maximization problem. The objective function becomes

\[
\min_S \sum_{i=1}^{m} tr \left( s_i^T c \right) - \logdet \left( s_i \hat{D} \right),
\]

\[
s.t. \quad s_i \geq 0, \quad s_i \hat{D} \succ 0,
\]

where \( \hat{D} \) is transformed dictionary tuned according to the action graph to approximate, \( G_i \), i.e., \( \hat{D} = [\hat{A}_j]_{j=1}^p \) with \( \hat{A}_j = G_i^{-1/2} A_j G_i^{-1/2} \). \( c \) is the vector of the traces of dictionary \( \hat{D} \): \( c_i = tr \hat{A}_i + \alpha \).

This is a convex optimization problem on \( \{ s_i | s_i \hat{D} \succ 0 \} \) known as max-det problem [133], which has an efficient interior-point solution. We use the cvx modeling framework \(^8\) to get the optimal values for \( S \).
Figure 5.1. Plot of the optimal sparse coding solutions. Notice the the sparseness of the coefficients.

5.4. Experimental Results

We use the KTH dataset to evaluate our approach. We split the videos of each category into training and testing, and build the dictionary $D$ from the training dataset. Action graphs are constructed for each video, and we randomly select $k$ ($p = Nk$, where $N$ is the number of actions) action graphs from each category and assemble them to obtain the dictionary $D$. Therefore, the dictionary is in the form of

$$D = [\mathcal{A}_{11}, ..., \mathcal{A}_{1k}, ..., |\mathcal{A}_{N1}, ..., \mathcal{A}_{Nk}]$$

For any given video, its action graph is decomposed using the dictionary and represented by the decomposition coefficients, $s_i$. Figure 5.1 shows two examples of the coefficients of two videos of different actions. For classification purpose, we get the maximum of decomposition coefficients of each category, $a_i$, and label the video with the category having the maximum coefficient as shown in the following equation:

$$a^* = \arg \max_a \{ \max_i \{ s_{ai} \} \}$$

---

8CVX Research: http://cvxr.com/
Figure 5.2. Average sparse coding coefficients $s_i$ for each category of videos

Figure 5.3. The maximum coefficients from the sparse coding.

Figure 5.2 shows the result from sparse coding optimization. For each testing video of categories shown in x-axis, we take the maximum optimized coefficients $s^*_a$ for each category $a$, $a \in \{box, clap, jog, run, walk\}$, i.e. $s^*_a = \max \{s_a\}$, and then average it over all the videos in the
same category to obtain \( \vec{s}_a \). Each vector \( \vec{s}_{box}, ..., \vec{s}_{walk} \) corresponds to one curve in the figure. For each curve, we can see the peak of the coefficients is always consistent with its actual type of actions. Figure 5.3 shows the decomposition coefficients from some sample videos. The shaded cells denote that the corresponding videos do not have the maximum coefficient and thus will be misclassified.

5.5. Summary

We present a sparse coding approach to decompose action graphs for analysing activities in videos. The action in a video is characterized by an action graph, which describe the temporal relationships between different types of mid-level trajectory clusters. The use of graphs instead of vectors as features keeps better structure information about the components of actions. The difficulty with variation in graphs is handled by using tensor sparse coding.
CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this dissertation, I conducted research to address two important problems for video-based activity analysis: the localization of spatio-temporal regions-of-interest (ROI) and spatio-temporal relationship representation for action recognition. The outcomes of this dissertation research may be used for many applications depending on the purposes, or combined together to form a “localization-then-recognition” pipeline.

I divided the research into three parts, including a 3D structure tensor-based regions-of-interest localization algorithm for wide-angle view videos, an action recognition model with combination of bag-of-words (BoW) representation and temporal relationships, and a new temporal relationship representation based on sparse coding.

6.1. A Nonparametric Approach for ROI Detection

Region-of-interest detection is a popular topic in image processing and video analysis. It can be used as a pre-processing step for tasks such as image segmentation or action recognition, while it can also be applied independently in scenarios such as anomaly detection and intelligent surveillance. In Chapter 3, we proposed a method to detect spatio-temporal ROI in a unifying manner. I selected 3D structure tensor as the descriptor of the motion at individual location and modeled the motion pattern by the distribution of features from 3D structure tensors. The estimation of the distribution is a kernel-based nonparametric approach. The innovation of the research lies in the following two aspects:

- I chose 3D structure tensors as descriptors for the motion. The 3D structure tensors can be constructed directly from pixel values but have more stable characterization than raw pixels;
- The motion patterns are modeled and estimated in a nonparametric manner, which alleviates the tuning of parameters for different applications;
- The proposed approach detects both spatial and temporal regions of interest with the same setting of parameters. Basically it is a data-driven approach.
6.2. Combining Temporal Relationships with BoW for Action Recognition

Spatial and temporal relationships are becoming the focus of video-based action analysis after traditional bag-of-words (BoW) model encounters bottleneck. I exploited the incorporation of spatial and temporal relationships for action recognition in Chapter 4. I constructed mid-level representation of actions (i.e., actionlets) by clustering dense trajectories with spatial proximity and feature similarity. The temporal relationship between actionlets are modeled using Allen’s temporal predicates and combined into BoW model. Two major work in this dissertation are:

- I proposed to use actionlets as mid-level representation of actions based on dense trajectories. This does not require the tracking of objects yet provide good description of short-term primitive actions.
- Based on Allen’s temporal predicates, temporal relationships between actionlets are modeled using 2D histograms, which are incorporated to the bag-of-words model for action recognition. We compared our approach with traditional bag-of-words model and Markov logic, and the results show the effectiveness of the proposed approach.

6.3. Novel Temporal Relationship Representation

A novel temporal relationship representation is presented in Chapter 5. I proposed Action Graph as a general representation of temporal relationships. It describes the temporal relationships between actionlets but in the form of laplacian matrices. This dissertation studies how to obtain distinctive features from action graphs based on sparse coding. This was motivated by the issues related to learning and recognition from 2D features, for which a new representation is presented. Correspondingly, this dissertation studies distance metrics for action graphs to categorize actions of videos. The major efforts lie in:

- A better representation for temporal relationships that is in SPD form. This enables efficient learning, and
- Decomposing temporal relationships based on sparse coding framework, from which features are obtained for action recognition.
To sum up, this dissertation studies the spatio-temporal information for video-based activity analysis, including region-of-interest localization and action recognition based on mid-level actionlets and action graphs.

6.4. Discussions on Future Work

Spatial and temporal information has been attracting increasingly attention in activity analysis in videos, as mentioned in Chapter 2. The research in this dissertation can be continued in many ways. They can be categorized into the following two aspects: 1) direct improvement of the methods in the dissertation, and 2) advancement of spatio-temporal relationship for video-based activity analytics.

The research can be improved in several ways. Firstly, instead of dense trajectories, local features may be exploited to construct mid-level actionlets. Although dense trajectories have shown better performance, it is a computationally intensive task to obtain and process them. New local feature-based methods are worthy of survey and study. Secondly, a randomly-selected action graphs are used to construct the codebook/dictionary in the sparse coding-based temporal relationship analysis; however, a better implementation could be based on learning of the dictionary automatically and adaptively.

Spatial and temporal relationships are playing a promising role in video-based activity analysis. For future research, modeling of the temporal relationship, including representation and learning, would be my main focus, as initiated in Chapters 4 and 5. I have been rethinking better temporal representations rather than matrices based on Allen temporal predicates. How to apply thus representations and develop corresponding learning framework would lead the research in this field. Last but not least, seeking a unifying way to incorporate spatial relationships is a straightforward yet important next step. In this dissertation, spatial relationships are not treated in the same way as temporal relationships, and they are employed at a different level to cluster the dense trajectories.

In a word, the spatial and temporal relationships will be the emphasis of future research besides investigation on new features for mid-level action representation.
APPENDIX

RELATED PUBLICATIONS
Most of the work introduced in this dissertation has their first appearance in publications. Below are related publications and their correspondence in chapters. Some are directly compiled to this dissertation while the others are research on the application of the methods used.


BIBLIOGRAPHY


79


via probabilistic voting, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 398–405. 23


[38] Olivier Duchenne, Armand Joulin, and Jean Ponce, A graph-matching kernel for object categorization, IEEE International Conference on Computer Vision (ICCV), 2011, pp. 1792–1799. 22


[40] R Gaborski, V Vaingankar, and V Chaoji, Detection of inconsistent regions in video streams, SPIE Human Vision and Electronic Imaging (2004). 10

[41] Utkarsh Gaur, Yingying Zhu, Bi Song, and A Roy-Chowdhury, A “string of feature graphs”
model for recognition of complex activities in natural videos, IEEE International Conference on Computer Vision (ICCV), 2011, pp. 2595–2602. 21, 23


[49] Chris Harris and Mike Stephens, A combined corner and edge detector, Alvey Vision Conference, 1988, pp. 147–151. 11


86


[99] Bruno A Olshausen et al., *Emergence of simple-cell receptive field properties by learning a sparse code for natural images*, Nature 381 (1996), no. 6583, 607–609. 21


88


[118] Jianbo Shi and Carlo Tomasi, *Good features to track*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1994, pp. 593–600. 11, 12


[125] Ju Sun, Xiao Wu, Shuicheng Yan, Loongfah Cheong, Tat-Seng Chua, and Jintao Li, *Hi-

[126] N. Sundaram, T. Brox, and K. Keutzer, Dense point trajectories by gpu-accelerated large displacement optical flow, European Conference on Computer Vision (ECCV), Sept. 2010. 16


[129] Kevin Tang, Li Fei-Fei, and Daphne Koller, Learning latent temporal structure for complex event detection, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 1250–1257. 1


[134] James Vlahos, Surveillance society: New high-tech cameras are watching you, Popular Machines (2008), 64–69. 43


[136] Haiyun Wang and Kaikuang Ma, Structure tensor-based motion field classification and op...
tical flow estimation, Joint Conference on Information, Communications and Signal Processing, and the Fourth Pacific Rim Conference on Multimedia., 2003, pp. 66–70. 27


a moving camera using motion decomposition of Lagrangian particle trajectories, IEEE International Conference on Computer Vision (ICCV), 2011, pp. 1419–1426. 16

[147] Tao Xiang and Shaogang Gong, Video behavior profiling for anomaly detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008), no. 5, 893–908. 10


[149] Bangpeng Yao and Li Fei-Fei, Action recognition with exemplar based 2.5d graph matching, European Conference on Computer Vision (ECCV), 2012, pp. 173–186. 23


[153] Elden Yu and J.K. Aggarwal, Human action recognition with extremities as semantic posture representation, IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2009. 44

[154] Fei Yuan, Gui-Song Xia, Hichem Sahbi, and Veronique Prinet, Mid-level features and spatio-temporal context for activity recognition, Pattern Recognition 45 (2012), no. 12, 4182–4191. 1, 16


[156] Bin Zhao, Fei-Fei Li, and Eric P. Xing, Online detection of unusual events in videos via dynamic sparse coding, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 3313–3320. 21


