TOWARDS ACCURATE GROUP ACTIVITY ANALYSIS IN VIDEOS: ROBUST SALIENCY DETECTION AND EFFECTIVE FEATURE MODELING

BY XINYI CUI

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Professor Dimitris N. Metaxas

and approved by

_____

_____

_____

_____

_____

New Brunswick, New Jersey

May, 2013

ABSTRACT OF THE DISSERTATION

Towards Accurate Group Activity Analysis in Videos: Robust Saliency Detection and Effective Feature Modeling

by Xinyi Cui

Dissertation Director: Professor Dimitris N. Metaxas

Human activity analysis is an important area of computer vision research today. The goal of human activity analysis is to automatically analyze ongoing activities from an unknown video. The ability to analyze complex human activities from videos has many important applications, such as smart camera system, video surveillance, etc. However, it is still far from an off-the-shelf system. There are many challenging problems and it is still an active research area. This dissertation focuses on addressing two problems: various camera motions and effective modeling of group behaviors.

We propose a unified and robust framework to detect salient motions from diverse types of videos. Given a video sequence that is recorded from either a stationary or moving camera, our algorithm is able to detect the salient motion regions. The model is inspired by two observations: 1) background motion caused by orthographic cameras lies in a low rank subspace, and 2) pixels belonging to one trajectory tend to group together. Based on these two observations, we introduce a new model using both low rank and group sparsity constraints. It is able to robustly decompose a motion trajectory matrix into foreground and background ones. Extensive experiments demonstrate very competitive performance on both synthetic data and real videos.

After salient motion detection, a new method is proposed to model group behaviors

in video sequences. This approach effectively models group activities based on social behavior analysis. Different from previous work that uses independent local features, our method explores the relationships between the current behavior state of a subject and its actions. An interaction energy potential function is proposed to represent the current behavior state of a subject, and velocity is used as its actions. Our method does not depend on human detection, so it is robust to detection errors. Instead, tracked salient points are able to provide a good estimation of modeling group interaction. We evaluate our algorithm in two datasets: UMN and BEHAVE. Experimental results show its promising performance against the state-of-art methods.

Acknowledgements

I am grateful to Professor Dimitris Metaxas for his advice and encouragement during my Ph.D. study. He has been an excellent advisor and always directed me towards fundamental research. He also exposed me to opportunities to do research in computer vision and machine learning. None of the work in this dissertation would have happened without him.

I would like to thank the other members of my doctoral committee: Prof. Vladmir Pavlovic, Prof. Tina Eliassi-Rad and Prof. Xiaolei Huang (Lehigh University) for their advice, help and valuable suggestions regarding this dissertation. It is honors for me to have each of them serve in my committee.

I also thank Dr. Rajat Raina (Facebook) and Vignesh Ganapathy (Google). They have been wonderful mentors to me. They have showed me the excellence of research and development experience during my summer internships.

Thanks also go to many professors at Rutgers that have helped and supported me in many aspects, especially to Prof. Doug DeCarlo, Prof. Ahmed Elgammal, Prof. Eric Allender, and Prof. William Steiger. Also thanks to other professors and researchers, especially to Prof. Leon Axel (New York University), Prof. Junzhou Huang (The University of Texas at Arlington).

Special thanks to my friends and colleagues from the Center for Computational Biomedicine Imaging and Modeling (CBIM). I benefited a lot from their friendship and help. They made my years at Rutgers a real pleasure.

Dedication

This dissertation is dedicated to my parents.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Human Activity Analysis

Human activity analysis is an important area of computer vision research today. The goal of human activity analysis is to automatically analyze ongoing activities from an unknown video. For example, a sequence of image frames. In the sample case when a video sequence contains only one clip of a human activity, the objective of the system is to label the video into its activity category. In more complex cases, when an input video is given, the system needs to detect starting and ending times of all occurring activities.

The ability to analyze complex human activities from videos has many important applications. Automated surveillance systems in public places like airports and subway stations require detection of suspicious activities as opposed to normal activities. For instance, an airport surveillance system must be able to automatically find suspicious activities like 'a person leaving a bag' or 'a person placing his/her bag in a trash bin'. Analysis of human activities also enables the real-time monitoring of patients, children, and elderly persons. The construction of gesture-based human computer interfaces and vision-based intelligent environments becomes possible as well with an activity analysis system. Table 1.1 lists a few application domains and potential usages of computer vision systems that understand people's behaviors.

There are various types of human activities. Depending on their complexity, human activities can be roughly categorized into three levels: gestures, actions and group interactions/activities. Gestures are elementary movements of a person's body part, and are the atomic components describing the meaningful motion of a person. 'Stretching an arm' and 'raising a leg' are good examples of gestures. Actions are single person

Application domain	Potential use
security, surveillance	automatic monitoring, abnormality detection
sports and entertainment	sport analysis
web application	movie/video retrieval
hospitals and medical applications	nursing home monitoring, smart surgery

Table 1.1: Application domains and potential use of computer vision systems that can understand people's behaviors.

activities, such as 'walking', 'waving', and 'punching'. group interaction/activites are human activities that involve two or more persons and/or objects. For example, 'a person stealing a suitcase from another' is a human-object interaction involving two humans and one object. A group activity is performed by conceptual groups composed of multiple persons and/or objects. 'A group of persons marching', 'a group having a meeting', and 'two groups fighting' are typical examples of them.

Now days, cameras are everywhere. The research of activity analysis is strong encouraged by the emerging number of cameras. But these cameras do not actually "see" things. For example, for surveillance cameras, we still reply on people to monitor the events and activities. A significant amount of progress on human activity analysis has been made in the past 10 years, but it is still far from being an off the shelf technology. Further, today's environment for human activity analysis keeps changing. The cameras were mostly fixed cameras and without pan-tilt-zoom adjustments. Today's cameras may be mounted on several types of moving platforms ranging from a moving car or a truck to an unmanned aerial vehicle (UAV). In addition, there are increasing number of freely moving cameras, such as smart phones and hand-held digital video cameras. Designing an activity analysis system to handle these cameras is an extremely challenging task.

1.2 Challenges

There are many challenging problems for group activity analysis.

- Few Pixels on Objects: In a scene with a lot of people, detection of individual objects becomes extremely hard as the number of pixels on the object decreases

with increasing number of people. The appearance information becomes further distorted due to the constant interaction among individuals making up the crowd. The interaction among a group of people makes it hard to detect or tracking each individual. See Figure 1.1 for example.

- Appearance Ambiguity: Ideally one would like to track all the visible objects throughout the scene. However, ambiguous appearance information resulting from too few pixels than desirable on the targets makes it difficult to persistently track the objects. Some state-of-art methods rely on manual correction of tracked trajectories for further group activity analysis. This however does not work for automatic video analysis.

- Various camera motions: The crowded scenes may not be recorded by an absolutely stationary camera. As the cameras become cheap, many videos are recorded by hand-held digital video cameras, smart phones or UAV. Even for a surveillance camera mounted in a building, it still suffers from small shakes by window or group vibrations. The analysis of activity from a moving platform poses many more challenges. Noise, tracking, and segmentation issues arising out of stabilization of video add to the difficulty of the problem of the analysis of activities. The camera motions makes it hard to separate the people behaviors from the scene.

- Representation of group behaviors: The behaviors in a group are the interactions among the participants. The individual centric representation of behaviors is not applicable for a group. How to build a model to effectively represent the group interaction is challenging.

1.3 Our Solutions

This dissertation focuses on two challenging problems: handling various camera motions and representation of group behaviors. We propose a salient motion detection method to find the moving area in a video sequence. The video sequence can be recorded by either a stationary camera or a freely moving camera. We also propose a new method

Figure 1.1: Video sequences with a lot of people. As people are severely occluded, it is hard to have accurate detection or tracking of each individual.

to model the behaviors of groups, by representing the group activity as the interactions of particles.

A new method is designed for salient motion detection [19]. This is based on two constraints: the low rank constraint on the salient motion area (foreground area), and the group sparsity constraint on the background area. Those constraints are inspired by two "sparsity" observations behind our method. When the scene in a video does not have any foreground moving objects, video motion has a low rank constraint for orthographic cameras [69, 89]. Thus the motion of background points forms a low rank matrix. For the salient motion regions, the foreground moving objects usually occupy a small portion of the scene. Thus this satisfied a sparsity constraint. In addition, when a foreground object is projected to pixels on multiple frames, these pixels are not randomly distributed. They tend to group together as a continuous trajectory. Thus these foreground trajectories usually satisfy the group sparsity constraint.

We use such information to differentiate independent objects from the scene. Based on these two observations, the video salient motion detection problem is formulated as a matrix decomposition problem. First, the video motion is represented as a matrix on *trajectory* level (*i.e.* each row in the motion matrix is a trajectory of a point). Then it is decomposed into a background matrix and a foreground matrix, where the background matrix is low rank, and the foreground matrix is group sparse. This low rank constraint is able to automatically model background from both stationary and moving cameras, and the group sparsity constraint improves the robustness to noise. We validate our approach on various types of data, *i.e.*, synthetic data, real video sequences

recorded by stationary cameras or moving cameras and/or nonrigid foreground objects. Experiments show that our method performs better than the recent state-of-the-art algorithms.

To model the group activities from video sequences, we propose a new algorithm to represent group activities by learning the relationships between the current behavior state of a subject and its actions [20]. This algorithms is based on the exploration of the reasons why people take different actions under different situations. Our goal is to explore the reasons why people take different actions under different situations.

An interaction energy potential function is defined to represent the current state of a subject based on the positions/velocities of a subject itself as well as its neighbors. Social behaviors are captured by the relationship between interaction energy potential and its action, which is then used to describe social behaviors. Then we use SVM to build a model for analyzing the group behaviors. We test the algorithm on two datasets UMN [2] and BEHAVE [1]. Results show that our method is more powerful to model behaviors in group activities, comparing to other state-of-art algorithms.

1.4 Main Contributions

The main contribution of this dissertation is fourfold.

1. A new model is proposed for salient motion detection in video sequences. For any given raw videos from a surveillance camera, our method is able to locate the moving regions and generate the dense tracked trajectories. This servers as the first step for group activity analysis.

2. Our framework is able to handle various types of video sequences, including rigid and non-rigid objects in stationary cameras, nominally moving cameras and moving cameras. Since our method is based on low rank and group sparsity constraints over multiple frames, it is robust to handle outliers along the video sequences.

3. A new feature representation is proposed for group activity analysis. This method is used to model behaviors in human group activities. As this approach models group activities using social behavior analysis, it is effective to describe the

interactions among a group of people.

4. Our algorithm is successfully applied to video datasets under surveillance cameras. The UMN and BEHAVE datasets contain video sequences of group interactions like walking together, running, fighting, panic, etc. Monitoring group events in a public area is important for video surveillance and smart camera system.

1.5 Organization

The remainder of this dissertation is organized as follows. Chapter 2 reviews the relevant works in two major areas: salient motion detection and action/activity analysis. For salient detection problem in Section 2.2, we start from image saliency detection, and then review the major algorithms in salient motion detection in video sequences. Since our work employ sparse methods to find the salient moving regions, we also briefly introduce sparse method in Section 2.3. Major research methods in action/activity analysis are introduced in Section 2.4. This section starts from surveys in action analysis from single persons, then discuss the major methods in group behavior analysis.

Given a video sequence as an input, our system first locates the salient motion regions. Chapter 3 introduces the proposed salient motion detection method in detail. This method is inspired by two observations from the background and foreground. Based on the observations, a unified and robust framework to effectively handle diverse types of videos is presented. The problem formulation and optimization framework are also discussed here. This framework is able to handle both videos from stationary, nominally moving cameras as well as moving cameras. We also evaluate the algorithm performance in this chapter using both synthetic data and real videos.

After having the salient motion regions, we can further analyze the group behaviors. Our proposed method for modeling behaviors in human group activities is presented in Chapter 4. This approach effectively models group activities based on social behavior analysis. The motivation comes from the way people interact in a public area. As our major contribution is the feature, how the feature is calculated and used is discussed here. To validate the performance, we test it on two datasets: UMN and BEHAVE. It

shows competitive performance on these datasets.

Finally we conclude and discuss the future work in Chapter 5.

Chapter 2

Related Work

2.1 Overview

Our work involves two major research areas: robust saliency detection and effective feature modeling. Thus this chapter reviews the relevant work in both field. In Section 2.2, we will review the major research works in saliency detection. Since saliency detection involves both work in images and videos, it starts from image saliency detection, then the major approaches in motion saliency detection in video sequences. The approach we propose for robust saliency detection uses sparsity analysis to find the salient moving regions. Thus we also give a brief survey of sparsity methods in Section 2.3. Section 2.4 discusses the mainstream work of feature modeling for action/activity analysis in video sequences.

2.2 Salient Motion Detection

Visual saliency is the ability of a vision system (human or machine) to select a certain subset of visual information for further processing. This mechanism serves as a filter to select only the interesting information related to current behaviors or tasks to be processed while ignoring irrelevant information. Recently, salient object detection has attracted a lot of interest in computer vision as it provides fast solutions to several complex processes. There are basically two major categories for saliency detection: saliency detection in images and salient motion detection in videos. This section will give a brief introduction to image saliency detection, and will then mostly focus on the research in salient motion detection.

Image saliency detection aims to detect the salient foreground regions. The saliency

detection system first detects the most salient and attention-grabbing object in a scene, and then it segments the whole extent of that object. The output usually is a map where the intensity of each pixel represents the probability of that pixel belonging to the salient object. This problem in its essence is a segmentation problem but slightly differs from the traditional general image segmentation. While salient object detection models segment only the salient foreground object from the background, general segmentation algorithms partition an image into regions of coherent properties. Some typical research methods in this area are Itti [42] and Hou [36]. Please refer to extensive reviews for more details [8].

A considerable amount of work has studied the problem of salient motion detection in video sequences. Salient motion detection is actually a broad concept. The goal of salient detection is to find the salient foreground regions from the background in a video sequence. Background subtraction and motion segmentation all fall into this concept. Since this dissertation compares both sub-domains, the rest of this section will discuss these two research areas.

Background subtraction aims to detect all foreground objects given a video sequence, and label the foreground areas as a binary mask. Many algorithms have been proposed for this problem. Here we review a few related work, and please refer to [71, 11] for comprehensive surveys.

The mainstream in the research area of background subtraction focuses on stationary cameras. The earliest background subtraction methods use frame difference to detect moving objects. It thresholds the difference between two/three consecutive frames. Large changes are considered as foreground [43, 44]. Many subsequent approaches have been proposed to model uncertainty in background appearance. W4 is a well known system to incorporate statistic models for the background subtraction problems [31]. It models the variance in a set of background images with the maximum and minimum intensity value and the maximum difference between consecutive frames. Pfinder [95] is based on Gaussian distribution models. The assumption is that the pixel value follows a Gaussian distribution, and a likelihood model is used to compare the likelihood of background and foreground for a particular pixel. The Mixture of Gaussians (MoG) [85]

assumes the color evolution of each pixel can be modeled. It is widely used in real systems [88]. Elgammal et al. [27] propose a non-parametric model. Sheikh and Shah consider both temporal and spatial constraints and build a joint spatial-color model in a Bayesian framework [82]. [22, 21, 24] use image saliency properties to find salient motion regions.

One important variation in stationary camera based research is the background dynamism. When the camera is stationary, the background scene may change over time due to many factors (*e.g.* illumination changes, waves in water bodies, shadows, etc). Several algorithms have been proposed to handle dynamic background [64, 117, 57, 46, 50, 18, 50].

All the above work assumes that the camera is stationary. Background subtraction under moving camera is more challenging since it is not straightforward to model or update foreground/background. The research for moving cameras has recently attracted people's attention. A popular way to handle camera motion makes strong assumption of the scene. [63, 34, 75] cancel the camera motion by estimating dominant background motion to identify foreground objects. However, these methods are based on a strong assumption that the background is able to be modeled effectively with a single plane, which is not generally valid. A more advanced approach is the combination of plane and parallax framework, where a homography is first computed to match the features in two consecutive frames and the residual pixels are further registered by parallax estimation [106]. This technique involves fewer restrictions than the homography-only based algorithms, but still assumes that there exists a dominant plane for matching by homography.

Recently, [81] has been proposed to build a background model using RANSAC to estimate the background trajectory basis. This approach assumes that the background motion spans a three dimensional subspace. Then sets of three trajectories are randomly selected to construct the background motion space until a consensus set is discovered, by measuring the projection error on the subspace spanned by the trajectory set. However, RANSAC based methods are generally sensitive to parameter selection, which makes it less robust when handling different videos. The goal of our approach is to propose a

unified framework to robustly handle diverse types of videos.

Motion segmentation aims to segment the trajectories into different motion segments. The trajectories are tracked using tracking algorithms from an input video sequence. Many motion segmentation algorithms have been proposed in recent years. Generalized Principal Component Analysis (GPCA) [92] is designed as generic subspace separation algorithms that do not place any restriction on the relative orientations of the motion subspaces. Local Subspace Affinity (LSA) [99] uses local information around each trajectory to create a pairwise similarity matrix that can then be segmented using spectral clustering techniques. This algorithm works well then the trajectory number is small. But the algorithm itself is computationally heavy, it is not able to handle dense trajectories. RANSAC based method is also proposed to segment motions [91]. The Hopkins 155 Dataset has been created with the goal of providing an extensive benchmark for testing feature based motion segmentation algorithms. The salient motion algorithm this dissertation proposes here is similar to motion segmentation. But our algorithm does not need to know the cluster number, which is not given in real applications.

2.3 Sparsity Methods

Sparsity methods have been widely studied recently. The basic idea is that a sparse signal can be recovered with high probability from a small number of its linear measurements [12, 26]. The problem of sparsity priors can be solved by either using greedy methods such as basis pursuit [17] and matching pursuit [58], or using $L1$ norm relaxation and convex optimization [12, 45, 28]. Sparsity methods have been used in many applications, such as face recognition [14], super resolution [100, 101], medical image segmentation [112, 113, 114], image annotation [110, 111] and MR image reconstruction [39, 38].

The idea of using group sparse structure to achieve better performance has attracted a lot of attention [107]. Theoretically proves are provided to show that group sparsity is superior to standard sparsity for strongly group-sparse signals [7, 40]. When the

underlying group structure is consistent with the data, a convincing theoretical justification has been provided to use group sparse regularization instead of regular sparse regularization. Group sparsity has been used in several vision problems including, but not limited to, human gait recognition [98] and image annotation [110].

[40, 37] employs spatial group sparsity to tackle background subtraction problem. It naturally extends the standard sparsity concept in compressive sending to dynamic group sparsity. This approach is motivated by the observation that in some practical sparse data the nonzero coefficients are often not random but tend to be clustered. By utilizing both the clustering and sparsity priors, better results can be achieved. A greedy sparse recovery algorithm is developed, which prunes data residues in the iterative process according to both sparsity and group clustering priors rather than only sparsity as in previous methods. This algorithm can recover stably sparse data with clustering trends using far fewer measurements and computations current state-of-the-art algorithms with provable guarantees. This algorithm can also adaptively learn the dynamic group structure and the sparsity number if they are not available in the practical applications. Our approach is inspired by this method in the sense of group sparsity. But there are two differences. First the group property used our method is on the temporal domain, constraining the tracking points from multiple frames to group together. It is not the neighboring pixels on the same frame. Second, the dynamic group sparsity is only able to solve background subtraction problem under stationary cameras, while our method is able to handle both stationary cameras and moving cameras.

Low rank constraints and matrix completion problems have been well studied in recent years [15] and applied to several vision problems, such as face recognition [14], face shadow removal [65] and image classification [108].

The basic idea is to recover a low rank matrix from only a small fraction of its entries, and by extension, from a small number of linear functionals. Robust PCA [14] tries to recover a low-rank matrix by minimizing nuclear norm. The basic idea is that given a data matrix with a low-rank component and a sparse component, it is possible to recover both the low-rank and the sparse components exactly by solving a very convenient convex problem called Principal Component Pursuit under some

suitable assumptions. It provides a principled approach to robust principal component analysis, which can recover the principal components of a data matrix, even though a positive fraction of the entries are arbitrarily corrupted or missing. This work has been successfully applied to detection of objects in a cluttered background scenario. It assumes that the stationary background satisfies a low rank constraint. However, this assumption does not hold when camera moves. In this dissertation, a new low rank constraint is introduced on moving cameras. The constraint is applied on the tracked trajectories from the scene in temporal domain, and it is able to handle salient motions from a freely moving cameras.

2.4 Modeling activity features

Human action/activity modeling in video sequences is a hot topic in the communities of computer vision and pattern recognition. Please see [72, 3] for full survey. The interest in the topic is motivated by the promise of many applications. Automatic analysis of videos enable more efficient video searching e.g. finding tackles in soccer matches, handshakes in news footage or typical dance moves in music videos. It is also important for automatic surveillance, e.g. monitoring shopping malls. Another example is to support aging in places for the elderly in smart homes. Interaction applications like human-computer interactions also benefit from the advances in automatic human action analysis.

In recent years, many algorithms have been proposed to improve the performance of action/activity analysis. A lot of research work focuses on finding better image representation and features extracted from the image sequences. Ideally, these should generalize over small variations in person appearance, background, viewpoint and action types. At the same time, the representations must be sufficient rich for robust action analysis. Using local descriptors or patches is a popular way to represent human actions. A video sequence is then represented by a collection of independent patches. Accurate localization and background subtraction are not required. The local representations are somewhat invariant to changes in viewpoint, person appearance and partial occlusions. A variety of features have been studied in recent years. 3D Haar-like features [55, 23] are

used to model pedestrian's movements. Encoded dynamic features are used to describe periodical movements [102]. Space-time interest points are the locations in space and time where sudden changes of movement occur in the video. Laptev and Lindeberg [48] extended the Harris corner detector [33] to 3D. Space-time interest points are those points where the local neighborhood has a significant variation in both the spatial and the temporal domain. Dollár et al. [25] uses dense sampling instead of sparse interest points for feature representation. This method applies Gabor filtering on the spatial and temporal dimensions individually. In addition to intensity and motion cues, Rapantzikos et al. [74] also incorporate color.

After local interest point detection, local descriptors are applied to summarize an image/video patch. The spatial and temporal size of a patch is usually determined by the scale of the interest point. Schuldt et al. [77] calculate patches of normalized derivatives in space and time. Niebles et al. [67] take the same approach but apply smoothing before reducing the dimensionality using PCA. Dollar et al. [25] experiment with both image gradients and optical flow. Please refer to Mikolajczyk et al. [62] for the survey on features.

How to model the relationship among local features is also very important. One way is to build grids over spatial/temporaldomain. Ikizler and Duygulu [41] sample oriented rectangular patches and bin them into a grid. Zhao and Elgammal [116] bin local descriptors around interest points in a histogram with different levels of granularity. Nowozin et al. [68] use a temporal instead of a spatial grid. Another way is to exploit correlations between local descriptors to construct higher-level descriptors. Scovanner et al. [78] construct a word co-occurrence matrix for a reduced codebook size. Liu et al. [53] uses a combination of the space-time features and spin images to represent the correlations of features. Yi and Pavlovic [80] use Isotonic Canonical Correlation Analysis for movement alignment and action anlaysis. These algorithms have been successfully applied to action analysis problems, focusing on single action with one person [77](*hand-waving, running...*). pair-wise action recognition (*answer phone*[49], *horse riding* [54]).

These works do not consider interactions among multiple people. For most of the

surveillance systems in public area, it is also important to identify group activities. Events like fighting or escaping often involve multiple people and their interactions.

Several algorithms for group activity modeling have been proposed in recent years. Different features are used for group activity: human body/body parts [59, 73], optical flow [5] and detecting moving regions [96]. Recently, Zhou *et al.* .[118] and Ni *et al.* *et al.* [66] use trajectory analysis to describe different group activities. However, these algorithms heavily depend on the accuracy of tracking trajectories. Especially when the camera is mounted on buildings and people are severely occluded, these algorithms need human intervention to correct the trajectories, which are not practical in real applications.

Modeling social behaviors of people is an important branch to represent group activity, and it has been widely used in evacuation dynamics, traffic analysis and graphics. Pedestrian behaviors have been studied from a crowd perspective, with macroscopic models for crowd density and velocity. On the other end, microscopic models deal with individual pedestrians. A popular model is the Social Force Model [35]. In the Social Force Model, pedestrians react to energy potentials caused by other pedestrians and static obstacles through a repulsive force, while trying to keep a desired speed and motion direction. Helbing *et al.* in [35] originally introduce it to investigate people movement dynamics. It is also applied to the simulation of crowd behavior [105], virtual reality and studies in computer graphics for creating realistic animations of the crowd [90].

Social behavior analysis has also attracted much attention in the computer vision community. Ali and Shah [4] use the cellular automaton model to track in extremely crowded situations. Antonini *et al.* [6] propose a variant of Discrete Choice Model to build a probability distribution over pedestrian positions in next time step. Scovanner and Tappen [79] learns pedestrians' dynamics and motions as a continuous optimization problem. Pellegrini *et al.* [70] propose a Linear Trajectory Avoidance (LTA) method to track multiple targets. Predictions of velocities are computed by the minimization of energy potentials. Recently, Mehran *et al.* [60] propose a method to model behaviors among a group of people. It represents the group patterns in a local region based on

moving particles. Wu *et al.* [97] uses chaotic invariants of Lagrangian Particle Trajectories to model group behaviors in crowded scenes. They have been successfully used in crowded scene modeling.

Different from the above work, our method is based on the relationship between the current state of a person and his/her reactions. It fully utilizes the information of interaction energy potential and the corresponding people's reactions, which contains comprehensive information to model the behaviors among a group of people.

Chapter 3

Salient Motion Detection

3.1 Introduction

Salient motion detection is an important step in many video analysis systems. It aims to find independent moving objects in a scene and filter out the unimportant area. The idea of saliency detection comes from human visual system, where the first stage of human vision is a fast but simple pre-attentive process. Salient motion detection can be used in many applications, such as the surveillance and monitoring of public facilities like train stations, underground subways or airports, monitoring patients in a hospital environment or other health care facilities, and other similar applications.

Recently, moving camera platforms have increased significantly, like cellular phones, vehicles, and robots. As a larger and larger percentage of video content is produced by moving cameras, the need for foundational algorithms that can isolate interesting areas in such video is becoming increasingly pressing. It is still a very challenging problem to robustly handle diverse types of videos. Here we propose a unified framework for salient motion detection, which can robustly deal with videos from stationary or moving cameras with various number of righd/non-rigid objects.

The proposed method for salient motion detection [19] is based on two sparsity constraints applied on foreground and background levels, *i.e.*, low rank [14] and group sparsity constraints [107]. It is inspired by recently proposed sparsity theories [13, 84]. There are two "sparsity" observations behind our method. *First*, when the scene in a video does not have any foreground moving objects, video motion has a low rank constraint for orthographic cameras [69, 89]. Thus the motion of background points forms a low rank matrix. *Second*, foreground moving objects usually occupy a small portion of the scene. In addition, when a foreground object is projected to pixels on

multiple frames, these pixels are not randomly distributed. They tend to group together as a continuous trajectory. Thus these foreground trajectories usually satisfy the group sparsity constraint.

These two observations provide important information to differentiate independent objects from the scene. Based on them, the video salient motion detection problem is formulated as a matrix decomposition problem. First, the video motion is represented as a matrix on *trajectory* level (*i.e.* each row in the motion matrix is a trajectory of a point). Then it is decomposed into a background matrix and a foreground matrix, where the background matrix is low rank, and the foreground matrix is group sparse. This low rank constraint is able to automatically model background from both stationary and moving cameras, and the group sparsity constraint improves the robustness to noise (see details in Sec. 3.2.2).

Our approach is validated on various types of data, *i.e.*, synthetic data, real video sequences recorded by stationary cameras or moving cameras and/or nonrigid foreground objects. Extensive experiments also show that our method compares favorably to the recent state-of-the-art methods.

The main contribution of the proposed approach is a new model using low rank and group sparsity constraints to differentiate foreground and background motions. This approach has three merits:

1. The low rank constraint is able to handle both static and moving cameras. It allows us to develop a unified algorithm to handle both stationary cameras and moving cameras.

2. The group sparsity constraint leverages the information of the points on the consecutive frames. By using the group information, rather than individual points, it makes the algorithm robust to random noise;

3. It is relatively not sensitive to parameter settings. This is of significant practical importance: the same parameter works well for all tested videos.

In the remainder of this chapter, the major method will be presented in Section 3.2, including the math formulation using sparse representation for background subtraction

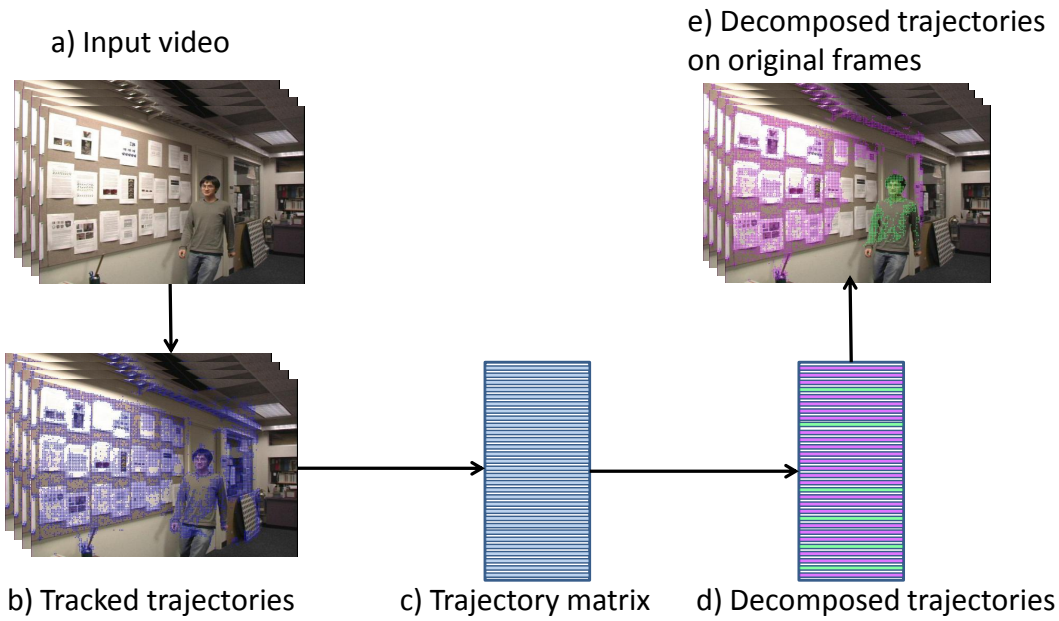

Figure 3.1: The framework. Our method takes a raw video sequence as input, then generate the dense point trajectories using a dense point tracking tool [86]. The dense point trajectories form a two dimensional data matrix. Our algorithm is then applied to decompose the matrix into foreground and background using the proposed low rank and group sparsity based model. The final labeled trajectories are visualized in the original frames.

in Section 3.2.2; the optimization framework in Section 3.2.3, and pixel-level labeling in Section 3.2.4. Section 3.3 shows the experimental results of our methods and comparisons with state-of-art methods on both synthetic data and real videos. Section 3.4 concludes this chapter.

3.2 Methodology

3.2.1 The Overview

Our salient motion detection algorithm takes a raw video sequence as input, then return a set of trajectories that are labeled as foreground or background motions. Figure 3.1 shows our framework. First, A dense set of points is tracked over all frames. We use an off-the-shelf dense point tracker [86] to produce the trajectories. These dense point trajectories form a two dimensional data matrix. Then a low rank and group sparsity based model is performed to decompose the data matrix into foreground and

background.

After the trajectory level separation, the trajectories can be further used to label the frames into binary foreground and background. motion segments are generated using optical flow [51] and graph cuts [9]. Then the color and motion information gathered from the recognized trajectories builds statistics to classify each motion segment as foreground or background.

3.2.2 Low Rank and Group Sparsity based Model

Notations: Given a video sequence, k points are tracked over l frames. Each trajectory is represented as

$$p_i = [x_{1i}, y_{1i}, x_{2i}, y_{2i}, ...x_{li}, y_{li}] \in \mathbb{R}^{1\times 2l},$$

where x and y denote the 2D coordinates in each frame. The collection of k trajectories is represented as a $k \times 2l$ matrix,

$$\phi = [p_1^T, p_2^T, ..., p_l^T]^T, \quad \phi \in \mathbb{R}^{k\times 2l}.$$

In a video with moving foreground objects, a subset of k trajectories comes from the foreground, and the rest belongs to the background. Our goal is to decompose tracked k trajectories into two parts: m background trajectories and n foreground trajectories. If we already know exactly which trajectories belong to the background, then foreground objects can be easily obtained by subtracting them from k trajectories, and vice versa. In other words, ϕ can be decomposed as:

$$\phi = B + F, \tag{3.1}$$

where $B \in \mathbb{R}^{k\times 2l}$ and $F \in \mathbb{R}^{k\times 2l}$ denote matrices of background and foreground trajectories, respectively. In the ideal case, the decomposed foreground matrix F consists of n rows of foreground trajectories and m rows of flat zeros, while B has m rows of background trajectories and n rows of zeros.

Eq. 3.1 is a severely under-constrained problem. It is difficult to find B and F

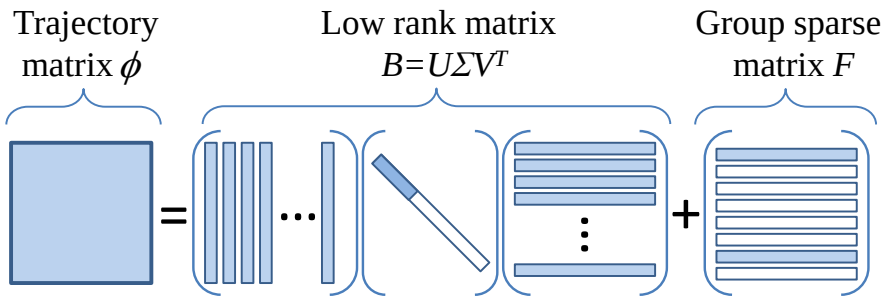

Figure 3.2: Illustration of our model. Trajectory matrix ϕ is decomposed into a background matrix B and a foreground matrix F. B is a low rank matrix, which only has a few nonzero eigenvalues (*i.e.* the diagonal elements of Σ in SVD); F is a group sparse matrix. Elements in one row belongs to the same group (either foreground or background), since they lie on one trajectory. The foreground rows are sparse comparing to all rows. White color denotes zero values, while blue color denotes nonzero values.

without any prior information. In our method, we incorporate two effective priors to robustly solve this problem, *i.e.*, the low rank constraint for the background trajectories and the group sparsity constraint for the foreground trajectories.

Low rank constraint for the background. In a 3D structured scene without any moving foreground object, video motion solely depends on the scene and the motion of the camera. Our background modeling is inspired from the fact that B can be factored as a $k \times 3$ structure matrix of 3D points and a $3 \times 2l$ orthogonal matrix [89]. Thus the background matrix is a low rank matrix with rank value at most 3. This leads us to build a low rank constraint model for the background matrix B:

$$rank(B) \leq 3, \tag{3.2}$$

Another constraint has been used in the previous research work using RANSAC based method [81]. This work assumes that the background matrix is of rank three: $rank(B) = 3$. This is a very strict constraint for the problem. We refer the above two types of constraints as the General Rank model (GR) and the Fixed Rank model (FR). Our GR model is more general and handles more situations. A rank-3 matrix models 3D scenes under moving cameras; a rank-2 matrix models a 2D scene or 3D scene under stationary cameras; a rank-1 matrix is a degenerated case when scene only has one point. The usage of GR model allows us to develop a unified framework to

handle both stationary cameras and moving cameras.

The experiment section (Sec. 3.3.2) provides more analysis on the effectiveness of the GR model when handling diverse types videos. We also compare the performance of our method using GR model and RANSAC based method[81] in Sec. 3.3.2 (see Tab. 3.2)

Group sparsity constraint for the foreground. Foreground moving objects, in general, occupy a small portion of the scene. This observation motivates us to use another important prior, *i.e.*, the number of foreground trajectories should be smaller than a certain ratio of all trajectories,

$$m < \alpha k, \tag{3.3}$$

where α controls the sparsity of foreground trajectories.

Another important observation is that each row in ϕ represents one trajectory. Thus the entries in ϕ are not randomly distributed. They are spatially clustered within each row. If one entry of the ith row ϕ_i belongs to the foreground, the whole ϕ_i is also in the foreground. This observation makes the foreground trajectory matrix F satisfy the group sparsity constraint:

$$\|F\|_{2,0} < \alpha k, \tag{3.4}$$

where $\|\cdot\|_{2,0}$ is the mixture of both L_2 and L_0 norm. The L_2 norm constraint is applied to each group separately (*i.e.*, each row of F). It ensures that all elements in the same row are either zero or nonzero at the same time. The L_0 norm constraint is applied to count the nonzero groups/rows of F. It guarantees that only a sparse number of rows are nonzero. Thus this group sparsity constraint not only ensures that the foreground objects are spatially sparse, but also guarantees that each trajectory is treated as one unit.

The traditional sparsity constraint, L_0 norm, has been intensively studied in recent years. However, it does not work well for this problem compared to the group sparsity one. L_0 norm treats each element of F independently. It does not consider any neighborhood information. Thus it is possible that points from the same trajectory are classified into two classes. In the experiment section (Sec. 3.3.1), we discuss the

advantage of group sparsity constraint over sparsity constraint through synthetic data analysis, and also show that this constraint improves the robustness of our model.

Based on the low rank and group sparsity constraints, we formulate our objective function as:

$$\left(\hat{B}, \hat{F}\right) = \arg\min_{B,F} \left(\| \phi - B - F \|_F^2 \right),$$
$$\text{s.t. } rank(B) \leq 3, \ \| F \|_{2,0} < \alpha k, \tag{3.5}$$

where $\| \cdot \|_F$ is the Frobenius norm. This model leads to a good separation of foreground and background trajectories. Figure 3.2 illustrates our model.

Eq. 3.5 only has one parameter α, which controls the sparsity of the foreground trajectories. In general, user-tuning parameter is a key issue for a good model. It is preferable that the parameters are easy to tune and not sensitive to different datasets. In the experiment section (Sec. 3.3), we show that the model is relatively insensitive to parameter selection.

Low rank constraints and Robust PCA have been recently used to solve vision problems [115, 14], including background subtraction at the pixel level [14]. It assumes that the stationary scenes satisfy a low rank constraint. However, this assumption does not hold when camera moves. Furthermore, that formulation does not consider any group information, which is an important constraint to make sure neighbor elements are considered together.

3.2.3 Optimization Framework

This subsection discusses how to effectively solve Eq. 3.5. The first challenge is that it is not a convex problem, because of the nonconvexity of the low rank constraint and the group sparsity constraint. Furthermore, we also need to simultaneously recover matrix B and F, which is generally a Chicken-and-Egg problem.

In our framework, alternating optimization and greedy methods are employed to solve this problem. We first focus on the fixed rank problem (*i.e.*, rank equals to 3), and then will discuss how to deal with the more general constraint of $rank \leq 3$.

Eq. 3.5 is divided into two subproblems with unknown B or F, and solved by using two steps iteratively:

Step 1: Fix B, and update F. The subproblem is:

$$\left(\hat{F}\right) = \arg\min_{F} \left(\| \phi' - F \|_F^2 \right), \text{ s.t. } \|F\|_{2,0} < \alpha k, \tag{3.6}$$

where $\phi' = \phi - B$.

Step 2: Fix F, and update B. The subproblem is:

$$\left(\hat{B}\right) = \arg\min_{B} \left(\| \phi'' - B \|_F^2 \right), \text{ s.t. } rank(B) = 3, \tag{3.7}$$

where $\phi'' = \phi - F$.

To initialize this optimization framework, we simply choose $B_{init} = \phi$, and $F_{init} = \mathbf{0}$. Greedy methods are used to solve both subproblems. To solve Eq. 3.6, we compute $\|F_i\|_2, i \in 1, 2, ..., k$, which represents the L_2 norm of each row. Then the αk rows with largest values are preserved, while the rest rows are set to zero. This is the estimated F in the first step. In the second step, ϕ'' is computed as per newly-updated F. To solve Eq. 3.7. Singular value decomposition (SVD) is applied on ϕ''. Then three eigenvectors with largest eigenvalues are used to reconstruct B. Two steps are alternatively employed until a stable solution of $\hat{B}$ is found. Then $\hat{F}$ is computed as $\phi - \hat{B}$. The reason of updating $\hat{F}$ after all iterations is that the greedy method of solving Eq. 3.6 discovers exact αk number of foreground trajectories, which may not be the real foreground number. On the contrary, B can be always well estimated, since a subset of unknown number of background trajectories is able to have a good estimation of background subspace. Thus we finalize $\hat{F}$ by $\phi - \hat{B}$. Since the whole framework is based on greedy algorithms, it does not guarantee a global minimum. In our experiments, however, it is able to generate reliable and stable results.

The above-mentioned method solves the fixed rank problem, but the rank value in the background problem usually cannot be pre-determined. To handle this undetermined rank issue, we propose a multiple rank iteration method. First, B and F are

initialized as $B_{init}^{(0)} = \phi$ and $F_{init}^{(0)} = \mathbf{0}^{k \times 2l}$. Then the fixed rank optimization procedure is performed on each specific rank starting from 1 to 3. The output of the current fixed rank procedure is fed to the next rank as its initialization. We obtain the final result $B^{(3)}$ and $F^{(3)}$ in the rank-3 iteration. Algorithm 1 shows this optimization framework in detail.

Given a data matrix of $k \times 2l$ with k trajectories over l frames, the major calculation is $O(kl^2 + l^3)$ for SVD on each iteration. Convergence of the fixed rank problem is achieved 6.7 iterations on average. Since we use a few frames to construct the trajectory matrix ($10 - 30$ frames in our framework), the value of l is much lower than k, the total trajectory number. The overall time complexity is $O(kl^2 + k^3)$, where $l << k$.

To explain why our framework works for the general rank problem, we discuss two examples. First, if the rank of B is 3 (*i.e.*, moving cameras), then this framework discovers an optimal solution in the third iteration, *i.e.*, using rank-3 model. The reason is that the first two iterations, *i.e.* the rank-1 and rank-2 models, cannot find the correct solution as they are using the wrong rank constraints. Second, if the rank of the matrix is 2 (*i.e.*, stationary cameras), then this framework obtains stable solution in the second iteration. This solution will not be affected in the rank-3 iteration. The reason is that the greedy method is used to solve Eq. 3.7. When selecting the eigenvectors with three largest eigenvalues, one of them is simply flat zero. Thus B does not change, and the solution is the same in this iteration. Note that low rank problems can also be solved using convex relaxation on the constraint problem [14]. However, our greedy method on unconstrained problem is better than convex relaxation in this application. Convex relaxation is not able to make use of the specific rank value constraint (≤ 3 in our case). The convex relaxation uses λ to implicitly constrain the rank level, which is hard to constrain a matrix to be lower than a specific rank value.

3.2.4 Pixel Level Labeling

The labeled trajectories from the previous step are then used to label each frame at the pixel level (*i.e.* return a binary mask for a frame). In this step, each frame is treated as an individual labeling task. First, the optical flow [51] is calculated between

Algorithm 1 Optimization framework to solve equation (3.5)

input: Trajectory matrix $\phi \in \mathbb{R}^{k\times 2l}$, sparsity weight α.
initialization: $B_{init}^{(0)} = \phi$, $F_{init}^{(0)} = \mathbf{0}^{k\times 2l}$.
optimization:
for $r_c = 1$ to 3 **do**
  $B_{init}^{(r_c)} = B^{(r_c-1)}$, $F_{init}^{(r_c)} = F^{(r_c-1)}$
  repeat
    update F:
    $(\hat{F}) = arg\min(\| \phi' - F \|_F^2)$, s.t. $\|F\|_{2,0} < \alpha k$
    $F_g = \|\phi'\|_2$
    Keep αK rows from ϕ' with the largest value in F_g, and set the rest to zeros
    update B:
    $(\hat{B}) = arg\min(\| \phi'' - B \|_F^2)$, s.t. $rank(B) = r_c$
    $[U, \Sigma, V] = SVD(\phi'')$
    $V_r = V[1:r]$
    $\hat{B} = \phi V_r V_r^T$
  until halting criterion true.
  $\hat{F} = \phi - \hat{B}$
end for
output: $\hat{B}, \hat{F}$.

two consecutive frames. Then motion segments are computed using graph cuts [9] on optical flow. The advantage of using optical flow instead of color for graph cuts is that it is able to find independent motion regions. Thus each motion segment is a unit from a rigid part on a moving object. After collecting the motion segments, the goal is to label each motion segment s as f or b, where f and b denotes the label of foreground and background.

There are two steps to label the segments. First, segments with high confidence belonging to f and b are selected. Second, a statistical model is built based on those segments. This model is used to label segments with low confidence.

The confidence of a segment is determined by counting the number of labeled f and b trajectories. If a segment only has one type of trajectories (*i.e.* either f or b), then it is a high confidence segment. The low confidence segments are those having no trajectories or containing both f and b ones. Thus it is hard to determine the label by simply counting trajectories.

To predict the labels of these low confidence segments, a statistical model is built for f and b based on high confidence ones. First, 20% pixels are uniformly sampled

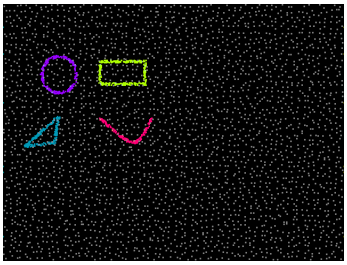

Figure 3.3: Synthetic data. The foreground is four moving shapes, and the background is a randomized grid (Each point is enlarged to have a better view).

on segments with high confidence. Each sampled pixel w is represented by color in hue-saturation space (h, s), optical flow (u, v) and position on the frame (x, y). The reason we use sampled points to build the model instead of the original trajectories is that the sparse trajectories may not cover enough information (*i.e.*, color and positions) on the motion unit. Uniform sampling covering the whole segment is able to build a richer model.

The probability of a segment s_i belonging to f or b is then evaluated using a kernel density function:

$$P(s_i|c) = \frac{1}{N \cdot |s_i|} \sum_{i=1}^{N} \sum_{j \in s_i} \kappa(e_j - w_i), \quad c \in \{f, b\} \tag{3.8}$$

where $\kappa(\cdot)$ is the Normal kernel, N is the total number of sampled pixels, and $|s_i|$ is the pixel number of s_i. For every pixel j lying on s_i, e_j denotes the vector containing color, optical flow and position. This formula defines the probability a segment belongs to the label of f or b. For a segment with low confidence, its label is assigned to f if $P(s|f) > P(s|b)$ and vice versa.

3.3 Experiments

To evaluate the performance of our algorithm, we conduct experiments on different data sources: synthetic data, real videos from both moving and stationary cameras. Its performance on the trajectory separation is evaluated by F-Measure, which is the harmonic mean of recall and precision. This is a standard measurement for many

state-of-art algorithm [11, 50]:

$$F = \frac{2 \cdot recall \cdot precision}{recall + precision}, \tag{3.9}$$

where recall is the ratio of the number of correctly classified foreground trajectories to the number of foreground trajectories in ground truth, and precision is the ratio of the number of correctly classified foreground trajectories to the number of trajectories classified as foreground. Since our major contribution is to separate background/foreground motions on trajectory level, thus our comparisons and analysis mainly focus on the first step: trajectory level labeling.

3.3.1 Evaluations on Synthetic Data

Experimental settings. A grid of background points and four shapes are generated (Figure 3.3). The homogeneous representation of each point is $(X, Y, Z, 1)$ as its position in the 3D world. Then the projected points (x, y) are obtained in a 2D image by

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = C \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

where C is a 3×4 camera projection matrix. The depth value Z in the 3D world for foreground shapes and background grid is 10 ± 5 and 20 ± 10, respectively. The foreground shapes move and the background grid stays still. Changing the camera projection matrix C simulates the camera movement and generates projected images. Two advantages of our method are demonstrated, *i.e.*, the robustness of the group sparsity constraint, and the insensitivity to parameter settings.

Group sparsity constraint versus sparsity constraint. We first compare the performance between the group sparsity constraint ($L_{2,0}$ norm) and traditional sparsity constraint (L_0 norm). The sparsity constraint aims to find a sparse set of nonzero elements, which is $\|F\|_0 < \alpha k \times 2l$ in our problem. $k \times 2l$ denotes the total number

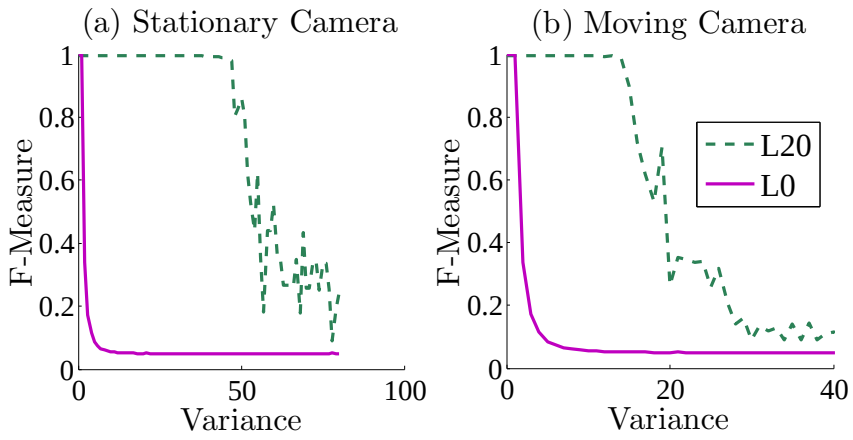

Figure 3.4: Comparison between group sparsity constraint ($L_{2,0}$ norm) and sparsity (L_0 norm) constraint. The left is synthetic data simulated with a stationary camera, and the right is simulated with a moving camera.

of nonzero elements. It is equivalent to the total number of nonzero elements in the group sparsity constraint. Note that the formulation with this L_0 sparsity constraint is similar to Robust PCA method [14]. The difference is that we use it at the trajectory level instead of pixel level.

Two sets of data are generated to evaluate the performance. One is simulated with a stationary camera, and the other is from a moving one. Foreground keeps moving in the whole video sequence. Random noise with variance v is added to both the foreground moving trajectories and camera projection matrix C. The performance is shown in Figure 3.4. In the noiseless case (*i.e.* $v = 0$), the motion pattern from the foreground is distinct from the background in the whole video sequence. Thus each element on the foreground trajectories is different from the background element. Sparsity constraint produces the same perfect result as group sparsity constraint. When v goes up, the distinction of elements between foreground and background goes down. Thus some elements from the foreground may be recognized as background. On the contrary, the group sparsity constraint connects the elements in the neighboring frames. It treats the elements on one trajectory as one unit. Even some elements on this trajectories are similar to the background, the distinction along the whole trajectory is still large from the background. As shown in the Figure 3.4, using the group sparsity constraint is more robust than using the sparsity constraint when variance increases. In Sec. 3.3.2,

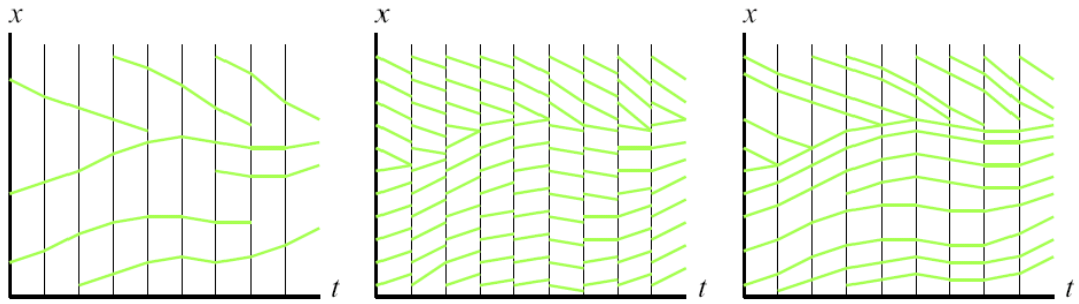

Figure 3.5: Demonstration of dense point tracking technique. Left: feature tracking; middle: optical flow; right: dense point tracking. Each diagram represents point correspondences between frames of a hypothetical sequence. Feature tracking is long-range but sparse. Optical Flow is dense but short-range. The dense point tracking is dense and long-range.

we will further show the results in real videos. The visualized $\hat{B}$ and $\hat{F}$ matrices will be also shown.

3.3.2 Evaluations on Real Videos

Experimental settings. We test our algorithm on publicly available videos from various sources. One video source is provided by Sand and Teller [76] (refer this as ST sequences). ST sequences are recorded with hand held cameras, both indoors and outdoors, containing a variety of non-rigidly deforming objects (hands, faces and bodies). It contains three video sequences ("*VHand*", "*VCars*", "*VPerson*"). They are high resolution images with large frame-to-frame motion and significant parallax. The average motion of a background point is 133.90 pixels for "*VHand*", 67.10 pixels for "*VCars*" and 90.93 pixels for "*VPerson*". Another source of videos is provided from Hopkins 155 dataset [91], which has two or three motions from indoor and outdoor scenes. These sequences contain degenerate and non-degenerate motions, independent and partially dependent motions, articulated and non-rigid motions. We also test our algorithm on some typical videos for traditional background subtraction: "*truck*" (stationary cameras). The trajectories in these sequences were created using an off-the-shelf dense particle tracker [86]. We will first evaluate the trajectory based labeling and then show the performance on the the pixel level labeling results.

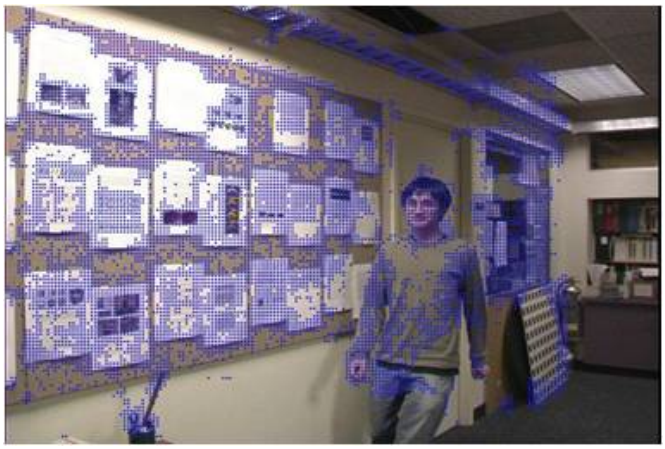

Figure 3.6: Dense point tracking technique (Sundaram et al. [86]) is used to generate long-term trajectories. The figure shows the tracked points. They are sampled by every 4 pixels.

Dense point tracking Given an input video, we first need to extract the trajectories of the pixels. We use a set of trajectories of dense points as the input, referred as dense point trajectories. Dense point tracking techniques become popular in very recent years [76, 86]. It combines two approaches: feature tracking [56] and optical flow [51, 10]. Feature tracking follows a sparse set of salient image points over many frames, whereas optical flow estimates a dense motion field from one frame to the next. Dense point tracking aims to produce motion estimates that are both spatially dense and temporally long-range. For an image point, it needs to know where the corresponding scene point appears in all other video frames (until the point leaves the filed of view or becomes occluded).

The current technique of dense point tracing makes it possible to generate long-term trajectories with a rich coverage. We use an off-the-shelf dense point tracking tool (Sundaram et al. [86]) for all videos. To initialize the tracking process, it first samples the pixels on the first frame. In our experiment setting, we sample pixels by every 4 pixels. Smaller number (e.g. sampling by every 2 pixels) generates denser trajectory set. But this is not necessary, as our algorithm can label each pixel as background/foreground in the second stage. Figure 3.6 shows an example.

To obtain the ground truth of these videos, we manually label each frame of those videos into a binary mask. The label tool we use here is the Human-Assisted Motion

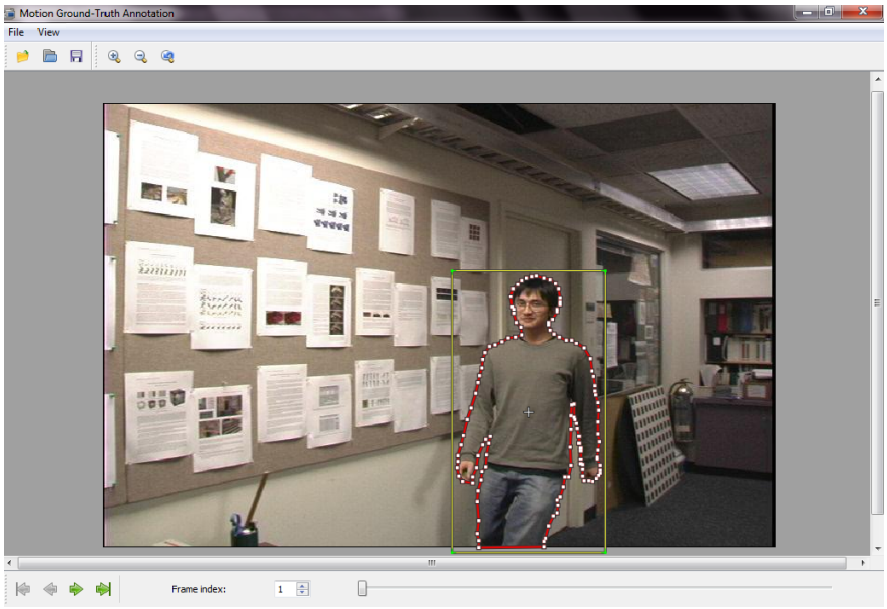

Figure 3.7: The Human-Assisted Motion Annotation tool by Liu et al [52]. For each frame, we generate binary mask as the ground truth.

Annotation tool by Liu et al. [52]. The advantage of this tool is that it has a built-in motion tracking algorithm. It can track the moving objects. By labeling the pixels on the object contour, this tool is able to generate a binary mask for the object, and then automatically track the contour to the next frame. When the automatic tracking drifts, it can be fixed by manual correction. This speeds up the whole manual labeling process. Figure 3.7 shows an example of our labeling tool interface.

Handling stationary and moving cameras. We first demonstrate that our approach handles both stationary cameras and moving cameras automatically in a unified framework, by using the General Rank constraint (GR) instead of the Fixed Rank constraint (FR). We use two videos to show the difference. One is "*VHand*" from a moving camera ($rank(B) = 3$), and the other is "*truck*" captured by stationary camera ($rank(B) = 2$). We use the distribution of L_2 norms of estimated foreground trajectories (*i.e.*, $\|\hat{F}_i\|_2, i \in 1, 2, ..., k$) to show how well background and foreground is separated in our model. For a good separation result, F should be well estimated. Thus $\|\hat{F}_i\|_2$ is large for foreground trajectories and small for background ones. In other words, its distribution has an obvious difference between the foreground region and the background region (see examples in Figure 3.8).

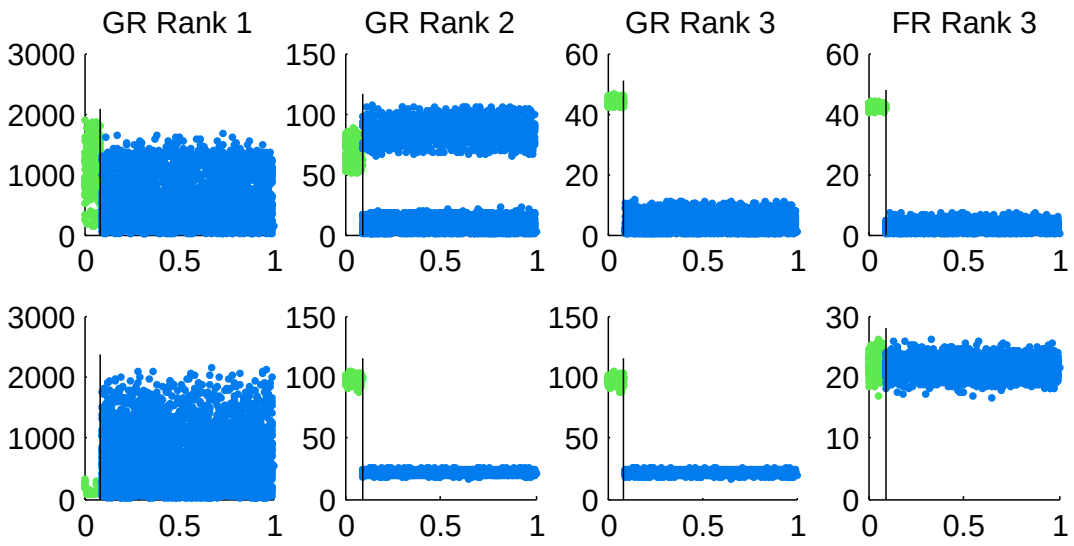

Figure 3.8: $\|\hat{F}_i\|_2$ distribution of the GR and the FR model. Green means foreground and blue means background. Separation means good result. Top row is from moving camera, and bottom row is from stationary camera. The first three columns are GR-1, GR-2 and GR-3, and the forth column is FR.

We use GR-$i, i \in 1, 2, 3$ to denote the optimization iteration on each rank value. $\|\hat{F}_i\|_2$ of each specific rank iteration is plotted in Figure 3.8. The GR method works for both cases. When the rank of B is 3 (the first row of Figure 3.8), the FR model also finds a good solution, since rank-3 perfectly fits the FR model. However, the FR constraint fails when the rank of B is 2, where the distribution of $\|\hat{F}_i\|_2$ between B and F are mixed together. On the other hand, GR-2 can handle this well, since the data perfectly fits the constraint. On GR-3 stage, it uses the result from GR-2 as the initialization, thus the result on GR-3 still holds. The figure shows that the distribution of $\|\hat{F}_i\|_2$ from the two parts has been clearly separated in the third column of the bottom row. This experiment demonstrates that the GR model can handle more situations than the FR model. Since in real applications it is hard to know the specific rank value in advance, the GR model provides a more flexible way to find the right solution.

Performance evaluation on trajectory labeling. We discussed the parameter sensitivity between our method and RANSAC based method (RANSAC-b) in Sec. 3.3.1.

	RANSAC Best	RANSAC Global	Our Method
VPerson	0.923 ± 0.123 $th=26, p=90\%$	0.786 ± 0.221 $th=21, p=70\%$	0.981 $\alpha=0.3$
VHand	0.976 ± 0.006 $th=16, p=70\%$	0.952 ± 0.168 $th=21, p=70\%$	0.987 $\alpha=0.3$
VCars	0.995 ± 0.003 $th=11, p=90\%$	0.867 ± 0.298 $th=21, p=70\%$	0.993 $\alpha=0.3$
cars2	0.893 ± 0.174 $th=21, p=90\%$	0.750 ± 0.291 $th=21, p=70\%$	0.976 $\alpha=0.3$
cars5	0.992 ± 0.017 $th=16, p=70\%$	0.985 ± 0.127 $th=21, p=70\%$	0.990 $\alpha=0.3$
people1	0.968 ± 0.097 $th=11, p=90\%$	0.932 ± 0.186 $th=21, p=70\%$	0.955 $\alpha=0.3$
truck	0.562 ± 0.317 $th=21, p=70\%$	0.351 ± 0.537 $th=21, p=70\%$	0.975 $\alpha=0.3$

Table 3.1: Quantitative results on ST video sequences (the average F-Measure with its standard deviation, and the parameter settings). Our method provides a deterministic solution. Thus the standard deviation is zero. *RANSAC Best* uses the optimal parameters for each video, while *RANSAC Global* uses a uniform setting which results in the best average performance.

We further evaluate the parameter sensitivity in real videos. Table 3.1 shows the quantitative results on RANSAC-b and our method. As RANSAC based method is sensitive to parameters, we show two types of results for RANSAC-b. One is from different optimal parameter settings for each individual video sequence (refer as *RANSAC-b Best*), and the other is from a uniform setting resulting in the best average performance for all video sequences (refer as *RANSAC-b Global*). *RANSAC-b Best* has reached very good performance, but the parameter settings for these video sequences are different (Table 3.1). Since many practical applications need a uniform setting to handle multiple videos, *RANSAC-b Global* is a fair way to compare, whose performance is relatively worse. In general, our algorithm outperforms them in terms of average F-Measure, with one parameter $\alpha = 0.3$ for all videos.

We also compare our method with three state-of-art algorithms: background subtraction algorihm using RANSAC (RANSAC-b) [81], Generalized GPCA (GPCA) [92],

	RANSAC-b	*GPCA*	*LSA*	*RANSAC-m*	*Std. Sparse*	*Ours*
VPerson	0.786	0.648	0.912	0.656	0.616	0.981
VHand	0.952	0.932	0.909	0.930	0.132	0.987
VCars	0.867	0.316	0.145	0.276	0.706	0.993
cars2	0.750	0.773	0.568	0.958	0.625	0.976
cars5	0.985	0.376	0.054	0.637	0.779	0.990
people1	0.932	0.564	0.087	0.743	0.662	0.955
truck	0.351	0.368	0.140	0.363	0.794	0.975

Table 3.2: Quantitative evaluation at trajectory level labeling. The numbers reported here are the average value on multiple runs. The variation is not reported here. Please refer to Table 3.1 for more information.

Local Subspace Affinity (LSA) [99] and motion segmentation using RANSAC (RANSAC-m) [91]. RANSAC-b is a state-of-art algorithm used for background subtraction problem [81]. GPCA, LSA and RANSAC-m are motion segmentation algorithms using subspace analysis for trajectories. The code is available with the Hopkins dataset. When testing these methods, we use the same trajectories as for our own method. The three motion segmentation algorithms ask for the number of regions to be given in advance. We provide the correct number of segments n, whereas our method does not need that. Motion segmentation methods separate trajectories into n segments. Here we treat the segment with the largest trajectory number as the background and rest as the foreground. Since LSA method runs very slow when using trajectories more than 5000, we randomly sample 5000 trajectories for each test video.

For *RANSAC-b* method, two major parameters influence the performance: projection error threshold th and consensus percentage p. Inappropriate selection of parameters may result in failure of finding the correct result. In addition, as *RANSAC-b* randomly selects three trajectories in each round, it may end up with finding a subspace spanned by part of foreground and background. The result it generates is not stable. Running the algorithm multiple times may give different separation of background and foreground, which is undesirable. In order to have a fair comparison with it, we grid search the best parameter set over all teste videos and report the performance under

the optimal parameters.

The quantitative results are shown in Table 3.2. The results on those video sequences are also demonstrated here (VCars: Figure 3.9, cars2: Figure 3.10, cars5: Figure 3.11, people1: Figure 3.12, VHand: Figure 3.13, truck: Figure 3.14, VPerson: Figure 3.15).

Our method works well for these videos. Take "*cars5*" for example. *GPCA* and *LSA* misclassify some trajectories. *RANSAC-b* randomly selects three trajectories to build the scene motion. On this frame, the three random trajectories all lie in the middle region. The background model built from these 3 trajectories do not cover the left and right region of the scene, thus the left and right regions are misclassified as foreground. *RANSAC-m* produces similar behavior to *RANSAC-b*. *Std. sparse* method does not have any group constraint in the consecutive frames, thus some trajectories are classified as foreground in one frame, and classified as background in the next frame. Note that the quantitative results are obtained by averaging on all frames over 50 iterations. Figure 3.11 only shows performance on one frame, which may not reflect the overall performance shown in Tab. 3.2.

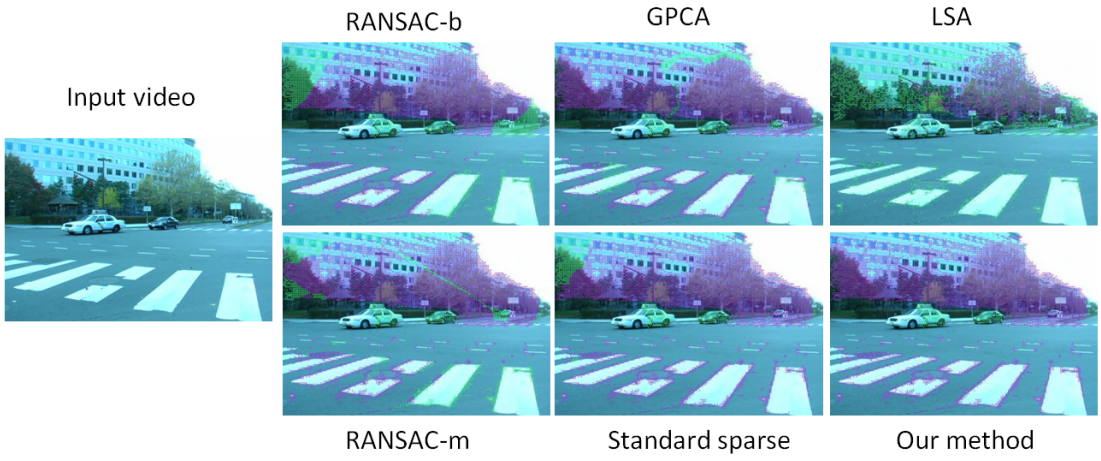

Figure 3.9: Performance comparison on video sequence "VCars"

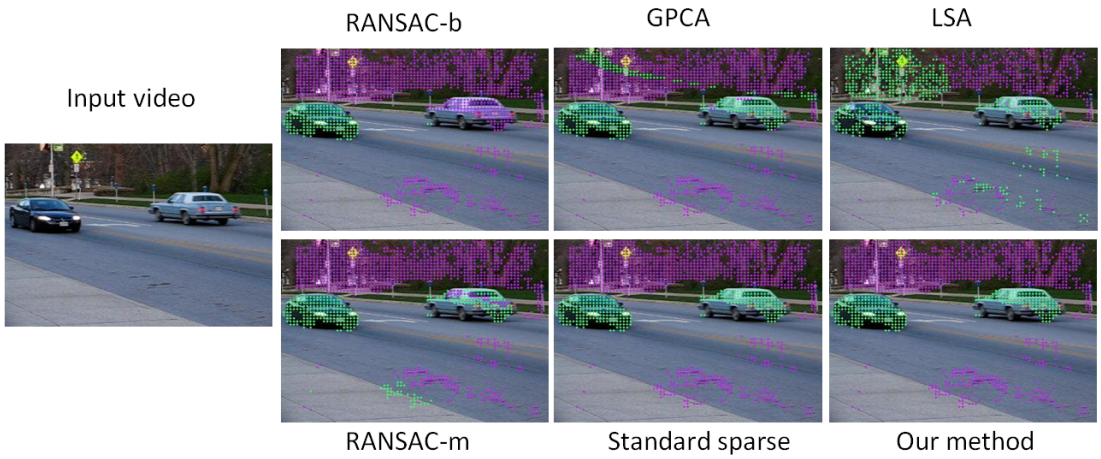

Figure 3.10: Performance comparison on video sequence "cars2"

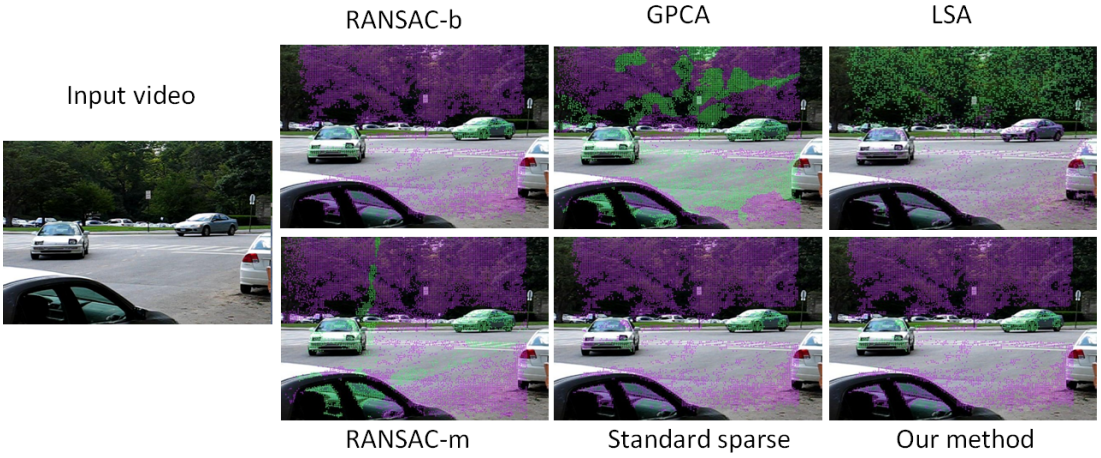

Figure 3.11: Performance comparison on video sequence "cars5"

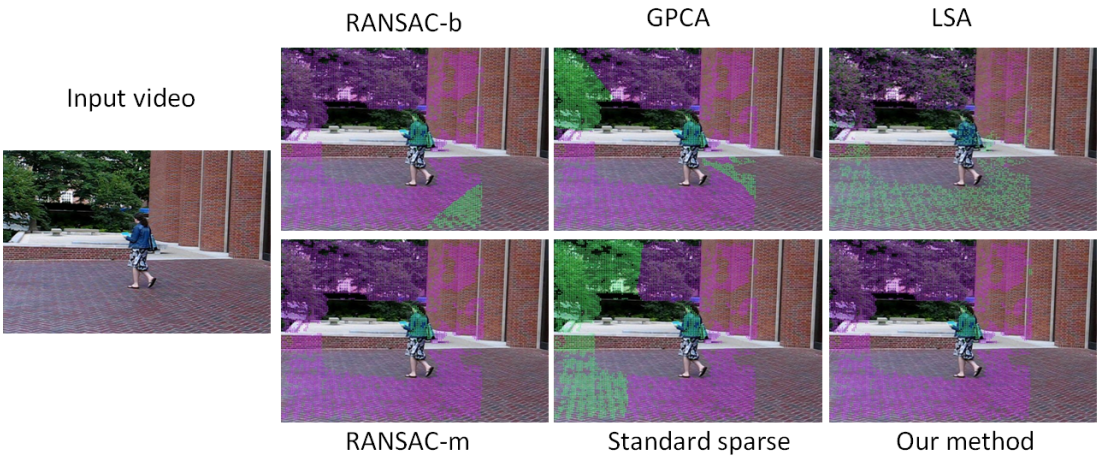

Figure 3.12: Performance comparison on video sequence "people1"

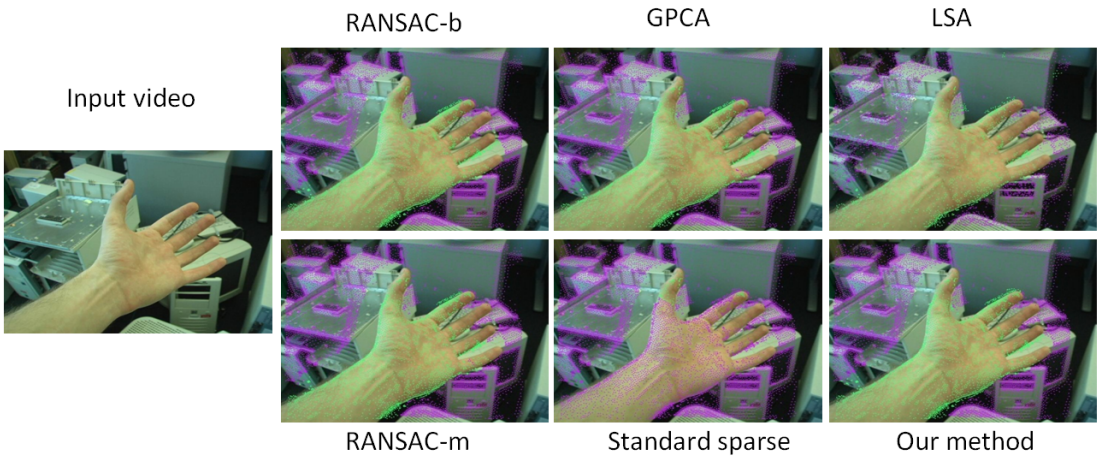

Figure 3.13: Performance comparison on video sequence "VHand"

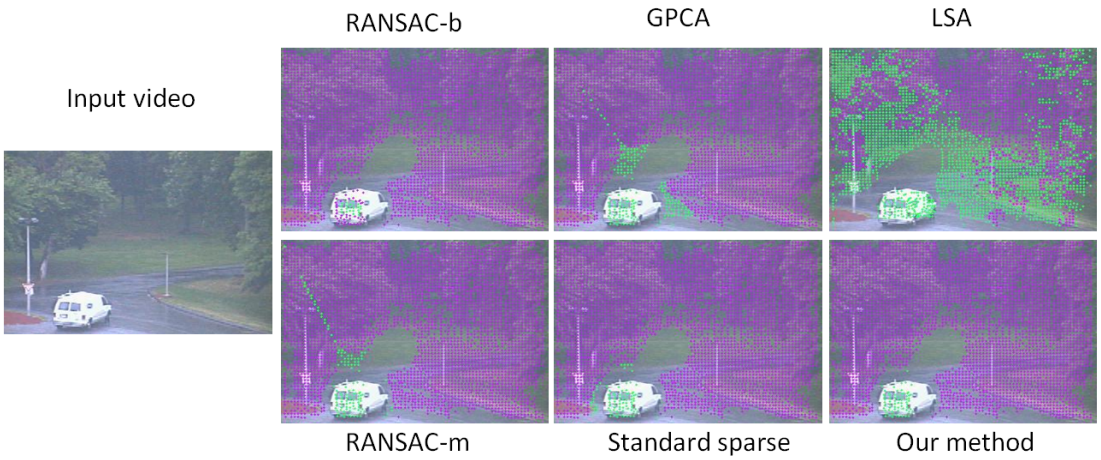

Figure 3.14: Performance comparison on video sequence "truck"

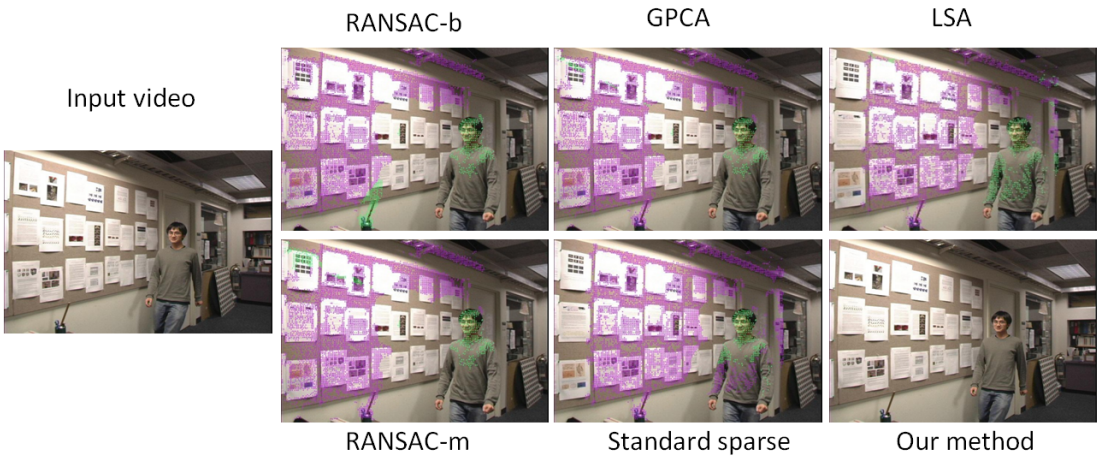

Figure 3.15: Performance comparison on video sequence "VPerson"

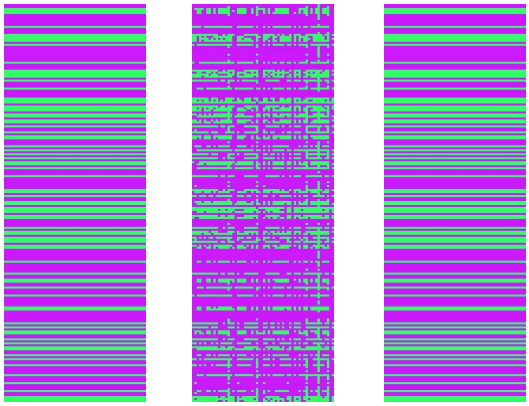

Figure 3.16: Visualization of the computed $\hat{B}$ and $\hat{F}$ matrices. Left: ground truth; middle: the result from L_0; right: the result from our method. As the computed $\hat{B}$ and $\hat{F}$ are exactly complimentary to each other, we show them in one matrix and use different color to denote the elements in $\hat{B}$ and $\hat{F}$. Green color shows the non-zero elements in $\hat{F}$ denoting the foreground elements; purple color shows the non-zero elements in $\hat{B}$ denoting the background elements. The size of the matrices is 1644 by 60. To have the best visualization, all foreground trajectories and 20% randomly selected background trajectories and shown here. They are also re-scaled to fit the space. Each row denotes one trajectory in the video, where each trajectory contains the x,y positions over 30 frames. In our method, the trajectories are classified either foreground or background. In L_0 methods, the discovered foreground elements scatter over the whole matrix.

Group sparsity constraint versus sparsity constraint in real videos. Our method works better than the regular sparsity method (refer to L_0 in this dissertation). Our method finds the correct region (on the person) as the foreground, while the L_0 finds not only the region on the person, but also part of the scene as the foreground. As discussed in Sec. 3.3.1, our method uses the group information to make it more robust to noise. On some frames in this "*VPerson*" video, the foreground and background motions are very similar. L_0 cannot differentiate the motions on such frames, since it is hard to classify the motions on one frame without considering the neighboring frames. Our method treats all frames on the trajectory as a whole unit. Even when the motions between foreground and background on some frames are not easy to differentiate, the whole foreground trajectory can still stand out of the scene motions. The visualization on the computed $\hat{B}$ and $\hat{F}$ matrices for "*VPerson*" video in Figure 3.16 also confirms our assumption. Our method returns a whole trajectory as either foreground or background. The foreground regions calculated from the L_0 method distributes the whole matrix.

It is able to correctly locate part of the foreground motions, but it also finds some elements from the background as the foreground.

RANSAC randomly selects three trajectories to build the scene motion. In this figure, the three random trajectories all lie in the left region. Since the video contains 3D structure with large depth, the motion between the left side of the scene and the right side is different. In RANSAC method, the scene motion model based on the 3 trajectories from the left side does not cover the right side. Thus it only finds the background on the left side of the region and classify the right side of the scene as foreground.

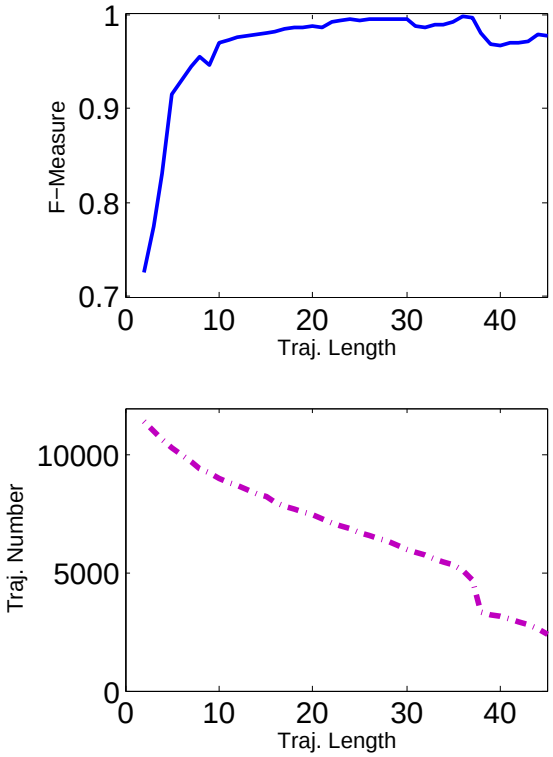

Figure 3.17: The influence of trajectory length l. Left: the performance change under l; right: the trajectory number change under l. When l increases, the performance increases as well, while the trajectory number drops constantly. A good tradeoff occurs around $l = 10$.

The influence of trajectory length. In our method, the trajectories are tracked over l frames. Longer trajectory length carries more information for foreground/background separation, thus it leads to higher performance. But many trajectories cannot be tracked for many frames, due to lost tracking, falling out of the scene or occlusion

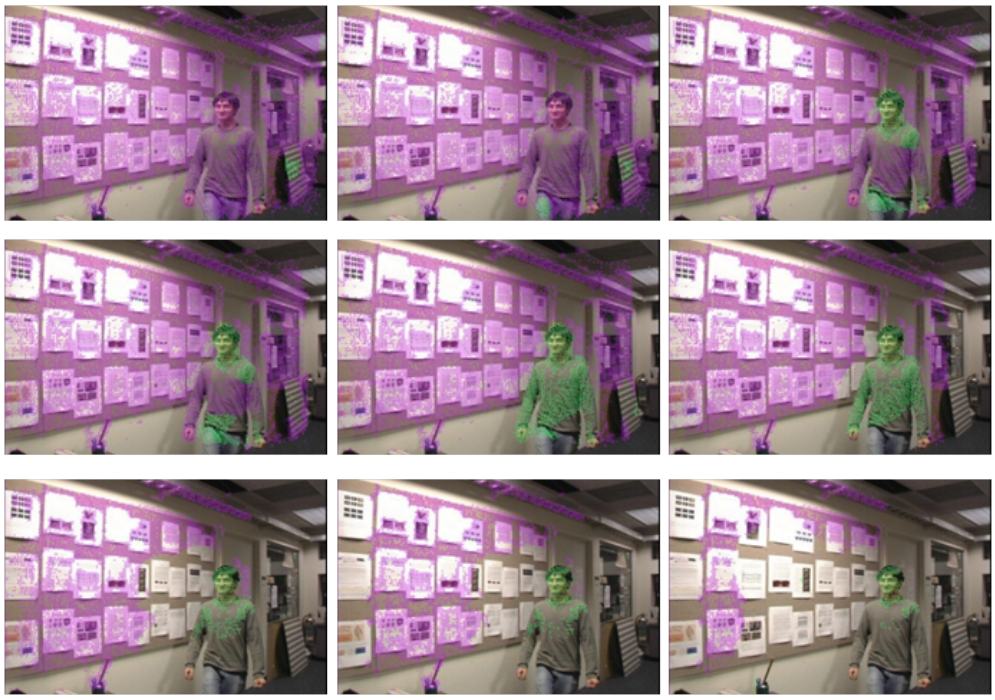

Figure 3.18: Demonstration of the influence of trajectory length l on a specific frame. When the trajectory length l goes up, the performance increases while the trajectory number drops.

by other objects. Using a long trajectory length number makes low coverage of the scene. Thus there is a tradeoff between high performance and trajectory coverage.

We conduct an experiment to show the influence of trajectory length on both performance and trajectory coverage. The result is shown in Figure 3.17. When the trajectory length l goes up, the performance increases. It reaches high and stable performance after $l = 9$. The trajectory number drops constantly along the trajectory length change. The separation results using different l is also shown in one specific frame 9 in Figure 3.18. In the first image ($l = 2$), the separation result is totaly wrong. This is because the motion between only 2 frames does not carry much information to separate foreground and background motions in this video. When l increase from $l = 3$ to $l = 7$, the accuracy of foreground becomes better and better. When $l = 9$, more than 90% trajectories are classified correctly. When l increases from $l = 10$ to $l = 45$, the separation performance remains good, but the most trajectories are gone. When $l = 10$, it is a good tradeoff between performance and trajectory coverage.

Figure 3.19: Performance on "VPerson", a video sequence under moving camera. From left to right: top row: one frame from the original video sequence, the ground truth we manually labeled, result on RANSAC-b; bottom row: MoG, Standard Sparsity discussed in the previous section, and our method. The performance of RANSAC-b is conducted when it reaches its optimal performance in the trajectory level separation step.

Performance evaluation on pixel level labeling. We also evaluate the performance at the pixel level to compare four methods: RANSAC-b [81], MoG [85], L_0 constraint and the proposed method. GPCA, LSA, RANSAC-m are not evaluated in this part, since these three algorithms do not provide pixel level labeling. Figure 3.19 shows the result on "VPerson", a video sequence under moving camera.

MoG works well for stationary cameras, but has unsatisfied performance in moving cameras (*e.g.* "*VPerson*"). This is because a statistical model of MoG is built on pixels of fixed positions over multiple frames, but background objects do not stay in fixed positions under moving cameras. RANSAC based method can accurately label pixels if the trajectories are well classified (*e.g.*, "*VPerson*"). However, it is also possible to build a wrong background subspace because RANSAC method is not stable and sensitive to parameter selection. Our method can robustly handle these diverse types of data, due to our generalized low rank constraint and group sparsity constraint.

As discussed in the previous section, there is no guarantee that RANSAC-b generates stable results on the trajectory level separation. This is due to the fact of the random

sampling of three trajectories in the initialization process. When the trajectory level separation step of RANSAC-b fails to generate satisfied results, the result on pixel level labeling is bad as well. In this experiment, since we want to focus on the pixel level labeling performance, we take the optimal result on the trajectory level labeling of RANSAC-b. Note that the pixel level result of our method is better than the one from RANSAC on "*VPerson*" video (Figure 3.19). The region around head is well segmented in our method with little background scene. This is because the pixel labeling in our approach treats each rigid moving part as a single unit (*i.e.* motion segments on optical flow). Thus the head region is well separated from the scene.

One limitation of our method is that it classifies the shadow as part of the foreground (*e.g.*, on the left side of the person in "*VPerson*" video). This could be further refined by using shadow detection/removal techniques [50].

Discussions One limitation of our algorithm is that it takes the input from the dense point tracker. So the performance of our algorithm really depends on the qualify of dense point tracker. The dense point tracking is still an open research area. It fails to track the objects when it moves very fast. Figure 3.20 shows an example of a squirrel jumping from a table. Since the action is fast, the dense point tracker fails to track it. Without the trajectories on the object, our algorithm is not able to give a good result on this.

Another limitation is that our algorithm is not able to handle missing elements in the trajectory matrix. The algorithm needs to take the trajectories that last for the entire l frames. Trajectories that do not last for the whole l frames are discarded. Figure 3.21 shows an example of this. The area marked in yellow square shows the region of missing trajectories. the left frame shows the trajectories original from the dense point tracker. The yellow region has a lot of trajectories. The right frame shows the result from our algorithm. The trajectories in the lower region are lost due to occlusion when the person is walking. The trajectories in the right region are lost due to the fact they are out of the scene. The trajectories in these regions do not last the entire l frames, so they are discarded and not shown in the frame. To overcome this problem, one solution would be to further introduce some techniques to handle missing

Figure 3.20: The dense point tracking result from a fast moving object: a jumping squirrel. Since the squirrel is moving very fast, the dense point tracker is not able to capture the movement.

Figure 3.21: The input of our algorithm is the trajectories with full length. The trajectories that do not last for the whole l frames are discarded. The area with yellow square shows the region of missing trajectories. Left: the original tracked trajectories; Right: the classified trajectories from our algorithm.

data, for example, matrix factorization. And the combine the strategy of handling missing data with our framework to cover the regions.

3.4 Summary

This dissertation proposed an effective approach to do salient motion detection for complex videos by decomposing the motion trajectory matrix into a low rank one and a group sparsity one. Then the information from these trajectories is used to further label foreground at the pixel level.

The proposed approach is a new algorithm to detect salient motions using low rank

and group sparsity constraints. By using the low rank constraint, the algorithm is able to handle both static and moving cameras. This gives us a unified algorithm to handle both stationary cameras and moving cameras. By using the group sparsity constraint, it brings the points on the consecutive frames together. Rather than analyzing individual points in the video sequence, the group constraint makes the algorithm robust to random noise. The algorithm is also relatively insensitive to parameter settings.

The trajectories recognized by the above model can be further used to label a frame into foreground and background at the pixel level. Motion segments on a video sequence are generated using fairly standard techniques (*i.e.* optical flow and graph cuts). Then the color and motion information gathered from the trajectories is employed to classify each motion segment as foreground or background.

Extensive experiments are conducted on both synthetic data and real videos to show the benefits of our model. The low rank and group sparsity constraints make the model robust to noise and handle diverse types of videos.

Our method depends on trajectory-tracking technique, which is also an active research area in computer vision. When the tracking technique fails, our method may not work well. A robust way is to build the tracking errors into the optimization formulation, so it is able to handle the tracking errors and detect salient motions at the same time. This is one of our future directions.

Chapter 4

Group Activity Analysis

4.1 Introduction

Group activity analysis plays an important role in video surveillance and smart camera systems. Various activities have been studied, including restricted-area access detection [47], car counting [29], detection of people carrying cases [32], abandoned objects [83], group activity detection [118, 66], social network modeling[104], monitoring vehicles [103], scene analysis [87] and so on. This dissertation focuses on modeling events in human group activities, which is a very important application for video surveillance. Figure 4.1 shows two sample frames. (a) shows a group of people fighting in the street, and (b) shows people running away from the scenes.

Figure 4.1: Abnormal event examples. (a) a group of people fighting; (b) People are panic, trying to run away from the scene.

We propose a new method to model group activities. We represent group activities by learning relationships between the current behavior state of a subject and its actions. The goal is to explore the reasons why people take different actions under different

Figure 4.2: Interaction energy potentials of two sample frames. Green arrow is the velocity; round dot denotes energy values. Red dot shows a low energy value and blue shows a high value.

situations [20].

In the real world, people are driven by their goals. They take into account of the environment as well as the influence of other people. We define an interaction energy potential function to represent the current state of a subject based on the positions/velocities of a subject itself as well as its neighbors. Figure 4.2 shows an example of interaction energy potentials and velocities. Section 4.2 gives the details of the definition. Social behaviors are captured by the relationship between interaction energy potential and its action, which is then used to describe social behaviors. The feature patterns indicate a group activity. The Interaction Energy Potentials are further represented by Bag-of-Words features and trained through machine learning algorithms.

Our method is validated on two datasets UMN [2] and BEHAVE [1]. Extensive experiments show that our method is more effective to model the behaviors in group activities, than the state-of-art algorithms.

The main contribution of the proposed approach is a new feature representation method using Interaction Energy Potential to model the group behaviors. This approach has three merits:

1. The Interaction Energy Potential is proposed to model the relationship among a group of people;

2. The relationship between the current state of a subject and the corresponding reaction is explored to model the group behavior patterns;

3. This method does not rely on human detection or segmentation technique, so it is more robust to the errors that are introduced by detectoin/segmentation techniques.

In the remainder of this chapter, the major method will be presented in Section 4.2, including the salient points detection and tracking in Section 4.2.2, Interaction Energy Potential formulation in Section 4.2.3, and feature representation and modeling in Section 4.2.4. The experimental results are shown in Section 4.3. The algorithm is tested on two datasets: UMN and BEHAVE. Comparing with the state-of-art methods, our method are more effective. Section 4.4 discuss the summary and future work.

4.2 Methodology

4.2.1 Overview

Here we propose a new method to model group interactions using interaction energy potentials. Our method takes a raw video sequence as input, and then label each video clip as normal or abnormal event. The system framework is summarized in Figure 4.3. First, salient moving points on the foreground moving regions are extracted. Second, an interaction energy potential is calculated for each point. Third, features are represented by relationships among interaction energy potentials and corresponding actions with a coding scheme. Finally, SVM is used to model the features.

4.2.2 Salient Point Detection and Tracking

The ideal case for human activity analysis is to track all the subjects and estimate their positions and velocities, but human detection and tracking is still a challenging problem. Instead we use local salient regions (local interest points) to represent subjects in a scene. The movements of subjects can be represented by the movements of salient

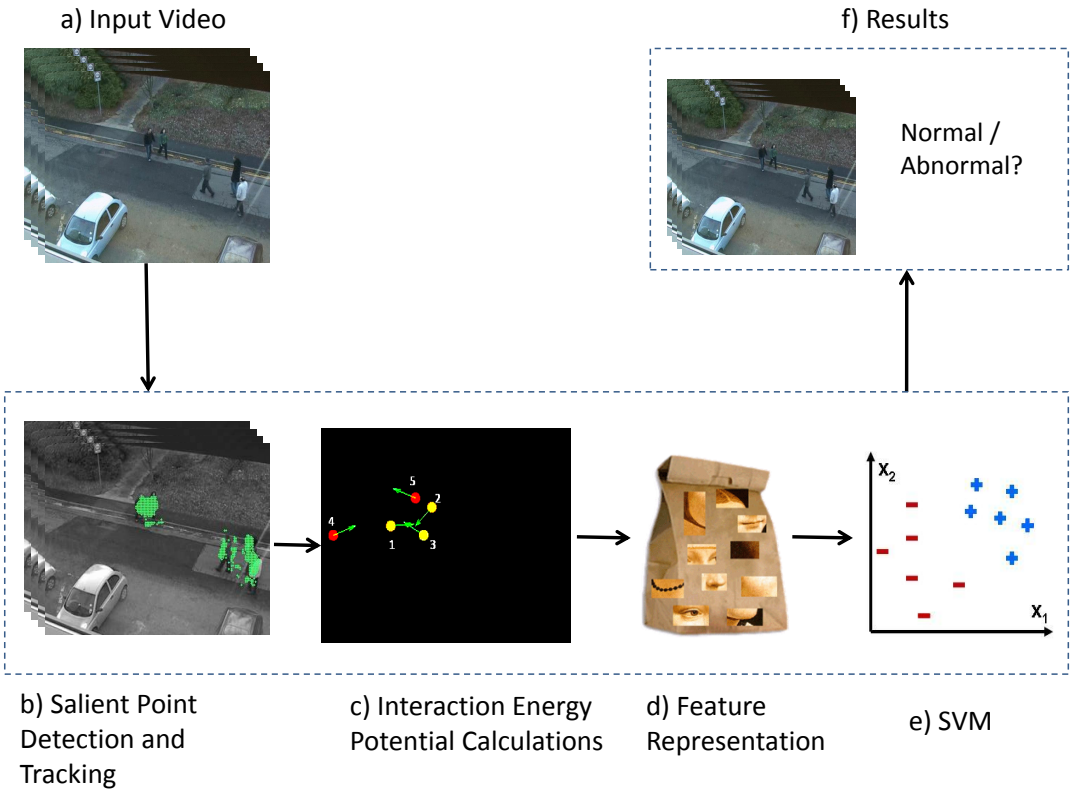

Figure 4.3: Flow chart: given an input clip, salient detection and tracking is performed first. Then Interaction Energy Potential is calculated on the tracked points. After wrapping up with feature representation, SVM is used to label each event.

moving points associated with the subjects, and interactions among the subjects can be implicitly embodied in the interactions among salient points. We use the method proposed in the previous chapter to detect the local salient points. Since the videos we use here are all under stationary cameras, so salient points we obtain are of good quality. The other way to generate the points are using [49] to detect the local spatio-temporal interest points (STIP), and then use the KLT tracker [56, 61] to track interest points. Figure 4.2 shows an example of salient point detection and tracking.

For each tracked points p_i, we record its positions $\{\mathbf{x}_i^0 ... \mathbf{x}_i^t ...\}$, where each $\mathbf{x}^t$ is a 2D vector, and its velocity $\mathbf{v}_i^t$ at time t is calculated by

$$\mathbf{v}_i^t = \frac{\mathbf{x}_i^{t+T} - \mathbf{x}_i^t}{T} \tag{4.1}$$

where T is the time interval. A point p_i is then modeled as $p_i = (\mathbf{x}_i^t, \mathbf{v}_i^t)$. Besides

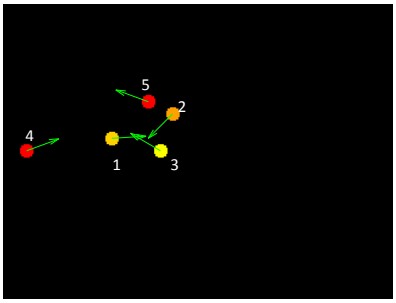

Figure 4.4: Toy examples. Five subjects, with their current velocities. Color denotes energy values. Red color denotes a low interaction energy potential value, while yellow denotes a high energy value. Taking perspective of subject 1, it has interactions with subject 2 and 3; ignores subject 4 subject and moves away from subject 5.

the self-representation of velocity, we also take into account of neighbor salient points, which implicitly represent interactions among subjects in group activities. Interactions among subjects are modeled by interaction energy potentials, which is addressed in the following section.

4.2.3 Interaction Energy Potentials

Given a set of salient points $S = \{p_i\}$ $(i = 1...n)$, energy potential E_i of p_i is calculated based on positions and velocities of its neighbor points. The calculation of the interaction energy potentials is inspired by the idea of social behaviors[70]: assuming that people are aware of the positions and velocities of other people at time t. Thus we can make a reasonable assumption that people can predict the movement of other people and have a general estimation about whether they would meet in the near future. This is also how people walk in the real world.

We first consider two subjects. Given two subjects s_i and s_j in a scene, we are now thinking from the perspective of s_i, and treating s_j as its neighbor. We define the current time as $t = 0$ and use $\mathbf{x}_i = \mathbf{x}_i^0$ for simplicity. If s_i proceeds with velocity $\mathbf{v}_i$, then it expects to have a distance $d_{ij}^2(t)$ from s_j at time t.

$$d_{ij}^2(t) = ||\mathbf{x}_i + t\mathbf{v}_i - (\mathbf{x}_j + t\mathbf{v}_j)||^2 \tag{4.2}$$

Minimal distance d_{ij} occurs at the time of closest point t^*, where

$$t^* = \max\{0, \arg\min d_{ij}^2(t)\} \tag{4.3}$$

where $\arg\min d_{ij}^2(t)$ can be obtained by setting the derivative of d_{ij} to zero with respect to time t. Then we obtain t^* as follows:

$$t^* = \max\{0, -\frac{(\mathbf{x}_i - \mathbf{x}_j) \cdot (\mathbf{v}_i - \mathbf{v}_j)'}{||\mathbf{v}_i - \mathbf{v}_j||^2}\} \tag{4.4}$$

By substituting t into Eq. 4.2, we can obtain the minimum distance d_{ij}^{*2} between subjects i and j as

$$d_{ij}^{*2} = d_{ij}^2(t^*).$$

d_{ij}^{*2} defines how far two subjects will meet based on the current velocities. If d_{ij}^{*2} is smaller than a distance threshold d_c, a close-distance meet would happen in the near future. If p_j has a close-distance from p_i, p_j is very likely to draw p_i's attention at this moment. We therefore build an interaction energy potential function based on their distance

$$E_{ij} = w_{ij}^c w_{ij}^\phi exp(-\frac{d_{ij}^2(t)}{2\sigma_d^2}) \tag{4.5}$$

$$w_{ij}^c = \begin{cases} 1 & d_{ij}^{*2} < d_c \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

$$w_{ij}^\phi = \begin{cases} 1 & \phi_{ij} < \phi_{view} \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

where σ_d is the radius of influence of p_j. The closer of p_j, the higher attention would p_i have. ϕ is the current angular displacement of p_j from the perspective of p_i. ϕ_{view} is the field-of-view, which is the angle displacement between the current moving direction

and the neighbor point direction. As people only see things in the front, ϕ_{view} controls how people see things. The Interaction energy potential E_{ij} describes the influence from p_j. E_{ij} is high when they are close, and it is minimal as their distance goes to infinity.

For the case of multiple subjects, the influence of all the other subjects can be modeled as an average of energy potential E_{ik}. The overall interaction energy for subject p_i is given by

$$E_i = \frac{1}{N} \sum_{k \neq i, E_{ik} > 0} E_{ik} \tag{4.8}$$

where N is the number of non-zero neighbor points. The Interaction energy potential E_i describes the current behavior state of subject p_i. Figure 4.4 shows an example of 5 points with their interaction energy potentials. Now we are taking perspective of subject p_1, with 4 neighboring points in the frame. p_1 is moving towards p_2 and p_3. As they are going to meet based on the current velocities, p_2 and p_3 have a high influence on p_1. p_4 is in the back, so p_1 does not see it. p_5 moves further away from p_1, so it does not draw p_1's attention at this moment. The total interaction energy potential of p_1 comes from p_2 and p_3. Next we take perspective of p_5. It moves away from all the other points, so its neighbors would not influence it at this time. It results in a low interaction energy value for p_5. The Energy potential is calculated for each subject from its own view. Then energy values are denoted by color in the figure. Yellow dot in Figure 4.4 shows a high energy value, while red dot shows a low energy value. In our method, E is calculated for each point. Figure 4.2 shows an example of detected points and corresponding interaction energy potentials.

4.2.4 Features Representation and Modeling

The Interaction energy potential reflects the current interaction with the surrounding of a person. Different from [70], our goal is to find reasons why people take actions and what situations make them take actions. This can be modeled by relationships between current states (interaction energy potential E) and actions (velocities v).

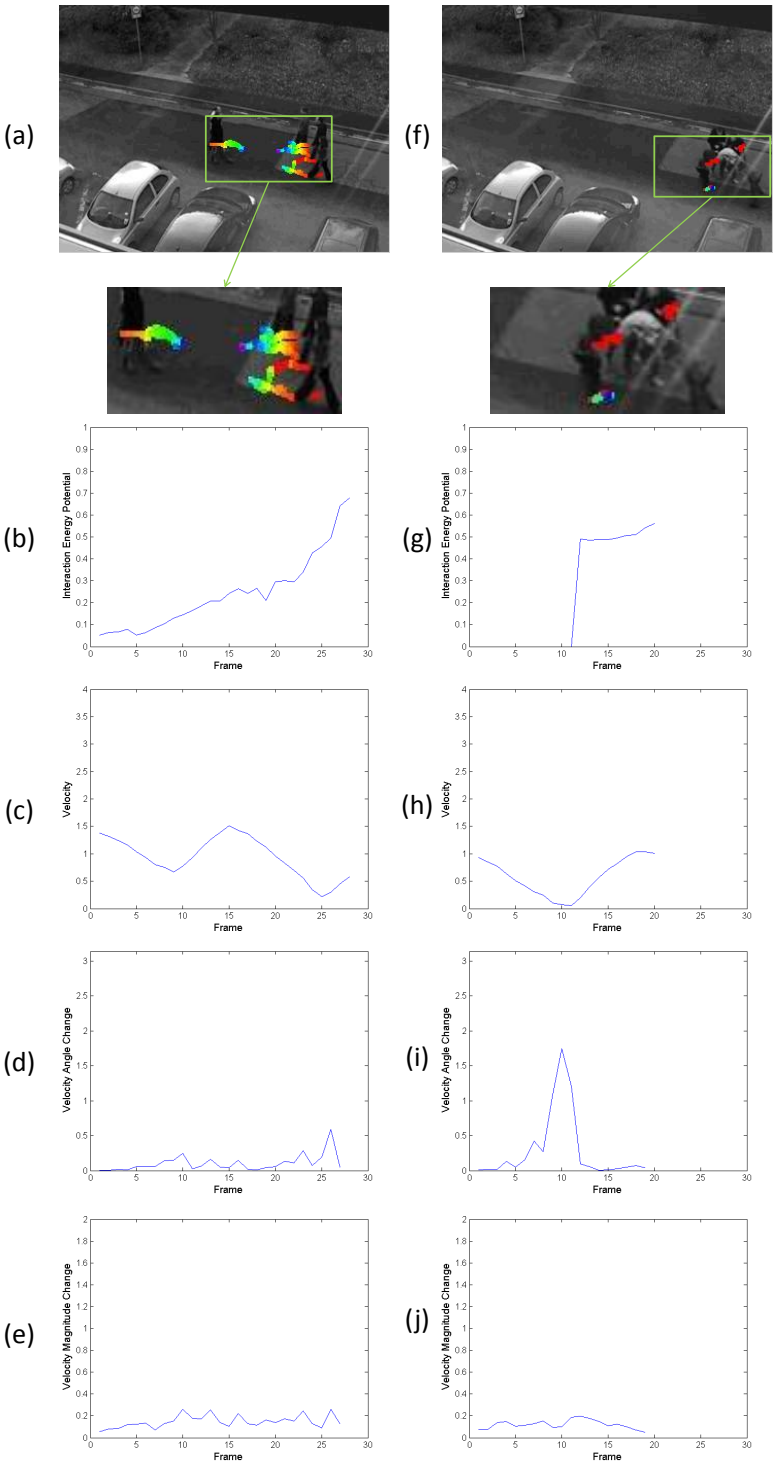

Figure 4.5: Two events. (a)group meeting event; (b) energy E of meeting; (c) velocity magnitude v_m of meeting; (d) velocity direction changing Δv_d of meeting; (e) velocity magnitude changing Δv_d of meeting; (f)fighting event; (g) energy E of fighting; (h) velocity magnitude v_m of fighting; (i) velocity direction changing Δv_d of fighting; (j) velocity magnitude changing Δv_d of fighting.

Figure 4.5 shows an example of the relationship between energy changing and velocity changing over time. In Figure 4.5, (a) is a group of people meeting. Color lines show energy changing through time. We choose one point and its trajectory for analysis. (b) shows its energy changing over time. As people move closer, the energy increases slowly. At the same time, v_m, Δv_d and Δv_m remain stable. They are shown in (c)(d)(e) respectively. This is a common event in the real world. People have their desires to meet, and they try to remain at a constant speed and direction. In contrast, (f) shows a group of people fighting. At time 10, Δv_d changes dramatically (shown in (i)) even with low interaction energy potential (shown in (g)). This indicates an uncommon pattern. A point changes its moving direction dramatically without an obvious reason.

Each local patch around the salient points is represented by Interaction energy potentials and optical flows. Then standard bag-of-words method is used. The bag-of-words model (BoW model) is a histogram representation based on independent features. It represents an image or a video as an orderless collection of local features. It has been widely used in image classification/retrieval and action analysis [93, 94, 109]. A bag of visual words is a sparse vector of occurrence counts of a vocabulary of local image features. There are two basic steps for BoW model: feature representation and codebook generation. Here Interaction Energy Potential is used as the feature representation. Then these feature vectors are converted to "codewords" (analogy to words in text documents), which also produces a "codebook" (analogy to a word dictionary). A codeword can be considered as a representative of several similar patches. Here we uses k-means clustering to generate the clusters. Then each patch in an image/video is mapped to a certain codeword through the clustering process and the image can be represented by the histogram of the codewords.

Soft assignment [30] is used here to generate the codebook. Soft assignment is a technique for representing images/videos as histograms by flexible assignment of image descriptors to a visual vocabulary. It has several advantages over hard assignment. Hard assignment associates each descriptor vector with the nearest visual word of a given dictionary. Whilst this provides with a reasonable expressive power, a single descriptor belongs to only one closest word in a dictionary. This yields a high quantization error.

Soft assignment mitigates this effect by allowing soft contribution of each descriptor to its closest words in a dictionary. After the BoW representation, each video clip is represented by a feature vector. Then we use SVM [16] to build a model and label each video clip.

The major computation in our algorithm is the calculation of interaction energy potential for each tracked points. For each tracked point, the value of IEP is determined by checking every other point. Thus the overall time complexity is $O(k^2)$, where k is the number of tracked salient points in our framework. The number of salient points for each video sequence is not large. It is in the scale of a few hundred.

4.3 Experiments

Figure 4.6: The UMN Dataset: people walking in a park.

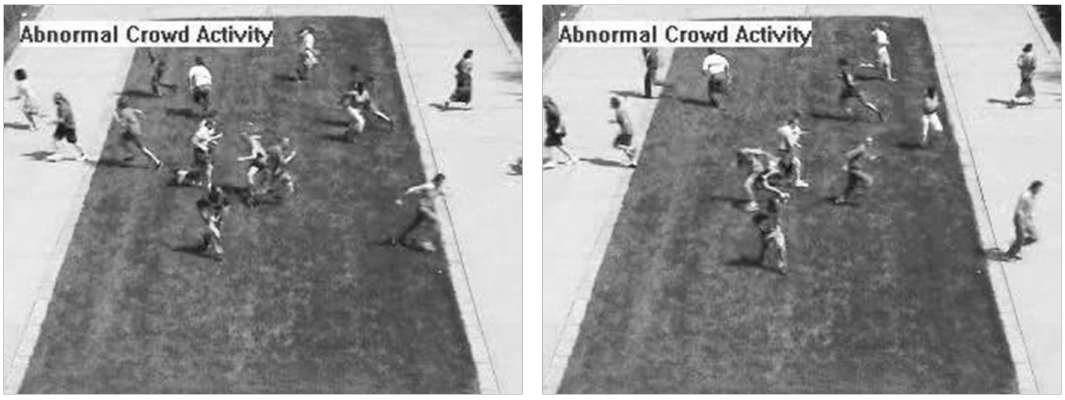

Figure 4.7: The UMN Dataset: people running away from the park.

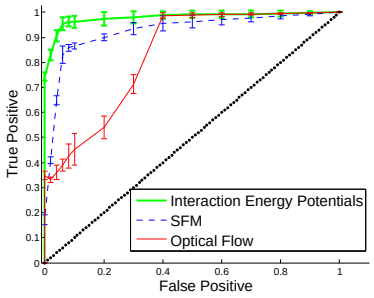

Figure 4.8: Performance on the UMN dataset.

To evaluate the performance of our algorithm, we conduct experiments on two datasets: UMN dataset [2] and BEHAVE dataset [1]. Details are shown in the following sections.

Method	Area Under ROC
Interaction Energy Potential	0.9795 ± 0.0144
SFM [60]	0.9468 ± 0.0146
Optical Flow	0.8317 ± 0.0138

Table 4.1: Quantitative results on the UMN dataset.

4.3.1 The UMN Dataset

This dataset is collected from University of Minnesota [2], which contains videos of 11 different scenarios of an escape event. The videos are shot in 3 different scenes, including both indoor and outdoor. Each video clip starts with an initial part of normal behaviors and ends with sequences of abnormal behaviors. Figure 4.6 shows some sample frames from a normal event. Figure 4.7 shows some sample frames from an event with all people running away from the scene. Scenes in this dataset are crowded, with about 20 people walking around. The videos are chopped into 765 clips. Each clip has 10 frames, containing either normal or abnormal event. Our job is to find the abnormal events, and label each clip as a normal/abnormal event. To obtain a reliable result, we random select 5 scenarios for training, and 6 for testing. Then the average and variance of the performance are reported.

We take optical flow features as the baseline, and also compare with Social Force [60], a state of art algorithm for group event analysis. Figure 4.8 and Table 4.1 reports the

experimental results. The results show that our algorithm is competitive with these state-of-art methods.

4.3.2 The BEHAVE Dataset

To further demonstrate the effectiveness of our method, we conduct experiments on another dataset: the BEHAVE Dataset. We collect an activity dataset from the BEHAVE dataset [1]. The BEHAVE dataset has many complex group activities, including meeting, splitting up, standing, walking together, ignoring each other, fighting, escaping as well as running. Scenarios contain various number of participants. The dataset consists of 50 clips of fighting events, and 271 normal events. All the activities in this dataset are common in the real world. The scene is moderately crowded. The length of tracked salient points are 27.81 frames in average. Figure 4.9 shows the frames from two normal events: two groups of people are passing by each other, and two groups of people are walking towards each other and meeting. Figure 4.10 shows the frames from two fighting scenes.

Figure 4.9: The BEHAVE dataset: samples from the normal events. Left: two groups of people passing by; Right: two groups of people meeting.

Figure 4.10: The BEHAVE dataset: people fighting on the street.

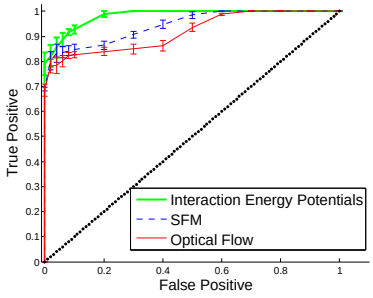

Figure 4.11: Results on BEHAVE dataset. Comparison of our method (green line) with Social Force [60] and Optical Flow.

Method	Area Under ROC
Interaction Energy Potential	0.9822 ± 0.0034
SFM [60]	0.9246 ± 0.0041
Optical Flow	0.9080 ± 0.0072

Table 4.2: Quantitative results on the BEHAVE dataset.

Finally, we compare our method with the optical flow based method and Mehran *et al.* 's method [60]. Figure 4.11 and Table 4.2 show our results comparing to these two methods. It shows that our Interaction Energy Potential does a better job to represent events in such complex group activities. It comes from the fact that our feature does not only consider the velocity distribution, but also utilizes the interaction among a group, which is able to improve the performance.

Discussions One of the limitations of our algorithm is that it is context dependent.

This is due to the fact that the group behaviors can be very different in many scenarios. For example, the events in an indoor scene like airport or bank are different from an outdoor field like soccer field. Figure 4.12 shows two examples. On the soccer field, running fast, chasing another player or scrambling for the ball are quite common. But this can be treated as a suspicious event in an airport, bank or subway. Since the definition of events is dependent on different locations, the model needs to be trained accordingly. The model used in different environment is context dependent. However, since our algorithm is designed for surveillance cameras that are usually mounted in a building or mounting structure for a long time, training a model should be only a one-time thing. One camera does not need to cover all the scenarios. In the future, if there are cases where the cameras need to be moved into different scenes, we can also make our model context aware. The basic idea is to combine a scene classification method into our framework. It can be done by combining scene features (static image features) into our model, or train a hierarchy model with a built-in scene classifier. In doing so, our model can carry both the scene information and the event modeling based on different events. The model can first detect the scene environment, and then use the corresponding knowledge from this scene environment.

Another limitation of this algorithm is when the camera position changes, the parameters need to be adjusted. Currently in our model, the parameters are based on the datasets where the cameras are mounted in a relatively high building. The current distance and the comfort distance are learnt accordingly. However, if the camera is mounted from a high-rise building as shown in Figure 4.13, the parameters need to be adjusted, since the distance setting is different. One solution to overcome this limitation is to learn the parameters in world coordinates. Then the distances in the world coordinates reflect the real distance that people are comfortable. This is more generic. When a camera is mounted, the image position is first converted to world positions, so the parameters do not need to be re-trained. This decouples the problem into a one-time parameter learning and camera calibration problem.

Figure 4.12: The definition of abnormal events is different depending on the scenarios. Chasing in a soccer field is common, but in an indoor scene like subway, it can be an abnormal event. The variety of abnormal events makes our model context dependent. Left: a subway station; Right: a soccer field.

4.4 Summary

We proposed a method to model group activities. The proposed algorithm explored the reasons why people take actions and what situations make people take actions. The relationships between the current behavior states and actions indicate normal/abnormal patterns. Pedestrians' environment is modeled by an interaction energy potential function. Different group activities are indicated in uncommon energy-velocity patterns. Our method does not depend on human detection or tracking algorithm. We conducted the experiments on the UMN dataset and the BEHAVE dataset. Results showed the effectiveness of our method, and it is more competitive with the state-of-art methods.

One future direction would be exploring more on the spatial information. In our approach, Bag-of-Words (BoW) model is used for the final feature representation. It represents an image as an orderless collection of local features. Though it has shown impressive levels of performance, it discards the spatial relationships of local features. In our method, the spatial information within each cropped window is implicitly considered. But the spatial layout among the cropped windows are not considered, due to the orderless Bag-of-Words feature representation. The spatial relationships between local image features are important in the sense that they provide a kind of 'linkage' information between independent image features. This will help us better understand

Figure 4.13: Street scene from a bird-of-view camera mounted in a high-rise building.

how the objects in the scenes are related to each other. To model complex group activities, the spatial relationships among the local areas can be strong signals. This can potentially improve the performance of group activities. A popular way to model spatial relationship is the Spatial Pyramid Match model (SPM) model. It works by placing a sequence of increasingly coarser grids over the image and taking a weighted sum of the number of matches that occur at each scale. Feature matches from finer scales are given more weight. Applying the SPM model to our feature representation is one of our future work.

Chapter 5

Conclusions

This dissertation aims to address two challenging problems in group activity analysis: various camera motions and feature modeling of group activity analysis. To address the first problem, we proposed a salient motion detection method to handle various camera motions. Given a video sequence under either stationary or moving cameras, the algorithm is able to detect the moving areas from the background. The general idea of this method comes from two "sparsity" observations. The motions of foreground moving objects satisfies a group sparsity constraint. At the same time, the motion of the background objects satisfies a low rank constraint under an orthographic cameras. Using these constraints, the salient motion detection problem is formulated as a matrix decomposition problem. Experiments are conducted on various types of data. Our algorithm performs better than the state-of-art algorithms. To effectively model the motions of group behaviors, a new feature representation is proposed. The interaction energy potential proposed in this feature is able to represent interactions among groups of people. The algorithm is tested on two datasets UMN [2] and BEHAVE [1]. Results show that our feature presentation is more effective comparing to other state-of-art methods.

There are some future directions to pursue toward the goal of group activity analysis. First, the salient motion detection method depends on trajectory-tracking technique, which is also an active research area in computer vision. When the tracking technique fails, our method may not work well. A robust way is to build the tracking errors into the optimization formulation, so it is able to handle the tracking errors and detect salient motions at the same time. This is one of our future directions.

Second, more group information can be applied to the salient motion detection

model. The group information used in our algorithm is the neighboring information along a tracked point. We call it temporal group information. Another strong group information is the neighboring information in spatial displacement. For example, a point on a car's window and a point on this car's roof should have very similar motion movements. This gives us further constraints for the foreground regions. How to incorporate the spatial group information into our formula is also interesting.

Third, more spatial information can be incorporated into the features. In our method, each cropped window incorporates the spatial information. But the spatial information among those windows is not considered, which is an important signal for complex group activities. We believe this can potentially improve the performance.

Fourth, we live in a world with a lot of data. The current datasets we deal with only contain a few categories with several hundreds of videos. How to do group activity analysis in huge datasets is a very challenging yet interesting problem. This is one of our future directions.

References

[1] Behave: http://homepages.inf.ed.ac.uk/rbf/behave.

[2] Unusual Crowd Activity Dataset: http://mha.cs.umn.edu/movies/crowd-activity-all.avi.

[3] J. Aggarwal and M. Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, 43(3):16:1–16:43, 2011.

[4] S. Ali and M. Shah. Floor fields for tracking in high density crowd scenes. In *Proceedings of the European Conference on Computer Vision*, pages 1–14, 2008.

[5] E. L. Andrade, S. Blunsden, and R. B. Fisher. Modelling crowd scenes for event detection. In *Proceedings of the International Conference on Pattern Recognition*, pages 175–178, 2006.

[6] G. Antonini, S. V. Martinez, M. Bierlaire, and J. P. Thiran. Behavioral priors for detection and tracking of pedestrians in video sequences. *International Journal of Computer Vision*, pages 159–180, 2006.

[7] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Structured sparsity through convex optimization. *Technical Report HAL*, 00621245, 2011.

[8] A. Borji and L. Itti. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):185–207, 2013.

[9] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[10] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proceedings of the European Conference on Computer Vision*, pages 25–36, 2004.

[11] S. Brutzer, B. Hoferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1937–1944, 2011.

[12] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[13] E. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[14] E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of ACM*, 58(1):1–37, 2011.

[15] E. J. Candes and B. Recht. Exact matrix completion via convex optimzation. *Foundations of Computational Mathematics*, 9:717–772, 2009.

[16] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[17] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001.

[18] L. Cheng and M. Gong. Realtime background subtraction from dynamic scenes. In *Proceedings of the International Conference on Computer Vision*, pages 2066–2073, 2009.

[19] X. Cui, J. Huang, S. Zhang, and D. N. Metaxas. Background subtraction using low rank and group sparsity constraints. In *Proceedings of the European conference on Computer Vision*, pages 612–625, 2012.

[20] X. Cui, Q. Liu, M. Gao, and D. N. Metaxas. Abnormal detection using interaction energy potentials. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 3161–3167, 2011.

[21] X. Cui, Q. Liu, and D. Metaxas. Temporal spectral residual: fast motion saliency detection. In *Proceedings of the conference on Multimedia*, pages 617–620, 2009.

[22] X. Cui, Q. Liu, S. Zhang, F. Yang, and D. N. Metaxas. Temporal spectral residual for fast salient motion detection. *Neurocomputing*, 86:24–32, 2012.

[23] X. Cui, Y. Liu, S. Shan, X. Chen, and W. Gao. 3D haar-like features for pedestrian detection. In *Proceedings of the International Conference on Multimedia and Expo*, pages 1263–1266, 2007.

[24] X. Cui, S. Zhang, J. Huang, X. Huang, D. Metaxas, and L. Axel. Left endocardium segmentation using spatio-temporal metamorphs. In *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pages 226–229, 2012.

[25] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS05)*, pages 65–72, 2005.

[26] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[27] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163, 2002.

[28] M. Figueiredo, R. Nowak, and S. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):586–597, 2007.

[29] N. Friedman and S. Russell. Image segmentation in video sequences: A probabilistic approach. In *Proceedings of Uncertainty in Artificial Intelligence*, pages 175–181, 1997.

[30] J. C. Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. Smeulders. Kernel codebooks for scene categorization. In *Proceedings of the European Conference on Computer Vision*, pages 696–709, 2008.

[31] I. Haritaoglu, D. Harwood, and L. Davis. W4: real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[32] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:809–830, 2000.

[33] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference*, pages 147–151, 1988.

[34] E. Hayman and J.-O. Eklundh. Statistical background subtraction for a mobile observer. In *Proceedings of the International Conference on Computer Vision*, pages 67–74, 2003.

[35] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical Review E*, pages 4282–4286, 1995.

[36] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[37] J. Huang, X. Huang, and D. Metaxas. Learning with dynamic group sparsity. In *Proceedings of the International Conference on Computer Vision*, pages 64–71, 2009.

[38] J. Huang, S. Zhang, and D. Metaxas. Efficient MR image reconstruction for compressed MR imaging. In *Medical Image Computing and Computer-Assisted Intervention*, volume 6361, pages 135–142, 2010.

[39] J. Huang, S. Zhang, and D. Metaxas. Efficient MR image reconstruction for compressed MR imaging. *Medical Image Analysis*, 15(5):670 – 679, 2011.

[40] J. Huang and T. Zhang. The benefit of group sparsity. *The Annals of statistics*, 38(4):1978–2004, 2010.

[41] N. Ikizler and P. Duygulu. Histogram of oriented rectangles: a new pose descriptor for human action recognition. *Image Vision Computing*, 27(10):1515–1526, 2009.

[42] L. Itti and C. Koch. Computational modeling of visual attention. *Nature reviews neuroscience*, 2(3):194–203, 2001.

[43] R. Jain and H. Nagel. On the analysis of accumulative difference pictures from image sequences of real world scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):206–214, 2009.

[44] Y. Kameda and M. Minoh. A human motion estimation method using 3-successive video frames. In *International Conference on Virtual Systems and MultiMedia*, pages 135–140, 1996.

[45] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l1-regularized least squares. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):606–617, 2007.

[46] T. Ko, S. Soatto, and D. Estrin. Warping background subtraction. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1331–1338, 2010.

[47] J. Konrad. Motion detection and estimation. *Handbook of Image and Video Processing, 2nd Edition*, 2005.

[48] I. Laptev and T. Lindeberg. Space-time interest points. In *Proceedings of the International Conference on Computer Vision*, pages 432–439, 2003.

[49] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[50] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikainen, and S. Li. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1301–1306, 2010.

[51] C. Liu. Beyond pixels: Exploring new representations and applications for motion analysis. *Doctoral thesis, Massachusetts Institute of Technology.*, 2009.

[52] C. Liu, W. Freeman, E. Adelson, and Y. Weiss. Human-assisted motion annotation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[53] J. Liu, S. Ali, and M. Shah. Recognizing human actions using multiple features. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[54] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos 'in the wild'. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1996–2003, 2009.

[55] Y. Liu, X. Chen, H. Yao, X. Cui, C. Liu, and W. Gao. Contour-motion feature (CMF): A space–time approach for robust pedestrian detection. *Pattern Recognition Letters*, 30(2):148–156, 2009.

[56] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[57] V. Mahadevan and N. Vasconcelos. Background subtraction in highly dynamic scenes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2008.

[58] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. In *IEEE Transaction on Signal Processing*, pages 3397–3415, 1993.

[59] G. Medioni, I. Cohen, F. Brémond, S. Hongeng, and R. Nevatia. Event detection and analysis from video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):873–889, 2001.

[60] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 935–942, 2009.

[61] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *Proceedings of the International Conference on Computer Vision*, pages 104–111, 2009.

[62] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65:43–72, 2005.

[63] A. Mittal and D. Huttenlocher. Scene modeling for wide area surveillance and image synthesis. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 2, pages 160–167, 2000.

[64] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh. Background modeling and subtraction of dynamic scenes. In *Proceedings of the International Conference on Computer Vision*, pages 1305–1312, 2003.

[65] Y. Mu, J. Dong, X. Yuan, and S. Yan. Accelerated low-rank visual recovery by random projection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2609–2616, 2011.

[66] B. Ni, S. Yan, and A. Kassim. Recognizing human group activities with localized causalities. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1470–1477, 2009.

[67] J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318, 2008.

[68] S. Nowozin, G. Bakir, and K. Tsuda. Discriminative subsequence mining for action classification. In *Proceedings of the International Conference on Computer Vision*, pages 1–8, 2007.

[69] S. Palmer. *Vision science: Photons to phenomenology*. MIT press Cambridge, MA., 1999.

[70] S. Pellegrini, A. Ess, K. Schindler, and L. V. Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proceedings of the International Conference on Computer Vision*, pages 261–268, 2009.

[71] M. Piccardi. Background subtraction techniques: a review. In *IEEE International Conference on Systems, Man, and Cybernetics*, volume 4, pages 3099–3104, 2005.

[72] R. Poppe. A survey on vision-based human action recognition. *Image Vision Computing*, 28(6):976–990, 2010.

[73] A. Prati, S. Calderara, and R. Cucchiara. Using circular statistics for trajectory shape analysis. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[74] K. Rapantzikos, Y. Avrithis, and S. Kollias. Dense saliency-based spatiotemporal feature points for action recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1454–1461, 2009.

[75] Y. Ren, C. Chua, and Y. Ho. Statistical background modeling for non-stationary camera. *Pattern Recognition Letters*, 24(1-3):183–196, 2003.

[76] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2195–2202, 2006.

[77] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Proceedings of International Conference on Pattern Recognition*, pages 32–36, 2004.

[78] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *International Conference on Multimedia*, pages 357–360, 2007.

[79] P. Scovanner and M. F. Tappen. Learning pedestrian dynamics from the real world. In *Proceedings of the International Conference on Computer Vision*, pages 381–388, 2009.

[80] S. Shariat and V. Pavlovic. Isotonic cca for sequence alignment and activity recognition. In *Proceedings of the International Conference on Computer Vision*, pages 2572–2578, 2011.

[81] Y. Sheikh, O. Javed, and T. Kanade. Background subtraction for freely moving cameras. In *Proceedings of the International Conference on Computer Vision*, pages 1219–1225, 2009.

[82] Y. Sheikh and M. Shah. Bayesian object detection in dynamic scenes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 1, pages 74–79, 2005.

[83] K. Smith, P. Quelhas, and D. Gatica-Perez. Detecting abandoned luggage items in a public space. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 75–82, 2006.

[84] J.-L. Starck, M. Elad, and D. Donoho. Image decomposition via the combination of sparse representations and a variational approach. *IEEE Transactions on Image Processing*, 14(10):1570–1582, 2005.

[85] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.

[86] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *Proceedings of the European Conference on Computer Vision*, pages 438–451, 2010.

[87] E. Swears and A. Hoogs. Functional scene element recognition for video scene analysis. In *IEEE Workshop on Motion and Video Computing (WMVC)*, pages 1–8, 2009.

[88] Y. Tian, M. Lu, and A. Hampapur. Robust and efficient foreground analysis for real-time video surveillance. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1182–1187, 2005.

[89] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[90] A. Treuille, S. Cooper, and Z. Popovic. Continuum crowds. *ACM Transactions on Graphics*, 25(3):1160–1168, 2006.

[91] R. Tron and R. Vidal. A benchmark for the comparison of 3-D motion segmentation algorithms. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[92] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 1, pages 621–628, 2003.

[93] C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 257–264, 2003.

[94] J. Willamowski, D. Arregui, G. Csurka, C. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors. In *Proceedings of the International Conference on Pattern Recognition Workshop on Learning for Adaptable Visual Systems*, volume 17, page 21, 2004.

[95] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 2002.

[96] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. Rehg. A scalable approach to activity recognition based on object use. In *Proceedings of the International Conference on Computer Vision*, pages 1–8, 2007.

[97] S. Wu, B. Moore, and M. Shah. Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2054–2060, 2010.

[98] D. Xu, Y. Huang, Z. Zeng, and X. Xu. Human gait recognition using patch distribution feature and locality-constrained group sparse representation. *IEEE Transactions on Image Processing*, 21(1):316–326, 2012.

[99] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and nondegenerate. In *Proceedings of the European Conference on Computer Vision*, pages 94–106, 2006.

[100] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang. Coupled dictionary training for image super-resolution. *IEEE Transactions on Image Processing*, 21(8):3467–3478, 2012.

[101] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873, 2010.

[102] P. Yang, Q. Liu, X. Cui, and D. N. Metaxas. Facial expression recognition using encoded dynamic features. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[103] Q. Yu and G. Medioni. Motion pattern interpretation and detection for tracking moving vehicles in airborne video. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2671–2678, 2009.

[104] T. Yu, S. Lim, K. Patwardhan, and N. Krahnstoever. Monitoring, recognizing and discovering social networks. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1462–1469, 2009.

[105] W. Yu and A. Johansson. Modeling crowd turbulence by many-particle simulations. *Physical Review E*, 76(4):046105, 2007.

[106] C. Yuan, G. Medioni, J. Kang, and I. Cohen. Detecting motion regions in the presence of a strong parallax from a moving camera by multiview geometric constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1627–1641, 2007.

[107] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journals of the Royal Statistical Society*, 68(1):49–67, 2006.

[108] C. Zhang, J. Liu, Q. Tian, C. Xu, H. Lu, and S. Ma. Image classification by non-negative sparse coding, low-rank and sparse decomposition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1673–1680, 2011.

[109] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: An in-depth study. *Technical Report RR-5737, INRIA Rhone-Alpes*, 2005.

[110] S. Zhang, J. Huang, Y. Huang, Y. Yu, H. Li, and D. N. Metaxas. Automatic image annotation using group sparsity. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 3312–3319, 2010.

[111] S. Zhang, J. Huang, H. Li, and D. N. Metaxas. Automatic image annotation and retrieval using group sparsity. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(3):838–849, 2012.

[112] S. Zhang, Y. Zhan, M. Dewan, J. Huang, D. N. Metaxas, and X. S. Zhou. Sparse shape composition: A new framework for shape prior modeling. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1025–1032, 2011.

[113] S. Zhang, Y. Zhan, M. Dewan, J. Huang, D. N. Metaxas, and X. S. Zhou. Towards robust and effective shape modeling: Sparse shape composition. *Medical Image Analysis*, 16(1):265 – 277, 2012.

[114] S. Zhang, Y. Zhan, and D. N. Metaxas. Deformable segmentation via sparse representation and dictionary learning. *Medical Image Analysis*, 16(7):1385 – 1396, 2012.

[115] Z. Zhang, X. Liang, and Y. Ma. Unwrapping low-rank textures on generalized cylindrical surfaces. In *Proceedings of the International Conference on Computer Vision*, pages 1347–1354, 2011.

[116] Z. Zhao and A. Elgammal. Human activity recognition from frame's spatiotemporal representation. In *Proceedings of the International Conference on Pattern Recognition*, pages 1–4, 2008.

[117] J. Zhong and S. Sclaroff. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In *Proceedings of the International Conference on Computer Vision*, pages 44–50, 2003.

[118] Y. Zhou, S. Yan, and T. Huang. Pair-activity classification by bi-trajectories analysis. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.