TAMING CROWDED VISUAL SCENES

by

SAAD ALI
B.S. Ghulam Ishaq Khan Institute
M.S. University of Central Florida

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the School of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2008

Major Professor: Mubarak Shah

ABSTRACT

Computer vision algorithms have played a pivotal role in commercial video surveillance systems for a number of years. However, a common weakness among these systems is their inability to handle crowded scenes. In this thesis, we have developed algorithms that overcome some of the challenges encountered in videos of crowded environments such as sporting events, religious festivals, parades, concerts, train stations, airports, and malls. We adopt a top-down approach by first performing a global-level analysis that locates dynamically distinct crowd regions within the video. This knowledge is then employed in the detection of abnormal behaviors and tracking of individual targets within crowds. In addition, the thesis explores the utility of contextual information necessary for persistent tracking and re-acquisition of objects in crowded scenes.

For the global-level analysis, a framework based on *Lagrangian Particle Dynamics* is proposed to segment the scene into dynamically distinct crowd regions or groupings. For this purpose, the spatial extent of the video is treated as a phase space of a time-dependent dynamical system in which transport from one region of the phase space to another is controlled by the optical flow. Next, a grid of particles is advected forward in time through the phase space using a numerical integration to generate a "flow map". The flow map relates the initial positions of particles to their final positions. The spatial gradients of the flow map are used to compute a Cauchy Green Deformation tensor that quantifies the amount by which the neighboring particles diverge over the

length of the integration. The maximum eigenvalue of the tensor is used to construct a forward Finite Time Lyapunov Exponent (FTLE) field that reveals the Attracting Lagrangian Coherent Structures (LCS). The same process is repeated by advecting the particles backward in time to obtain a backward FTLE field that reveals the repelling LCS. The attracting and repelling LCS are the time dependent invariant manifolds of the phase space and correspond to the boundaries between dynamically distinct crowd flows. The forward and backward FTLE fields are combined to obtain one scalar field that is segmented using a watershed segmentation algorithm to obtain the labeling of distinct crowd-flow segments. Next, abnormal behaviors within the crowd are localized by detecting changes in the number of crowd-flow segments over time.

Next, the global-level knowledge of the scene generated by the crowd-flow segmentation is used as an auxiliary source of information for tracking an individual target within a crowd. This is achieved by developing a *scene structure-based force model*. This force model captures the notion that an individual, when moving in a particular scene, is subjected to global and local forces that are functions of the layout of that scene and the locomotive behavior of other individuals in his or her vicinity. The key ingredients of the force model are three floor fields that are inspired by research in the field of evacuation dynamics; namely, *Static Floor Field* (SFF), *Dynamic Floor Field* (DFF), and *Boundary Floor Field* (BFF). These fields determine the probability of moving from one location to the next by converting the long-range forces into local forces. The SFF specifies regions of the scene that are attractive in nature, such as an exit location. The DFF, which is based on the idea of active walker models, corresponds to the virtual traces created by the movements of nearby individuals in the scene. The BFF specifies influences exhibited by

the barriers within the scene, such as walls and no-entry areas. By combining influence from all three fields with the available appearance information, we are able to track individuals in high-density crowds. The results are reported on real-world sequences of marathons and railway stations that contain thousands of people. A comparative analysis with respect to an appearance-based mean shift tracker is also conducted by generating the ground truth. The result of this analysis demonstrates the benefit of using floor fields in crowded scenes.

The occurrence of occlusion is very frequent in crowded scenes due to a high number of interacting objects. To overcome this challenge, we propose an algorithm that has been developed to augment a generic tracking algorithm to perform persistent tracking in crowded environments. The algorithm exploits the contextual knowledge, which is divided into two categories consisting of *motion context* (MC) and *appearance context* (AC). The MC is a collection of trajectories that are representative of the motion of the occluded or unobserved object. These trajectories belong to other moving individuals in a given environment. The MC is constructed using a clustering scheme based on the Lyapunov Characteristic Exponent (LCE), which measures the mean exponential rate of convergence or divergence of the nearby trajectories in a given state space. Next, the MC is used to predict the location of the occluded or unobserved object in a regression framework. It is important to note that the LCE is used for measuring divergence between a pair of particles while the FTLE field is obtained by computing the LCE for a grid of particles. The *appearance context* (AC) of a target object consists of its own appearance history and appearance information of the other objects that are occluded. The intent is to make the appearance descriptor of the target object more discriminative with respect to other unobserved objects, thereby reducing the possible confusion

between the unobserved objects upon re-acquisition. This is achieved by learning the distribution of the *intra-class* variation of each occluded object using all of its previous observations. In addition, a distribution of *inter-class* variation for each target-unobservable object pair is constructed. Finally, the re-acquisition decision is made using both the MC and the AC.

To my loving parents

ACKNOWLEDGMENTS

I must begin by thanking my advisor Dr. Mubarak Shah for his guidance and relentless support. His commitment to hard work was a great source of inspiration over all these years, and motivated me to always do my best. I enjoyed working in the demanding research environment provided by him. I would like to thank my colleagues with whom I interacted and worked on joint projects, including, Omar Javed, Saad M. Khan, Yaser Sheikh, Vladimir Reilly, Paul Scovanner, Jingen Liu, Arslan Basharat, Mikel Rodriguez, Alper Yilmaz, Asaad Hakeem, and Fahd Rafi. I also want to thank Dr. Xin Li for all the discussion sessions that greatly helped me in shaping my research. I also want to thank my Ph.D. committee that includes Dr. Marshall Tappen, Dr. Kenneth Stanley, and Dr. Kiminobu Sugaya. However, this list would be incomplete without acknowledging the constant support and encouragement of my family, especially, my parents.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

The objective of this work is to overcome the challenges posed by high density crowded scenes. Despite the concerted effort of the computer vision research community, intelligent surveillance systems that process video feeds from real-world scenes like train stations, airports, city centers, malls, concerts, rallies, sporting events, etc., have not yet attained the desirable level of applicability and robustness. This is largely due to the algorithmic assumptions about the *density of objects* in a scene that are often violated in the real world environment.

This thesis develops methods that address some of the critical aspects of handling a crowded visual scene. It adopts a top down approach, and starts by performing a global level analysis that automatically locates dynamically distinct crowd regions/ groupings present in the scene in terms of *crowd-flow segments*. This global knowledge is then employed, not only to localize abnormal behaviors in the crowd, but also to facilitate other related tasks such as the tracking of individuals within the crowd. In addition, this thesis explores the utility of *context information* for persistent tracking and re-acquisition of objects in crowded scenes.

It is important to note that ‘crowded visual scene’ is a generic term that refers to any visual scene that contains a high density of objects. These objects can be of a variety of types including but not limited to people, cars, a school of fish etc. For example Figure 1.1 (a) & (b) show examples of ‘crowded visual scenes’ that contain crowds of people. In contrast, Figure 1.1 (c) &

Figure 1.1: Few instances of crowded visual scenes containing objects of different modalities. (a) & (b) Crowded visual scenes containing people. (c) Crowded visual scenes containing cars. (d) Crowded visual scenes containing a school of fish.

(d) respectively show examples of 'crowded visual scenes' where cars and a school of fish are the objects of interest. Although, the primary focus of this work is on the scenes that contain crowds of people, results will be reported on scenes with other types of objects as well to emphasize the generic nature of the techniques developed in this thesis.

1.1 Motivation

Large gatherings of people at locations such as train stations, airports, city centers, malls, transportation terminals, concerts, political rallies, sporting events, religious festivals etc., pose significant challenges to public safety management officials from the scene monitoring point of view. Often at *events* involving a large gathering, crowds of people move through confined spaces such

(a) (b) (c)

Figure 1.2: Instances of events that involve thousands of people. (a) Participants in the yearly New York City marathon. (b) A crowd at a train station. (c) A gathering of pilgrims circling around the Kabba in Mecca.

as city streets, overhead bridges or narrow passageways. Some example scenarios of this type of situation are illustrated in Figure 1.2. Figure 1.2(a) shows people participating in the yearly New York City Marathon. Approximately 37,000 people participated in the 2005 event. Figure 1.2(b) displays a scene where the crowd is moving towards the exit at a busy train station. Similarly, Figure 1.2(c) shows a scene where thousands of pilgrims are circling around the Kabba in Mecca. It is quite obvious that incident free management of such huge gatherings is a daunting task simply due to the shear numbers of people involved in these events.

One way to reduce the incidence of any catastrophic event in situations involving large crowds is through better coordination and remodeling of the expected bottleneck areas. However, numerous events of stampedes in the recent past have shown that coordination between public safety organizations and remodeling alone, cannot solve the problem of the management of large crowds. For instance on January 2005, 265 people lost their lives in a stampede near a temple in Maharash-

tra, India. Another such event happened in the PhilSports Arena in Philippines on February 2006 where 74 spectators were killed. On another occasion, 270 and 251 pilgrims were killed at Jamarat Bridge in Makkah in May 1994 and February 2004, respectively. This led to the redesigning of the approach to the bridge and the exit points, but unfortunately on January 2006, 345 lives were again lost at the same bridge due to a stampede. Since most of these gatherings are constantly monitored by a network of cameras, I believe that computer vision algorithms can make a significant contribution towards the management of large crowds. By analyzing and studying the behavior of the crowd through a vision system, it is possible to work in consonance with the public safety officials to ensure the safety of the public With the technological development in the field of computer vision in the past decade, one would assume that computer vision based algorithms should be able to predict congested spots, abnormal flows, crowding, and any other out of ordinary situation in its infancy, which will allow the human resources on the ground to take quick remedial action.

Tracking people in crowds, and inference about their individual and collective behavior, is a problem that arises in a variety of different contexts. For instance, at *locations* such as train stations, airports, city centers, malls, transportation terminals, etc., security personal watching the video-feed might be interested in tracking a few suspicious individuals within the crowd, to keep an eye on their activities. An example scenario is demonstrated in Figure 1.3, where during the morning rush hour at Liverpool train station, the majority of the people are walking along the red trajectories, but a person within the green ellipse is strolling suspiciously in the opposite direction. In crowded situations it is quite common to lose track of target objects due to severe occlusion arising from both the interaction of targets object with other members of the crowd and

Figure 1.3: Three frames of the video showing morning rush hour at Liverpool train station. The red trajectories are depicting the dominant direction along which the crowd is moving. However, the individual in green ellipse is walking suspiciously in the direction opposite to the dominant flow. A security personal will be interested in tracking this person among the crowd to gather more information about his behavior.

the structure of the scene. Therefore, it would be necessary to have a repertoire of algorithms that can help in overcoming these difficulties by using all the sources of information that can be extracted from the given scene.

Another motivating factor for developing algorithms specifically for crowded scenes is the absence of automated surveillance systems for crowded situations. The main components of current state of the art video surveillance systems often perform tasks such as the localization of moving objects; persistent tracking of targets; understanding of scene semantics; and application of the scene semantics in conjunction with tracking data for the detection of abnormal events and suspicious behaviors. However, a common weakness of these systems is their inability to handle crowded scenes. As soon as the density of the objects in the scene increases, a degradation in their

performance is usually observed. A quick glance at the research literature and industrial applications reveal that automated surveillance systems for crowded situations are almost non-existent. Limited, if any, research effort has been spent in building computer vision systems that can model high density crowded scenes and provide useful information for public safety officials. One obvious reason for the lack of effort in this direction is the complexity and challenges inherent in the problem. I examine some of these challenges in the next section.

1.2 Challenges

Successful techniques for handling a crowded visual scene must address a variety of problems:

- **Choice of Granularity**: The mechanics of human crowds is complex as a crowd exhibits both dynamics and psychological characteristics which are often goal directed. In addition, a scene may contain a high or extremely high density crowd. This makes it very challenging to come up with an appropriate level of granularity for modeling the dynamics of a crowd. Should a pixel based model, individual based model or something in between be used? This question needs to be answered for appropriate modeling of crowds in videos.

- **Representation of Abnormal Behavior**: Interactions between participants are indiscernible in crowded scenarios, and therefore individual centric representation of abnormal behaviors in crowds is implausible. In addition, an abnormal behavior or situation in a high density crowded scene often spreads very quickly, which makes it even more challenging to develop

a general appreciation of the abnormal situation by gleaning information from an individual's behavior.

- **Few Pixels on Objects**: In crowded situations, detection of individual objects becomes extremely hard as the number of pixels on the object decreases with the increasing density of the objects in the scene. The appearance information becomes further distorted due to the constant interaction among individuals making up the crowd (Figure 1.2 (a) - (c)).

- **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 target objects makes it difficult to persistently track the objects. This entails looking into other available sources of information to disambiguate the appearance information.

- **Effects of Terrain & Scene Features**: Physical characteristics of a scene can act as sources of occlusion resulting in the loss of observations of the target objects. However, if properly handled, the characteristic features of the scene can be used as cues for the tracking and reacquisition of objects.

1.3 Nomenclature

Many terms that are employed to describe phenomenon related to crowded visual scenes in this thesis are used in a some what loose manner in the literature. So to avoid confusion, I first provide definitions/explanation of the terms:

- The term *Crowded Visual Scene* is used to refer to a video stream that contains a high density of objects. The density itself is divided into three categories, namely: moderate, high, and extremely high.

- *Segmentation* - the task of dividing a given crowded visual scene into dynamically distinct crowd regions/groupings.

- *Flow Segment* - corresponds to a crowd region that moves in a coherent fashion in terms of user defined constraints. Each flow segment represents motion information which is global in nature and also has an associated physical interpretation for the given scene.

- The term *Abnormal Event* and *Abnormal Behavior* are used interchangeably, referring to a region of the scene where the behavior of the crowd is different from its learnt patterns.

- *Context* - refers to the contextual knowledge present in the crowded scene. It will be preceded by word ‘appearance’ or ‘motion’ to emphasize the modality of the context.

- *Scene Structure* - is composed of the characteristics of a scene which can be ‘physical’ or ‘virtual’ in nature. The physical characteristics correspond to scene features such as physical barriers, walls, entry & exit points etc. While the virtual characteristics correspond to the scene information such as virtual barriers separating different crowd groupings, distance to the exit locations, dominant paths etc.

- *Occlusion* - refers to inter-object occlusion resulting from the interactions of objects among themselves, and intra-object occlusion resulting from the interactions of objects with the scene.

1.4 Contributions

In this thesis, I have developed crowd-flow segmentation, abnormal event detection, target tracking and target reacquisition algorithms for crowded visual scenes. Unlike the traditional methods of processing a surveillance video, I start by performing a global level analysis to generate a representation of the scene which captures both the dynamics of the crowd and the structure of the scene. The global level analysis eliminates the need for low level change detection algorithms, and individual object localization/tracking. In particular, this is achieved by developing a crowd-flow segmentation framework which employs Lagrangian particle dynamics to uncover the spatial organization of the crowd. The segmentation information is then used to detect any temporal change in the behavior of the crowd enabling the localization of abnormal events/behaviors. Next, the crowd segmentation information is used in conjunction with the scene structure to develop a tracking algorithm that can be used to track an individual object of interest within the crowd. For this purpose, the structure of the scene is encoded in terms of 'floor fields', which are used to constrain the likely locations that an object can pursue while moving in the scene. Finally, I propose a target reacquisition algorithm that is employed to reduce the incidence of broken trajectories resulting from frequent occlusions in the crowded scenes. The proposed re-acquisition algorithm makes use

of the contextual information by building the appearance and motion context of the target object, which is subsequently used for re-acquiring the object when it re-appears.

After summarizing our contributions, I shall next introduce the crowd segmentation, abnormal event detection, target tracking, and re-acquisition approaches in more detail.

1.4.1 Crowd-Flow Segmentation

The first algorithm developed in this thesis performs crowd flow segmentation of the video depicting a crowded scene, and uses it for the detection of any abnormal event taking place in the crowd. It starts by treating the spatial extent of the video as the phase space of a non-autonomous (or time dependent) dynamical system, where transport/motion from one region of the phase space to the other is controlled by the optical flow. The idea is that the optical flow of a general scene will help in revealing the regions of qualitatively different dynamics in the phase space of the dynamical system (which is a video in this case). These different regions will reflect the distinct crowd groupings emerging from the spatio-temporal interactions of the members of the crowd with each other and with the physical world.

At the heart of our approach is the idea of *Lagrangian Particle Dynamics* which is used to uncover the spatial organization of the crowd. Traditionally Lagrangian Particle Dynamics refers to examining a cloud of particles as it mixes and is transported under the action of a time dependent optical flow field. The implication of using multiple optical flow fields to examine the temporal behavior of particles is that it helps in assimilating/integrating the motion information over longer durations of time. This is important for the analysis of complex temporal behaviors

or structures exhibited by a moving crowd. Next, the key theoretical notion that I use is the existence of *Lagrangian Coherent Structures (LCS)* [5] in the phase space, which are the invariant manifolds of a time dependent dynamical system. Roughly speaking, Coherent Structures are separatrices/material lines that influence the kinematics of the particle cloud over finite time intervals, and divides the flow, and in turn the phase space, into dynamically distinct regions, where all the particles within the same region have a similar fate, or in other words coherent behavior. Therefore, these structures can help in revealing the geometry of the crowd-flow in the video where they map to the boundaries of different crowd segments/regions.

The LCS are located using a Lyapunov Exponent approach which employs individual particle trajectories to generate a finite-time Lyapunov exponent (FTLE) field. Ridges in this field are a good estimate of LCS, and act as edges separating dynamically distinct crowd regions from each other. I compute two types of LCS: 1) Attracting LCS and 2) Repelling LCS. The attracting LCS, represented by a forward FTLE field, are computed by advecting the particle cloud forward in time, while the repelling LCS, represented by a backward FTLE field, are computed by advecting the particle cloud backward in time. The two FTLE fields are combined to generate a scalar field which is used in a watershed segmentation scheme to generate dynamically distinct crowd-flow segments.

Next, I use the crowd-flow segmentation information to detect change in the behavior of the crowd from its learnt pattern. Our formulation of the crowd-flow segmentation allows us to accomplish this task by simply detecting the presence of new flow segments from one time instant to the next. This is true because the difference in the dynamics of any part of the crowd-flow will give

rise to LCS exactly at the location where the change in dynamics is taking place. New LCS will eventually give rise to new flow segments, and by detecting these new segment one can pinpoint which part of the flow is deviating from its normal behavior.

1.4.2 Tracking Individual Targets in Crowded Scenes

The second algorithm developed in this thesis performs tracking of an individual object in a crowded scene. In our formulation of the problem, a crowd-flow is seen as a collection of mutually interacting particles. This is a reasonable assumption, because when people are densely packed, individual movement is restricted, and members of the crowd can be considered as granular particles interacting with each other. To track a specific individual in the crowd, I model the instantaneous movement of that person (particle) using a matrix of preferences which contain the probabilities of a move in a certain preferred direction. The probabilities take into consideration multiple sources of information, ranging from the appearance of individual target to the structure of the scene.

The influences resulting from the structure of the scene are represented in terms of a *scene structure based force model*. This model captures the notion that an individual, moving in a particular scene, is subjected to forces that are functions of the geometry of that scene and the locomotive behavior of other individuals in the immediate vicinity. The scene structure is incorporated into the tracking algorithm by introducing a concept of *floor fields*, which models the interactions between pedestrians and their preferred direction of movements by transforming the long ranged forces into the local ones. The transition probability of a tracked person depends on the strength of the floor

field in his/her neighborhood, such that transitions in the direction of larger fields are preferred. For instance, a long range force that is compelling the individual in a crowd to move towards the exit door can be converted into a local force by increasing the instantaneous probability of move in that direction.

I compute three such floor fields, namely: a 'Static Floor Field' (SFF), a 'Boundary Floor Field' (BFF), and a 'Dynamic Floor Field' (DFF). Here, the SFF field specifies the regions of space which are more attractive, e.g. an exit, dominant direction of motion etc.; while the BFF specifies the regions in the scene which are more repulsive e.g. barriers, no-go areas etc. The DFF corresponds to the virtual traces created by the movements of individuals in the scene, and in turn influences the motion of the individual being tracked. These floor fields taken together represent the 'scene structure force model'.

For any given scene, the SFF is computed only once during a learning period and it does not change with time. It is constructed by using a 'sink seeking' scheme that computes the distance to the nearest exit in the scene for every pixel location. The distance is defined as the number of steps required to exit the scene by wading through a smoothed optical flow field generated by the crowd-flow. In order to move from one point to the other, the velocity at that point is estimated as the weighted sum of the velocities of its neighbors. The weights are computed using a kernel density method. The DFF is computed at each time instant by using a sliding window of frames where the optical flow is computed between the consecutive frames thus resulting in a stack of optical flow fields. Next a grid of particles is overlaid over the first flow field. The particles are advected through the volume of optical flow fields using a numerical integration scheme. During

the advection, whenever a particle jumps from one pixel location to one of the neighboring pixels, the value of the interaction between these two pixels is increased by one. Repeating the process for all the pixels allows calculation of the DFF for the current time instant. The computation of the BFF is based on the crowd-flow segmentation algorithm described in the previous section where the boundaries of the crowd-flow segments represent physical and virtual barriers present in the scene. In order to generate the BFF, an edge map is created from the segmentation by retaining only the boundary pixels of each segment. The closest distance to barrier of each pixel is determined by computing a distance transform of this edge map. The distance transformed edge map represents the BFF.

1.4.3 Target Re-acquisition

The third algorithm developed in this thesis performs the reacquisition of target objects in the presence of an occlusion. The phenomenon of occlusion is very frequent in crowd videos due to high density of distracting objects in the scene. Sometimes physical features of the scene and camera motion also cause occlusion, resulting in the loss of visibility of the tracked object. Since the trajectories of moving objects are critical for understanding their behavior, any missing information will result in a significant degradation in the accuracy of any event recognition algorithm that uses these trajectories. Therefore, the objective of the re-acquisition algorithm is to reduce the incidence of missing information by augmenting the capability of the tracking algorithm to reacquire the target object after occlusions.

The main thrust of our proposed re-acquisition algorithm is to exploit the contextual knowledge present in the scene. I divide this contextual knowledge into two categories, namely *motion context* (MC) and *appearance context* (AC). The MC is based on the insightful observation that *the locomotive behavior of an object (e.g. people, vehicles) in a given environment provides information about the locomotive behaviors of nearby objects that are in the same environment.* The AC is based on the notion *that when a target object re-appears after undergoing occlusion, its appearance will have to be discriminated with respect to the appearance of other unobserved objects*.

Specifically, the (MC) is a collection of trajectories which are representative of the motion of the occluded/unobserved object. These trajectories belong to other moving objects in a given environment. The MC is constructed using a clustering scheme based on the Lyapunov Characteristic Exponent (LCE), which measures the mean exponential rate of convergence or divergence of the nearby trajectories in a given state space. Next, the MC is used to predict the location of the occluded/unobserved object in a regression framework. The *appearance context* (AC) of a target object consists of its own appearance history and the appearance information of the other objects which are currently occluded. The intent is to make the appearance descriptor of the target object more discriminative with respect to the other unobserved objects, thereby reducing the possible confusion between the unobserved objects at re-acquisition. This is achieved by learning the distribution of *intra-class* variation of each occluded object using all of its previous observations. In addition, a distribution of *inter-class* variation for each target-unobservable object pair is constructed. Finally the re-acquisition decision is made by using both the MC and the AC.

1.5 Organization of the Thesis

The structure of the thesis is as follows: **Chapter 2** reviews existing literature that focuses on handling different aspects of crowded visual scenes. It also discusses related approaches from other fields such as fluid dynamics, oceanography, and crowd dynamics. **Chapter 3** presents the crowd-flow segmentation framework, discusses the assumptions, and details the steps involved in the mathematical modeling of crowded scenes. Results are shown on very challenging sequences gathered from a variety of online resources. **Chapter 4** introduces the tracking algorithm that is specifically designed for tracking individuals in crowded scenes. The chapter discuss the steps involved in the construction of the floor fields, and show how they can be integrated into a tracking methodology. **Chapter 5** develops a target re-acquisition algorithm and elaborates on the regression-based prediction framework. The thesis is concluded in **Chapter 6** with a summary of contributions and description of future work.

CHAPTER 2
LITERATURE REVIEW

In this chapter I review the methods that have been developed to handle different aspects of crowded visual scenes. I have divided the chapter into two main parts. In the first part, I cover the algorithms and techniques that are used for detection, tracking, and event analysis in crowded scenes. I also describe a popular technique of modeling crowd/pedestrian flow dynamics. In the second part of the chapter I target the literature on object association under different camera setups.

2.1 Analysis of Crowded Scenes

The research on the analysis of crowded visual scenes can be categorized on the basis of the specific task that each work is trying to solve. The solutions to these tasks may use the methodologies developed in the field of computer vision or in other related research disciplines. I categorize these tasks as follows:

- Detection in (of) Crowds
- Tracking in Crowds
- Events in Crowds
- Modeling Crowd-Flow Dynamics

The research literature relevant to each of these categories is discussed in detail now.

2.1.1 Detection in (of) Crowds

The aim of this task is to develop algorithms that are capable of either localizing the individuals making up the crowd, or the crowd itself, in images and videos. It also includes methods that try to estimate the density of the crowd or explicitly count the number of people in the crowd.

2.1.1.1 *Detection of Individuals in Crowd*

The goal of this task is to *detect individual persons that are part of the crowd*. A number of vision algorithms have been proposed to achieve this goal [12, 13, 14, 15]. Zhao *et al.* [12] proposed a model based on a segmentation scheme for localizing people in a crowded scene. The problem was posed in a Bayesian framework where each person was localized by maximizing the posterior probability of matching 3D human shape models with the foreground blobs. The initial human hypotheses were generated by detecting the locations of heads in the foreground blobs. This method works well on a low density crowded scene but (it) is not scalable to high density situations where quite often the complete human body is not visible. In Wu *et al.* [13], an image-based detection approach is proposed which uses part based detectors consisting of edgelet features to localize individuals in crowds.

Detection of individuals using interest points and their trajectories has also been explored by a number of researchers in previous years. For instance, Tu *et al.* [14] proposed a global annealing optimization framework for segmenting individuals in crowds using clustering of interest points based on their geometric association with each other. The detection was only performed on images where the crowd was being viewed from above. This camera setup limits the types of scenes that

can be handled by their approach. Brostow *et al.* [15] proposed a Bayesian clustering scheme for grouping trajectories based on their space-time proximity. Their method tracked simple image features and probabilistically grouped them into clusters representing independently moving entities/individuals. The space-time proximity and trajectory coherence through the image space were used as the probabilistic criterion for the clustering.

In [16], Huang *et al.* proposed a stereo-based head detection algorithm for human detection in crowds. The algorithm was based on the assumption that at public places like airports, railway stations, shopping centers etc, the camera usually looks at a scene from a high position, and, since every person in the crowd occupies a 3D volume in space, the human heads are isolated from each other even in crowds. Their algorithm consisted of three steps: first, a scale adaptive filtering was performed to extract hypotheses of head like objects; second, a perspective correction was performed to suppress spurious hypotheses which had much higher or lower than average human height; third, mean-sift was used to locate human heads in the likelihood map. The results were reported on a low density scene. In [17], Faulhaber *et al.* proposed a different method based on Haar-wavelet features for head detection in crowded scenes. They used the assumption that heads of pedestrians in crowds form a texture which can be distinguished from the scene background by using wavelet features.

Recently, Dong *et al.* [18] has developed an algorithm that detects individual pedestrians from a foreground blob generated by a background subtraction algorithm. In a crowded situation such a blob often contains more than one person. Their algorithm used Fourier descriptors and an indexing scheme that mapped the observed descriptors to the parameter set explaining the shape.

There is another set of research work which employs laser range scanners for detection of moving objects in crowded situations. For instance, in [19], Fod *et al.* used multiple laser ranger scanners at waist height to localize objects. The background modeling of the laser scan image was performed to detect foreground blobs belonging to the moving entities. Again in [20], Zhao *et al.* employ a laser range scanner at the ground level for detection and tracking. At the ground level, each person generates two point clusters corresponding to two feet. These two clusters were grouped together using distance constraints to detect an individual. Another related method for crowd detection using a laser range scanner was proposed by Cui *et al.* [21].

The main limitation of these methods is that they are inherently designed to perform detection in videos of a low density crowd. These methods tend to be impractical when the number of objects present in the scene is large and the objects interact in complex manner, as shown in Figure 1.1. The computed features such as interest points, location of heads, foreground blobs, and color histograms also become noisy and unreliable. To overcome this shortcoming, I contend that in a scene of a high density crowd, detection of individual objects may not be necessary, and therefore, modeling the crowds at a global or holistic level is more practical. For this purpose, I propose a crowd-flow segmentation algorithm in Chapter 3, which is capable of locating the dynamically distinct crowd segments in a scene, and uses it for abnormal event detection and behavior analysis.

2.1.1.2 *Crowd Detection/Segmentation*

As mentioned before, in a scene containing a high density crowd, detection of individual objects in videos or images may not be possible. However, groups of people that share some feature, e.g.

direction of motion, appearance, collective behavior etc., can be identified more easily. In the past few works have attempted to do this. For instance, Riesman *et al.* [22] proposed a method to detect crowds in a video stream. The main thrust of their idea was to analyze xt slices of spatio-temporal volume of the given video to compute probability distribution of left and right inward motion, and then use these distributions to infer the crowd location.

Chan *et al.* [23, 24] proposed to segment the videos of crowded environments using a representation based on mixture of dynamic textures. A dynamic texture is defined as a sample from a stochastic process over space and time. The idea behind using a mixture of dynamic textures is that a video can be perceptually decomposed into multiple regions, each of which belongs to a semantically different visual process. In case of a crowd video, these different visual processes will correspond to different groups of people, or in case of a highway video, to traffic moving in opposite directions. Specifically, Chan *et al.* [23, 24] developed a generative model of a dynamic texture mixture, where a collection of video sequences (or video patches) were modeled as samples from a set of underlying dynamic textures. They also derived an expectation-maximization (EM) algorithm for maximum-likelihood estimation of the parameters of the dynamic texture mixture.

These approaches for crowd segmentation do not take into account the goal-directed nature of human crowds. Large crowds of pedestrians at sporting events, religious festivals, train-stations etc. can be described as goal directed and rational because the members of the crowds have clear knowledge of what and where their goals lie. I incorporate this observation into the crowd segmentation framework where segments are distinguished from each other on the basis of the fate

of the particles belonging to that segment. The particles with similar fate have similar goals, and, therefore, characterize a distinct group of the crowd in a given scene.

2.1.1.3 *Crowd Density Estimation*

The estimation of number or density of people in an area under surveillance is very important for the problem of crowd monitoring. The initial research efforts ([25, 26, 27, 28, 29, 30, 31]) tried to address the crowd density estimation problem in the early to late nineties. Global image features such as foreground pixels, textures, edges, optical flows etc., were often utilized in this body of work. For instance, Davies *et al.* [25] estimated the number of foreground pixels or number of edge pixels from the image, and used them in a linear regression framework to estimate the number of people in the scene. Coinaz *et al.* [26] used sizes of foreground regions and ratio of foreground to background regions as features, and trained a fuzzy classifier that classified the scene into one of five categories: no people, a few people, some people, many people, overcrowding. In the same vein, Cho *et al.* [27] and Schofield *et al.* [28] trained neural networks to classify the level of a crowd. In [29], real-time estimation of crowd density was carried out by extracting a set of features which included a number of edge points, a number of maxima in the edge point histogram, and the sum of the amplitudes of the maxima in the edge point histogram. Although these pixel-based techniques were simple and fast, they are not reliable when the crowd density is high.

The texture-based methods ([30, 31]) followed which used crowd images of different densities as different texture patterns, and estimated the crowd density by texture analysis schemes. In [30], texture measures were extracted from the images through gray level dependence matrices, straight

line segments, Fourier analysis, and fractal dimensions. The estimations of crowd densities were given in terms of the classification of the input images into five classes of densities (very low, low, moderate, high and very high). This method was extended in a later work [31] where the Minkowski fractal dimension was used for crowd density estimation. The utility of wavelet features for density estimation was also explored in the work of [32] and [33].

More recently, trajectory based information has been used for crowd density estimation and counting ([34, 35]). In [34], Rabaud *et al.* counted the number of people by segmenting the moving objects in a dense crowded scene. This is achieved by clustering a rich set of extended tracked features where spatial and temporal conditioning was used to overcome the fragmented nature of the tracks. In a parallel work, Antonini *et al.* [35] developed a trajectory clustering scheme for crowd counting. They used several data representations (Independent Component Analysis, time series, Maximum of Cross Correlation) and compared different distance/similarity measures (Euclidian, Longest Common Subsequence, Hausdroff) under a common hierarchical clustering framework. The hierarchy consisted of three stages: length clustering, spatial clustering and pedestrian counting.

A number of crowd counting algorithms have been developed that take into account feature normalization to deal with the perspective projection and different camera orientations. In this regard, [36, 37] described a viewpoint invariant learning method for counting people in crowds from a single camera. In [38], the density of persons was estimated by counting the foreground pixels with the weights based on perspective correction. These algorithms provide the benefit of viewpoint invariance in addition to easy deployment with at a new site.

In a slightly different flavor, [39] developed a system that counted people in a crowded scene using a network of multiple image sensors. They introduced a geometric algorithm for computing bounds on the number and possible locations of people using the silhouettes which were obtained from each sensor through background subtraction.

On the practical side, a number of commercial *crowd counting systems* have been developed. Albiol *et al.* [40] designed a vision system for the Spanish Railway Company to determine the number of people who get in and out of a train carriage. Zhang *et al.* [41] worked on an automatic pedestrian counting method for an escalator or a moving walkway. They used a model-specified directional filter to detect object candidate locations followed by a novel matching process to identify the pedestrian head positions. Terada *et al.* [42] employed stereo images for crowd counting at gate entrances. Similarly, using a camera hung from the ceiling of the gate, [43] proposed a real-time scheme to detect and track the people moving in various directions with a bounding box enclosing each person. Furthermore, [44] presented an automatic bi-directional people counting method dedicated to passing through a gate or door. Harasee *et al.* [45] used skin color modeling, iterative face detection, and tracking for people counting in transport vehicles. Bozzoli *et al.* [46] developed a system to estimate the number of people passing through a gate in a public area such as a metro or a railway station.

Our objective in this thesis is to analyze the crowds at a holistic level, and, therefore, I do not explicitly estimate the number of people or their density in the scene.

2.1.2 Tracking in Crowds

Tracking is one of the highly researched areas in the field of computer vision. Most tracking algorithms proposed over the years focus on the general problem of tracking, without specifically addressing the challenges of a crowded scene. In this section, I review the tracking methodologies that are specifically designed for crowded situations. The readers interested in a detailed review of the state of the art in tracking are referred to a recent survey by Yilmaz *et al.* [47].

To start with, few of the detection methodologies discussed previously have been used for tracking in crowded scenes as well. For instance, Zhao *et al.* [12] used the initial detection of people in crowds to initialize the ellipsoid-based human shape models and color histograms to carry out tracking. Similarly, Brostow *et al.* [15] tracked and clustered features points over-time, and, therefore, were able to generate a separate trajectory for each individual.

There is another interesting and relevant body of work that tries to track sparse crowds of ants [48], hockey players [49], crowds of clumped people [50, 51, 52], or a dense flock of bats [53] and biological cells [54]. In [48], Khan *et al.* employed a Markov chain Monte Carlo based particle filter to deal with interactions among targets in a crowded scenario. They used the intuitive notion that in a crowded situation the behaviors of targets are influenced by the proximity and/or behavior of other targets. The interactions among the targets were modeled by a Markov Random Field (MRF) based motion priors that were learnt on the fly using an MCMC sampling. The results were reported results on videos of interacting insects. Cai *et al.* [49] proposed a mutli-target tracking algorithm for tracking hockey players in a video. Using their approach, they were able to robustly track multiple targets and correctly maintain identities in the presence of background clutter, cam-

era motions and mutual occlusion between targets. The approach consisted of a modified particle filtering algorithm where they introduced a global nearest neighbor data association algorithm for assigning Ada-boost based detections to the existing tracks for the proposal distribution. In addition, the mean-shift algorithm was embedded into the particle filter framework to stabilize the trajectories of the targets for robust tracking during mutual occlusions.

Work of Gennari *et al.* [50] is aimed at scenarios where large numbers of targets form natural groups which can be efficiently tracked together. In their method, groups were defined on the basis of the position and velocity of targets. They used a set of merging and splitting rules which were embedded into a Kalman filtering framework for tracking multiple groups. In cases where groups of different velocities cross, a general methodology for matching measurements to groups was introduced. In [51, 52], Lin *et al.* advocated a different paradigm for tracking groups of people by treating them as a near-regular texture (NRT). The NRT is defined as a geometric and photometric deformation of a regular texture. For tracking purposes, the NRT was nested in a lattice-based MRF model of a 3D spatio-temporal space. Next, the tracking algorithm used the topological invariant property of the dynamic NRT by combining a global lattice structure that characterizes the topological constraint among multiple textons (people) and an image observation model that handles local geometry and appearance variations.

Recently, Betke *et al.* [53] proposed an algorithm to track a dense crowd of bats in thermal imagery. They combined multiple techniques such as multi-target track initiation, recursive Bayesian tracking, clutter modeling, event analysis, and multiple hypotheses filtering for this purpose. Impressive results were obtained by tracking up to approximately eight hundred thousand

bats. Tracking of multiple interacting and crowded objects has been attempted in the area of biological cell tracking as well. For instance, Li *et al.* [54] has recently developed an algorithm for tracking thousands of cells in phase contrast time-lapse microscopy images. The tracking was performed in two stages where at the first stage a track compiler operating in a frame-by-frame manner was producing intermediate tracking results, called track segments, which were linked into cell trajectories at the second stage by a track linker overseeing the entire tracking history. Another approach for tracking in crowded scenes using selective visual attention is proposed by Yang *et al.* [55]. In their algorithm, the early selection process extracts a pool of attentional regions that were defined as the salient image regions which have good localization properties, and the late selection process dynamically identified a subset of discriminative attentional regions through a discriminative learning of the historical data on the fly. They demonstrated tracking of complex targets in real-world sequences and movie clips.

Most tracking algorithm described so far only use low level image information for tracking purposes. Surprisingly little has been done in exploiting high-level cues for human detection and tracking in complex crowded situations. One of the few works on this topic is that of Antonini *et al.* [56] which used discrete choice models (DCM) [57] as motion prior to predict human motion patterns and fused this model in a visual tracker for improved performance.

The utility of multi-camera setups for tracking in crowds has also been explored. In this regard, Khan *et al.* [58] presented a homography constraint to fuse information from multiple views using geometrical constructs and resolved occlusions by localizing people on multiple scene planes. Mittal *et al.* [59] developed a system using multiple synchronized cameras for detection and track-

ing of multiple people in a crowded environment. There are several tracking approaches which specifically address the problem of occlusion. The traditional approach for detection of occlusion is by detecting blob merging [60]. The feature point based approaches define the occlusion as the disappearance of the point being tracked [61]. In recent years, tracking techniques using object contours [62, 63] and appearances [64, 65], which represent and estimate occlusion relationships between objects by using the hidden variables of depth ordering of objects toward the camera, have been proposed.

A crowded scene has a number of characteristics which makes the direct application of above-mentioned tracking algorithms extremely difficult. First, in high density crowds it is hard to discern individuals from each other, and therefore ownership of the features (color, spatial templates, interest points, contours, etc.) cannot be computed reliably. Second, severe occlusions occur due to interactions among the members of the crowd; therefore, even if reliable features are computed, tracking over longer durations of time is difficult. In order to overcome these difficulties, our tracking algorithm uses higher level knowledge about the scene which is the novel aspect of the algorithm. In other words, above-mentioned tracking methodologies are object centric, and do not exploit any high level knowledge that may aid the tracking algorithm. In the tracking algorithm, I incorporate the high level knowledge of the scene and the behavior of the crowd into the tracking algorithm by computing a number of *floor fields*. Another major difference is that the traditional crowd tracking algorithms are designed for low density scenes and are not extendable to a high density scene where it is difficult to determine the ownership of the features. Our algorithm, on the

other hand, performs tracking of individuals in high density crowded scenes containing *hundreds or thousands of people*.

2.1.3 Events in Crowds

Analysis of crowd behaviors is an important problem. It can be dealt with at the individual level where the event of interest is defined in terms of individual objects, or it can be defined at a global level where the behavior of the crowd is modeled at an extended spatial scale.

In the literature, the analysis of the global level behavior is often carried out by using the motion information described in terms of optical flow. This is different from the approaches which employ change detection algorithms to first detect foreground blobs and then use it for behavior analysis. Using the optical flow based motion information Velastin *et al.* [66] and Davies *et al.* [25] developed a block matching scheme to estimate the motion trends of the crowds. Specifically, they used frequency distribution of velocity directions for this purpose. Similarly, Bouchafa *et al.* [67] also used a block matching scheme for crowd monitoring in subway stations. The underlying assumption of their algorithm was that for detection of any abnormal activity the knowledge about direction of crowd motion is essential. In addition to block matching, they tested their method with two other ([68, 69]) optical flow algorithms as well. In a later work Bouchafa *et al.* [70] used the same technique for detection of abnormal individual or crowd motion in one-way subway corridors. Using the motion information, Yin [71] conducted a detailed study and showed that accurate estimation of crowd movements can be obtained through appropriate settings of the operating parameters (size of block, size of search window).

In the work of Andrade *et al.* [72, 73], the crowd behavior is characterized at a global level also by using the optical flow of the video sequence. During the learning stage, a reduced order representation of the optical flow was generated by performing Principal Component Analysis (PCA) on the flow vectors. The top few eigenvectors were used as the representative features and spectral clustering was performed to identify the number of distinct motion patterns present in the video. The features in the clustered motion segments were used to train different HMMs which were later used for event detection in crowds.

In [74], Boghossian *et al.* proposed to model the dynamics of the scene under consideration for the prevention of crowd-related emergencies in large crowds. They started by estimating the optical flow and clustered the optical flow vectors based on direction and magnitude to segment different crowds. Next they detected a number of events using a technique based on Hough voting space. The types of event detected by their method include circular flow paths close to site exits indicating trapped crowds; crowd-flow diverging from a point to all directions, which might indicate a potential danger (fights, fire etc.); obstacles in the flow paths that might correspond to injured pedestrians or deliberate flow disturbances. In their later work an automatic monitoring system was proposed for detecting overcrowding conditions on platforms of underground train services [75], and for determining the congestion levels on the platform [76].

Paragons *et al.* [38] proposed an MRF-based approach for real-time subway monitoring by carrying out change detection and congestion estimation. Their solution consisted of two steps: The first step was a change detection algorithm that distinguished the background from the foreground by using a discontinuity preserving MRF-based approach. In the MRF model, information

from different sources (background subtraction, intensity modeling) was combined with spatial constraints to provide a smooth motion detection map. In the second step the computed change detection map was combined with a geometry module to perform a soft auto-calibration to estimate a measure of congestion in the observed area (platform).

Recently Ke *et al.* [77] have proposed an algorithm for detection of single actor based events in crowded scenes. They handled the artifacts resulting from the partial occlusions and a cluttered environment. The recognition itself was performed by a part-based matching of a volumetric representation of an event against over-segmented spatio-temporal video volumes. The shape and flow features were used for the encoding information contained in each volume. Pham *et al.* [78] demonstrated event detection capabilities in thermal imagery of a crowded environment. Their method was based on the detection and segmentation of individuals within groups of people using a combination of several weak classifiers in a boosting algorithm.

The assumption of our abnormal event detection algorithm (Chapter 3) is that in a real crowd scene one cannot easily specify beforehand or train particular labels for behavior analysis. Therefore, the events are classified as normal or abnormal behavior without having any specific label for them. I believe this type of crude labeling will help in pinpointing the locations where the behavior of the crowd has changed, and will allow the safety management officials to take the remedial actions. Our abnormal event algorithm uses the crowd segmentation instead of optical flow which is one major difference with respect to the body of work described above.

2.1.4 Modeling Crowd-Flow Dynamics

Modeling pedestrian flow dynamics particularly in crowds has been a major topic of research in sociology and behavioral sciences. A number of models have been proposed for this purpose over the years. One well established way to study crowd dynamics is by discrete simulation of individual pedestrians. Discrete simulation forms a very useful numerical tool for practical applications, but as a research tool it suffers from lack of analytical tractability that makes deriving general results difficult. Some reference work that follow this line of modeling include [79, 80].

In the case of Cellular Automaton (CA) model [81], the local movements of the pedestrian are modeled with a matrix of preferences which contains the probabilities for a move, related to the preferred walking direction and speed, toward adjacent directions. Schadschneider [82] introduced the interesting concept of floor fields to model the long-ranged forces. These floor fields have their own dynamics (diffusion and decay), which are modified by pedestrians and in turn modify the matrix of preferences, thus simulating interactions between individuals and the geometry of the system. Simple behavioral rules are implemented (turning directions, obstacle avoidance) in order to reproduce more complex collective phenomena [83]. Several other approaches have been proposed and I refer the interested reader to Bierlaire *et al.* [84] for a detailed survey of the literature. It is important to note that in our work I am learning the floor fields directly from the observed data instead of designing them manually as has been the practice in the stimulation community.

Another famous pedestrian behavior model is the Social Force Model [79, 85]. In this model an individual is subject to long-ranged forces and his/her dynamics follow the equation of motion,

similar to Newtonian mechanics. The model has been empirically tested in a wider range of environmental settings and has shown compelling simulations of complex human group behaviors like lane formation, exit clogging, collision avoidance etc. Due to its simplicity and intuitive nature the social force model has become a pedestrian behavior model of choice in the behavioral sciences community.

A popular way of modeling crowds is in terms of attributes of a fluid. The examples include the work by Huges *et al.* [86, 87] and Henderson *et al.* [88]. In [86, 87], Huges *et al.* have developed a set of governing equations for high density crowd flows using the following three hypotheses: 1) the speed of a pedestrian depends on the density of the surrounding pedestrians, 2) pedestrians have a common sense of task and 3) pedestrians try to minimize their estimated travel time. These sets of equations are then used for studying the effects of barriers on the flow of the crowd. However, in this type of modeling it is assumed that the crowd will involve only a single pedestrian type which means these method are not be able to handle situations where multiple interacting crowd flows are present. Furthermore, these methods rely on the availability of accurate data about the crowd density in the scene. Unfortunately, there are no reliable means to measure such physical quantities using the video data, which makes these approaches impractical for a general scene.

2.2 Object Association

The proposed re-acquisition algorithm is related to a variety of previous works in the areas of track linking in UAV videos, multi-camera object association, appearance modeling, and context

modeling. In this section, I provide an overview of these related works and highlight similarities and differences of our approach.

2.2.1 Track Linking in Moving Cameras

With the growing interest in the area of aerial surveillance, vision researchers have explored techniques to perform object association or track linking in videos taken by UAVs. For monocular aerial cameras, Amitha *et al.* [91] recently proposed a framework, which builds upon their previous work [92], for linking tracks across occlusions . The object association problem is solved in two stages: In the first stage, one to one correspondence is established between the tracks seen at two different time intervals. The pairwise association probabilities are computed using temporal ordering, proximity of forward and backward estimation of the object's position, and similarity of appearance templates. The second stage improves the correspondence for splitting and merging of objects. Although I am employing the same camera setup, our approach differs from their work in several important ways: Amitha *et al.* [91] do not employ contextual knowledge of an object's kinematics or appearance for linking tracks. In addition, only a restricted set of object kinematics are managed using a linear motion model for prediction. This clearly is not the case in the real world where objects move along arbitrary paths. Our approach, on the other hand, can handle different types of motions by adapting to the kinematics of objects in the given scene through motion context.

Another related area of research has focused on associating tracks across multiple moving aerial cameras [93, 94]. In [93], Sheikh *et al.* proposed a method to correspond objects across

un-calibrated cameras that are mounted on aerial vehicles. It is assumed that the cameras have overlapping FOV, and a maximum likelihood estimate of object correspondence is computed using a graph theoretic approach. While in [94], the restriction on overlap between the FOVs is relaxed and track correspondences are established by fitting kinematic polynomial models to object trajectories. The parameters of polynomial and inter-camera homographies are estimated simultaneously in an Expectation Maximization algorithm. Again, our approach is different from their method as I do not explicitly impose any particular kinematic model on object trajectories, rather I infer the type of the motion from the contextual information.

2.2.2 Multi-Camera Object Association

Over the years, a number of algorithms have been proposed for associating objects in a multiple *stationary* camera setup. Although I am dealing with the problem of association in a single moving camera setup, the techniques proposed for stationary camera setups are worth mentioning. I will consider two scenarios: i) multiple stationary cameras with overlapping FOVs, and ii) multiple stationary cameras with non-overlapping FOVs.

2.2.2.1 *Multiple Stationary Cameras with Overlapping FOVs*

A large body of work has addressed the object association problem for this setup, beginning with the work of Nakazawa *et al.* [95], in which a state transition map was constructed that linked regions observed by one or more cameras along with a number of action rules to consolidate information between cameras. Cai *et al.* ([96]) proposed a method to track humans across a

distributed system of cameras, employing motion analysis on 3D geometry of the camera setup where the spatial matching was based on the Euclidean distance of a point with its corresponding epipolar line. Bayesian Networks for combining multiple cues were employed in a variety of papers. In [97], Chang *et al.* used Bayesian networks to combine geometry (epipolar geometry, homographies and landmarks) and recognition (height and appearance) based modalities to match objects across multiple sequences. Bayesian networks were also used by Dockstader *et al.* in [98] to track objects and resolve occlusions across multiple calibrated cameras. In [99], Khan *et al.* proposed an approach that avoided explicit calibration of cameras and instead used constraints on the field of view lines between cameras, learnt during a training phase, to track objects across the cameras. Weiming *et al.* ([100]) used the principal axis of foreground blobs for matching people across uncalibrated cameras. For this purpose, a relationship was established between the principal axis of a person in two views and the 3D "ground point" of the person.

2.2.2.2 *Multiple Stationary Cameras with Non-Overlapping FOVs*

Non-overlapping FOVs allow coverage of a far wider area. A number of algorithms have been proposed for carrying out object association within such a camera setup. An example of this is the work by Huang *et al.* [101], who developed a probabilistic appearance based approach for tracking vehicles across consecutive cameras on a highway. Constraints on the motion of the objects across cameras were first proposed by Kettnaker *et al.* [102], in which positions, object velocities and transition times across cameras were used in a setup of known path topology and transition probabilities. In [103], Collins *et al.* used a system of calibrated cameras with a environment model to

track objects across multiple views. The method proposed by Javed *et al.* ([104]) used a supervised framework for learning the camera topology and path probabilities of objects using Parzen windows and did not assume any site model. In [105], Stauffer *et al.* tracked objects across multiple cameras with both overlapping and non-overlapping FOVs, building a correspondence model for the entire set of cameras. Recently, Glibert *et. al* [106] proposed an unsupervised framework for learning camera topology.

Our proposed algorithm has several important differences from the above mentioned body of work. First, due to the nature of the setting in which airborne cameras operate, the topographic, appearance and learning based constraints of multi-camera object association are not easily extendable. Most of the aforementioned methods require a learning period to estimate various parameters of the system, however the UAVs on which cameras are mounted usually fly over an area only once, and therefore reliable learning of unknown parameters is not possible. Second, the multi-camera setups (overlapping and non-overlapping) that use motion patterns to acquire objects across camera views often address very restrictive cases. For instance, the solution suggested by Hunag *et al.* [101] is confined to setups where the cameras are placed along the side of a single path so that the movement of the objects is pre-determined, and can be predicted by linear motion models. Third, the above mentioned methods try to link tracks across occlusions and do not attempt to track them while they are occluded, as has been observed by [91]. Fourth, there is no notion of 'context' in the aforementioned algorithms, which means only the motion of the target object is used to make predictions about its potential future location. In other words, these algorithms ignore the state of the surrounding environment when making these predictions.

2.2.3 Appearance Modeling

Another technical component of our work is related to modeling of appearance of vehicles in aerial cameras. A number of papers have proposed solutions in this regard. For instance the work of Ying *et al.* [107, 108] used edge based measures to establish the association of cars across different cameras. The novel feature of their work was that instead of explicitly modeling the appearance of cars, they posed the problem as one of computing same-different probabilities. Guo *et al.* [109] proposed an alternative framework where objects were aligned and line based features were employed to match the objects with large pose variation. Ozge *et al.* ([110]) combined shape and appearance features for matching cars in aerial cameras. In [111], Javed *et al.* handled change in appearance across cameras by learning a brightness transfer function from a small training set. All of these approaches fall short of utilizing the contextual knowledge. In our proposed method, I am exploiting the contextual information available in the form of appearance history of not only the target object and but also other objects that are are present in the scene.

2.2.4 Context Modeling

A number of approaches have explored the utility of spatial context for modeling the appearance of targets for tracking purposes. The spatial context is defined as the features (e.g. color, interest points etc.) of the surrounding background of the object. This spatial context assists in making the distinction between the target and the surrounding background. For instance, in [112] those color features were selected for encoding the appearance, which were most discriminative with respect to the local background window. Similarly, [113] maintained an online foreground-background

discriminating function as the objective function during the target search. The spatial context was augmented by a temporal context for a fixed camera multi-target tracking in a parallel work by Nguyen *et al.* ([114]). The temporal context was constructed by integrating the entire history of target appearance using the probabilistic principal component analysis (PPCA) algorithm. The notion of temporal context is closely related to our definition of AC. However, the main difference is that I am incorporating not only the appearance history of the target object, but also the appearance history of other objects which are currently occluded. In addition, I use motion context which was not explored in [114].

The brief overview of the research literature underscores the fact that no attempt has been made so far to use the rich contextual knowledge present in a scene in terms of motion and appearance of inter-related objects. The is precisely where the main contribution of our re-acquisition algorithm (Chapter 5) lies, and I will show through experimental verification, the validity and usefulness of using the contextual knowledge. Now, in the next Chapter I describe the crowd-flow segmentation algorithm which is the first stage of processing the video of a high density crowded environment.

CHAPTER 3
A LAGRANGIAN PARTICLE DYNAMICS APPROACH FOR CROWD-FLOW SEGMENTATION

Crowd flow segmentation generates a global representation of the scene by locating all the distinct crowd regions/groups that are present in the scene. The emphasis of the approach is on locating those crowd groupings that are dynamically distinct and spatio-temporally dominant. In the later part of the chapter, the applicability of crowd segmentation is demonstrated on the task of abnormal event detection within crowds.

3.1 Overview

To achieve the goal of crowd-flow segmentation the proposed algorithm assumes that the spatial extent of the video is a phase space of a non-autonomous (or time dependent) dynamical system, in which transport from one region of the phase space to another is controlled by the optical flow. The idea is that, by observing the transport phenomenon under the influence of the time dependent optical flow, the regions of qualitatively different dynamics in the phase space will be revealed. These different regions of the phase space will have a one-to-one correspondence with the distinct crowd groupings emerging from the spatio-temporal interactions of the members of the crowd with each other and with the physical world.

The discovered crowd regions/groupings are called “Flow Segments”. To avoid any confusion, it is pertinent at this point to emphasize the distinction between the terms “Flow Segment” and “Optical Flow”. Traditionally, optical flow represents motion information that is local in space and time. There is no high level interpretation associated with this local information. In contrast, a “Flow Segment” represents a motion trend that is global in nature and has an associated physical interpretation in the context of the given scene, e.g., in a crowd video a “Flow Segment” represents a group of people whose behavior, in terms of dynamics, is distinct. A scene can have an arbitrary number of “Flow Segments” and each “Flow Segment” can have any arbitrary shape.

The crowd-flow segmentation algorithm developed in this chapter makes use of recent advances in the areas of nonlinear dynamical systems [1][2], fluid dynamics, [3][4][5] and turbulence theory [6][7]. The basis of the idea is to use Lagrangian Particle Dynamics to uncover the spatial organization of the flow field by examining a cloud of particles as it mixes and gets transported over time under the action of the optical flow generated by the crowd motion. At the conceptual level, the implication of using time-dependent optical flow fields to examine the temporal behavior of particles is that it helps in assimilating/integrating the motion information over longer periods of time. This integration is important for the analysis of complex temporal behaviors or structures exhibited by a moving crowd. In practical terms, the advection of particle cloud quantifies the transport between different regions of the phase space and, therefore, helps in revealing the representative characteristics of the phase space, such as locations of the barriers, mixing properties, sources, and sinks. As the phase space is directly related to the crowd video (a non-autonomous dynamical system), these characteristics have a direct relationship with the physical properties of the crowded scene,

such as physical and virtual barriers in the scene, the direction in which the crowd is going, the number of different crowd segments, and the locations at which segments merge or split.

However, in the proposed construction, I do not have to explicitly locate all these characteristic features of the phase space; instead, I use the key theoretical notion of *Lagrangian Coherent Structures (LCS)* [5], which are the invariant manifolds of the phase space. Roughly speaking, Coherent Structures are separatrices/material lines that influence the kinematics of the particle cloud over finite time intervals, and they divide the flow, and in turn the phase space, into dynamically distinct regions, where all the particles within the same region have a similar fate or, in other words, coherent behavior. The notion of coherent structure is extendable to phase spaces of the crowd videos, where they map to the boundaries of dynamically distinct crowd regions/groupings. Intuitively, coherent structure is to flow data what "edge" is to image data. Note that when coherent structures are studied in terms of quantities derived from particle trajectories, they are named as Lagrangian Coherent Structures (LCS).

Now, a fundamental question is how to locate LCS in the given phase space. Several approaches have been proposed to compute LCS based on whether the underlying dynamical system is periodic [10], aperiodic [9], or quasi-periodic. The crowd movements fall under the category of aperiodic motion (or time-dependent motion), since often in a given scenario there is no constraint on the dynamics of the crowd that includes both speed and direction. The Lyapunov Exponent approach is adopted in this work to locate the LCS. The Lyapunov exponent measures the exponential rate of convergence or divergence between two particle trajectories. For a given crowd video, I use a grid that covers the optical flow field of the video and compute the finite-time estimate of Lyapunov

exponents for trajectories starting at each point of the grid. This process returns a finite-time Lyapunov Exponent (FTLE) field over the phase space. It has been shown by Haller [5] that the coherent structures appear as ridges in the FTLE field and govern the mixing and transport of the particles.

Therefore, I treat ridges, which are coherent structures of the phase space, in this field as edges that separate flow segments that have different dynamics from each other. I compute two types of LCS: 1) attracting LCS and 2) repelling LCS. The attracting LCS, represented by a forward FTLE field, are computed by advecting the particle grid forward in time, while the repelling LCS, represented by a backward FTLE field, are computed by advecting the particle cloud grid in time. The two FTLE fields are combined to generate a scalar field that is then used in a watershed segmentation scheme to generate dynamically-distinct, crowd-flow segments. Note that only finite-time estimates of Lyapunov exponents are of interest, because the flow field in any physical context only has a finite time to operate on the particle/tracer. In most cases, these finite time estimates turn out to be a good estimate of the infinite-time exponents. The steps of the crowd-flow segmentation algorithm are summarized in the block diagram of Figure 3.1.

In the next section, I describe the mathematical notations, provide formal definitions, and explain some of key concepts in more detail. The nomenclature of Shadden *et al.* [8] is used for this purpose.

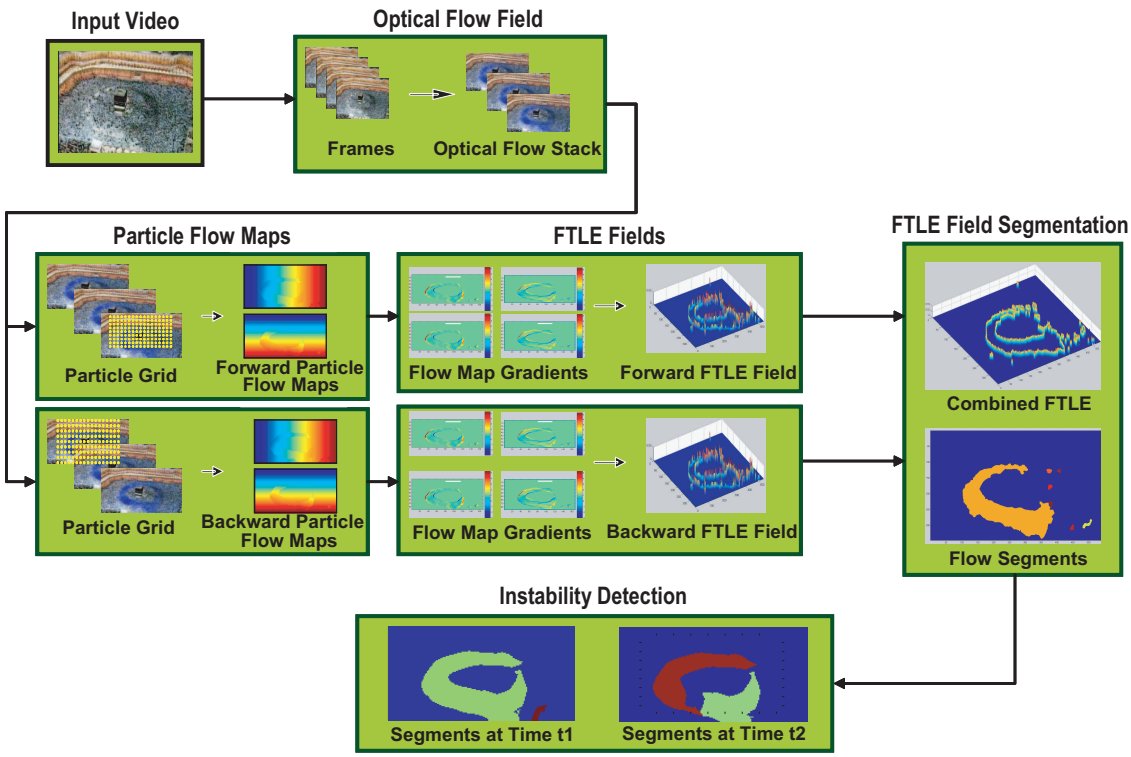

Figure 3.1: Block diagram of the crowd-flow segmentation algorithm. (1) The input is a video of a crowded scene. (2) Computation of optical flow from the frames of the video. (3) Forward and backward advection of particle grid resulting in forward and backward particle flow maps. (4) Computation of respective FTLE fields from the forward and backward particle flow maps. (5) Fusion of forward and backward FTLE fields and label assignment using the watershed segmentation algorithm. (6) Detection of abnormal events (or crowd-flow instabilities).

3.2 Definitions and Notations

Let a compact set $D \subset \mathbb{R}^2$ be the domain of the phase space under study. This domain corresponds to the 2D-spatial extent of the crowd video. Next, define a time-dependent optical flow field $\mathbf{v}(\mathbf{x},t)$

on D that satisfies C^0 and C^2 continuity in time and space, respectively. The C^0 and C^2 assumptions are required to keep the optical flow field smooth. Here, t corresponds to the t-th frame of the video. Then a particle trajectory $\mathbf{x}(t : t_0, \mathbf{x}_0)$, starting at point $\mathbf{x}_0$ at time t_0 can be defined as a solution of

$$\dot{\mathbf{x}}(t; t_0, \mathbf{x}_0) = \mathbf{v}(\mathbf{x}(t; t_0, \mathbf{x}_0), t), \tag{3.1}$$

$$\mathbf{x}(t_0; t_0, \mathbf{x}_0) = \mathbf{x}_0, \tag{3.2}$$

where $\dot{\mathbf{x}}$ is the time derivative. It can also be observed that a trajectory, $\mathbf{x}(t : t_0, \mathbf{x}_0)$, of a particle depends on the initial position $\mathbf{x}_0$ and the initial time t_0. From the above mentioned continuity constraints of optical flow, **v(x**,t), it follows that the particle trajectory, $\mathbf{x}(t : t_0, \mathbf{x}_0)$, will be C^1 in time and C^3 in space.

Since the goal is to analyze the transport properties of the phase space and, in turn, the underlying crowd, the solution of Equation 3.1 can be viewed as a transport device or map that takes particles from their initial position $\mathbf{x}_0$ at time t_0 to their position at time t. Formally, this solution is referred as a "flow map," denoted by $\phi_{t_0}^t$, and that satisfies:

$$\phi_{t_0}^t : D \to D : \mathbf{x}_0 \mapsto \phi_{t_0}^t(\mathbf{x}_0) = \mathbf{x}(t; t_0, \mathbf{x}_0). \tag{3.3}$$

In addition, the flow map $\phi_{t_0}^t$ satisfies the following properties:

$$\phi_{t_0}^{t_0}(\mathbf{x}) = \mathbf{x}, \tag{3.4}$$

$$\phi_{t_0}^{t+s}(\mathbf{x}) = \phi_s^{t+s}(\phi_{t_0}^s(\mathbf{x})) = \phi_t^{t+s}(\phi_{t_0}^t(\mathbf{x})). \tag{3.5}$$

These properties follow directly from the existence and uniqueness theorem that allows one to conclude that there exists only one solution to a first-order differential equation that satisfies the

given initial condition. The next section describes the concept of FTLE field, and also discusses the steps involved in its computation from the flow map ϕ.

3.2.1 Finite Time Lyapunov Exponent Field

As described in Section 3.1, the crowd segments are located by first locating the LCS, and the localization of LCS in turn requires computation of the FTLE field. The Lyapunov exponent is an asymptotic quantity that measures the extent to which an infinitely-close pair of particles separate in an infinite amount of time. In the theory of dynamical systems, it is used as a tool for measuring the chaoticity of the system under consideration by measuring the rate of exponential divergence between the neighboring trajectories in the state/phase space. Traditionally, for any given dynamical system, $\dot{x} = f(x)$, the maximum Lyapunov characteristic exponent is defined as $\gamma = lim_{t\to\infty}\chi(t)$, with

$$\chi(t) = \frac{1}{t} ln \frac{\mid \xi(t) \mid}{\mid \xi(0) \mid}, \tag{3.6}$$

where $\xi(t)$ is the current state of the system, while $\xi(0)$ is the initial state of the given system. These states are usually obtained by solving the differential equation controlling the evolution of the system.

When Lyapunov exponent analysis is performed over a grid of particles over finite times, it generates a FTLE field. In our formulation, the state of the system is defined as the maximum possible separation between a particle and its neighbors. Essentially, this means that the Lyapunov exponent now can be defined as a ratio of the initial separation to the maximum possible separation between the particle and its neighbors. Using this definition of the Lyapunov exponent, FTLE field

$\sigma_T(\mathbf{x}_0, t_0)$ can be computed using the flow map $\phi_{t_0}^{t_0+T}$, which contains the final locations of the particles at the end of advection. The flow map, as mentioned earlier, quantifies the transport properties of the phase space by taking a particle from the initial position, $\mathbf{x}_0$, at time t_0 to its later position at time $t_0 + T$.

One important point to note is that the FTLE does not capture the instantaneous separation rate, but rather measures the average, or integrated, separation rate between trajectories. This distinction is important because, in time-dependent complex crowd flows, the instantaneous optical flow is not very informative. However, by accounting for the integrated effect of the crowd-flow using particle trajectories in the FTLE field, I hope to extract information that is more indicative of the actual transport behavior.

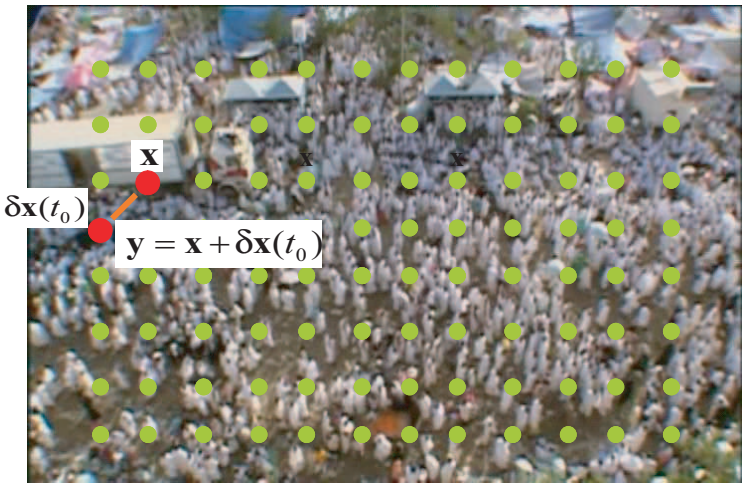

Figure 3.2: Computation of FTLE. The initial separation between particle x and y $= \mathbf{x}+\delta\mathbf{x}(0)$ is $\delta\mathbf{x}(0)$. In order to compute the FTLE between them, I need to find out the magnitude of the final separation between after a time interval T.

The formal derivation of the expression of FTLE proceeds as follows [6, 8]. Consider a particle $\mathbf{x} \in D$ at initial time t_0 (Figure 3.2). Following advection, the position of the particle after a time interval T is $\mathbf{x} \mapsto \phi_T^{t_0+T}(\mathbf{x})$. Now, when advected through the flow, any arbitrary particle that is infinitesimally close to **x** at time t_0 will behave in a manner similar to **x** locally in time. However, as the advection time increases the distance between these neighboring particles will change. Now, if I represent the neighboring particle by $\mathbf{y} = \mathbf{x} + \delta\mathbf{x}(0)$ (Figure 3.2), where $\delta\mathbf{x}(0)$ is an arbitrarily-oriented unit vector, then after a time interval T, the distance between them becomes:

$$\delta\mathbf{x}(t_0+T) = \phi_{t_0}^{t_0+T}(\mathbf{y}) - \phi_{t_0}^{t_0+T}(\mathbf{x}) \tag{3.7}$$

$$= \frac{d\phi_{t_0}^{t_0+T}(\mathbf{x})}{d\mathbf{x}}\delta\mathbf{x}(0) + O(\|\delta\mathbf{x}(0)\|^2). \tag{3.8}$$

Since the distance $\delta\mathbf{x}(0)$ is infinitesimally small, I can drop the higher order terms in the Taylor series expansion of the flow map around the location **x**. The magnitude, $\| \delta\mathbf{x}(t_0+T) \|$, of the final separation can be computed by taking the standard L_2 norm

$$\|\delta\mathbf{x}(t_0+T)\|_2 = \left\| \frac{d\phi_{t_0}^{t_0+T}(\mathbf{x})}{d\mathbf{x}}\delta\mathbf{x}(0) \right\|_2 . \tag{3.9}$$

I am interested in finding out the maximum possible separation between the particle, **x**, and all its neighbors, which, in other words, means that I seek to maximize $\| \delta\mathbf{x}(t_0+T) \|_2$ over all possible to choices of $\delta\mathbf{x}(0)$:

$$\| \delta\mathbf{x}(t_0+T) \|_2 = \max_{|\delta\mathbf{x}(0)|=1} \left\| \frac{d\phi_{t_0}^{t_0+T}(\mathbf{x})}{d\mathbf{x}}\delta\mathbf{x}(0) \right\|_2 . \tag{3.10}$$

Using the operator norm, the above equation can be written as:

$$\| \delta\mathbf{x}(t_0+T) \|_2 = \max_{|\delta\mathbf{x}(0)|=1} \left\| \frac{d\phi_{t_0}^{t_0+T}(\mathbf{x})}{d\mathbf{x}}\delta\mathbf{x}(0) \right\|_2 = \left\| \frac{d\phi_{t_0}^{t_0+T}(\mathbf{x})}{d\mathbf{x}} \right\|_2 . \tag{3.11}$$

The right-hand side of the above equation is the matrix L_2 norm that can be computed simply by using the standard property that states that, for any matrix A, the matrix L_2 norm is the square root of the maximum eigenvalue of the positive definite symmetric matrix $A^T A$. If I consider $A = \frac{d\phi_{t_0}^{t_0+T}(\mathbf{x})}{d\mathbf{x}}$, then $A^T A$ is

$$\Delta = A^T A = \frac{d\phi_{t_0}^{t_0+T}(\mathbf{x})}{d\mathbf{x}}^* . \frac{d\phi_{t_0}^{t_0+T}(\mathbf{x})}{d\mathbf{x}}, \tag{3.12}$$

where superscript '*' refers to the transpose operator. It is interesting to note that Δ is also known as the finite time version of the Cauchy-Green deformation tensor. The quantity $\frac{d\phi_{t_0}^{t_0+T}(\mathbf{x})}{d\mathbf{x}}$ is the spatial gradient tensor of the flow map. The maximum eigenvalue of Δ is represented by $\lambda_{max}(\Delta)$.

Now, knowing the magnitude of the maximum possible separation, $\lambda_{max}(\Delta)$, and the initial separation, $\delta\mathbf{x}(0)$, between the particle and its neighbors, I can compute the FTLE field, σ, with a finite integration time T corresponding to point $\mathbf{x} \in D$ at time t_0 as:

$$\sigma_{t_0}^T = \frac{1}{T} \ln \sqrt{\lambda_{max}(\Delta)}. \tag{3.13}$$

Since, $\delta\mathbf{x}(0)$ is a unit vector, I eliminated it from the above equation. The above quantity is computed for each $\mathbf{x} \in D$ to obtain the entire FTLE field at time t_0.

3.2.2 Lagrangian Coherent Structures

The LCS corresponds to the boundaries between the crowd flows of distinct dynamics. They appear as ridges in the FTLE field of the video. The relationship between ridges in the FTLE field and the LCS can be explained in the following way. If two regions of a phase space have

qualitatively different dynamics, then I expect a coherent motion of particles within each region, and, therefore, the eigenvalues of Δ will be close to 1, an indication that the fate of nearby particles is similar inside the region. At the boundary of the two regions, particles will move in incoherent fashion, and, therefore, will create much higher eigenvalues. These higher values will make the ridge prominent in the FTLE field and point to the locations of the LCS.

I compute two types of LCS, namely "Attracting Lagrangian Coherent Structures" (ALCS) and "Repelling Lagrangian Coherent Structures" (RLCS). The former will emphasize those boundaries between the crowds from which, in a given time interval (*forward in time*), all nearby particle trajectories separate; the later will emphasize those boundaries between the crowds from which in a given time interval (*backward in time*), all nearby particle trajectories separate. For the computation of ALCS, the particle grid is initialized at the first optical flow field and advected forward in time, followed by the computation of forward FTLE field. For the computation of RLCS, the particle grid is initialized at the last optical flow field and advected backward in time, followed by the computation of backward FTLE field.

3.3 Crowd-Flow Segmentation - The Algorithm

In this section, I bring together all the concepts explained so far and describe the algorithmic steps involved in carrying out the crowd-flow segmentation. A block diagram in Figure 3.1 provides the higher-level view of the steps and the data flow.

3.3.1 Optical Flow Computation

Given a video sequence, the first task is to compute the optical flow between the consecutive frames of the video. I employ two different schemes for this purpose. The first scheme consists of a block-based correlation in the Fourier domain. The process starts by selecting a square block centered at the same pixel location of two consecutive frames F_1 and F_2, of the given video. The pixel values in both blocks are mean normalized, and a correlation surface is constructed by performing cross correlation in the frequency domain. The peaks are located in the correlation surface and are used to calculate the displacement. Note that all the pixels inside a block are assigned the same displacement value. The process is repeated for all possible blocks in the given frame. Local outliers in the displacement vectors are replaced in a post-processing step, by using adaptive local median filtering. The removed vectors are filled by interpolation of the neighboring velocity vectors. A typical size of the block employed in our experiments is 16×16 pixels. The second scheme that I used was proposed in [115] where grey value constancy, gradient constancy, smoothness, and multi-scale constraints were used to estimate a high-accuracy optical flow.

To analyze the crowd-flow in a given interval of T frames, I pool the optical flow fields, $\mathbf{v}(1), \mathbf{v}(2), \ldots, \mathbf{v}(T)$, to generate a 3D volume of optical flows. To simplify the notation, I have removed the dependence of $\mathbf{v}$ on location $\mathbf{x}$. This 3D volume of optical flow is used to advect the particles, where parameter T is used as the integration time. I use the symbol B_t^{t+T} to represent a the 3D volume of optical flow fields $\mathbf{v}(t), \mathbf{v}(t+1), \ldots, \mathbf{v}(t+T)$. Figures 3.3, 3.4, 3.5, and 3.6 show color-coded optical flows computed from different sequences in our data set.

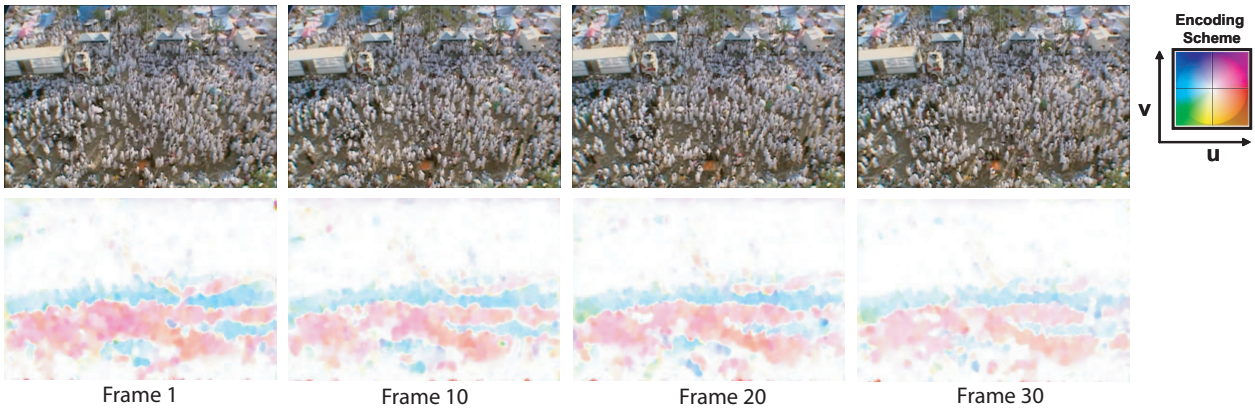

Figure 3.3: Examples of optical flow fields computed by using the algorithm of [115]. Top Row: Frames of the video. Bottom Row: Color-coded optical flow for the corresponding frames.

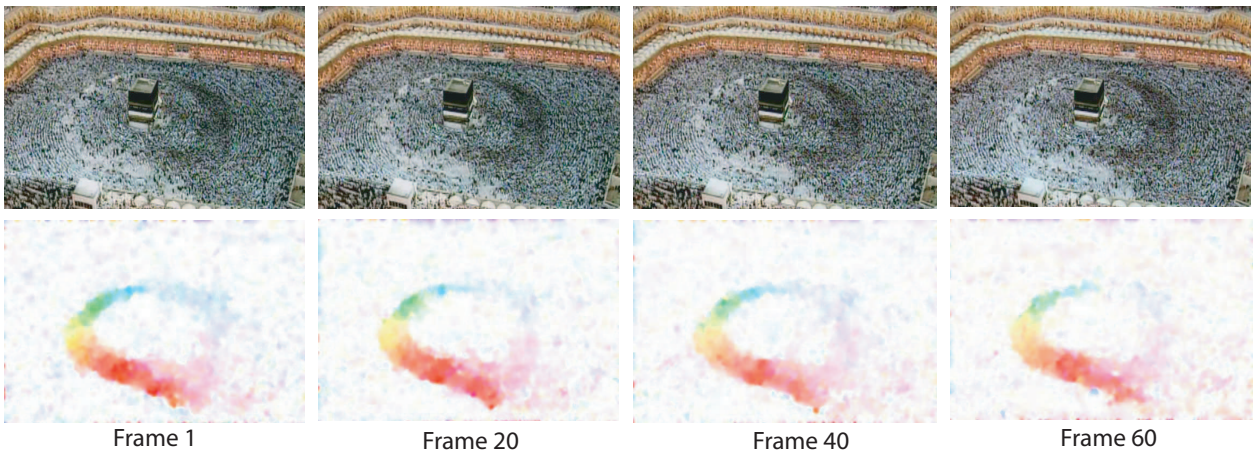

Figure 3.4: Examples of optical flow fields computed by using the algorithm of [115]. Top Row: Frames of the video. Bottom Row: Color-coded optical flow for the corresponding frames.

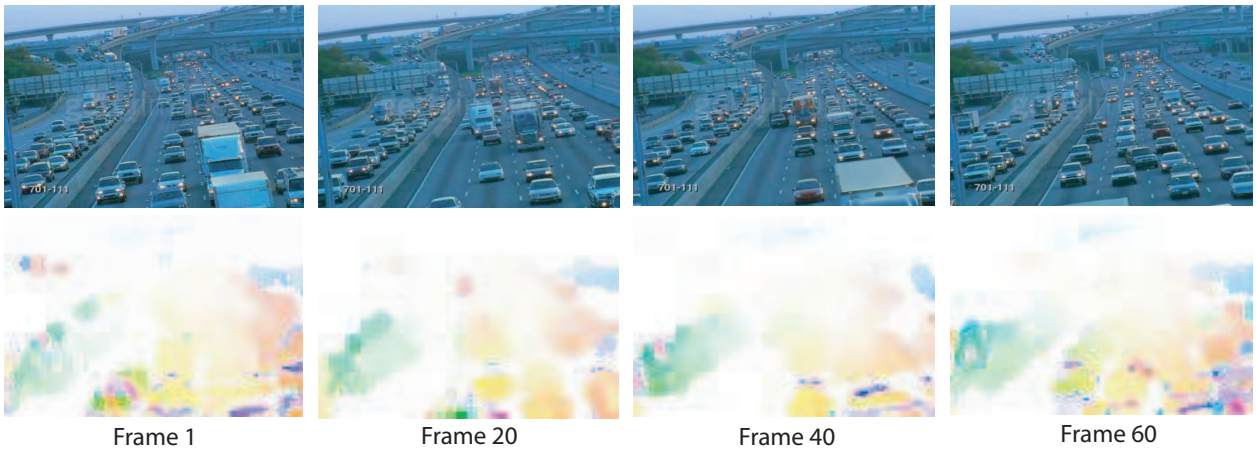

Figure 3.5: Examples of optical flow fields computed by using the block-based correlation algorithm. Top Row: Frames of the video. Bottom Row: Color-coded optical flow for the corresponding frames.

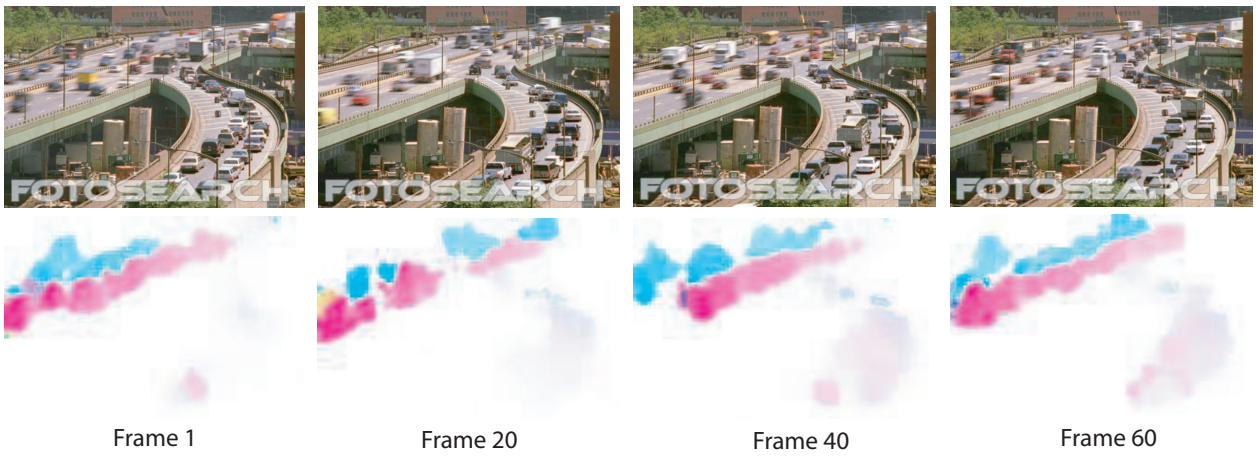

Figure 3.6: Examples of optical flow fields computed by using the block-based correlation algorithm. Top Row: Frames of the video. Bottom Row: Color-coded optical flow for the corresponding frames.

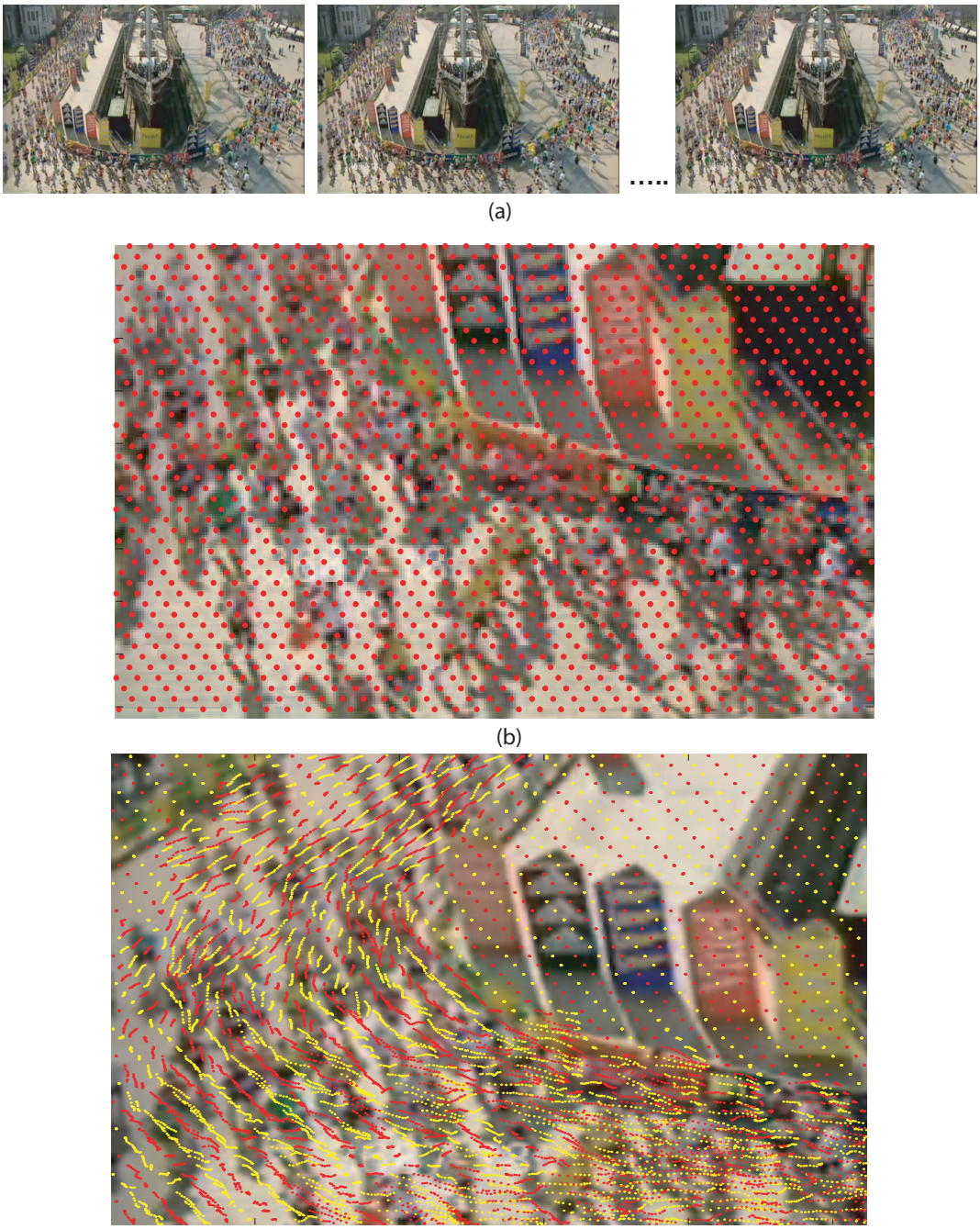

Figure 3.7: The particle advection process. (a) Frames from the input video. (b) A grid of particles is overlaid on the flow field of the input sequence. (c) Trajectories of the particles are obtained by advecting them through the flow field.

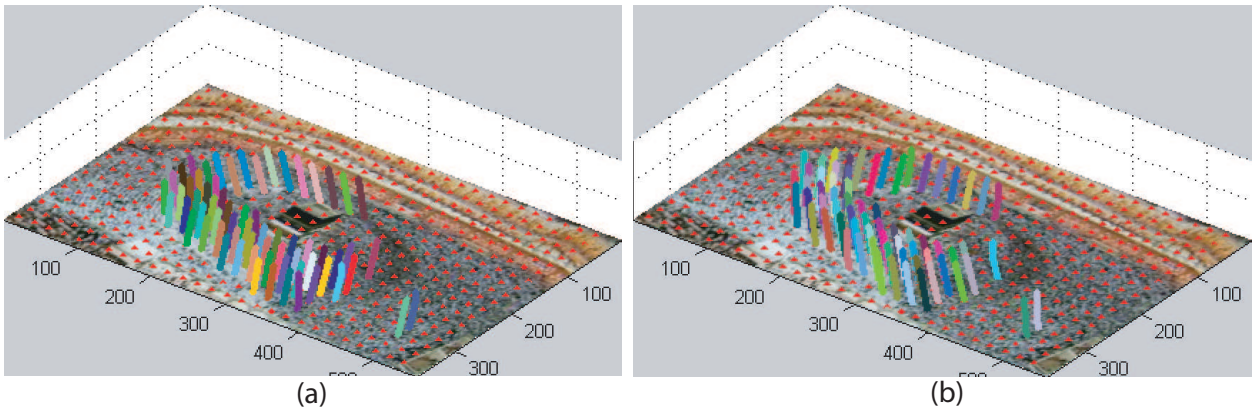

Figure 3.8: (a) The Lagrangian trajectories obtained by forward integration. (b) The Lagrangian trajectories obtained by backward integration.

3.3.2 Particle Advection

The next step is to advect a grid of particles through the 3D volume of flow fields, B_t^{t+T}, that corresponds to the time interval t to $t+T$. I start by launching a grid of particles over the first optical flow field, $\mathbf{v}(t)$, in B_t^{t+T}. Ideally, the resolution of the grid should be the same as the number of pixels in each frame of the video. An example of this Cartesian mesh of particles placed over the flow field of a crowd video and the trajectories of particles are provided in Figure 3.7.

Next, the Lagrangian trajectory $[x(t+T;t,x_0,y_0), y(t+T;t,x_0,y_0)]$ corresponding to a particle at grid location (x_0, y_0) is computed by solving the ordinary differential equations numerically:

$$\frac{dx}{dt} = u(x,y,t), \frac{dy}{dt} = v(x,y,t), \tag{3.14}$$

subject to the initial conditions $[x(0), y(0)] = (x_0, y_0)$. $t+T$ represents the time up-till which I want to compute the trajectory. I use the fourth order Runge-Kutta-Fehlberg algorithm along with cubic interpolation [11] of the velocity field to solve this system. The backward particle advection

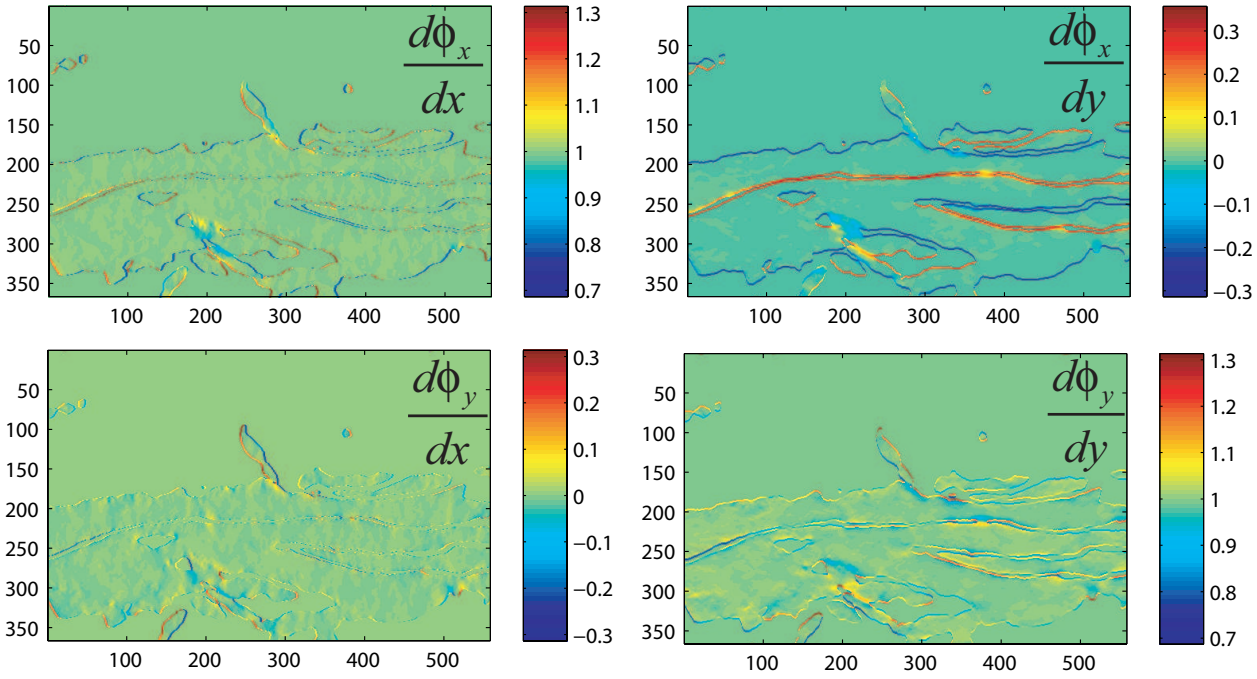

Figure 3.9: The spatial gradients of the particle flow maps for the sequence shown in Figure 3.3.

is carried out by initializing the grid of particles over the last optical flow field $\mathbf{v}(t+T)$ in the 3D volume of optical flow fields B_t^{t+T}. The direction of the optical flow vectors is reversed for the backward integration. Figure 3.8(a) provides a visualization of the Lagrangian trajectories obtained by forward integration, while Figure 3.8(b) provides the visualization of the Lagrangian trajectories obtained by the backward integration. The length of integration, $T = 50$, was used for this purpose.

Note that, in our case the domain D is not closed and trajectories can leave the domain. The particles that leave the domain are not advected anymore, and their last available positions are kept in the flow map. That is, I do not perform any re-seeding of the particles if they leave the domain.

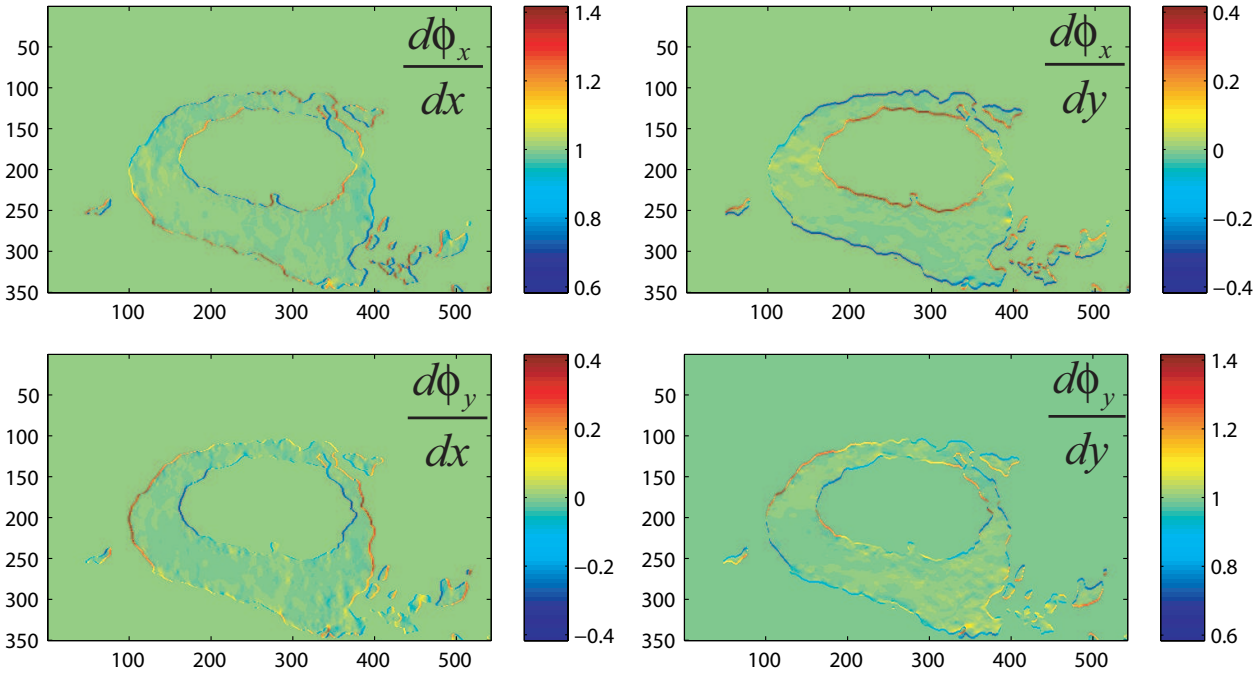

Figure 3.10: The spatial gradients of the particle flow maps for the sequence shown in Figure 3.4.

3.3.3 Particle Flow Maps and FTLE Field

During the forward and backward integration, a separate pair of flow maps, namely ϕ_x and ϕ_y, is maintained for the grid of particles. These flow maps are used to relate the initial position of each particle to its later position obtained after the advection process. This way, the particle flow maps integrate the motion over longer durations of time, which is lacking in the instantaneous optical flow. Here, the first map, ϕ_x, keeps track of how the x coordinate of particles is changing, and, similarly, ϕ_y keeps track of the y coordinate of particles. I use notation ϕ_x^f and ϕ_y^f to refer explicitly to forward flow maps, and ϕ_x^b and ϕ_y^b to refer explicitly to backward flow maps. When the explicit references are not important, I omit the superscripts.

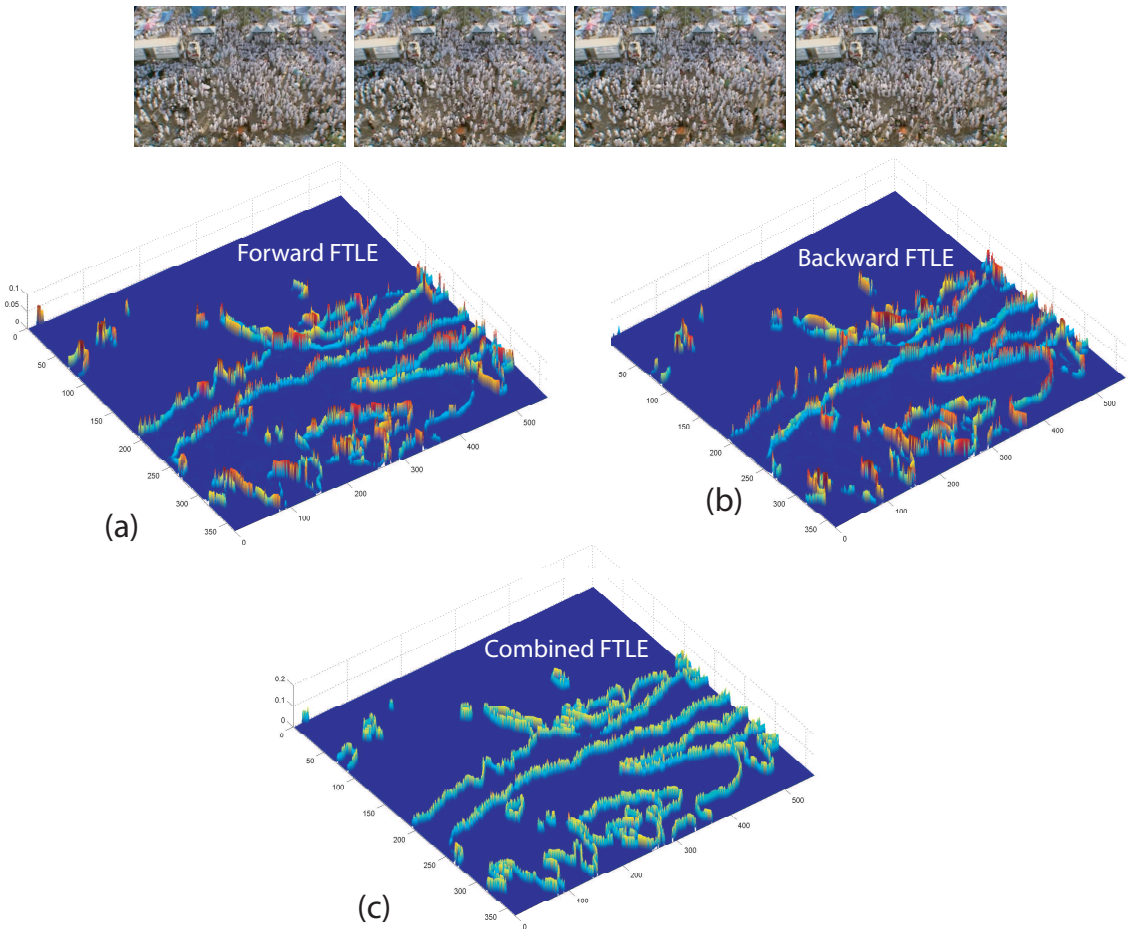

Figure 3.11: FTLE field for the sequence shown at the top. The sequence has multiple groups of people intermingling with each other. The ridges are prominent at the locations where the neighboring crowd groups have dynamically distinct behavior. (a) The forward FTLE field obtained by the forward integration of particles. (b) The backward FTLE field obtained by the backward integration of particles. (c) The combined FTLE field.

At the start, these maps are populated with the initial positions of the particles, which are the pixel locations at which the particle is placed. The particles are then advected under the influence

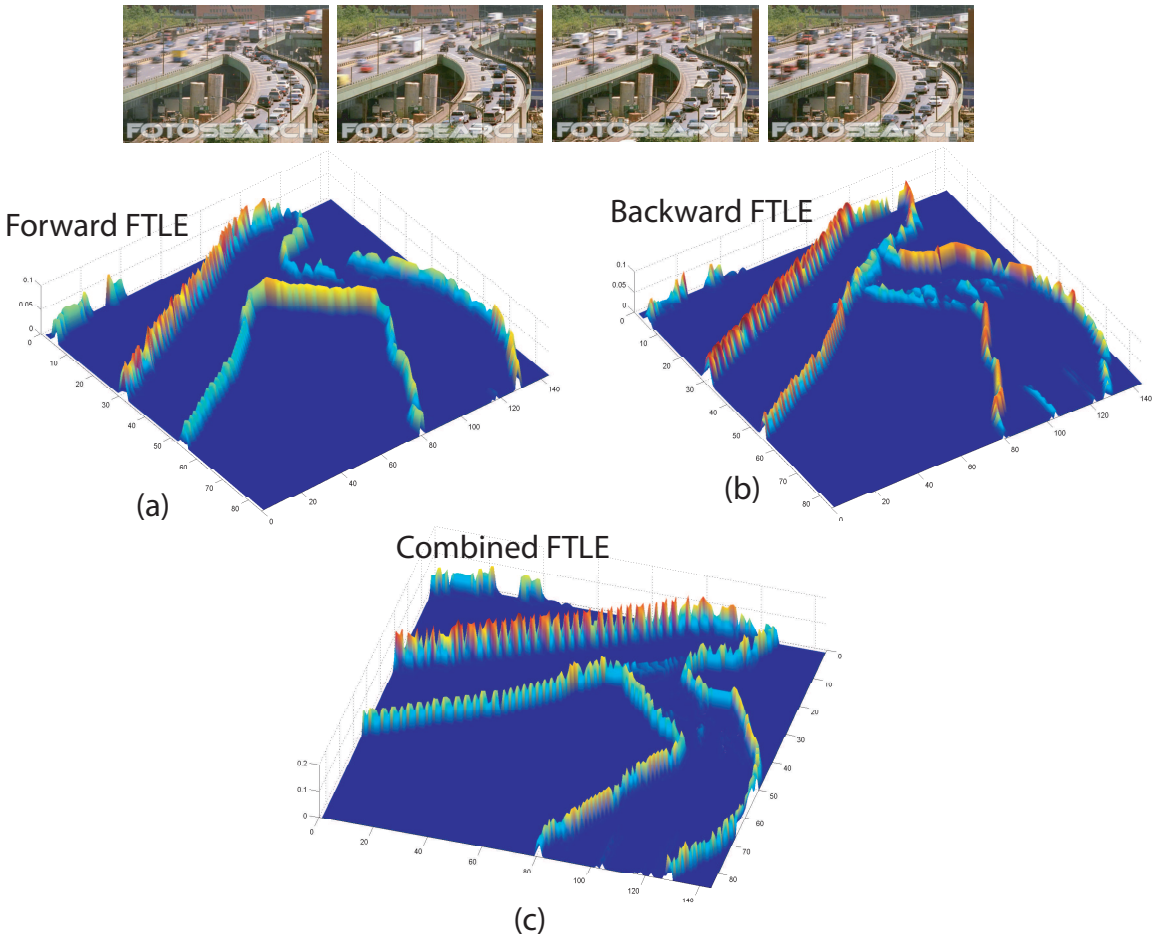

Figure 3.12: FTLE field for the sequence shown at the top. The sequence has multiple lanes of traffic, and the traffic from the ramp is merging onto the main highway. (a) The forward FTLE field obtained by the forward integration of particles. Note that no LCS are present at the intersection of the ramp and the highway. (b) The backward FTLE field obtained by the backward integration of particles. Note that LCS have now appeared at the intersection of the ramp and the highway. (c) The combined FTLE field.

of B_t^{t+T} using the method described in Section 3.3.2. The positions of the particles are updated until the end of the integration time length T.

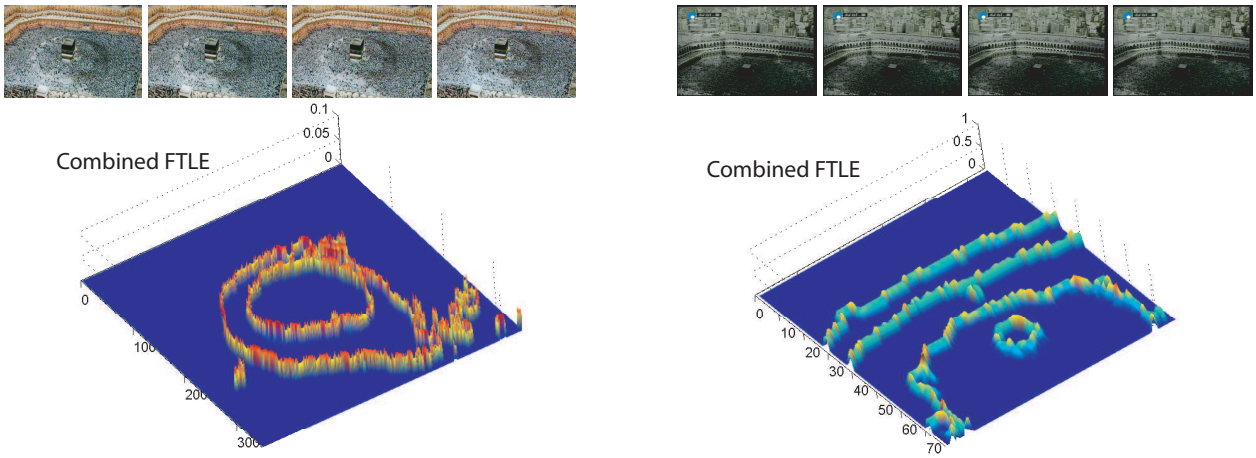

Figure 3.13: The combined FTLE fields for the sequences shown at the top.

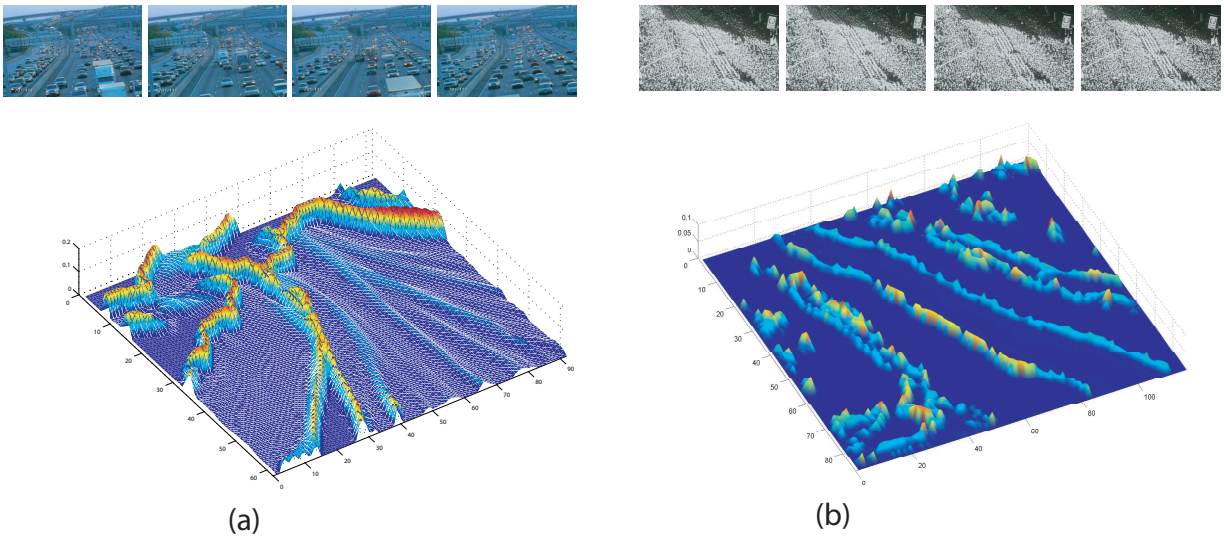

Figure 3.14: The combined FTLE fields for the sequences shown at the top.

The computation of the FTLE field from the particle flow maps requires computation of the spatial gradients of the particle flow maps, i.e., $\frac{d\phi_x}{dx}$, $\frac{d\phi_x}{dy}$, $\frac{d\phi_y}{dx}$, and $\frac{d\phi_y}{dy}$. This step is accomplished by using a finite differencing approach for taking derivatives. Figures 3.9 and 3.10 show spatial gradients of particle flow maps for two different sequences in the data set. It can be observed

that a high gradient is present where the neighboring particles are behaving differently over the length of the integration. The Cauchy-Green deformation tensor is computed by substituting the spatial gradients of the particle flow maps in Equation 3.12. Finally, the FTLE field is computed by finding the maximum eigenvalue of the Cauchy-Green deformation tensor and plugging it in Equation 3.13. Figures 3.11-3.14 show a number of FTLE fields corresponding to different crowd sequences in our data set. In these examples, the combined FTLE field is obtained by adding the forward and backward FTLE fields. It can be observed that ridges in these fields (Figures 3.11-3.14), which point to the location of LCS, are very prominent, and, therefore, can be used to separate regions of the crowd-flow that are dynamically distinct from each other.

The utility of computing forward and backward FTLE fields becomes obvious from the analysis of the FTLE fields shown in Figure 3.12. In this video sequence traffic from the ramp is merging onto the main highway. When the particles are advected forward in time, no LCS appear at the intersection of the ramp and the main highway (Figure 3.12(a)). The reason is that the particles at the intersection move forward coherently in time as the destinations of the underlying traffic flow on the ramp and the main highway are the same. But when these particles are advected backward in time, the LCS appear at the intersection (Figure 3.12(b)) since the particles at the intersection do not have the same destination backward in time because the underlying traffic is originating from different locations. In other words, by backward integration, I am able to take into account the origin of the flow in addition to its destination. This capability is important to completely resolve different crowd-flow segments present in the scene. This point will become clearer when I present the segmentation results in a later section.

3.3.4 FTLE Field Segmentation

The LCS in the FTLE field can be treated as the watershed lines dividing individual catchment basins. Each catchment basin represents the distinct crowd grouping that is present in the scene. The catchment basins are homogeneous in the sense that all the particles belonging to the same catchment basin have the same origin and destination. To generate a distinct labeling for each catchment basin, I employ the watershed segmentation algorithm [119]. The final segmentation map is created by removing those segments where the magnitude of the flow is zero. I call such segments "vacuum segments." Note that, due to the unique strength of the FTLE field based representation, I do not have to pre-specify the number of crowd-flow segments. This way, I am able to overcome the problem of specifying the number of segments or clusters which is common in most of the clustering and segmentation algorithms [121].

3.4 Flow Instability

In this section, I demonstrate the applicability of the crowd-flow segmentation on the task of abnormal event detection in crowds. This is in line with the goal of the thesis, which emphasizes the use of global level knowledge to help solve more complex, low-level vision tasks.

Given the crowd-flow segmentation information, I define the problem of locating the abnormal behavior (also called flow instability) as the problem of detecting the change in the number of flow segments over time. Recall that the boundaries between flow segments are reflected as LCS in the corresponding FTLE field. Now any change in the behavior of the crowd will cause new LCS to appear in the FTLE field exactly at the location of the change. These new LCS will eventually give

rise to new flow segments that were not there before. By detecting these new flow segments, I can identify the locations in the scene where the behavior of the crowd is changing and can term it as the abnormal behavior.

In formal terms, I establish correspondence between the flow segments that are generated from two consecutive blocks of the video for detecting new flow segments. Let us represent the segmentation maps of the two blocks by S_1 and S_2. The shape of a flow segment is represented by a Gaussian distribution over the spatial coordinates of pixels belonging to that segment. The mean of the Gaussian is initialized to the mean of the spatial coordinates, while the variance is initialized to the variance of the coordinates of the boundary pixels. A voting scheme is then employed for establishing correspondence between flow segments of the segmentation maps S_1 and S_2. Each pixel of a flow segment in S_2 votes for one of the flow segments in S_1. A flow segment from S_2 corresponds to a flow segment in S_1, if the majority of the pixels of the segment in S_2 have voted for that flow segment in S_1. A flow segment in S_2 whose correspondence cannot be established is "flagged" as an unstable flow (or abnormal) region. On the other hand, if the pairwise correspondence between all flow segments is found, it is assumed that the dynamics of the underlying crowd has not changed. Note that the spatial probability distributions of flow segments are constructed in a learning stage during which it is assumed that the behavior of the crowd is normal.

3.5 Experiments and Discussion

This section discusses the experimental setup and the data sets used in the experiments. It also presents the segmentation results along with a discussion of the interpretation of the results.

Figure 3.15: Example of sequences used in our experiments.

3.5.1 Data Sets and Experimental Setup

I have tested our approach on videos taken from the stock footage web sites (Getty-Images [136], Photo-Search), and Video Google [135]. Two types of crowded scenarios are covered in these videos: the first scenario consists of scenes involving the high-density crowds, while the second scenario consists of high-density traffic scenes. Traffic scenes can be treated as a close approximation of the motion of crowds of people and, therefore, provides us with useful data for testing the performance of the proposed algorithm. Another set of videos were taken from the National Ge-

ographic documentary, entitled “Inside Mecca,” which covers the yearly ritual of Hajj performed by close to two million people. Therefore, this event provides a unique opportunity for capturing data about the behavior of large gatherings of people in a realistic setting. Figure 3.15 shows key frames from some of these sequences.

For each video, the optical flow was computed by using the algorithms previously described in Section 3.3.1. The computation of the optical flow was performed at a coarser resolution than the resolution of the image to reduce the computational cost. Next, a grid of particles was placed over the flow field. The resolution of the grid was kept the same as the number of pixels on which the flow field was computed. The forward and backward particle flow maps were generated using the advection algorithm described in Section 3.3.2. The corresponding FTLE fields were computed from the spatial gradient tensor of the flow maps using Equation 3.13. The backward and forward FTLE fields were fused to generate a combined FTLE field. The fusion was carried out by adding the values of both fields. Finally, the segmentation was performed by using the watershed segmentation algorithm.

3.5.2 Segmentation Results

This section presents qualitative analysis of the results obtained on different video sequences. Figures 3.16- 3.24 show the segmentation results on all the sequences in the data set.

The first sequence, shown in Figure 3.16, was extracted from the National Geographic documentary entitled “Inside Mecca”. The sequence depicts thousands of people circling the Kabba in a counter-clockwise direction. In this case, the group of people circling in the center is part of the

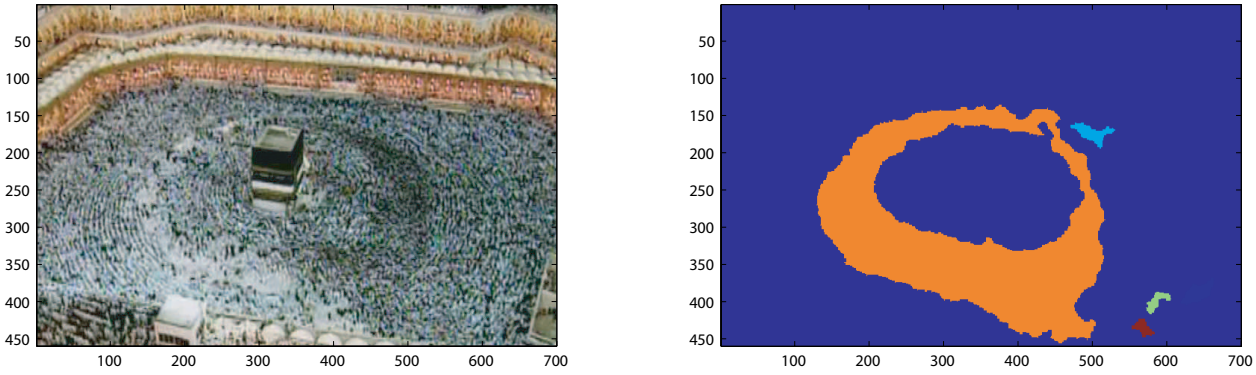

Figure 3.16: The flow segmentation result on a video taken from the National Geographic documentary "Inside Mecca." Left: A frame from the video. Right: The crowd-flow segmentation mask.

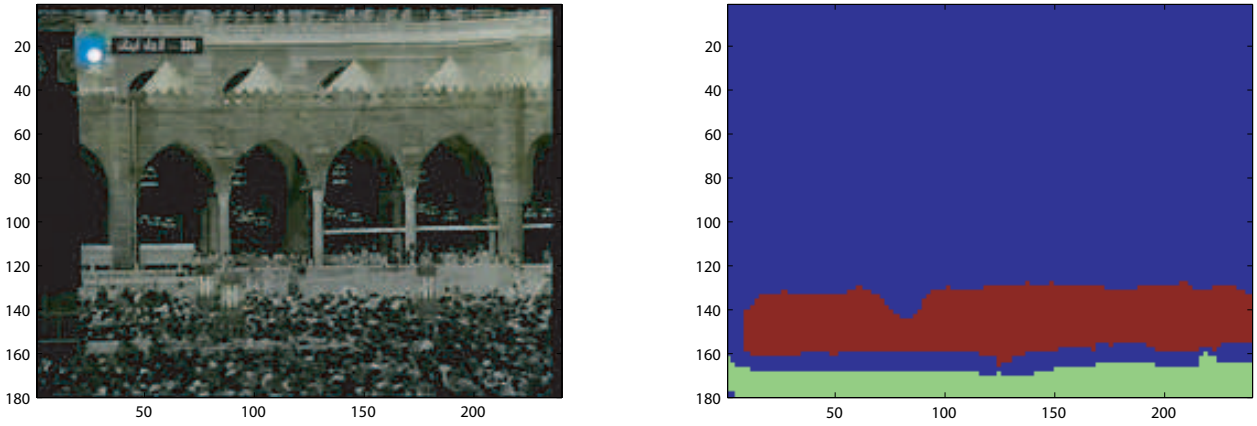

Figure 3.17: The flow segmentation result on a video from "Video Google." Left: A frame from the video. Right: The crowd-flow segmentation mask.

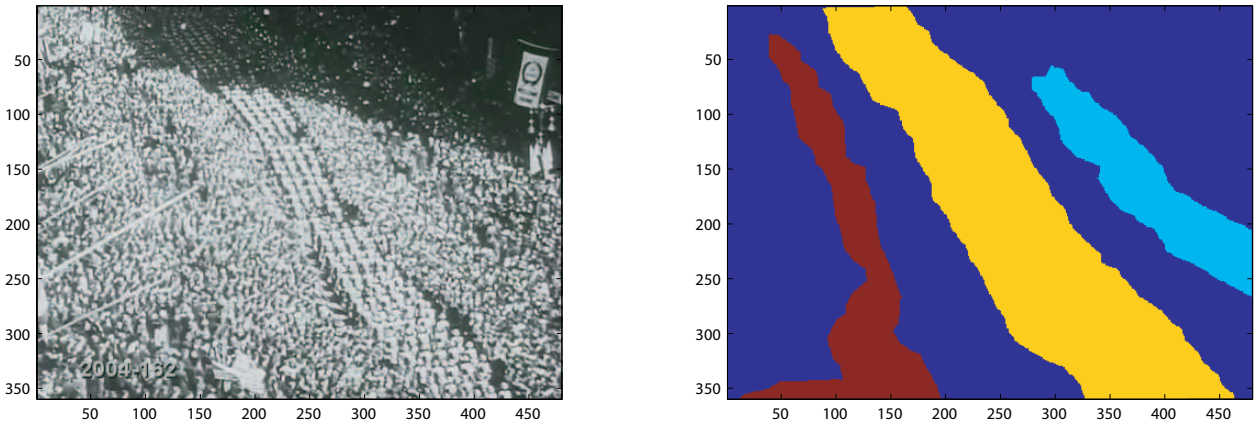

Figure 3.18: The flow segmentation result on a video taken from the stock footage web site "Getty Images." Left: A frame from the video. Right: The crowd-flow segmentation mask.

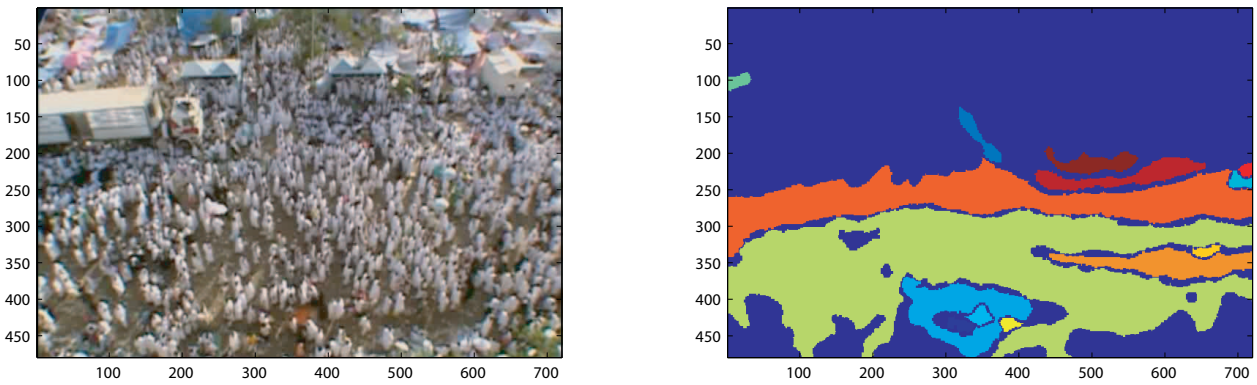

Figure 3.19: The flow segmentation result on a video taken from the National Geographic documentary "Inside Mecca." Left: A frame from the video. Right: The crowd-flow segmentation mask.

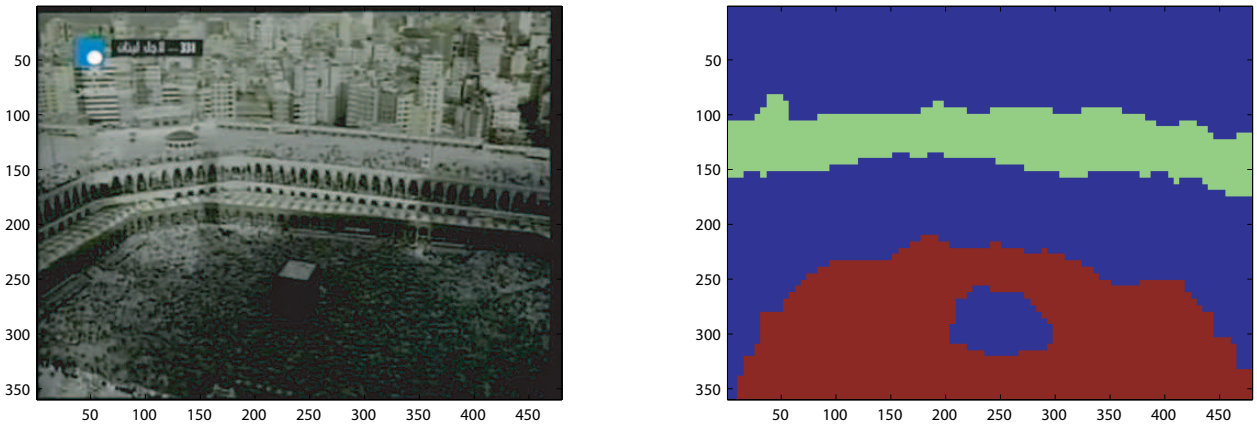

Figure 3.20: The flow segmentation result on a video from "Video Google." Left: A frame from the video. Right: The crowd-flow segmentation mask.

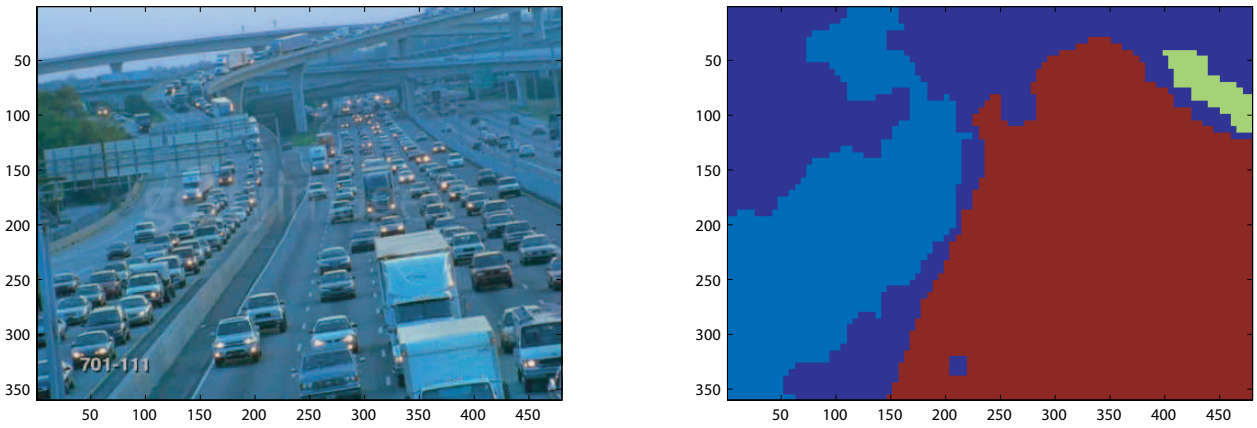

Figure 3.21: The result of the flow segmentation on a high-density traffic scene. This segmentation was obtained by using both the forward and backward FTLE fields. Left: A frame from the video. Right: The crowd-flow segmentation mask.

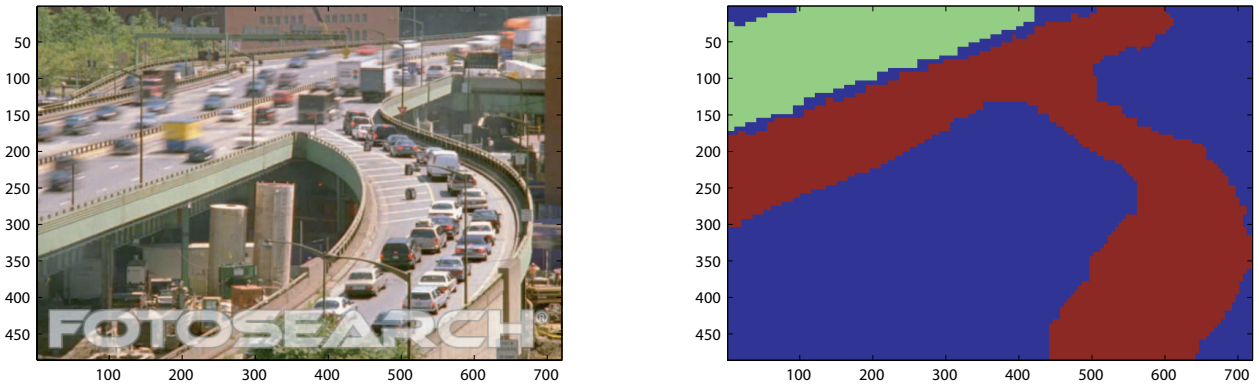

Figure 3.22: Result of the flow segmentation on a high-density traffic scene. The segments correspond to group of cars that are behaving dynamically different from each other.

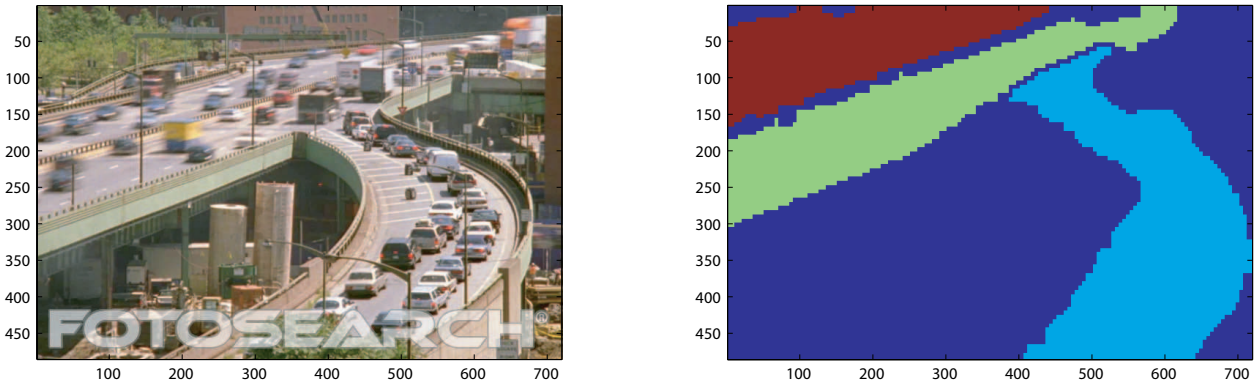

Figure 3.23: The result of the flow segmentation on a high-density traffic scene. This segmentation was obtained by using only the forward FTLE field. Left: A frame from the video. Right: The crowd-flow segmentation mask.

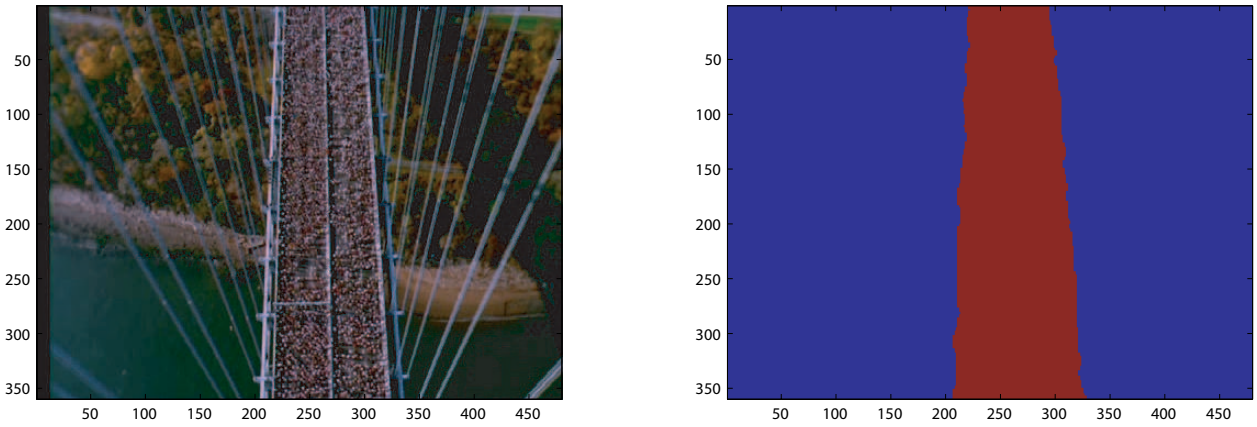

Figure 3.24: The result of the crowd-flow segmentation on a marathon sequence. Left: A frame from the video. Right: The crowd-flow segmentation mask.

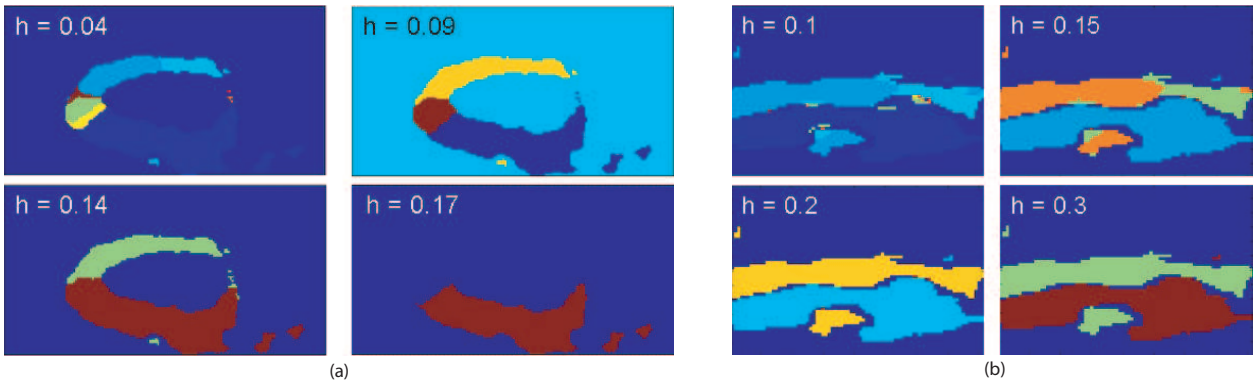

Figure 3.25: A comparison with respect to the mean shift segmentation. (a) The segmentation obtained for the sequence shown in Figure 3.16. (b) The segmentation obtained for the sequence shown in Figure 3.19.

same flow segment because of its common dynamics and desirable goal. The optical flow field of the crowd motion offers a unique challenge as one can observe from the color-coded optical flow shown in Figure 3.3. The different colors emphasize that the flow vectors along the circular path have different directions and magnitudes. This means that a simple clustering of these vectors will not allow us to assign these vectors to the same cluster when, in fact, they all belong to one cluster. The result is shown in Figure 3.25(a), where mean-shift clustering was used to cluster the optical flow vectors ((u, v)) extracted from the instantaneous optical flow field. The clustering results are shown for different choices of the band-width parameter. But even with different values of the band width, the mean-shift is not able to correctly localize the circular segment. However, using our method where I integrate the motion information over longer durations of time, I am able to correctly segment the complex crowd motions (Figure 3.16). The LCS structures previously shown in Figure 3.13(a), show that the dynamic behavior of the crowd moving in a circle is preserved by emphasizing the boundaries of the coherent flow regions. Another result of a similar type of motion is presented in Figure 3.20. In this case, there was an additional group of people that was walking on top of the roof. Our method was able to localize this additional crowd-flow segment as well.

The next result that I would like to discuss is shown in Figure 3.19. This sequence contains complex motion dynamics as there are several groups of people that are intermingling with each other and moving in various directions. The challenges posed by this sequence are different in that the mixing barriers between various crowd groupings must be correctly located. The segmentation result shown in Figure 3.19 demonstrate that I am able to localize most of the distinct crowd

groupings that were present in the scene. The discovered barriers between the crowd groupings can be observed in the combined FTLE field shown in Figure 3.11(c). The barriers which appear in the form of ridges in the FTLE field, encapsulate each crowd group. A comparison is again performed with the mean-shift clustering approach (Figure 3.25(b)), but, again, the mean shift is not able to localize all the crowd-flow segments. This again points to the utility of integrating motion information over longer periods of time, which helps to get a better picture of the crowd motion. Some other example results on sequences involving groups of people are presented in Figures 3.17, 3.18, and 3.24.

Next, I discuss the segmentation results on a high-density traffic sequence (Figure 3.22). The results on this sequence highlight the utility of using both forward and backward integration of particles through the 3D volume of optical flows. In this sequence, vehicles are moving in two opposite directions on the main highway, while a flow of traffic is merging onto the main highway from the ramp. The challenge in this sequence is to find the right membership of the flow generated by the traffic on the ramp by resolving its origin and destination. If I only use the forward integration, it is obvious that all the particles initialized over the ramp will have the same fate as the particles on the main highway. This means that the traffic on the ramp will become part of the flow generated by the lane on the right-hand side of the highway. Another way to look at the forward integration is from the viewpoint of flow continuity, where out-going flux on the ramp is equal to the additional flux received by the highway at this location. The segmentation result shown in Figure 3.22 validates the above observation where same labeling is being assigned to the ramp and to the right lane of the main highway. This ambiguity can be resolved by the addition of

the backward integration of particles. Since they are considered backwards in time, the particles on the two sections of the road do not share the same origin or, in other words, the outgoing flux is not equal to the flux received by the two sections of the road. The segmentation result shown in Figure 3.23 demonstrates that by using both forward and backward integration of particles, a flow segmentation that is more refined is obtained. The result on another traffic sequence is shown in Figure 3.21.

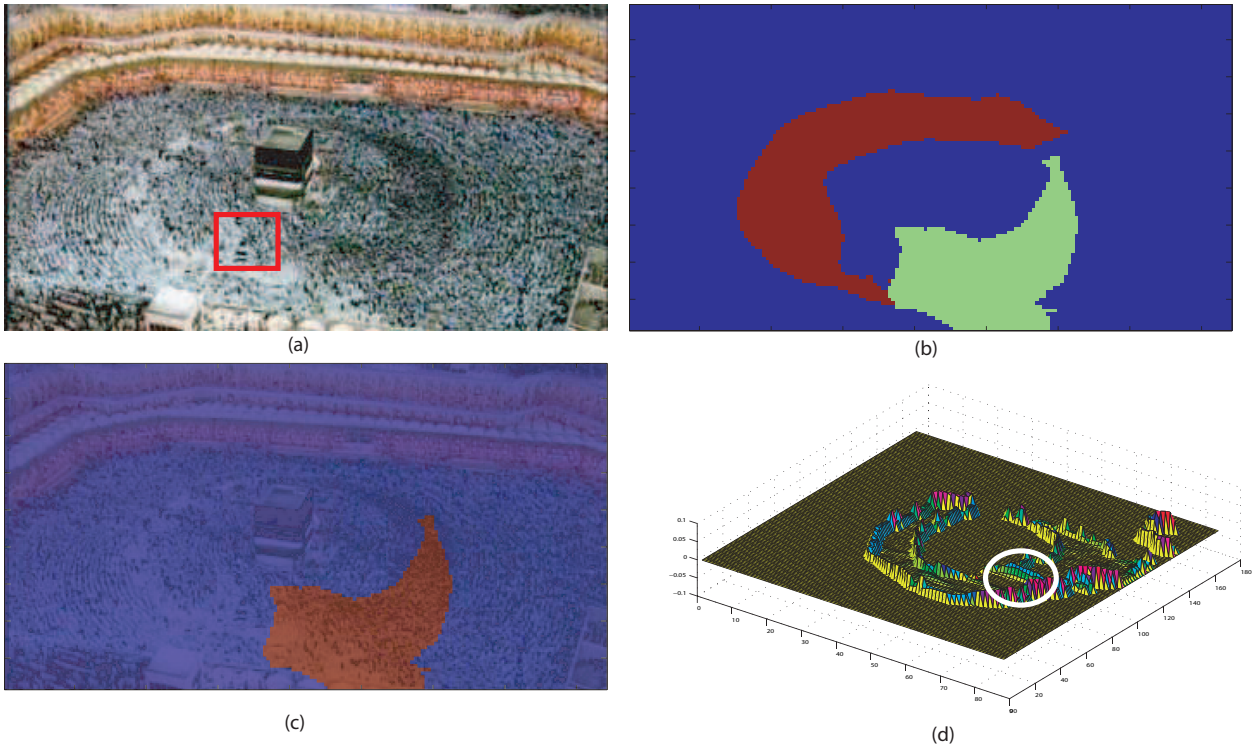

Figure 3.26: (a) Bounding box shows the location at which the instability was created by flipping the image patch. (b) Outcome of the flow segmentation algorithm. (c) Instable flow region is detected and highlighted on the video sequence. (d) The FTLE field corresponding to the video sequence with synthetic instabilities. Emergence of new LCS can be observed within the white circle.

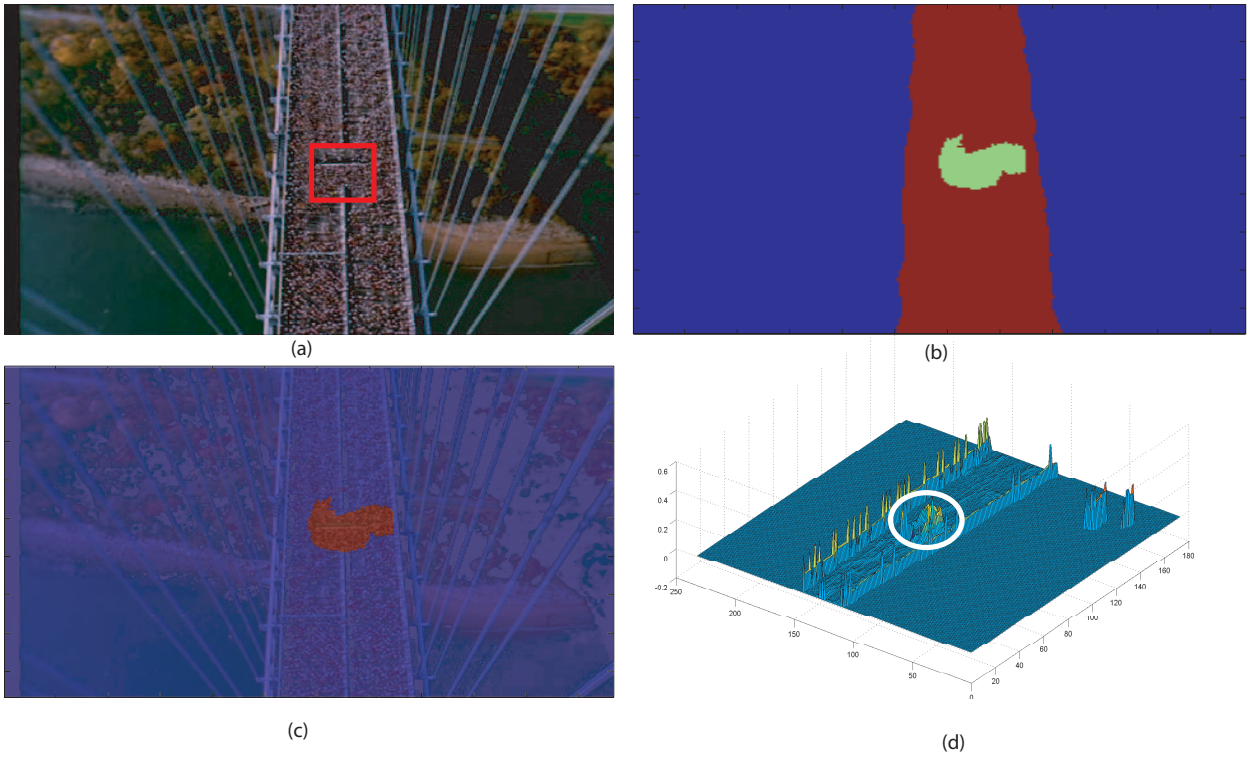

Figure 3.27: (a) Bounding box shows the location at which the instability was created by rotating the image patch. (b) Outcome of the flow segmentation algorithm. (c) Instable flow region is detected and highlighted on the video sequence. (d) The FTLE field corresponding to the video sequence with synthetic instabilities. Emergence of new LCS can be observed within the white circle.

3.5.3 Abnormal Event Detection Experiments

A second set of experiments was performed to test our approach for the detection of flow instability or an abnormal event occurring in the crowd. In the absence of publicly available videos that may contain shots of a stampede or other types of disturbances, I have created our own sequences by

inserting synthetic instabilities into the original video sequences. For each experiment, the original video sequence was used during the learning stage to compute the crowd-flow segmentation corresponding to the normal flow of the crowd.

After the learning stage, the next set of frames on which I perform the flow segmentation was taken from the corresponding video sequence that contains the synthetic instability. Synthetic instability is created by randomly choosing a location over the moving crowd and then placing a bounding box of fixed size around that location. The patch of the image within the bounding box is either flipped or rotated to change the flow behavior at that location. Two examples of this process are shown in Figures 3.26(a) and 3.27(a). The correspondence of the flow segments that were generated from the frames of the synthetic sequence is established with the learned set of segments using the procedure described previously. Figures 3.26 and 3.27 show the results of these experiments.

In case of the first sequence (Figure 3.26), the instability has created a barrier in the flow that resulted in the breakup of the original segment into two parts, as shown in Figure 3.26(b). The segment for which the correspondence cannot be established is flagged as a potential unstable flow region in Figure 3.26(c). The emergence of new LCS in the FTLE field, shown in Figure 3.26(d) (circled in white), validates the observation that any change in the dynamics of the flow will result in the emergence of new LCS that can be used to locate the instabilities.

The second sequence, shown in Figure 3.27, captures a bird's-eye view of the New York City marathon. In this case, the synthetic instability was placed at the location shown in Figure 3.27(a). Again, our algorithm was able to locate and flag it as a potential problem region as demonstrated

in Figures 3.27(b) and 3.27(c). The FTLE field of this sequence (Figure 3.27(d)) again shows the presence of LCS structures at the location of the instability.

3.6 Summary

This chapter has developed a framework for segmenting scenes of crowds of people into regions that are dynamically distinct using Lagrangian particle dynamics. For this purpose, the spatial extent of the video was treated as a phase space of a non-autonomous dynamical system in which transport from one region of the phase space to the other was controlled by the optical flow. Next, a grid of particles was advected forward and backward in time through this phase space and the amount by which the neighboring particles diverged was quantified by using a Cauchy-Green deformation tensor. The maximum eigenvalue of this tensor was used to construct a Finite-Time Lyapunov Exponent (FTLE) field, which revealed the time-dependent invariant manifolds of the phase space called Lagrangian Coherent Structures (LCS). The LCS in turn divided the crowd-flow into regions of different dynamics.

The strength of this approach lies in the fact that it bypasses the need for low-level detection of individual objects altogether, which will be impossible in a high-density crowded scene, and generates a concise representation of the complex mechanics of human crowds using only the global analysis. I also demonstrated that this global knowledge about different crowd groupings can be used to localize abnormal behaviors that are taking place in the scene.

The next goal is to use the information generated by the crowd-flow segmentation to solve the task of tracking individual objects within the crowd. This type of capability will allow the operator

who is observing the video stream to focus on a single suspicious person instead of worrying about tracking all the moving objects in the crowded scene. In the next chapter, I develop this capability by proposing a tracking algorithm that uses the information from the crowd segmentation.

CHAPTER 4
TRACKING INDIVIDUAL TARGETS IN CROWDED SCENES

Global information about the scene generated by crowd-flow segmentation can be used as an aid to solve the more complex task of tracking a single person within a high density crowd. In this chapter, I describe a tracking approach for *high density crowd scenes* containing hundreds and thousands of people at a time that employs crowd-flow segmentation as one of the cues. The approach is based on the observation that the *behavior of an individual in a crowded situation is a function of collective behavioral patterns evolving from the space time interaction of a large number of individuals among themselves and with the structure of the scene*. Therefore, these collective behavioral patterns can be channeled in as an auxiliary source of information, which may help in constraining the likely locations/paths taken by individuals in the given scene. I developed a model called the 'scene structure force model' to directly incorporate these influences. This model captures the notion that an individual, when moving in a particular scene, is subjected to forces that are functions of both the layout of that scene and locomotive behavior of other individuals in his/her vicinity. I demonstrate some examples of high density crowded scenes in which tracking is performed in Figure 4.1.

In our tracking algorithm, the crowd is treated as a collection of mutually interacting particles. This is a reasonable assumption, because when people are densely packed, individual movement

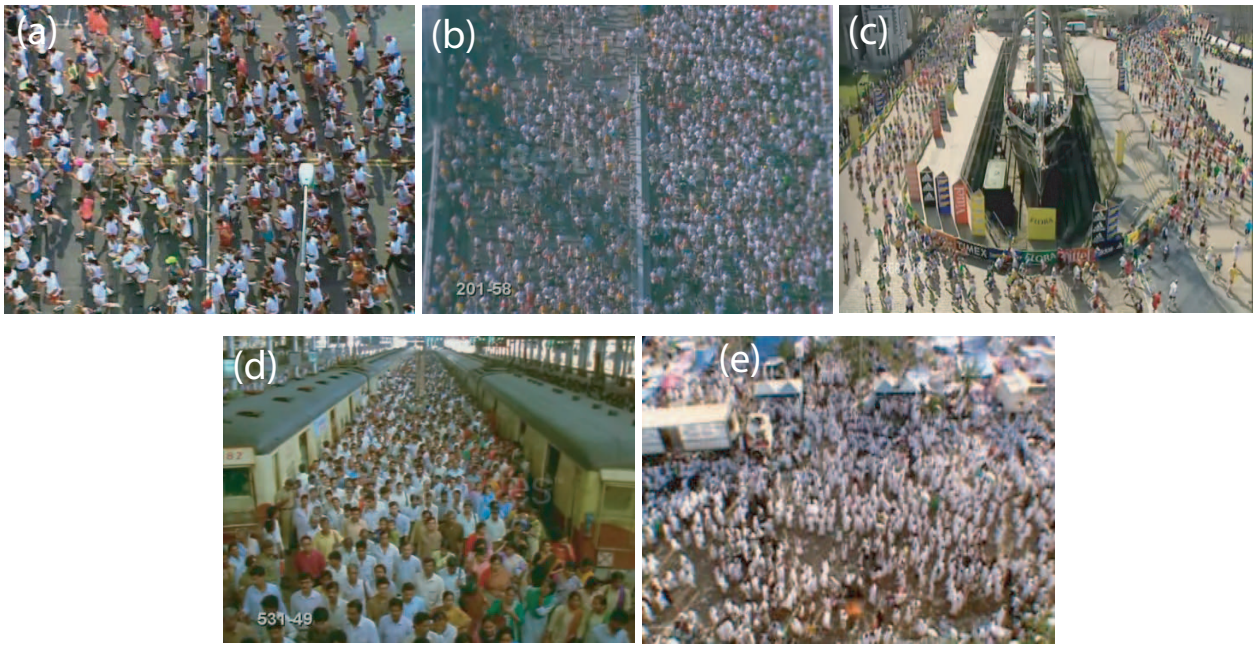

Figure 4.1: Examples of high density crowded scenes. (a)-(c) Hundreds of people participating in marathon races. (d) A scene from a densely packed railway station in India. (e) A group of people moving in various directions.

is restricted, and members of the crowd can be considered as granular particles. For tracking a specific individual in the crowd, I model the instantaneous movement of that person (particle) with a matrix of preferences containing the probabilities of a move in a certain preferred direction. The probabilities take into consideration multiple sources of information, including appearance of the target individual and structure of the scene. The scene structure is incorporated by introducing a concept of *floor fields*, which model the interactions between individuals and their preferred direction of movements by transforming long ranged forces into local ones. The transition probability of a tracked person then depends on the strength of the floor field in his/her neighborhood. For instance, a long range force that will compel the individual in a crowd to move towards the exit

door, can be converted into a local force by increasing the instantaneous probability of a move in that direction. The concept of floor field itself is inspired from the field of evacuation dynamics ([89, 90]), where floor fields are *manually* designed to simulate behaviors of pedestrians in panic situations. I compute three floor fields, namely: a 'Static Floor Field' (SFF), a 'Boundary Floor Field' (BFF), and a 'Dynamic Floor Field' (DFF). Here, the SFF field specifies the regions of space that are more attractive, e.g. an exit and dominant direction of motion; while the BFF specifies the regions in the scene which are more repulsive e.g. barriers and no-entry areas. The DFF corresponds to the virtual traces created by the movements of individuals which are abstracted in the form of particle trajectories, and in turn influences the motion of the individual being tracked.

4.1 Tracking Framework

The crowd-flow in the scene is treated as a collection of mutually interacting particles. Therefore, given a video $E = [f_1, f_2, \ldots, f_N]$, where N is the total number of frames, the image space is discretized into small cells where each cell is occupied by a single particle $o_{\mathbf{x}i}$. Here, $\mathbf{x}i = (xi, yi)$ is the coordinate of the ith pixel at which the particle is located. At the highest resolution each cell corresponds to a single pixel. For tracking, the target individual is represented by a set of particles $\mathcal{P} = [\ldots, o_{\mathbf{x}i}, \ldots]$ (red particles in Figure 4.2(a)). Next, an appearance template, H, of the target is computed using the pixels corresponding to particles $o \in \mathcal{P}$. The target moves from one cell (pixel) to the next at discrete time steps, $t \rightarrow t + 1$, according to a transition probability that determines the likely direction of the motion. These transition probabilities are determined by using two factors: 1) the similarity between the appearance templates at the current location and

the next; 2) the influence generated by the layout of the scene and the behavior of the crowd at and around the target individual, as captured by the floor fields. Formally, if the individual is currently at cell i, then the probability of moving to a neighboring cell j is:

$$p_{ij} = Ce^{k_D D_{ij}} e^{k_S S_{ij}} e^{k_B B_{ij}} R_{ij}, \tag{4.1}$$

Where:

- D_{ij} is the influence of the DFF
- k_D is the coupling strength of the tracked object to the DFF
- S_{ij} is the influence of the SFF
- k_S is the coupling strength of the tracked object to the SFF
- B_{ij} is the influence of the BFF
- k_B is the coupling strength of the tracked object to the BFF
- R_{ij} is the similarity measure between the initial appearance template H and the current appearance template of the target computed at the location j
- C is a normalization constant

A pictorial representation of the matrix of the preferred walking direction is shown in Figure 4.2(c). Next, I describe the algorithm for computing S_{ij}, D_{ij}, and B_{ij} from the respective floor fields.

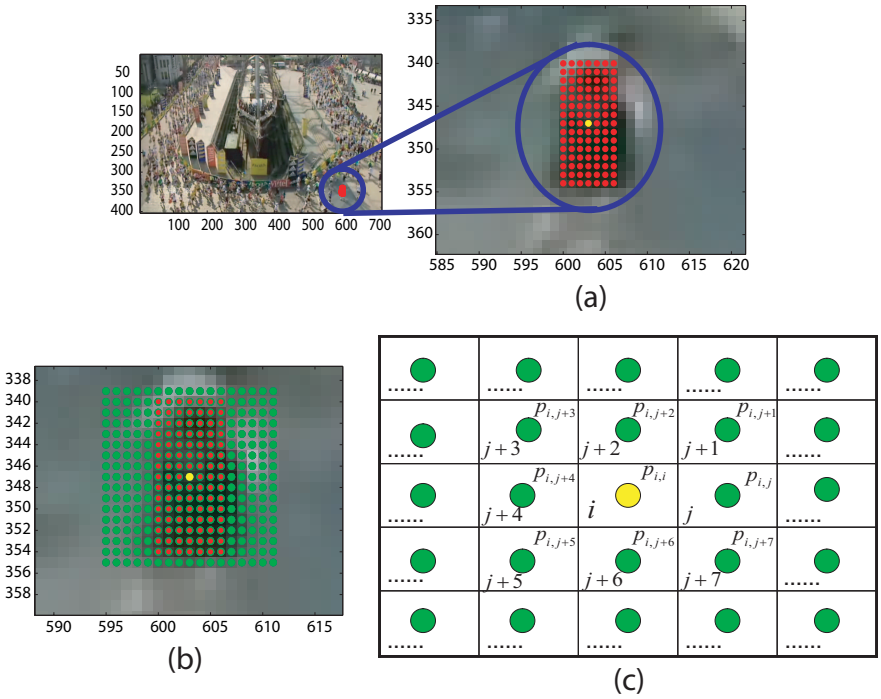

Figure 4.2: (a) Particles $o \in \mathcal{P}$ belonging to the individual I want to track. The yellow particle is the center cell. (b) The green particles represent the search area around the yellow particle. (c) The matrix of preferred walking directions. Each value in the matrix represents the probability of moving from the center cell i to the surrounding cell. The transition probability p_{ij} is computed by using Equation 4.1.

4.1.1 Static Floor Field - S_{ij}

The SFF is aimed at capturing the constant properties of the scene which are attractive in nature. These constant properties include the preferred areas such as dominant path often taken by the crowd as it moves through the scene, preferred exit locations etc. In our framework, for a given scene, the SFF is computed only once during the learning period which spans initial $M << N$

frames. The steps involved in the computation of the SFF are as follows: i) Computation of a point flow field; ii) Sink Seeking.

4.1.1.1 *Point Flow Field*

A 'point flow field' is representative of the instantaneous changes of motions present in the video. Each vector in this field is a 4-dimensional vector obtained by augmenting the local flow vector with the position information. The new vector is referred to as a 'point flow vector', hence the name 'point flow field'. Using the first M frames of an input video, $E = [f_1, f_2, \ldots, f_M]$, a dense optical flow can be computed between consecutive frames using the method of [129]. Next, for each cell (or pixel) i, a point flow vector, $Z_i = (X_i, V_i)$, is computed that includes both location $X_i = (x_i, y_i)$ and the optical flow vector $V_i = (v_{x_i}, v_{y_i})$. Note that V_i is the mean of $(M-1)$ optical flow vectors computed at pixel i from the first M frames of the video. All flow vectors averaged over M frames of the video then constitute the 'Point Flow Field' which represents the smoothed out motion information of the video in that interval. This smoothed motion information helps in computing the dominant properties (paths, exits) of the scene which is one of the primary objectives of the SFF. Figure 4.3(a) shows flow vectors generated for a marathon video using the dense optical flow computation [129]. The resulting point flow field is given in Figure 4.3(b).

4.1.1.2 *Sink Seeking Process*

Next, the point-flow field is used to discover the regions in the scene which are more attractive. These regions are called 'sinks'. The idea behind the sink seeking process is that the behavior of

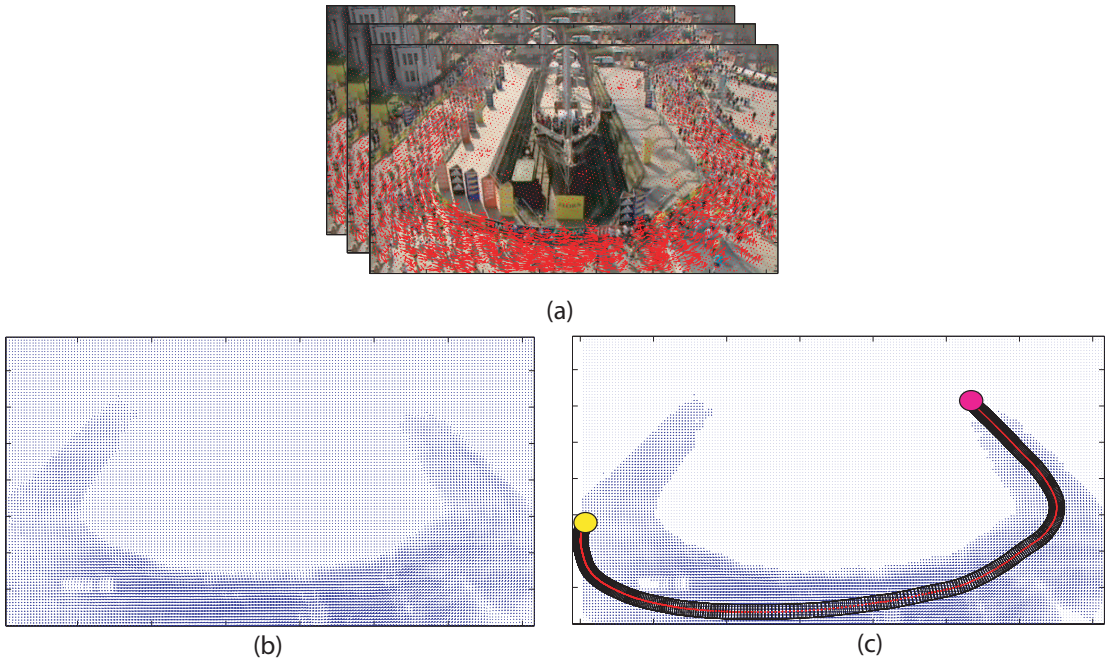

Figure 4.3: (a) Dense optical flow for frames $[f_1, f_2, \ldots, f_M]$ that represent the learning period. (b) The computed point flow field. (c) Sink seeking process. The yellow circle represents the initial location, while the red circle shows the corresponding sink. Black windows represent the area used to weight the local velocity and propagate the sink seeking process. The red trajectory represents the 'sink seeking path', while the number of black windows represents the corresponding number of sink steps.

large crowds of pedestrians in locations such as sporting events, religious festivals, train-stations etc., can be described as goal directed and rational because the members of the crowd have clear knowledge of what and where their goals lie [87]. Therefore, if I know the locations of the sinks, which are the desired locations (or goals) pursued by the crowd, then, for any given point in the scene I can compute a local force representing the tendency of the individual at that point to move

towards the closest sink. This local force will be a function of the shortest distance to the sink in terms of the appropriate distance metric.

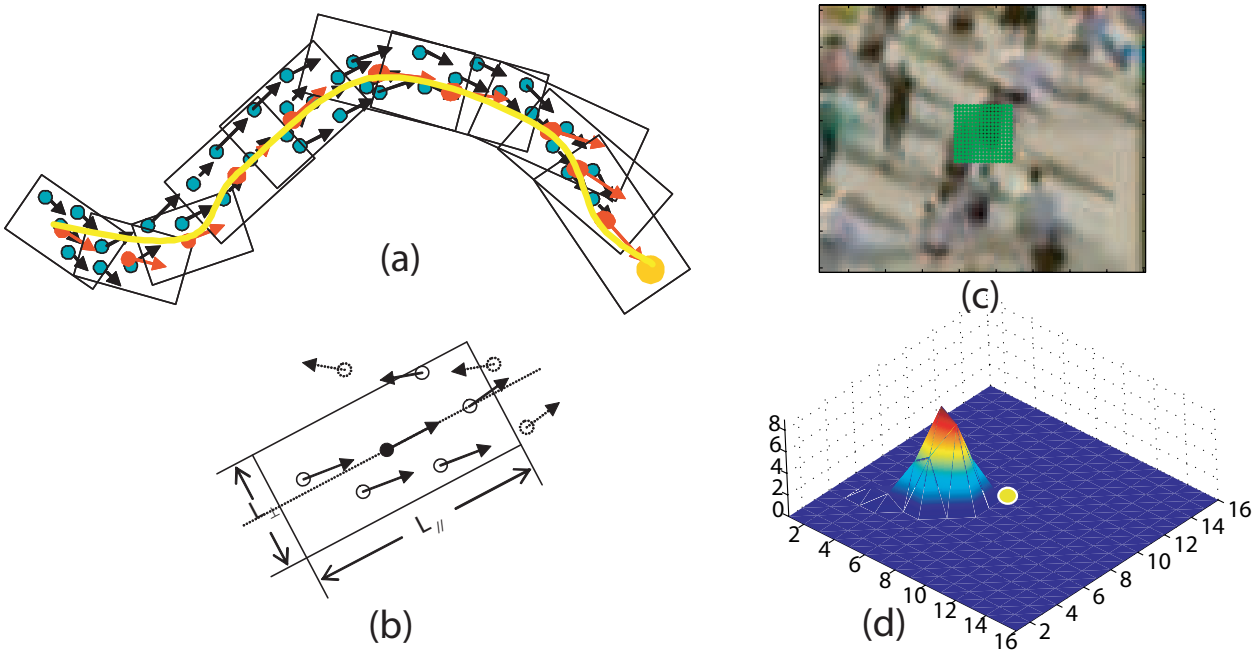

Figure 4.4: (a) Sink seeking (red: the states of the point flow in the sink seeking process, orange: the sink, rectangles: sliding windows, yellow: the sink path); (b) Sliding window (solid circle: the point flow under consideration; rectangle: sliding window; hollow circles: neighboring points; dotted circles: non-neighboring points). (c) The region at which I am interested in computing the DFF. (d) The computed DFF where the yellow circle represents the pixel i. In this case, the DFF is representing the strength of the relationship between the pixel i and other pixels.

To compute sinks and shortest distances, I initialize a grid of particles over the point flow field of the scene. Then, a particle dropped at a non-zero velocity location has the tendency to move to a new position under the influence of the neighboring point flow vectors. It then moves from the

new position to the next one and continues this process. To optimally combine the influence of the neighboring point flow vectors, the velocity at each new position is re-estimated as the weighted sum of its neighboring velocities (Figure 4.4(a)). The weights are computed using a kernel density method. If all the weights are below a threshold, which implies the new velocity is not significant enough to drive the particle to the next position. Therefore, the particle will stop, and the process of pursuing a new location is discontinued. I call this process the *sink seeking process*, and the last state (stopping state) of the process is called the *sink*. The corresponding path taken by the particle to reach the sink is called the *sink path*(Figure 4.3(c) and 4.4(a)). The length of the sink path is a quantification of the minimum number of steps required to reach the closest exit location in the scene. The number of steps taken during the sink-seeking process to reach the sink is called *seek steps*. This is also the distance metric used for representing the shortest distance. Note that the sink seeking process is carried out for each point in the point-flow field, thus generating one sink path per point. Formally, the 'sink seeking process' can be described as follows: Suppose $\{Z_1, Z_2, \cdots, Z_n\}$ is the point flow field of the video, where the state of the point i is defined as: $\widetilde{Z}_{i,t} = (\widetilde{X}_{i,t}, \widetilde{V}_{i,t}), t = 1, 2, ...$, and computed as:

$$\widetilde{Z}_{i,1} = Z_i, \quad \widetilde{X}_{i,t+1} = \widetilde{X}_{i,t} + \widetilde{V}_{i,t}, \tag{4.2}$$

$$\widetilde{V}_{i,t} = \frac{\sum_{n \in Neighbor(\widetilde{X}_{i,t})} V_n W_{t,n}}{\sum_{n \in Neighbor(\widetilde{X}_{i,t})} W_{t,n}}, \tag{4.3}$$

$$W_{t,n} = \exp\left(-\left\|\frac{\widetilde{V}_{t-1} - V_n}{h_{t-1}}\right\|^2\right), \tag{4.4}$$

In the previous equations, it is clear that the new position of a point only depends on the location and velocity at the previous state. However, the new velocity $\widetilde{V}_{i,t+1}$ depends not only on

the previous velocity but also on the observed velocities of its neighbors, which represents the motion trend of a local group. In this work, I employ the kernel based estimation that is similar to the mean shift approach [132]. However, there is one important difference. In mean shift tracking, the *appearance* of pixels in a small neighborhood of the object is used to determine the location of the object in the next frame. In our approach, I use *the location and the velocity* of neighboring points in the point flow field to determine the next location. There are other methods proposed in the literature for locating sources and sinks in the scene ([116]), however, they do not provide the shortest distance for each point in the scene. This distance is essential for our algorithm in order to compute the local SFF force. The following is the pseudo code of the sink seeking algorithm:

Algorithm 1: Sink Seeking Algorithm

Input: a set of n points $\{Z_i = (X_i, V_i)\}, i = 1, 2, ..., n$ in a video.

Output: the corresponding sinks $\{Z_i^*\}, i = 1, 2, ..., n$.

1 **for** *each point* i **do**

2 Initialize $t = 1$, $\widetilde{V}_{i,1} = V_i$ and $\widetilde{X}_{i,1} = X_i$;

3 Increment $t = t + 1$ and set $\widetilde{X}_{i,t} = \widetilde{X}_{i,t-1} + \widetilde{V}_{i,t-1}$;

4 Find the neighbors of $\widetilde{X}_{i,t}$ and compute the $W_{t,n}$;

5 **if** $\max_n W_{t,n} \geq T$ **then**

6 Compute $\widetilde{V}_{i,t}$ according to equation (4.4) and go to step 2

7 **else**

8 Set the sink $Z_i^* = (\widetilde{X}_{i,t-1}, \widetilde{V}_{i,t-1})$.

9 **end**

10 **end**

4.1.1.3 *SFF Generation*

The SFF is finally generated by using the sink steps for each sink path. I find the location (x, y) (in the image space) at which each sink path starts, and then place the value of corresponding 'sink step' at that location. Figure 4.5(d) shows the computed SFF for the sequence in Figure 4.1(c). It is interesting to note that the shape of the SFF emphasizes the notion that if you place a particle at any location it will roll down towards the exit. This is exactly what the goal oriented dynamics of the crowd in this scene represents. In the tracking algorithm, the shape of the SFF translates into a force in the direction that requires the minimum number of steps to reach the nearest exit location. That is, the difference between the values in cell i and j in this field is the measure of the S_{ij} parameter of Equation 4.1. Other SFFs are also shown in Figure 4.6.

4.1.2 Boundary Floor Field - B_{ij}

The purpose of the BFF is to capture influences generated by barriers/walls in the scene. When people are moving in confined spaces they tend to move away from the walls, and it is this repulsive effect that the BFF tries to capture. The computation of the BFF requires the localization of physical as well as virtual barriers in the scene. The virtual barriers arise from the presence of dynamically distinct crowd flows in the scene. The computation of the BFF is performed after a set time interval ΔT_B, and works on a group of frames defined by the parameter N_B. That is, computation of the BFF at time t uses frames $[f_t, f_{t+1}, \ldots, f_{t+N_B}]$.

The computation of the BFF is based on the crowd-flow segmentation algorithm proposed in Chapter 3. Recall that physical and virtual barriers in the scene are represented by the ridges in

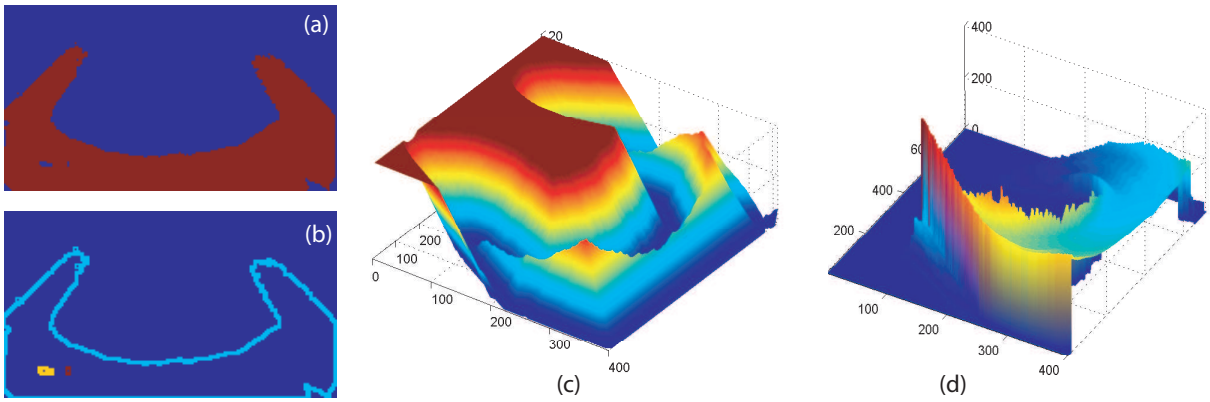

Figure 4.5: (a) Crowd-flow segmentation obtained by the method described in Chapter 3. (b) The edge map obtained from the segmentation. (c) The boundary floor field for the sequence shown previously in Figure 4.1(c). The higher values in the field represent the decreasing effect of the repulsive potential generated by the barriers. In this case, the barrier effect vanishes for distances greater than 20 pixels. (d) The static floor field computed by our algorithm for the sequence shown in Figure 4.1(c).

the Finite Time Lyapunov Exponent (FTLE) Field. The FTLE field was then used to compute a segmentation map where different labels represented different crowd-flow segments. In order to generate the BFF, I use this segmentation map and compute an edge map by retaining only the boundary pixels of each segment. Next, the closest distance to the wall/barrier for each pixel is determined by computing the distance transform of this edge map. An example of this process is show in Figures 4.5a-c. Note that, for a distance larger than a certain threshold, the barrier effect vanishes completely. This vanishing effect is represented by the flattening of the surface (the red region) in Figure 4.5(c). The difference between the values in cell j and i represents the value of B_{ij} in Equation 4.1. A few examples of BFFs are presented in Figure 4.6.

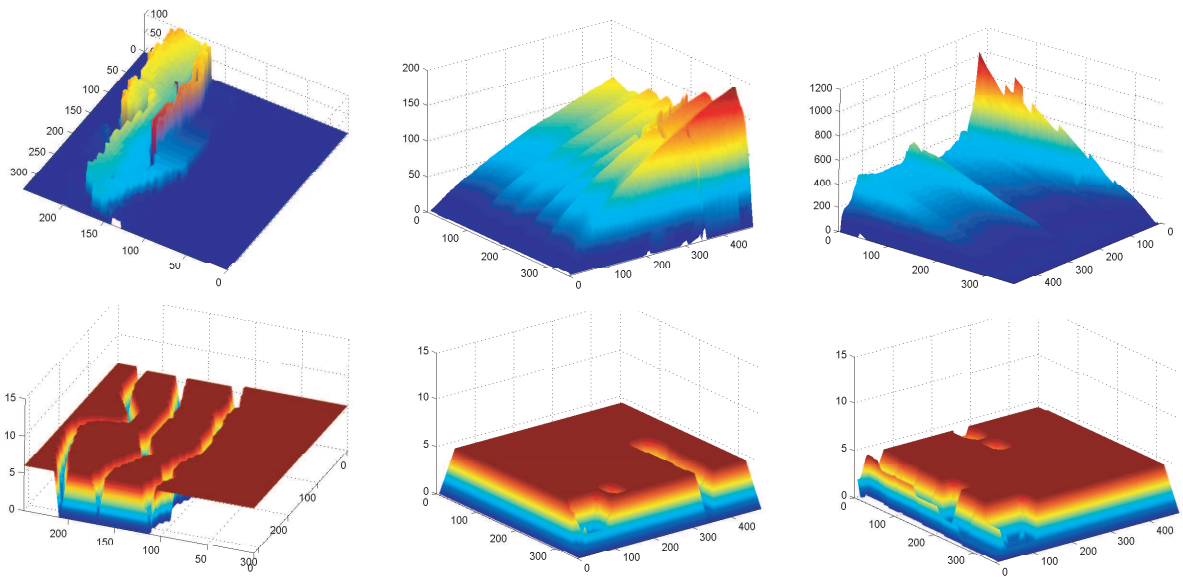

Figure 4.6: The SFFs (top) and BFFs (bottom) of various sequences. Left: For the sequence in Figure 4.1(e). Center: For the marathon sequence in Figure 4.1(a). Right: For the marathon sequence in Figure 4.1(b).

4.1.3 Dynamic Floor Field - D_{ij}

The objective of the DFF is to capture the behavior of the crowd around the individual being tracked. The instantaneous information about the crowd motion is an important cue for constraining likely future locations. This is even more important when the tracked individual acts suspiciously and does not obey the layout of the scene represented by the SFF and the BFF. The idea of the DFF is inspired by the active-walker models [117, 118] used for the simulation of trail formation.

In our framework, the instantaneous interaction among the members of the crowd is abstracted by using a particle based representation. For a given scene, the DFF is computed at each time

instant by using a sliding window of N_D frames. That is, for computing the DFF at time t, I use frames $F_D = [f_t, f_{t+1}, \ldots, f_{t+N_D}]$. I first compute the optical flow between consecutive frames in F_D, and stack them together to generate a 3D volume of optical flow fields. Next, a grid of particles is overlaid on the first flow field of the volume and numerically advected. During the advection, whenever a particle jumps from a cell (pixel) i to one of the neighboring cells j, the value of interaction between these cells (pixels) is increased by one. That means the DFF, D, can only have non-negative integer values. In addition, this construction results in one DFF per cell (pixel), where each DFF captures the strength of the dynamic interaction between the target pixel i and remaining pixels in the scene. A visualization of the DFF is shown in Figures 4.4(c)-(d). Since the DFF is meant to capture the local interaction of particles around the tracked individual, the Figures 4.4(c)-(d) represent the shape of the DFF, but only in that local neighborhood. The peak in Figure 4.4(d) represents the location where most particles end up if they pass through the yellow cell.

4.2 Experiments and Discussion

A detailed experimental analysis was performed on three marathon sequences shown in Figure 4.6. In addition, qualitative results are shown for a busy train-station sequence. In all the experiments, tracking started by selecting a rectangular region around the target object and using it to compute the gray-level appearance template. At each time instant, the next position of the target was chosen according to Equation 4.1, where the matrix of preferences around the current target location consisted of twice the size of the selected rectangular region. The appearance similarity was com-

puted using normalized cross correlation and the template was progressively updated at each time instant. I set the values of $k_S, k_D,$ and k_B equal to 0.02 for all experiments. The tracking results were stable for small changes in the values of these coupling factors. I used the first 50 frames of each sequence to construct the SFF. To compute the BFF and the DFF, the values of $N_B = 20$ and $N_D = 5$ were used.

Figure 4.7: Chips used for tracking. (a) Marathon-1. (b) Marathon-2. (c) Marathon-3.

4.2.1 Marathon-1

This sequence (Figure 4.1(a)) captures participants in a marathon from an overhead camera. It is a difficult sequence due to the severe occlusion among the participants, and the similar looking outfits worn by most athletes. The sequence has 492 frames, but each athlete, on average, remains in the field of view for 120 frames. I manually selected 199 individuals, shown in Figure 4.7(a), from various frames for tracking. The average size of the selected chip was 14×22 pixels.

A set of trajectories generated by our tracing algorithm is shown in Figure 4.8(a). In total, I was able to track 143 out of 199 individuals without any tracking error i.e. correct label was

maintained throughout the time duration for which the athlete was in the FOV. The number of frames for which 199 athletes were tracked is provided in Figure 4.12(a). The quantitative analysis of the tracking was performed by generating ground-truths trajectories for 50 athletes, which were selected randomly from the initial set of 199 athletes. The ground-truth shows that these 50 athletes were visible for an average of 77 frames, and our algorithm tracked them for an average of 72 frames. This is summarized by the first 50 bars in the graph of Figure 4.12(d). The average tracking error is summarized by the first 50 green bars in the graph of Figure 4.13(a). A qualitative visualization of the accuracy of the tracking is demonstrated in the first row of Figure 4.9, where red trajectories are the ground-truth and yellow trajectories are from our algorithm. The tracking failure on this sequence resulted in situations when the target was completely occluded either by another athlete or by the street-light in the scene. Since I did not use any prediction mechanism, I could not recover from full occlusion. However, partial occlusion was handled amicably by our tracker. Some tracking failures are shown in Figure 4.10(a).

4.2.2 Marathon-2

This sequence (Figure 4.1(c)) also involves a marathon. However, the camera is installed on a high-rise building to increase the FOV. As a result, the number of pixels on each individual is fewer. In addition, there are drastic illumination changes when athletes move into the shadow of the neighboring buildings. This sequence has 333 frames. I manually selected 120 individuals (Figure 4.7(b)) from various frames for tracking. The average size of the selected chip was 13 $\times$ 16 pixels.

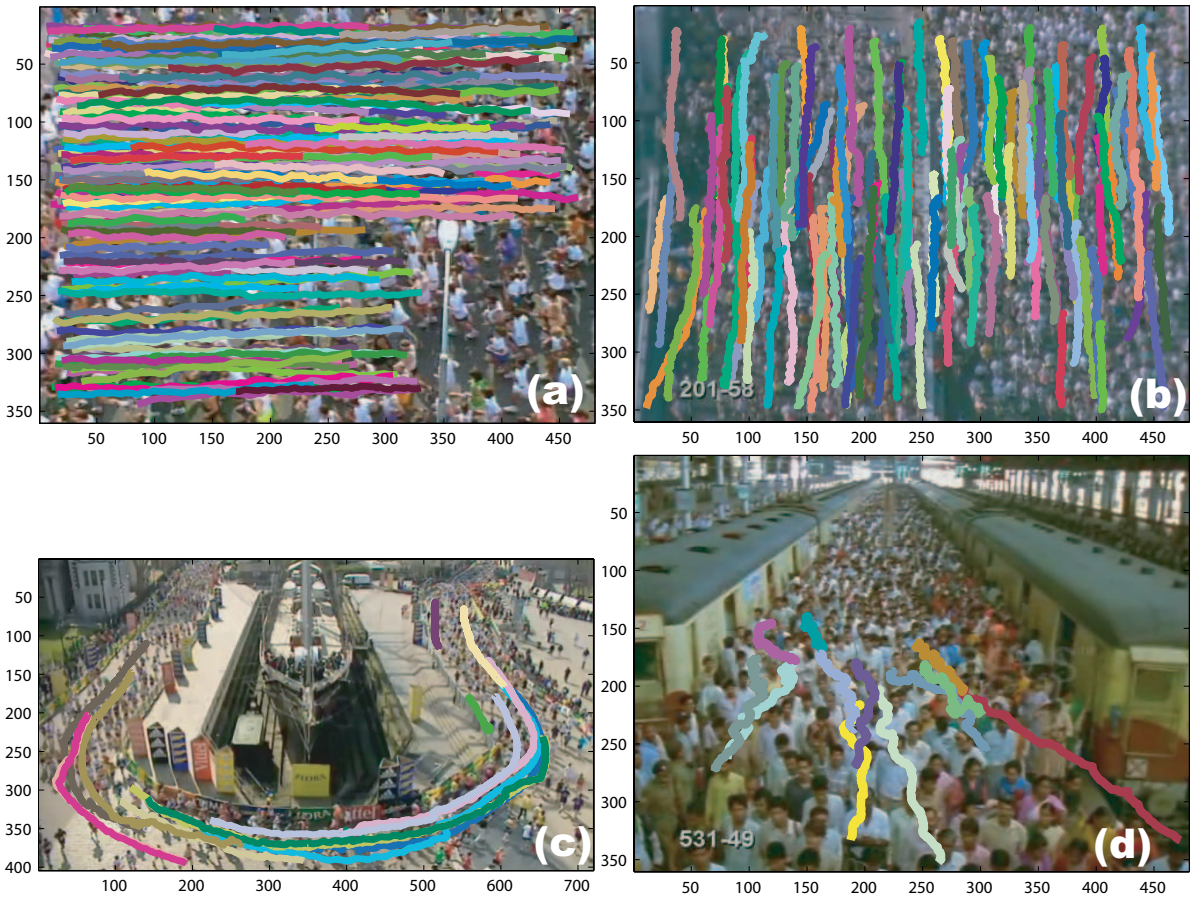

Figure 4.8: Displays trajectories of individuals which were accurately tracked by our method. (a) Marathon-1. (b) Marathon-2. (c) Marathon-3. (d) Train Station.

A set of trajectories generated by our tracing algorithm is shown in Figure 4.8(b). In total, I tracked 117 of the 120 individuals correctly. The number of frames for which each individual was tracked by our method is shown in Figure 4.12(b). A quantitative analysis was performed by generating ground-truths for 20 athletes. The length of ground-truth trajectories and trajectories generated by our tracker is summarized by bars 51-70 in Figure 4.12(d). A qualitative comparison with the ground-truth for some of the trajectories is presented in the second row of Figure 4.9.

The average tracking error is summarized by the green bars (51-70) in the graph of Figure 4.13(b). It can be observed that our tracking was very accurate in most cases, and able to overcome the illumination changes with the aid of the DFF and the SFF. Some tracking failures in this sequence are shown in Figure 4.10(b).

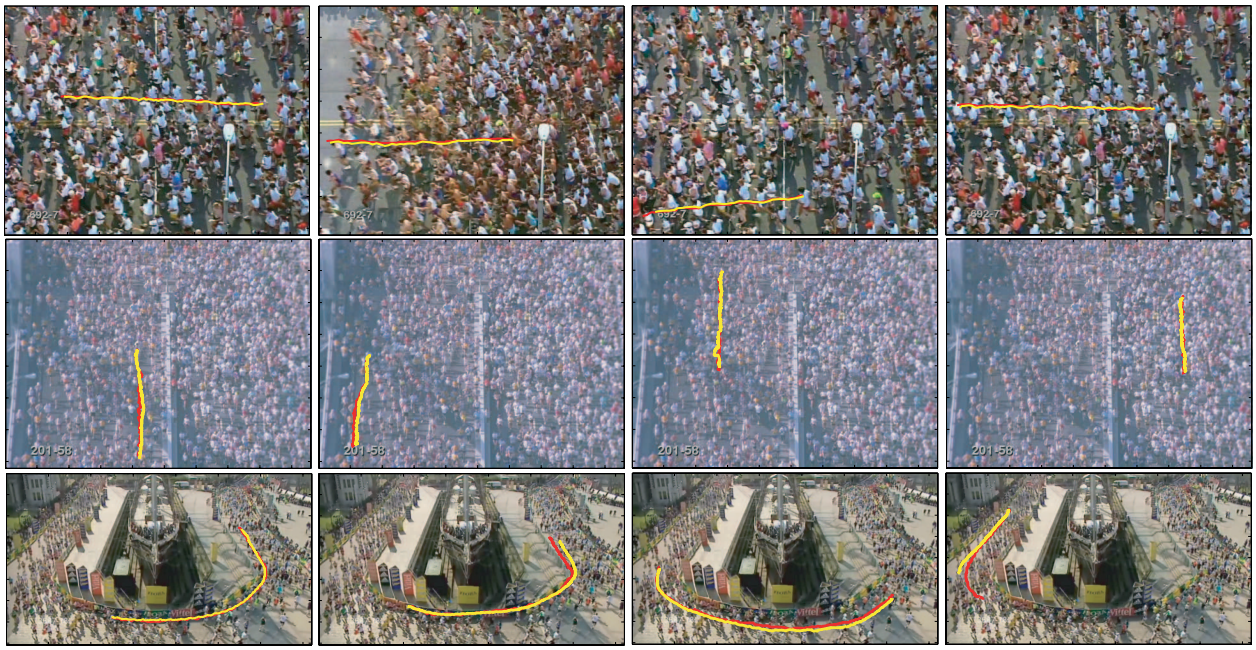

Figure 4.9: A comparison of tracking (yellow tracks) with the ground-truth (red tracks).

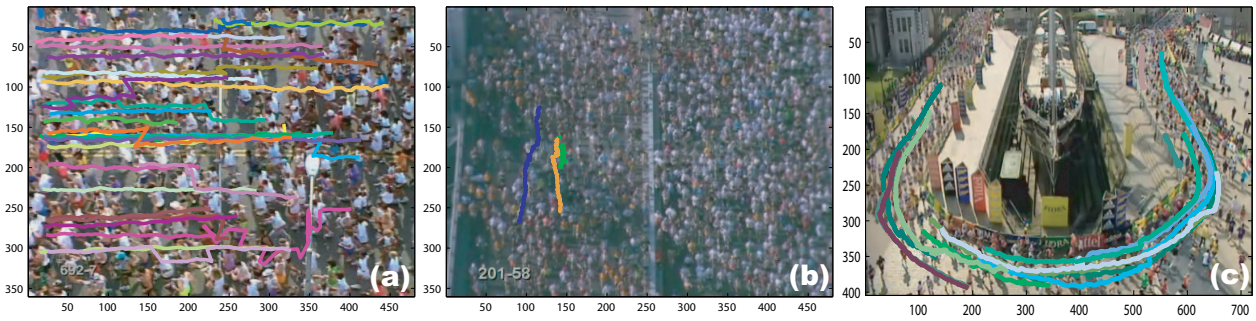

Figure 4.10: The failure cases. (a) Marathon-1. (b) Marathon-2. (c) Marathon-3.

4.2.3 Marathon-3

The third sequence (Figure 4.1(c)) is extremely challenging due to two factors: 1) appearance drastically changes due to the U-shape of the path; 2) the number of pixels on target varies due to the perspective effect. The fewer number of pixels make it more difficult to resolve even the partial occlusions. The sequence is 453 frames long. For tracking, I manually selected 50 individuals (4.7(c)). The average size of the selected chip was 14×17 pixels. In total, I was able to track 38 of the 50 individuals without any tracking error (Figure 4.8(c)). The number of frames for which each individual was tracked and a comparison with the 15 ground-truth trajectories is summarized in Figure 4.12(c) and Figure 4.12(d), respectively. The average tracking error is summarized by bars 71-85 in Figure 4.13(a).

In this sequence, I also performed the tracking of an individual who was moving in the direction opposite to the normal flow of the crowd. The result is shown in Figure 4.11(b). I would like to emphasize that our method is able to track this person due to the presence of the DFF, which captures the instantaneous motion information. Therefore, I was able to track this individual for 400 frames, even when he was behaving differently than the crowd. In addition, I performed tracking on a busy train-station sequence shown in Figure **??**(d).There, I tracked 20 individuals. The qualitative results are shown in Figure 4.8(d).

4.2.4 Analysis

On of our results will be discussed in detail to provide an intuitive insight as to how the floor fields are helping in improving the tracking accuracy. For this purpose, I picked a track from Marathon-

3, where the athlete was wearing a black shirt and running away from the camera (Figure 4.2(a)). During the course of tracking, the appearance became ambiguous due to other neighboring athletes who were also wearing black shirts. Figure 4.11(a) (top-left) shows the similarity surface obtained by matching the appearance template in a 16×16 neighborhood, for one of those instances. The surface was relatively flat, showing a lack of a good match for the tracked person in the current frame. If I was to use only this surface, there was a high probability that the tracker would jump onto one of the neighboring athletes wearing the same clothing. However, floor fields helped in resolving this ambiguity, as visible from the final decision surface (Figure 4.11(a) bottom-right). The DFF shown in Figure 4.11(a)(top-right) guided the tracker by emphasizing the direction taken by most particles from the current location of the target. I am able compute this reliably, because the DFF integrates the motion information over a small interval, and therefore, does not make a hasty decision. Similarly, the SFF shown Figure 4.11(a)(bottom-left) allowed the tracker to take into consideration the direction which would take the target to the exit point. In short, the DFF and the SFF together helped in resolving the appearance ambiguity, and allowed our tracker to maintain the correct label. Note that, in this example, the BFF was not playing any part as the individual was running on the flat surface of the BFF.

4.2.5 Mean-Shift Comparison

I then performed a quantitative analysis by comparing the results with a color based mean-shift tracker. The comparison is performed with respect to the ground-truth generated for the three marathon sequences. The mean-shift was initialized using the same regions and appearance was

updated during the course of tracking. The tracking error was computed as the average distance in term of pixels from the ground-truth location over the entire video. The results are summarized in Figure 4.13(a). The green bars in the graph correspond to the average error of our tracking algorithm, while the yellow bars correspond to the average error committed by the mean-shift tracker. It can be observed that our method works much better than the mean-shift tracking method. This verifies our initial observation that in videos of high density crowds, appearance alone is not a reliable cue, and therefore, other sources of information present in the scene should be exploited.

4.2.6 Contribution of Floor Fields

This experiment was performed to test the contribution of floor fields towards the accuracy of tracking. The comparison is performed using ground-truth trajectories from Marathon-1, for which I obtained accurate tracking using all three floor fields. There were 35 such trajectories in total. Next, I ran the tracker multiple times by first turning off the SFF and the BFF, and then by turning off the DFF and the BFF. The error was computed in a manner similar to the mean-shift experiment. The graph in Figure 4.13(b) shows the comparison. It can be observed that I obtained the minimum error by using all three fields. This points to the utility of using all floor fields together.

4.3 Summary

In this chapter, I have presented an algorithm for tracking individual targets in high density crowded scenes containing hundreds or thousands of people. Tracking in such a scene is extremely challenging, due to the small number of pixels on targets, ambiguous appearance resulting from dense

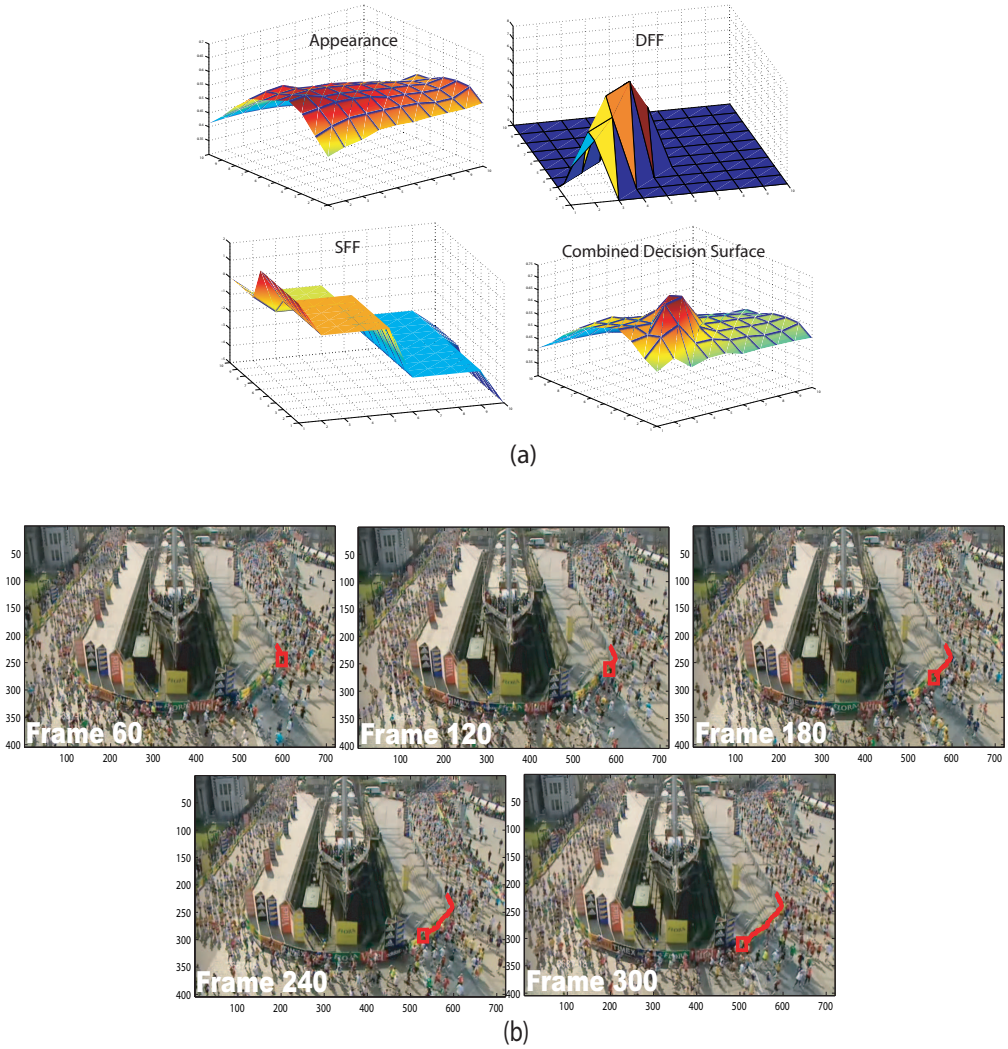

Figure 4.11: (a) (top-row from left to right) Appearance similarity surface and the local DFF. (bottom-row from left to right) The local SFF and the final decision surface obtained by merging appearance, the DFF, and the SFF according to Equation 4.1. (b) Tracking when the individual is going against the flow of the crowd.

packing, and severe inter-object occlusions. The novel aspect of the proposed tracking algorithm to overcome these challenges was called the *scene structure based force model*. This force model

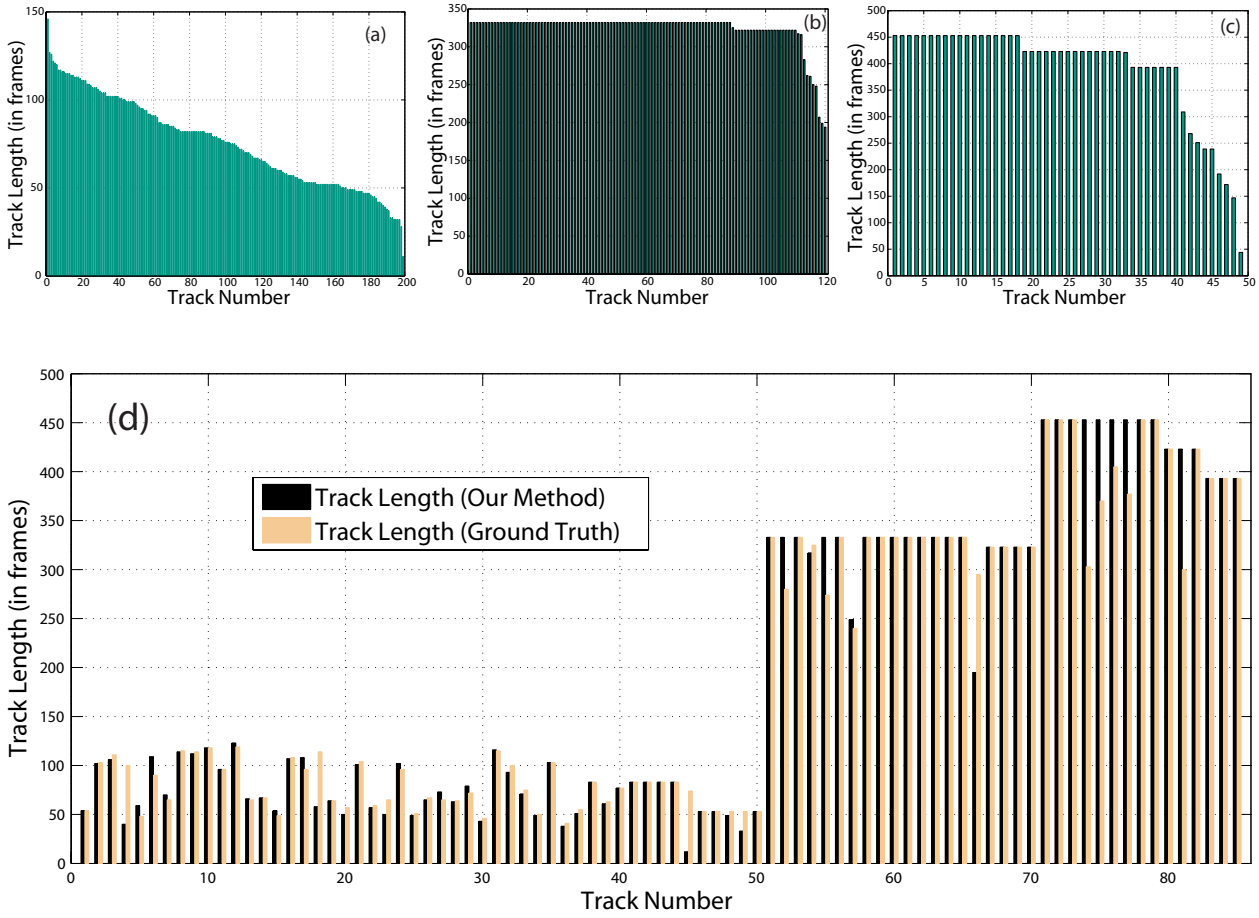

Figure 4.12: The number of frames for which the target was tracked. (a) Marathon-1. (b) Marathon-2. (c) Marathon-3. (d) A comparison of track lengths using the ground-truth: 1 to 50 Marathon-1; 51-70 Marathon-2; 71-85 Marathon-3.

captured the notion that an individual, when moving in a particular scene, is subjected to global and local forces that are functions of the layout of that scene and the locomotive behavior of other individuals in his/her vicinity.

The key ingredients of the force model were three floor fields, inspired by the research in the field of evacuation dynamics, namely, *Static Floor Field* (SFF), *Dynamic Floor Field* (DFF), and

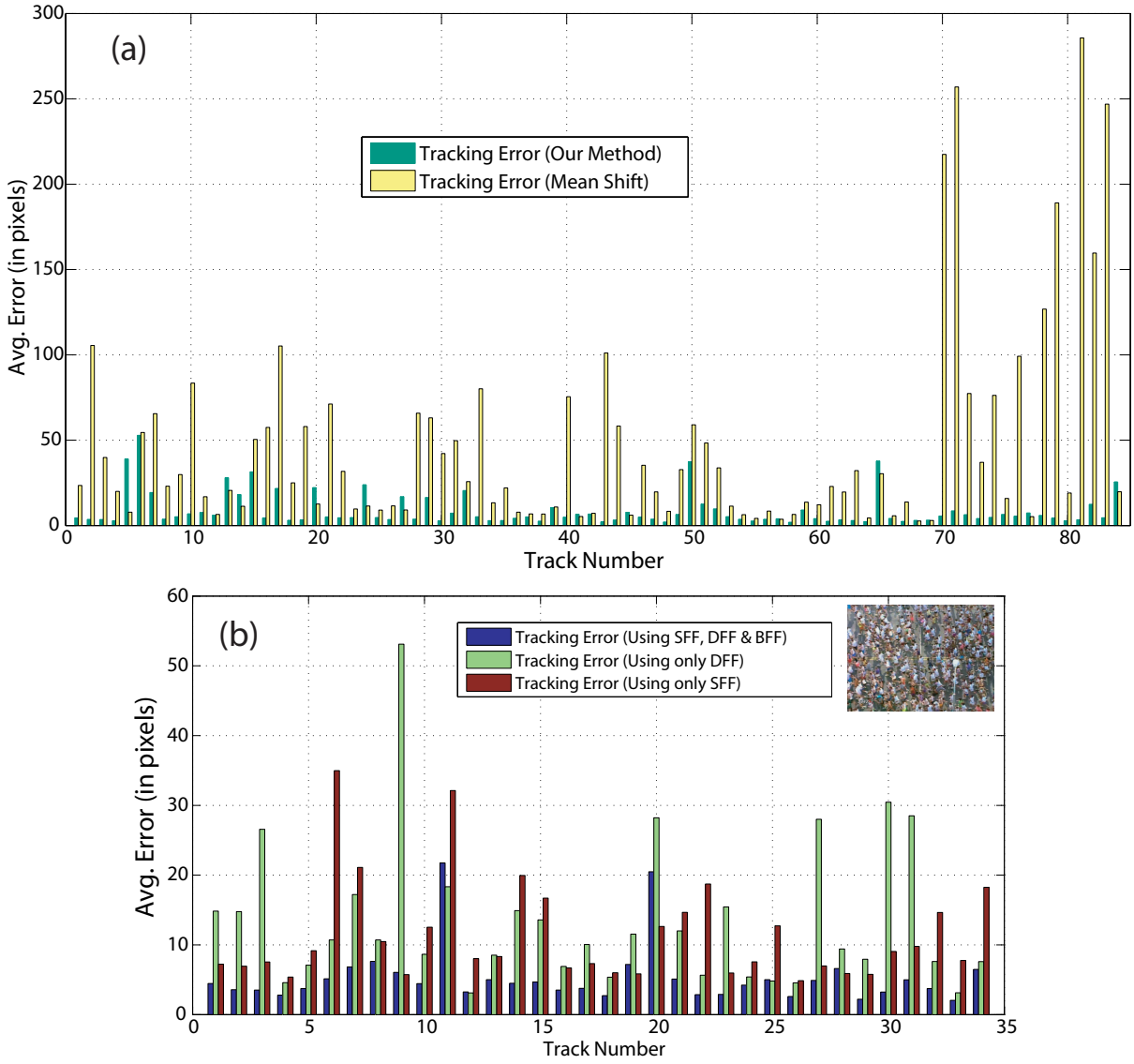

Figure 4.13: (a) Comparison of the tracking error of our method against the mean-shift tracker. The bars represent the average error over the entire track. The length of the tracks is given in Figure 4.12(d) (1 to 50 Marathon-1; 51-70 Marathon-2; 71-85 Marathon-3). (b) Contribution made by different floor fields towards the tracking accuracy.

Boundary Floor Field (BFF). These fields determine the probability of moving from one location to another by converting the long-range forces into local ones. The SFF specified regions of the scene which were attractive in nature e.g. an exit location. The DFF, which was based on the idea of active walker models, corresponded to the virtual traces created by the movements of nearby individuals in the scene. The BFF specified influences exhibited by the barriers in the scene e.g. walls, no-entry areas. By combining cues from all three fields with the available appearance information, I was able to track individuals in high density crowds. Results were reported on real-world sequences of marathons and railway stations containing thousands of people. A comparative analysis, with respect to the appearance based mean-shift tracker was also conducted by generating the ground truth. The results of this analysis demonstrate the benefit of using floor fields in crowded scenes.

In the next chapter, I propose a target re-acquisition methodology, with an aim to reduce the incidence of broken trajectories resulting from frequent occlusions and limited field of view of the camera.

CHAPTER 5
TARGET RE-ACQUISITION IN CROWD AND AERIAL VIDEOS

In this chapter, I propose an algorithm that has been developed to augment a generic tracking algorithm to perform persistent tracking in crowded and other types of scenes. The term 'persistent tracking' refers to the ability of the tracking algorithm to maintain the correct label of an object when the object is occluded or when it leaves and re-enters the field of view (FOV) of the camera. The occurrence of occlusion is very frequent in crowded scenes due to a high number of interacting objects. This makes it necessary for the tracking algorithm to have a re-acquisition capability. Since, trajectories of moving objects are critical for interpretation of their behavior, any missing information will result in a significant degradation in the accuracy of any event recognition modules using these trajectories. In this Chapter, without loss of generality, I initially setup the reacquisition problem for moderate to high density traffic scenes which are viewed either by moving aerial cameras or cameras mounted on high rise buildings. In the later part of the Chapter, I will show the applicability of the re-acquisition on a high density crowded scene containing people as well.

The main focus of our proposed re-acquisition idea is to utilize the contextual knowledge present in the scene in order to re-acquire previously tracked objects. Generally, the contextual knowledge consists of the information that is necessary to understand and interpret the meaning of a process/event taking place in a scene. For the purpose of this thesis, this contextual knowledge is

divided into two categories, *motion context* (MC) and *appearance context* (AC). The MC is based on the insightful observation that *the locomotive behavior of an object (e.g. a car, a person) in a given environment provides information about the locomotive behaviors of nearby objects (e.g. other cars or persons) that are in the same environment.* This is true because cars moving along the same stretch of a road are subjected to similar constraints, for example which path they can take, the shape of the path, road conditions, and speed restrictions. Therefore, the motion of one car contains information that can be used to interpret how neighboring cars will behave, thus providing the necessary MC. I can apply this same idea towards re-acquisition, that is to predict the movement of an occluded car by using the motion of other cars that are part of its MC. Figure 5.1 provides a pictorial description of the concept of MC. All the cars circled in yellow are part of the MC of the car circled in red. This is valid for the following two reasons: first, the yellow cars have a spatial relationship with respect to the red car that remains relatively unchanged over a short time; second, all the yellow cars and the red car are subjected to same physical constraints such as road conditions, road shape and direction of motion. Therefore, when the red car goes under the bridge, the contextual knowledge consisting of how the yellow cars are behaving can be used to predict the future location of the red car. It is important to note that since in most cases our objects of interest are moving cars, I will use terms ‘objects’ and ‘cars’ interchangeably.

The AC is based on the notion *that when a target car re-appears after undergoing occlusion, its appearance will have to be discriminated with respect to the appearance of other unobserved cars*. This is true because the car that appears in a FOV of an aerial camera could be either a new car, or one of the cars from the set of currently occluded cars. By constructing an appearance

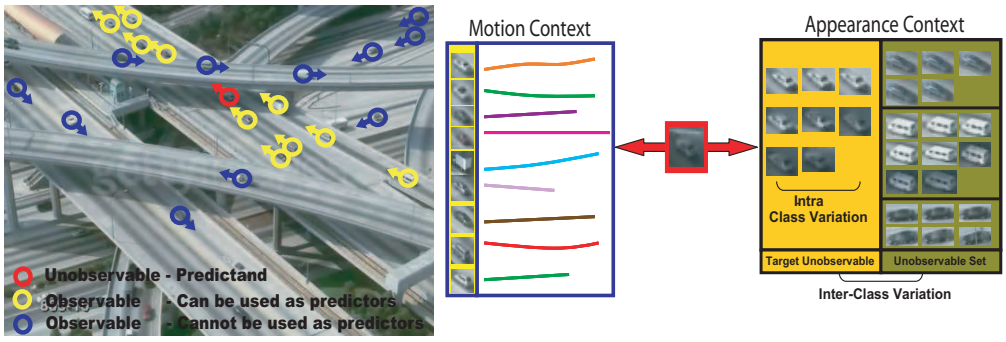

Figure 5.1: The figure illustrates the concept of *Motion and Appearance Context*. The motion context of a car, which in this example is circled in red, is defined by the cars that have motion dynamics similar to that of the selected car. In this case, these cars are circled in yellow. The cars circled in blue are not part of the motion context of red car, because the blue cars have motion dynamics which are different from the red car. Tracks corresponding to the yellow cars, which are used for predicting the motion of the red car, are shown in the blue rectangle on the right. The appearance context of the red car consists of the other cars which are currently unobservable. These cars are shown in the green rectangle on the far right, where I have multiple observations for each car. The yellow rectangle displays the observations of the red car. The appearance context of the red car is then computed using intra and intra class variations of the red car with itself and with the unobservable cars respectively.

descriptor for a car that takes into consideration its own appearance history and the appearance of other unobserved cars, I hope to build a more discriminative representation. This will make it easier to establish the correspondence between the appearance of the newly detected car and that of the cars from the occluded set. It is important to note that our definition of *appearance context*

is different from the definitions currently used in the literature, where the *appearance context* is defined exclusively in terms of the foreground vs. background appearance features. Our definition extends the idea of AC by taking into consideration not only the previous observations of the target car itself, but also the appearance of other cars in the scene.

Figure 5.1 provides a pictorial description of the concept of AC. The car circled in red is about to become occluded. All the observations related to the appearance of this car until this point are displayed in the yellow rectangle. While the red car is undergoing occlusion, there are other cars in the scene which are currently unobserved and have not been reacquired yet. These cars are shown in the green rectangle. The AC requires that the appearance model of the red car be discriminative with respect to the cars in the green rectangle, since only these cars are approaching the re-acquisition stage. Therefore, the appearances of the cars in the green rectangle can be used to build a more discriminative appearance descriptor for the red car.

Before moving on to the next section, I would like to emphasize that in our proposed algorithm the MC and the AC provide a unifying theme which states that information from other objects in the environment can be utilized to solve the re-acquisition problem.

5.1 Overview

The proposed concept of MC is implemented in a regression framework, which is inspired by the research conducted in the field of oceanography for search and rescue operations at sea [122, 123, 124, 125]. The goal in that scenario is to narrow the search area based on the best prediction of the lost object, given its initial position and the mean current (velocity field) of the sea. However,

prediction just based on the mean current becomes difficult due to the presence of large velocity fluctuations. Such fluctuations will cause the object to drift away from the track predicted by the mean velocity field. In order to overcome this problem, one can utilize other floating objects in the same area, for example debris or human made floating devices, which can be tracked or observed by satellites. The tracks of these objects can be used to make a prediction about the potential location of the lost object since they are both subjected to similar physical forces.

I have mapped this to the scenario of aerial videos by defining the object (car) on the ground which has either gone out of the FOV of the camera or has undergone occlusion due to some terrain feature, as the object that I want to predict. Let us call this object 'predictand'. Tracks from other objects (cars) which are either currently present in the FOV of the camera or have moved along the same path in the past, can be used to estimate the likely location of the 'predictand'. Let us call all such objects (cars) 'predictors'. I use tracks of objects that are currently present in the FOV of the camera because they move along the same stretch of the road, and, therefore, are subjected to similar constraints in terms of, e.g., the path they can take, the shape of the path, the road conditions and the speed restrictions. Tracks from the past objects are also used because they provide potential observations about the likely paths taken by objects that passed through the current scene.

Let $r_1(0), r_2(0), ..., r_N(0)$ be the starting positions of N objects, $\{O_i\}_{i=1}^{N}$, in the image plane at time $t = 0$. Corresponding to each O_i, I have a set $C_i \subset O$, of cardinality $M \leq N$, defining the MC. Then, assume that the trajectories of first $p = N - 1$ objects (cars) $r_1(t), r_2(t), ..., r_p(t)$ are observed during the time interval $(0, T)$, while the trajectory of the last object, $r_N(t)$, is not

observed. The problem is then defined as making a prediction about the location of the unobserved object O_N, given trajectories of predictors in C_N and the initial 'predictand' position. The optimal prediction in the mean square sense is $E \mid \widehat{r}_N(T) - r_N(T) \mid \rightarrow min$ [122], where $\widehat{r}_N$ is given by the conditional expectation $\widehat{r}_N(T) = E(r_N(T) \mid r_1(t), r_2(t), ..., r_p(t), 0 \leq t \leq T)$ [126], based on all the observations. However, this expectation is difficult to find explicitly, as observed by Piterbarg *et al.* [122]. To overcome this problem, I resort to a regression framework which estimates the future location of the 'predictand' by employing the data from the predictors in a least square sense. Note that, for clarity purposes, the above mentioned equations assume that all p objects are in the set C_N.

I solve the problem of finding the optimal predictors by employing a concept known as Lyapunov Characteristic Exponent (LCE) from the Theory of Linear and Non-Linear Dynamical Systems. LCE measures the mean exponential rate of convergence or divergence of nearby trajectories in a state space. It is used for measuring sensitivity of a dynamical system to initial conditions. A track generated by a moving object in a given scene can be treated as a trajectory taken by a dynamical system through the phase space which is defined by the position and time variables (r, t). Tracks of any two objects moving in a scene can be considered as two different trajectories taken by the dynamical system in this phase space. I can get a measure of dynamical similarity between these trajectories by computing the LCE. Tracks showing high similarity can be used as predictors for each other

For implementing the AC I maintain a set of objects $U \subset O$ at all times t. The appearance of an object O_i, currently undergoing occlusion, is encoded in terms of intra-class appearance variations,

which are obtained by using all previous observations of O_i, and pairwise inter-class appearance variation with respect to other objects in the set U. Therefore, if l is the cardinality of the set U, then for each $O_i \in U$ I have $(l-1)$ observation sets of inter-class variation. Next, I consider each inter- and intra-class variation as an observation of a single dimension random variable drawn from a Gaussian probability density (PDF) specific to that inter/intra class variation set. The means and standard deviations of $(l-1)$ inter-class and one intra-class Gaussian PDFs are computed using the corresponding observations of inter- and intra-class variation. In order to establish whether a newly detected blob, Q, is O_i, first observations of intra- and inter-class variation of Q are computed using the observations of O_i and other objects in the set U, respectively. Then the similarity is computed by the Bhattacharya metric which quantifies how well the observations of intra- and inter-class variation of B are described by the PDFs of inter- and intra-class variations of object O_i. Note that, by employing inter-class variations for object re-acquisition, I am able to incorporate the appearance of other objects into the model. Also, by directly modeling the differences in the appearances of cars rather than differences with respect to a mean class appearance, I hope to capture the physical variation in the shape of the car over time. I believe that such variations have a number of 'inter-class' and 'intra-class' properties that are functions of the make, model and color of the car, and, therefore, can be used to develop a discriminative representation.

5.2 Framework

In this section, the theoretical and implementation details of the proposed framework are presented. I begin by presenting the modeling details of MC and then explain the modeling details of AC.

5.2.1 Modeling Motion Context

Let N be the total number of objects observed until time T, which are represented by the set O. Let $V_T \subseteq O$ be the objects that are visible in the current frame of the video. $P \subset O$ is the set of objects whose locations are being predicted. Corresponding to each $O_i \in P$, I have a set $C_i \subset O$ of objects that act as the predictors for O_i. Last, I maintain a set of trajectories R, where $r_i(0, ..., T)$ is the trajectory corresponding to object O_i. Our goal is to predict the next location, $r_i(t)$, of object $O_i \in P$ at time t where $t > T$, given its last location $r_i(T)$. Note that $r_i(t)$ is a vector consisting of image location $[x_i(t), y_i(t)]$. The starting and the current locations $r_j(T)$ of all M predictors in C_i are also known.

Given this information the current location of the object $O_i \in P$ can be predicted by using the following regression model:

$$r_i(t) = A(t)r_i(T) + b(t) + z_i(t), \tag{5.1}$$

where $A(t)$ and $b(t)$ are an unknown 2x2 matrix and 2-vector, respectively. The variable $z_i(t)$ represents a stochastic process with zero mean uncorrelated for fixed T.

The unknown matrix $A(t)$ and vector $b(t)$ for O_i are computed using the initial and the current locations of predictors in the set C_i. There are six unknown parameters, i.e., four entries of matrix A and two entries of vector b. Therefore, I must have at least three or more predictors in the set C_i to solve the system. If I have less than three predictors, I resort to appearance only re-acquisition, which is explained later.

The least square estimates of $A(t)$ and $b(t)$ are obtained by the equation: $\widehat{A}(t) = \frac{S(t)}{S^{-1}(T)}$ and $\widehat{b}(t) = m(t) - \widehat{A}(t) - m(T)$, where $m(t) = \frac{1}{p}\sum_{i=1}^{p} r_i(t)$ is the center of mass of the predictor cluster. The value of $S(t)$ is calculated using the relation $S(t) = \sum_{i=1}^{p}(r_i(t) - m(t))(r_i(0) - m(0))'$. Finally, the obtained estimator is used to predict the unobservable object O_i using $\widehat{r}_i(t) = m(t) + S(t)S(0)^{-1}(r_i(0) - m(0))$.

The most significant step in the above formulation is the computation of the set C_i for each object O_i. Once I have this set, finding the solution of the above equation is straightforward. The next section will discuss the theory and the algorithm behind the computation of predictor set C_i.

5.2.2 Selecting Predictors

The predictors for each occluded and unobservable object are selected using a methodology based on the concept of Lyapunov Characteristic Exponent (LCE). I now briefly describe LCE before moving on to the algorithm for predictor selection that utilizes it.

5.2.2.1 *Lyapunov Characteristic Exponent*

The LCE is a tool for measuring the chaoticity in dynamical systems ([127][128]). It measures the rate of exponential divergence between neighboring trajectories in a phase space of the dynamical system. For a given dynamical system, $\dot{x} = f(x)$, the LCE is defined as $\gamma = lim_{t\to\infty}\chi(t)$, with $\chi(t) = \frac{1}{t}ln\frac{|\xi(t)|}{|\xi_0|}$. $\xi(t)$ is the current state of the dynamical system, obtained by solving the differential equation that controls the evolution of the system through the phase space; $\xi(0)$ is the initial state of the system. The value of γ, which is close to zero, represents a system that is stable

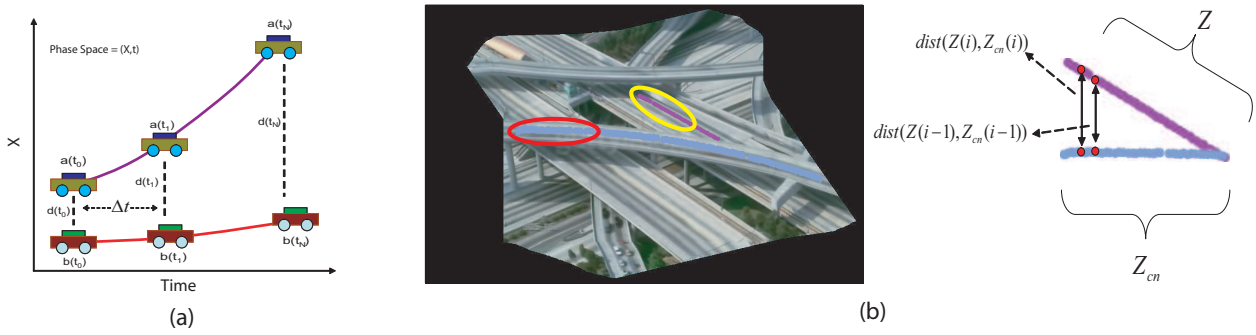

Figure 5.2: (a) Initially, cars a and b are $d(t_0)$ units apart. Over $(\Delta t \times N)$, in a series of time steps from t_0 to t_N, the two cars move until the distance between them becomes $d(t_N)$ units. This divergence is quantified by LCE and can be calculated using Equation 5.3. (b) A portion of potential predictor trajectory (shown in the red ellipse) is first normalized with respect to the predictand trajectory (shown with the yellow ellipse). Next, at each time step i, the Euclidian distance is computed between the corresponding points of the two trajectories. To compute LCE, these Euclidian distances are accumulated over the entire length of the trajectory using Equation 5.3.

to small perturbations in its initial conditions, while $\gamma > 0$ signals the presence of chaos in the system. The value $\chi(t)$ is called the Lyapunov Characteristic Exponent. In practical computations it is not possible to take the limit to infinity. Therefore, I follow the evolution of the system up to a pre-determined number of steps.

Another related issue is that I typically do not have information about the differential equations that govern the temporal evolution of a dynamical system. This is the case in the scenario discussed in this Chapter in which the state of the observed scene is governed by some unknown differential

Figure 5.3: The results of the predictor selection procedure. (a) Shows the set of trajectories that have been observed so far in this scene. This set contains trajectories generated from observed objects as well as trajectories that have been predicted in the past. (b) Shows the predictand trajectory for which I want to select the predictors. (c) Predictor selection result returned by our selection procedure. It is evident that our procedure was able to select the objects whose motion dynamics are similar to the predictand trajectory.

equations. To overcome this problem, I use an alternative approach proposed by Wolf et. al. [133] for computing the LCE, in which the definition of LCE is replaced by

$$\chi(t) = \frac{1}{N} ln \frac{d_t}{d_0}. \tag{5.2}$$

Here, d_t is the distance between two trajectories in the phase space at time t. These trajectories were initially separated by distance d_0. A pictorial description of LCE computation using the above equation is given in Figure 5.2(a).

5.2.2.2 *Predictor Selection*

Using LCE to compute predictor set C_i for a given predictand $O_i \in P$ requires special consideration. First, the 'predictand trajectory', $r_i(t)$, is labeled as the reference trajectory and the locations of all the other trajectories in G, where G is the set of all tracks, are computed relative to the reference trajectory. Pairwise LCE are computed between the reference trajectory and the remaining trajectories, one at a time, using Equation 5.3.

In order to accurately predict the position $r_i(t)$ of the unobserved object O_i at time t, I must select a set of meaningful predictors C_i to be used in the regression framework. The term 'meaningful predictor' is used to emphasize the object that *is following* or *has followed* a track similar to that of the predictand during time interval $t_c = (t_s, t_e)$ where $0 < t_s < t_e$ and $t_s < t_e < T$. This implies that a section $r_j(t_s^j, t_e^j)$ of trajectory $r_j(t) \in G$ during some time $t_c^j = (t_s^j, t_e^j)$ is similar to the track $r_i(t)$ of the predictand in $F = (t_e^j - t_s^j)$ frames of the sequence, where F is a parameter defining the observation window ($F = 10$ is used for all experiments). Note that superscript j in t_s^j and t_e^j refers to the fact that there is a separate start and end time corresponding to each potential predictor track $r_j(t)$. In order to assemble a set of predictors for a particular predictand, I search all the tracks in G for a section $r_j(t_s^j, t_e^j)$, in which the motion of the object is most similar to that of the predictand in terms of the LCE.

In order to compute the LCE, I define a track section Z of predictand O_i as $Z = r_i(t - F, t)$, where t is the current time. Then, for every possible section $t_c^j = (t_s^j, t_e^j)$ of the potential predictor's track $r_j(t)$, I extract a section $Z_c = r_j(t_s^j, t_e^j)$ (Figure 5.2(b)). Next, I normalize track section Z_c with respect to the predictand's track section Z, so that they both start at the same point. Let us

call the normalized track Z_{cn}. Then, I compute the LCE between the predictand's and the object's track sections Z and Z_{cn} as:

$$\gamma(Z, Z_{cn}) = \sum_{i=2}^{F} ln \frac{dist(Z(i), Z_{cn}(i))}{dist(Z(1), Z_{cn}(1))}, \tag{5.3}$$

where $dist(Z(i), Z_{cn}(i))$ is the Euclidian distance between two track points $(x(i), y(i))$ and $(x_{cn}(i), y_{cn}(i))$ given by $dist = \sqrt{(x(i) - x_{cn}(i))^2 + (y(i) - y_{cn}(i))^2}$. (Figure 5.2(b))

If $\gamma(Z, Z_{cn}) \leq \alpha$, where α (12 for all the experiments) is an empirically derived threshold, I add object O_j to the set of predictors C_i of predictand O_i. I also add t_e^j to a new set B_i, where the cardinalities of B_i and C_i are the same. The closer the value of γ is to 0, the more similar the motion of predictor O_j, between frames t_s^j and t_e^j, is to the motion of the predictand between frames $t - F$ and t. The process of computing the LCE between a reference trajectory and a potential predictor trajectory is summarized in Figure 5.2(b).

Once an object has been added to the predictor set C_i of predictand O_i, I maintain the same predictor in set C_i until the predictor's and predictand's tracks begin to diverge. Therefore, for all subsequent frames, I call the above procedure to ensure that the motion of predictor $O_j \in C_i$, between frames $t_s^j + p$ and $t_e^j + p$, is similar to the motion of the predictand O_i between frames $t - F + p$ and $t + p$, where p is the frame counter. If the tracks of the predictand and the predictor begin to diverge, I remove the predictor from the set. Figure 5.3 shows a result for the reference trajectory shown in Figure 5.3(b); I was able to construct the corresponding predictor set shown in Figure 5.3(c) from the input tracks displayed in Figure 5.3(a).

5.2.3 Modeling Appearance Context

The appearance model of each object O_i is constructed using the contextual knowledge about which objects are in the set P. An example of the set P is shown in the leftmost column of Figure 5.4 where I have three cars in the set. In general, the set P can have any number of cars. The observations (image chips) of O_i are represented by a set $\Phi_i = (\mathbf{x}_n \mid n = 1, ...L)$ (leftmost column in Figure 5.4), where L is the total number of observations. For each Φ_i, I generate a set H_i (center column in Figure 5.4), in which each element $h_n \in H_i$ is a three-dimensional (RGB) color histogram. For each H_i, I define a variable q, which is a function $q = f(h_a, h_b)$ of a pair of observations, $h_a \in H$ and $h_b \in H$. In the scenario described, the function f is the histogram intersection of h_a and h_b. Next, I compute a vector K_i, which contains the values of histogram intersection, q, for all possible pairs of observations in H_i. This way I generate the observations of *intra-class appearance variation* of object O_i. (Figure 5.4)

Similarly, for O_i, observations of *inter-class appearance variation* are computed with respect to each $O_j \in P$. For this purpose, RGB histograms of observations, Φ_i and Φ_j, are employed to compute vector Ω_{ij}. Ω_{ij}, which contains the values of histogram intersection between all possible pairs of histograms in H_i and H_j. (Figure 5.4) The same process is repeated to obtain observations of inter-class variation with respect to the remaining object in the set P.

Assuming the differences in the values of histogram intersections originate from additive Gaussian noises, I construct a Gaussian probability density function for observations of intra-class variation (K_i) and observation of inter-class variation (Ω_{ij}). This is completed by using the standard formulas to compute the means and standard deviations of the observations (Figure 5.4). Finally,

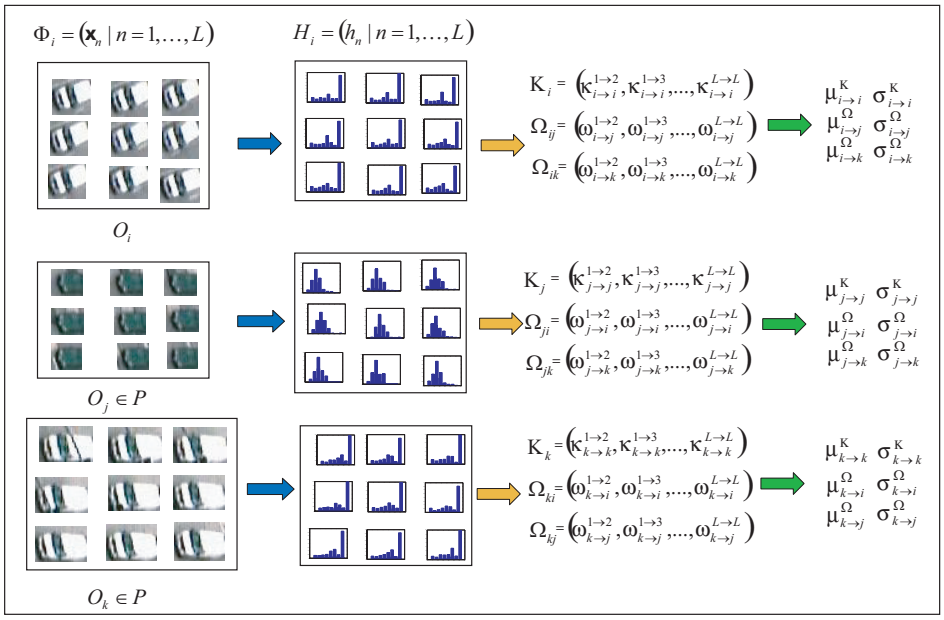

Figure 5.4: Modeling of AC when the set P contains three cars, O_i, O_j and O_k. The first column displays the observations (chips) of each of these cars until current time T. Then, each observation of the car is encoded in terms of an RGB color histogram, as shown in the second column. Vectors of inter- and intra-class variations between these objects are computed by performing histogram intersection. Finally, the mean and standard deviations of the values in these vectors are computed, which summarize the inter- and intra-class variation information, as shown in the fourth column.

each object, $O_i \in P$, is represented by a set of Gaussian PDFs, where one Gaussian PDF encodes the variation in the appearance of the object with respect to itself, and the remaining PDFs encode the variation in the appearance of the object with respect to other objects in the set P.

If a new object O_k is detected and I want to determine whether if it is an unobservable object from the set P, I proceed as follows: observations (image chips) belonging to the object O_k are extracted and an RGB color histogram, s, is computed for every observation. For each $O_j \in P$, I compute a vector of variations by computing histogram intersection between all possible pairs of observations of O_j and O_k. I again model each observation in the vector as a Gaussian PDF by computing its mean and standard deviation. This provides a set of Gaussian PDFs (one for each $O_i \in P$) for the new object. Our goal is to select the object O_i from the set P whose inter- and intra-class variation distributions are closer to the variation distributions of the new object O_k. To do this, for every object, $O_i \in P$, I compute the average Bhattacharya distance between its own variation distributions and variation distributions of O_k. The object O_i is reacquired as O_k if the average Bhattacharya distance for O_k is a minimum of:

$$\zeta(O_i, O_k) = \frac{1}{|P|} \sum_{j=1}^{|P|} \frac{1}{4} \frac{(\mu_i^j - \mu_k^j)^2}{\sigma_i^j + \sigma_k^j} + \frac{1}{2} \log \frac{\left| \frac{\sigma_i^j + \sigma_i^k}{2} \right|}{\sqrt{\left|\sigma_i^j\right| \left|\sigma_i^k\right|}}. \tag{5.4}$$

5.3 Target Re-acquisition

For every new object O_i that appears in the scene, I search the set of predicted objects, P, for objects that satisfy the motion and appearance context constraints. The search is limited to those objects whose predicted tracks lie within a certain radius around the centroid of the new object O_i. The region is defined in terms of a Euclidian distance threshold, η, as shown in Figure 5.5(a). An additional motion constraint, defined in terms of Lyapunov exponent γ, is also applied at the re-acquisition stage. It is used to filter out the trajectories within the search area that do not agree with the dynamical behavior of the predicted trajectories.

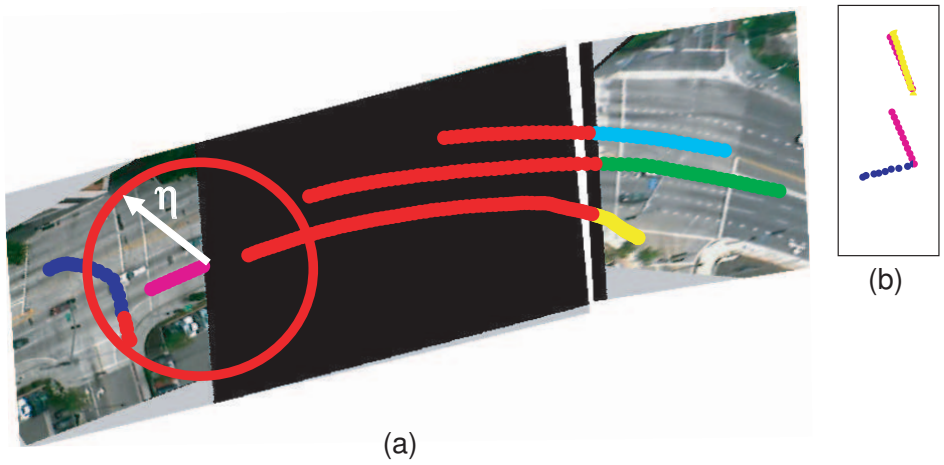

Figure 5.5: Visualization of the re-acquisition procedure. (a) The red circle represents the search area around the new object O_i. The black portion represents the part of the scene through which the trajectories are predicted using the MC (Section 5.2.1). There are four trajectories represented by the colors yellow, green, cyan, and blue. The red portion of these trajectories represents the predicted portion. (b) Computation of Lyapunov exponent at the re-acquisition stage. The trajectory of the new object O_i is represented by the pink track section. To compute the motion constraint, it is normalized with respect to the two predicted tracks that are within the search area.

This motion constraint is computed as follows: Gather t observations of the new object O_i (shown by the pink trajectory in Figure 5.5(a)) and normalize them with respect to predicted trajectories that are within the search area. There are two such trajectories for the example described in Figure 5.5(a), and their corresponding normalizations are shown in Figure 5.5(b). After the normalization, the LCE is computed using Equation 5.3. Trajectories that have an LCE above a

Algorithm 2: Target Re-acquisition

foreach *incoming frame* f **do**
  Get set of objects, V_f, in the current frame;
  foreach *new object* O_k *in* V_f **do**
    Find if O_k is in P by computing distance between current location of O_k and last predicted location of P_i within $(f - u)$ frames;
    Find if motion of O_k is similar to that of P_i by computing γ by using Equation 5.3;
    If $distance < \eta$ AND $\gamma < \alpha$, set $O_k = P_i$;
    Remove P_i from P;
  end
  foreach *old object* O_l *in* V_f **do**
    Update the predictor set C_i by calling *PredictorSelection*$(r_l(0, ..., f), C_i, B_i, G)$
  end
end

threshold, α, are removed from the set of potential target objects. From the remaining trajectories, I reassign that label to the trajectory that lies closest to the new object O_i in terms of Euclidian distance. For example, in Figure 5.5(b), the LCE between the pink and the blue trajectories is above the threshold, therefore the blue trajectory is removed from the set of potential target objects. The yellow trajectory has an LCE that is below the threshold and its last predicted position (i.e., the red portion of the yellow trajectory in the black region) is closer to the starting position of the pink trajectory, therefore, the yellow label will be assigned to the pink trajectory.

Algorithm 3: PredictorSelection

Input: $\mathbf{R}(\mathbf{0}....,\mathbf{f}), \mathbf{C}, \mathbf{B}, \mathbf{G}, \mathbf{O}$, Returns: C

foreach *track* $r_i(0,..,f-1)$ *in* G *except the track* $R(0,...,f-1)$ **do**

    if O_i not member of C then for each time instance $j = F$ to $(f-1)$;

        $Z = R(f-F, f)$, $Z_c = r_i(j-F, j)$;

        $Z_{cn} = normalize(Z, Z_c)$;

        Compute γ by using Equation 5.3;

        if $\gamma < \alpha$ then add O_i to C and add j to B;

    else

        Pick entry corresponding to C_i from B, call it j;

        $Z_c = r_i(j-F+1, j+1)$, $Z = R(f-F, f)$;

        $Z_{cn} = normalize(Z, Z_c)$;

        Compute γ using Equation 5.3;

        if $\gamma > \alpha$, then remove C_i from C and add B_i from B, otherwise $B(i) = B(i) + 1$;

end

However, assigning labels purely on the basis of MC-based prediction cannot handle all possible scenarios. For example, if the behavior of the cars while out of the FOV of the camera is different from the behavior of the cars in the FOV of the camera, MC-based may fail to assign the labels correctly. Then, there is the possibility that the cars may have changed positions while out of the FOV of the camera. Similarly, if the predicted positions of two cars are spatially near each other and if there is error present in the prediction, then once again label reassignment based

purely on the MC based prediction may be incorrect. To overcome these situations, I use AC to aid the re-acquisition. A subset of unobservable objects is selected that is within the search area and that satisfies the motion constraint. However, rather than selecting the object that best satisfies the spatial proximity, I select the object that best matches with the AC of the objects in the set P as described in Section 5.2.3. A pseudo-code of the re-acquisition algorithm is provided in the above table.

5.4 Experiments and Results

In this section, I present systematic evaluations of the proposed re-acquisition algorithm. In Section 5.4.1, I perform qualitative and quantitative evaluations of the algorithm on aerial videos taken from the VIVID data set and GettyImages, which is a stock footage web-site. In Section 5.4.4, the experiments are performed on a multi-camera data set that consists of videos taken from a high-rise building and simulates the setting of aerial videos. I also compare our re-acquisition results with the results obtained using the motion prediction model of [91].

5.4.1 Re-acquisition in Aerial Videos

5.4.1.1 *Data Set*

The first set of aerial video sequences were taken from the DARPA VIVID data corpus. The VIVID sequences contain a convoy of cars that is being followed by the UAV. However, due to the rapid motion of the UAV, a particular car remains visible in the FOV of the camera only for a short duration of time. In the absence of a re-acquisition methodology, the resulting tracks are broken.

The second set of aerial videos is downloaded from Getty-Images, which is a stock footage web site. These sequences contain footage of busy road intersections, in which the average number of visible cars varies from 20 to 30 in each frame. The cars move along multiple paths at any given time. These sequences are challenging not only from the point of view of track completion but also predictor selection. The types of occlusions in these videos include single overhead bridges and multiple overhead bridges.

5.4.1.2 *Detection and Tracking*

The detection and tracking *within* each sequence are done automatically. Detection of all the moving objects is a challenge in these videos due to the small sizes of the objects. For automatic detection and tracking, I used the COCOA system [120], which performs these tasks in two stages. First, the incoming frames are aligned to the reference frame using the direct method for image alignment [130]. In the second stage, an algorithm based on frame differencing and motion history image [131] is used to detect the regions containing moving objects. I observed that fusion of results obtained by using both techniques returns far better results for object detection. I also tried background modeling [134], but results were not as satisfactory.

5.4.2 Qualitative Results

The experiments on aerial videos were conducted in the following manner. The detected objects were tracked, and once they become unobservable, their positions were predicted using the MC algorithm discussed in Section 5.2.1. When a new object enters the FOV of the camera, I match

it with all predicted tracks within the search radius using motion and appearance constraints, as described in Section 5.3. The qualitative results of re-acquisition on two different tracks, each taken from a separate video of the VIVID data set, are shown in Figure 5.6.

Figure 5.6(a) shows a scenario in which the motion of the convoy of cars is linear. The different colors in Figure 5.6(a) represent different labels that were assigned to our target car every time it re-entered into the FOV of the aerial camera. I predicted the motion of the target car through the unobservable region by using its MC, which was composed of the motion of the other cars in the convoy. This allowed us to correctly assign it the same label (i.e., color) when it reappeared, as shown in the Figure 5.6(b). The predicted portion in Figure 5.6(b) is shown in blue color.

Figures 5.6(c)-(d) show re-acquisition results on a rather challenging maneuver. In this sequence, the convoy of cars made a U-turn, and the aerial camera was only able to track part of the convoy. In the absence of the re-acquisition module, the tracking algorithms generated four tracklets for the target car (Figure 5.6(c)). However, our prediction algorithm was able to assign the correct label every time the car reappeared in the FOV of the camera (Figure 5.6(d)). Although in this case, the prediction was still along the linear stretch of the motion, I was able to overcome the *change in the direction of the motion* after the U-turn using the MC. The cars that composed the MC were behaving in a manner similar to the target car and therefore, it was possible for the algorithm to infer the correct dynamics of the motion. I would like to stress here that a prediction model based on linear motion assumption, for example the work by Perera *et al.* [91], will not be able to handle a change in the direction of motion, because it uses the motion information of the target object in isolation. Note that in Figure 5.6(d), the noisiness of the tracks near the matching

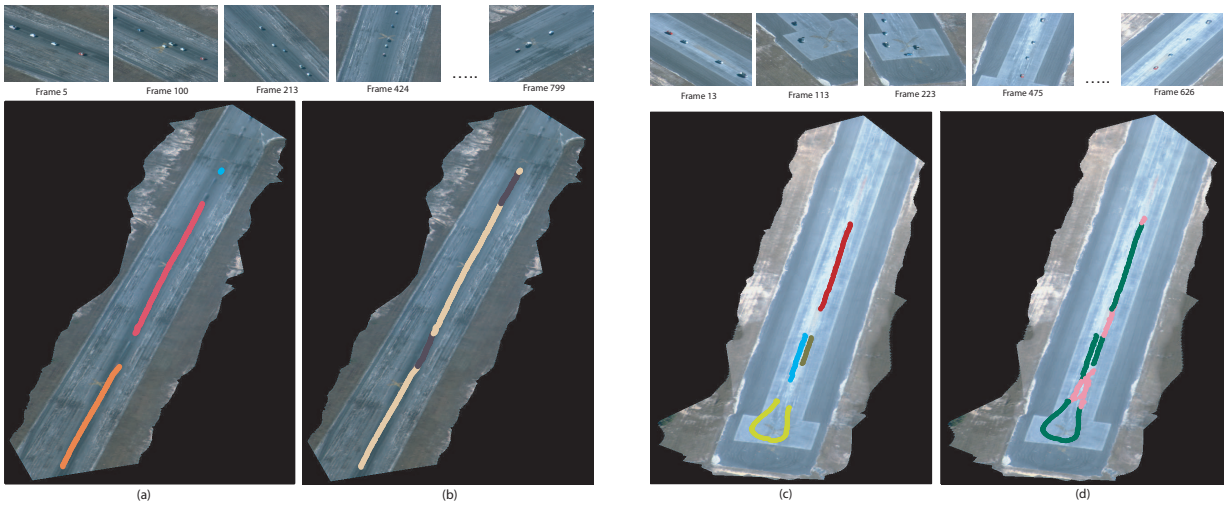

Figure 5.6: Target re-acquisition for linear motion. The tracks of the cars are overlaid onto the mosaic of the aerial sequence for better visualization. The top row shows a number of frames from the video sequence. (a) Track sections belonging to the same car are assigned different colors in the absence of prediction. (b) Our algorithm is able to assign the same color (light brown) to the target car every time it reappears in the FOV of the camera. Note that our target car leaves and re-enters the FOV twice, and I was able to maintain the correct label. (c) Another scenario where the target car performs a U-turn and becomes unobservable at three different time instances during the course of tracking. The four tracklets belong to the same car, but different colors are assigned in the absence of prediction. (d) Our algorithm was able to assign the same color (green) to the car every time it reappears in the FOV. In this scenario, the car moves along a non-linear trajectory, but the predicted portion contains only the linear motion.

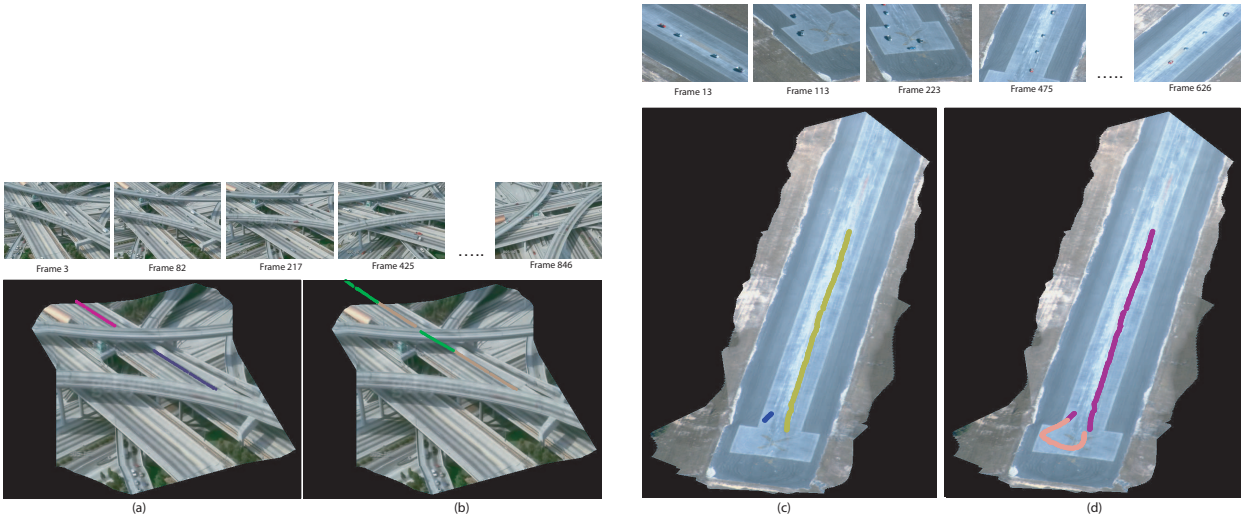

Figure 5.7: A target re-acquisition result where the tracks of the cars are overlaid onto the mosaic of the aerial sequence for better visualization. The top row shows a number of frames from the video sequence. (a) Figure shows re-acquisition on a busy road intersection, where cars are moving along different paths and in different directions. Tracks belonging to the same object were assigned different colors in the absence of prediction. (b) Our algorithm was able to assign the same color to the object as it reappeared on the other side of the overhead bridge. (c) Figure shows re-acquisition for a non-linear motion case. Tracks belonging to the same object were assigned different colors in the absence of prediction. (b) Our algorithm was able to assign the correct label by accurately predicting the motion of the car along the U-turn.

points is due to the error present in the prediction model. A detailed analysis of the error model is presented in the next section. The error in the prediction model can result either in a gap or an over-

lap between the predicted and the matched tracks, depending on whether the model overestimated or underestimated the velocity of the unobserved object.

The re-acquisition result of an aerial sequence from the Getty-Images website is shown in Figures 5.7(a)-(b). This scenario contains a static occlusion in the form of an over-head bridge instead of a dynamic occlusion caused by the motion of the aerial platform. In addition, cars were moving along different paths that made it challenging to learn the valid MC. The re-acquisition in Figure 5.7(b) demonstrates that by learning a valid MC, I can predict the motion through the regions where I do not have any observed data, for example under the bridge. This is possible because the MC also contains the inter-object relationships, for example, how far ahead the unobserved car is with respect to the cars that are part of the MC. The algorithm tries to maintain these relationships during the prediction step and is then able to re-acquire the car when it appears from the other side.

Finally, I show a re-acquisition result for the non-linear motion case, which demonstrates how robust and rich our algorithm is. The results are shown in Figures 5.7(c)-(d), where the prediction is made along the non-linear portion of the track, which was unobserved due to motion of the aerial platform. This result emphasizes the point that our prediction algorithm is model independent and infers the motion directly from the environment using the MC. This is not possible in other algorithms available in the current literature (e.g. [91][94]). In Figure 5.7(c), the two tracklets belong to the same object and are correctly repaired by predicting the motion along the U-turn as shown in Figure 5.7(d).

5.4.3 Quantitative Results

Quantitative analysis was performed by analyzing the prediction accuracy of our context-based motion prediction model. The prediction error of our model was a function of how well the motion of an unobservable object is modeled by the selected predictors. In order to analyze the prediction accuracy, I proceeded as follows: For a given sequence with short and rare periods of occlusion and enough potential predictors, I allowed an object to be observed for F number of frames. After a brief period of observation, I artificially added the object to the list of unobservable objects and predicted its position using our MC based algorithm. The AC was not used in this case. At the end of the sequence, I computed the distance between the predicted position p_i and the actual position t_i of the object for every position of the predicted track of length N. Then, for each sub-interval of length n of the track, I computed the mean prediction error as: $MPE_n = \frac{1}{n}\sum_{i=1}^{n} dist(p_i, t_i)$. The mean prediction error computed for linear motion and non-linear motion is shown in Figure 5.8. It can be observed that the mean prediction error increases with the duration of the prediction. The increase is greater when the object undergoes non-linear motion, as shown in Figure 5.8(e), indicating the difficulty of the regression model to predict the position of the object. However, the error is still sufficiently small enough for the track re-acquisition to work, as demonstrated in Figure 5.7(d). I computed the variance of the error (Figure 5.8(c) and Figure 5.8(f)) for each sub-interval of length n as: $Variance_n = \frac{1}{n}\sum_{i=1}^{n}(MPE_n - dist(p_i, t_i))^2$.

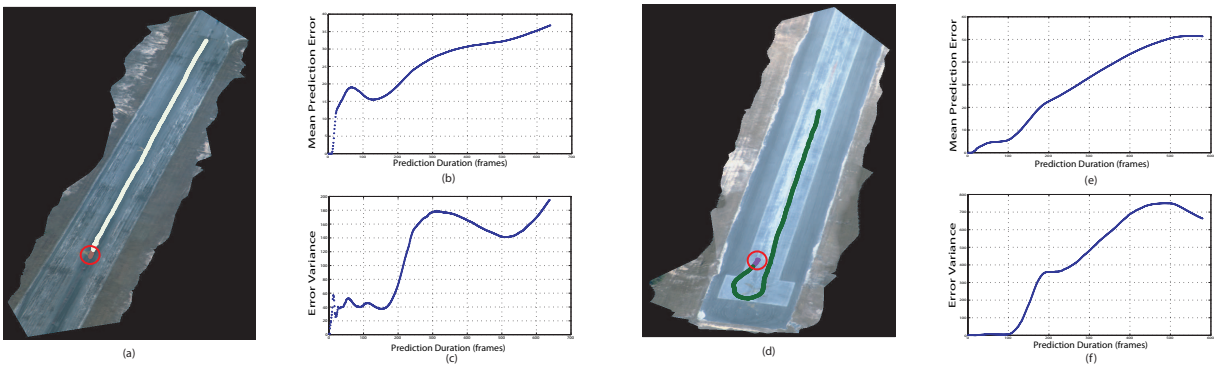

Figure 5.8: Estimation error of the prediction algorithm. (a)-(c) The estimation error in the case of linear motion. (a) The portion circled in red is the observed segment, while the remaining portion of the track is predicted using the algorithm. (b) The mean distance of the predicted track to the actual track. The error increase with the increase in the duration of the prediction. (c) The variance of the distance around the mean distance. (d)-(e) The estimation error in the case of non-linear motion. (d) The track circled in red is the observed portion, while the remaining portion of the track is predicted. (e) The mean distance of the predicted track to the actual track. (f) The variance of the distance around the mean distance.

5.4.4 Re-acquisition in a Multi-Camera Data Set

The second set of experiments was performed on the multi-camera data set generated by the Next Generation Simulation (NGSIM) program [137]. This data set contained video sequences of the traffic flow on a section of Lankershim Boulevard in Los Angeles, California, on June 16, 2005. The data was collected over two 15-minute intervals using five video cameras that were mounted on a 36-story building. The cameras were numbered 1 to 5, and camera 1 recorded the southernmost section, and camera 5 recorded the northernmost section of the study area. The FOV of each

camera had a small overlap with the FOV of the adjacent cameras. One frame from each of these 5 cameras is shown in Figure 5.9. This data set was employed to test our re-acquisition algorithm for two main reasons: first, a camera on a high-rise building accurately simulated the characteristic of a camera mounted on an aerial platform. These characteristics included a wide FOV and a small number of pixels on targets; second, the ground truth was available with the data set that allowed precise quantitative verification of the performance of our algorithm in this real world setting.

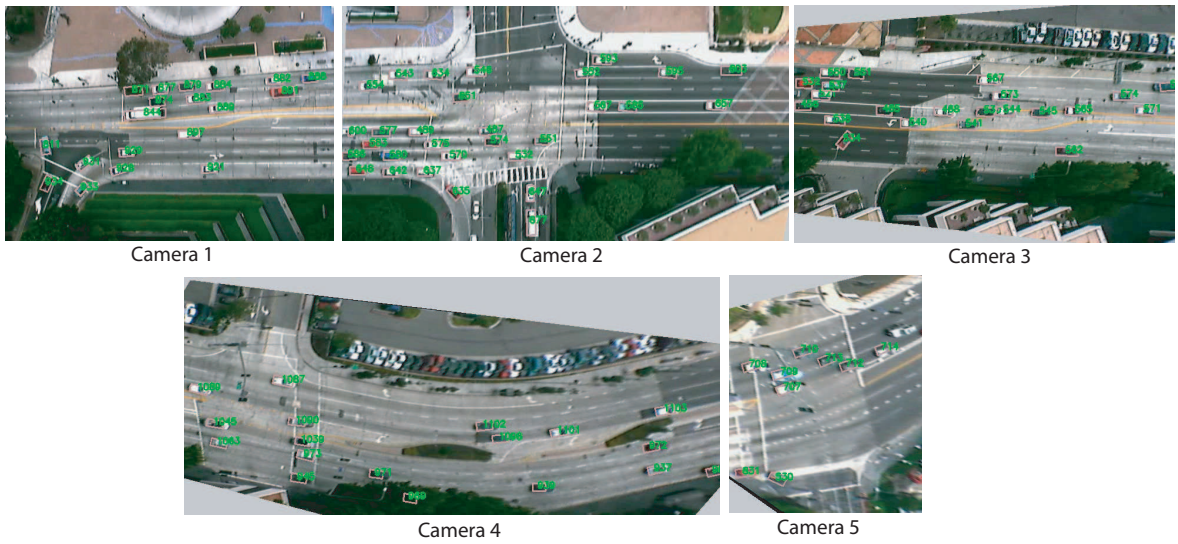

Figure 5.9: One frame from each of the videos recorded by Cameras 1 to 5, mounted on a 36-story building at Lankershim Boulevard. Each vehicle was tracked consistently across the five cameras. The bounding boxes illustrate the labels assigned to each vehicle.

The available ground truth was generated originally by the Next Generation Vehicle Interaction and Detection Environment for Operations software [137]. This program automatically detected and tracked the vehicles and transcribed the trajectory data into a database. All vehicles maintained their labels across different cameras. The data provided (X, Y) coordinates of each vehicle, using

the California State Plane Coordinate System Zone 5 - NAD83, every 0.1 second. For each tracked vehicle, I had 24 pieces of information that included vehicle id, trajectory in state plane coordinate system, vehicle type (1 - motorcycle, 2 - auto, 3 - truck), vehicle width and length, vehicle entry and exit points, camera FOV in which the vehicle was visible at any time during its passage through the scene.

I have performed the following three experiments on this data set: 1) Experiments evaluating the performance of MC. 2) Experiments evaluating the contribution of AC. 3) Experiments evaluating the contribution of a number of initial predictors on the prediction accuracy.

5.4.4.1 *Evaluation of Motion Context*

The goal here was to compare the performance of the MC-based prediction against the linear motion based prediction of [91]. The predictions were performed in the state plane coordinate system (units: feet). I used both 15-minute videos generated by cameras 4 and 5 for the experiment. A synthetic dynamic occlusion was introduced after observing the scene for a short time. The dynamic occlusion was introduced so our setup could accurately simulate a scene observed by an aerial camera. Due to constant motion of an aerial camera, it often observes a particular region of the scene for a short time before moving on to another area. Similarly, I allowed our cameras to observe the scene for a short time before introducing the occlusion. The combined FOV of cameras 4 and 5 is shown in Figure 5.10(a), while the FOV with synthetic occlusion is shown in Figure 5.10(b). Note that the synthetic occlusion spans a portion of both camera 4 and camera 5.

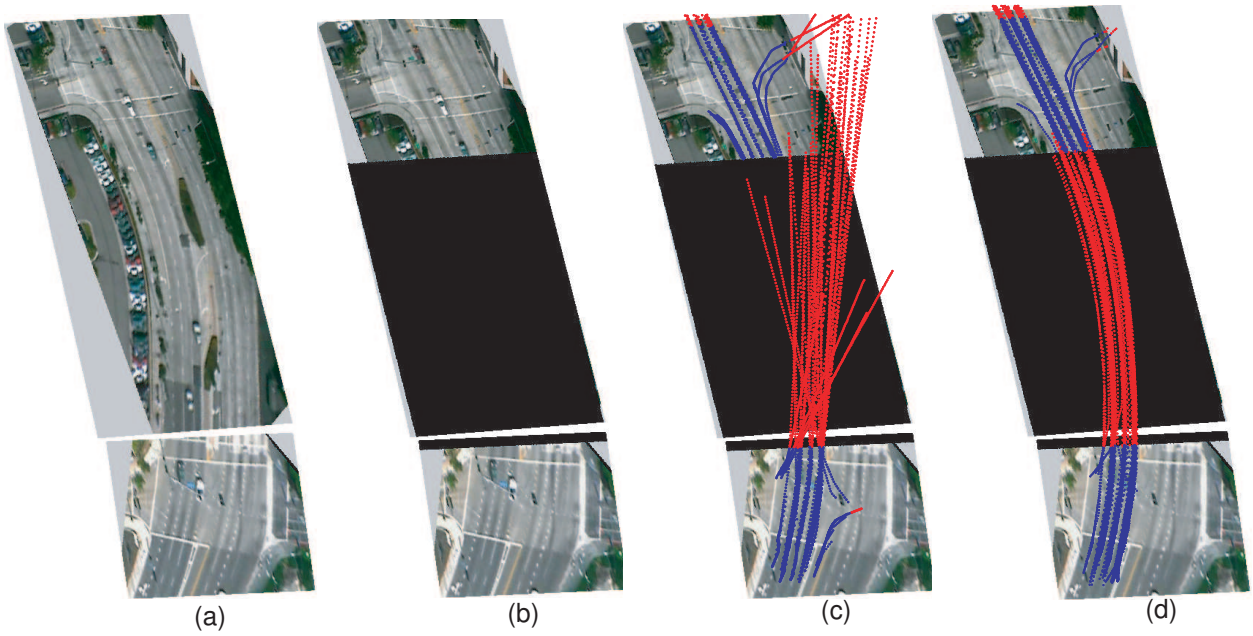

Figure 5.10: (a) The combine FOV of camera 4 and 5 of the NGSIM data set. (b) A synthetic dynamic occlusion was introduced in the combined FOV of camera 4 and 5 to simulate the characteristics of an aerial camera. (c) Plot of predicted tracks without using the MC information. (d) Plot of predicted tracks using the MC information.

The initial observations of the cars to be re-acquired were made in camera 5 (the bottom camera in Figure 5.10(b)) and re-acquisition was performed in camera 4 (the top camera in Figure 5.10(b)). Overall, there were 982 cars that I had to accurately re-acquire using the complete 30-minute data. For each car undergoing occlusion, the MC was computed using the methods described in Section 5.2.1.

The first experiment tested the practicality of MC for re-acquisition purposes against the algorithm that does not use the MC. For this purpose, I first ran the algorithm to re-acquire the cars

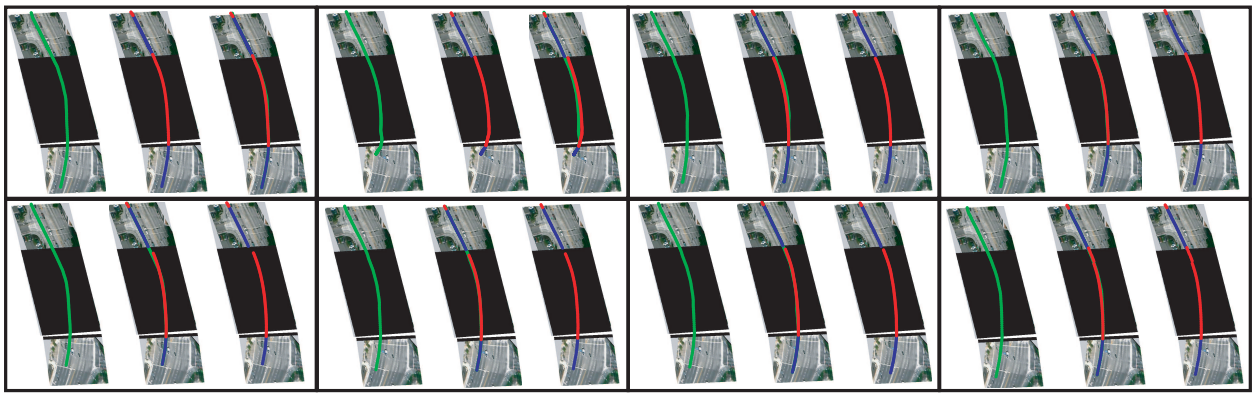

Figure 5.11: Qualitative performance of the MC-based prediction algorithm for *correctly reacquired* tracks on the NGSIM data set. Each block corresponds to one track. Within each block, the leftmost column shows the ground truth track, the center column shows the predicted track, and the rightmost column shows the predicted track superimposed over the ground truth track.

undergoing occlusion using the MC information. Next, I used the linear prediction (LP) algorithm which does not use any MC information and tried to re-acquire the same set of cars. The results of this experiment are summarized in plots of trajectories in Figures 5.10 (c) and (d), which show that our algorithm can accurately predict the potential path of the cars (Figure 5.10(d)) in comparison to the LP algorithm (Figure 5.10(c)). In this scenario the linear motion model-based prediction fails due to the curvilinear nature of the path taken by the cars as they moved through the occlusion. Without information about the type of path, the LP algorithm is unable to adapt to the situation. Conversely, our algorithm made use of the MC to adapt to the situation and made an accurate prediction about the path pursued by the occluded objects. For clarity purposes, I show separate plots of predictions for some of the trajectories in Figure 5.11 where each block belongs to one

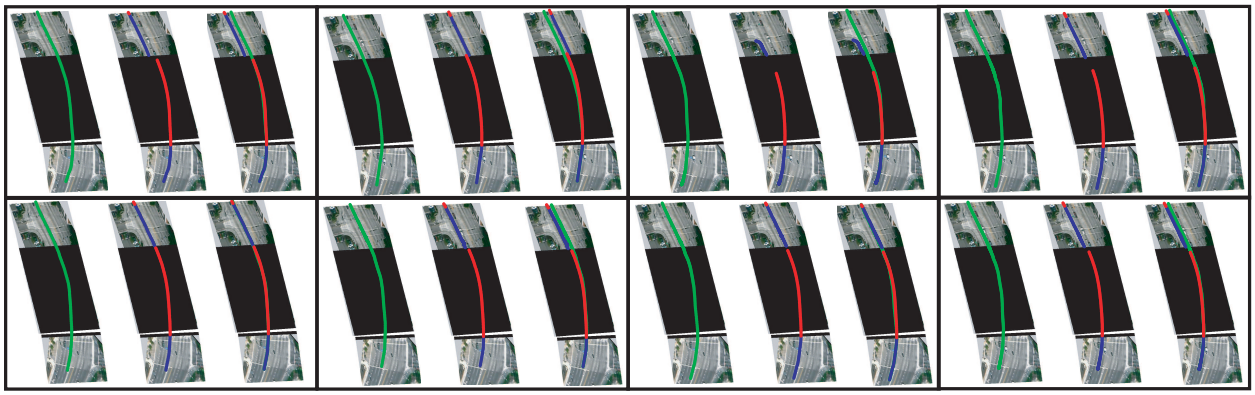

Figure 5.12: Qualitative performance of the MC-based prediction algorithm for *incorrectly reacquired* tracks on the NGSIM data set. Each block corresponds to one track. Within each block, the leftmost column shows the ground truth track, the center column shows the predicted track, and the rightmost column shows the predicted track superimposed over the ground truth track.

trajectory. There are three plots (Leftmost: ground-truth, Center: Predicted using MC, Rightmost: Overlay of predicted track on the ground-truth) for each trajectory in the block. The proximity of the predicted tracks to the ground-truth demonstrates the fact that MC was helpful in re-acquiring the target. To further emphasize the efficacy of the MC-based prediction, I show plots of the trajectories that were re-acquired incorrectly in Figure 5.12. It is important to note in Figure 5.12 that even in the case of incorrect re-acquisition, the error committed by our algorithm is small. However, in the case of LP, the error of incorrect re-acquisition is far larger as shown in Figure 5.10(c).

Further *quantitative* analysis was performed by comparing the re-acquisition rate of the MC-based prediction against the LP over the entire data set. The superior performance of our algorithm was validated, as shown in the graph in Figure 5.13(a). On the x-axis of this graph, I show the radius of the neighborhood in which I searched for the object when it reappeared from occlusion. On the y-axis, I show the re-acquisition rate. The green and red curves show the performance of the MC-based and the linear motion prediction algorithms, respectively. The performance of our algorithm improves with an increase in the search radius, but decreases after a certain value (60 feet) until it becomes parallel to the performance of the linear motion prediction algorithm. This is because an increase in the search radius forced us to perform the comparison against more cars that are in the neighborhood, thus raising the odds of making a mistake. In other words, the increase in the search radius had the effect of a brute force search for the correct car in that region, which effectively means the MC information is not utilized. As the search region started encompassing the location predicted by the linear motion model, the performance of the two algorithms becomes the same. This graph, therefore, shows that utilizing the MC information assists in accurately constraining the probable locations of the unobserved cars resulting in higher re-acquisition rates.

Furthermore, I analyzed the relationship between prediction error and prediction duration of the two models (MC and LP). The prediction error of our model was a function of how accurately the motion of an unobservable object was modeled by the selected predictors. In order to analyze the prediction accuracy I proceed in a similar manner as described in Sub-Section 5.4.3. The plots of mean prediction errors computed for a number of correctly acquired tracks are shown in Figure 5.14(a). In these plots, the x-axis contains the frame numbers, while the y-axis contains the error

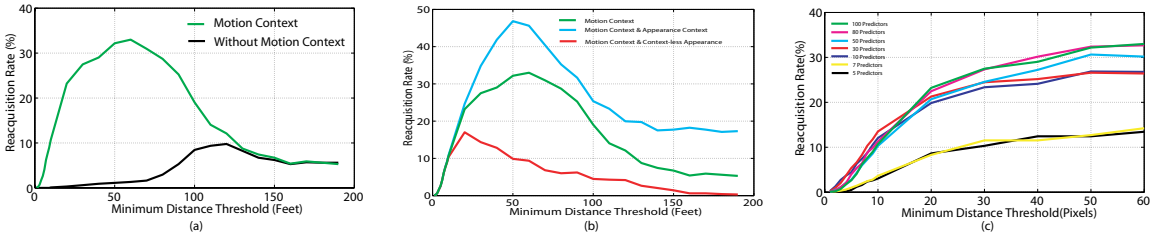

Figure 5.13: (a) Plot of the re-acquisition rate against the re-acquisition distance threshold with (Green) and without (Black) MC. (b) Plot of the re-acquisition accuracy with respect to the re-acquisition distance threshold when both AC and MC were used (Cyan), when only MC was used (Green), and when MC and context-less appearance model was used (red). (c) Figure shows the re-acquisition accuracy with respect to the re-acquisition distance threshold when MC was used with different number of prior observations. 5 (Black), 7 (Yellow), 10 (Blue), 30 (Red), 50 (Cyan), 80 (Magenta), 100 (Green).

in terms of pixels. The mean prediction error increases with the duration of the prediction for both models. However, the increase is greater for LP (black curves) than for MC-based prediction (green curves), which highlights the much higher reliability of our algorithm. I have also plotted the mean prediction errors for a number of tracks that were *incorrectly re-acquired* in Figure 5.14(b). Again, it is important to note that even in cases where our algorithm incorrectly reacquired an object, the error was far smaller than the error by the LP algorithm. Table 5.1 and Table 5.2 summarize the predication error over the entire data set (982 cars) for correctly and incorrectly re-acquired cars, respectively. For correctly re-acquired cars, our algorithm made an average error of 7.83 feet after making prediction for 80 frames, as compared to an average error of 28.53 feet by the linear motion

prediction. Similarly, for incorrectly reacquired cars, our algorithm made an average error of 20.86 feet after making prediction for 80 frames, as compared to an average error of 31.76 pixels by the linear motion predication.

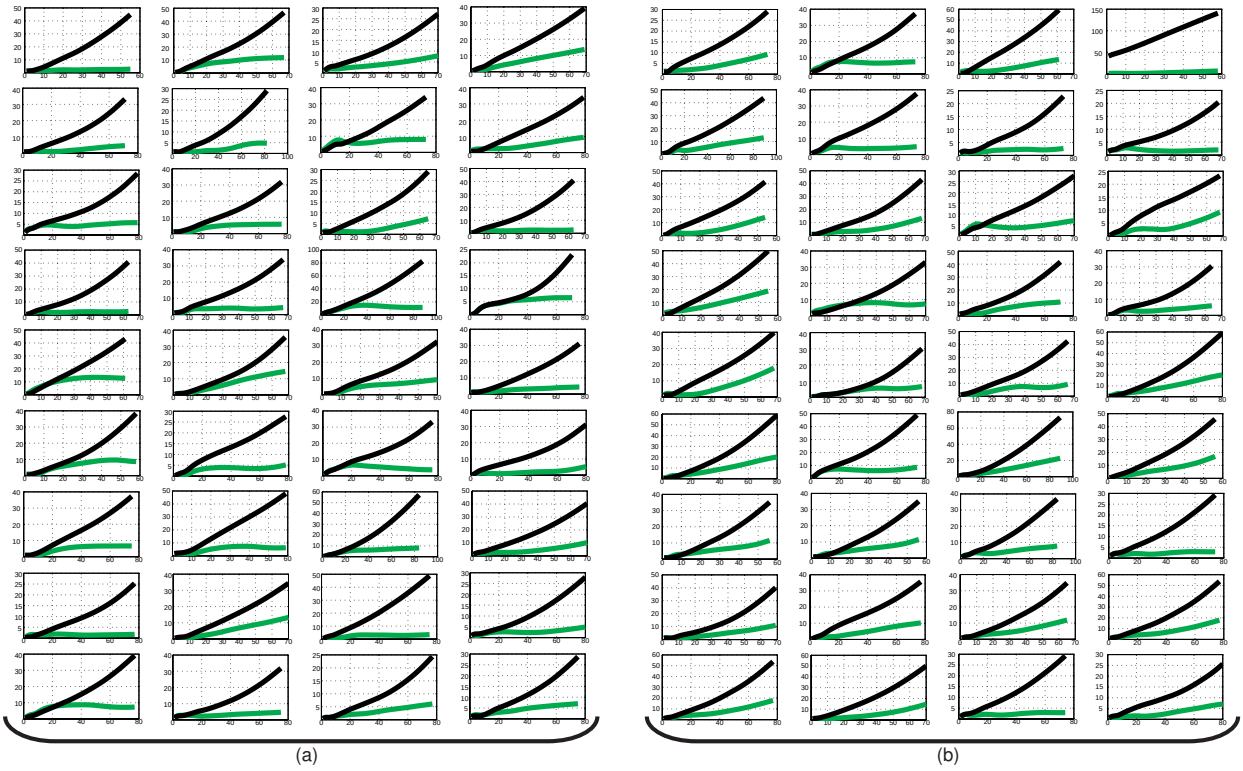

Figure 5.14: (a) Mean distance error between the correctly re-acquired tracks and the ground-truth for cases where the MC was used (Green), and where the MC was not used (Black). (b) Mean distance error between the incorrectly reacquired tracks and the ground-truth for cases where the MC was used (Green) and where the MC was not used (Black).

Table 5.1: Summary of prediction error of two models (MC and LP) over the entire data set for correctly reacquired cars. The columns show the average error committed by the respective algorithm at the end of 20, 40, 60, and 80 frames. The error is in feet.

Prediction Model	**Avg. Error**-(20)	**Avg. Error**-(40)	**Avg. Error**-(60)	**Avg. Error**-(80)
LP	7.51	14.66	23.4	28.53
MC	5.54	8.21	10.23	7.83

Table 5.2: Summary of prediction error of two models (MC and LP) over the entire data set for incorrectly reacquired cars. The columns show the average error committed by the respective algorithm at the end of 20, 40, 60, and 80 frames. The error is in feet.

Prediction Model	**Avg. Error**-(20)	**Avg. Error**-(40)	**Avg. Error**-(60)	**Avg. Error**-(80)
LP	7.03	13.98	22.49	31.76
MC	5.54	9.57	13.74	20.86

5.4.4.2 *Evaluation of Appearance Context*

Next, I performed an experiment to determine the contribution of the AC to the re-acquisition rates. In the NGSIM data set, additional work is required to obtain observations (chips) for every vehicle. The position of every vehicle in the ground-truth was in the state plane coordinate system. In the ortho-rectified views of the scene, the appearance of the car was corrupted by the visualization of the bounding box and its associated vehicle id (Figure 5.9). Therefore, the observation of every vehicle must be acquired from the raw video (without the bounding box visualization) of the scene. For this purpose, I proceeded as follows: To obtain an observation for a particular

vehicle, I first warped its state-plane position to the ortho-rectified view of the scene, and then warp its ortho-rectified position to the raw view. In order to compute homography between the state-plane and the ortho-rectified view, I used correspondences between the state-plane ground-truth positions of several vehicles and their corresponding positions in the ortho-rectified view. Next, prominent image features, such as dividing lane intersections, were used to manually compute the homography between the ortho-rectified and the raw views of the scene. The chips of the cars that passed through the study area during the first 15 minutes are shown in Figure 5.15. Figure 5.15(a) shows the chips of the cars just before they enter into the occluded region, while Figure 5.15(b) shows the chips of the cars when they reappear from the occluded region.

Now, when a car undergoes occlusion, I constructed its AC using all its previous observations, employing the method described in Section 5.2.3. During occlusion, the position of the car was predicted using the MC-based prediction. For re-acquisition, both MC and AC were used, meaning that when I searched for a car at the other end of the occlusion, I confirmed that it satisfied both the motion- and appearance-based constraints as described in Section 5.3. The process was repeated for all the cars in the data set. The results of this experiment are summarized in Figure 5.13(b), where I plotted the re-acquisition rates as a function of the search radius. The blue curve shows the re-acquisition rates when both the MC and the AC were used. The green curve shows the re-acquisition rates when only the MC was used. This outcome demonstrates that the AC is helping to improve the re-acquisition rate. On closer analysis, it was observed that the AC is particularly helpful in cases where two or more cars simultaneously entered the occluded region next to each

Figure 5.15: (a) Chips of the cars just before they enter into the occluded region during the first 15 minutes of the video. There are 386 cars in total. (b) Chips of the cars as they reappear from the occluded region during the first 15 minutes of the video.

other. In this case, when the cars reappeared at the other end, the AC helped the algorithm to overcome the uncertainty present in the MC-based prediction.

Next, I performed an experiment to test the utility of using *context* in the appearance model. For this purpose, when a car undergoes occlusion, I constructed its context-less appearance model

(an RGB histogram) using its most recent observations. No inter- and intra-class variations are employed in this case, hence there was no contextual information present in the appearance model. During occlusion, the position of the car was predicted using the MC-based prediction. For re-acquisition, both MC and context-less appearance model were used, meaning that when I searched for a car at the other end of the occlusion, I confirmed that it satisfied the motion-based constraint and its appearance also matched with the context-less appearance model. The process was repeated for all the cars in the data set. The results of this experiment are summarized by the red curve in Figure 5.13(b). It is evident that the context-less appearance model degraded the performance of the entire prediction framework. The main reason is that without the context information, the appearance model was not rich enough to discriminate between the cars as they reappeared after the occlusion. This clearly demonstrates the benefit of using the context information when building the appearance model.

It is important to note that our main objective in above mentioned experiments was to show the utility of context information in building discriminative representations, and not to show that the color features are the best features. Within our AC framework, other sophisticated features which are proposed in the literature ([108][110]) can be integrated to achieve even better performance.

5.4.4.3 *Number of Predictors*

The last experiment was conducted to test the influence of the prior observation on the accuracy of the MC-based prediction. The prior observation corresponds to the number of tracks that were observed before the introduction of the dynamic occlusion. This was the initial number of tracks

that were part of the set C_i for object O_i. The remaining experimental setup was the same as that described in Section 5.4.4.1. I plotted the re-acquisition rates for different numbers of prior observations in Figure 5.13(c). The re-acquisition performance improved with an increase in the number of prior observations. This was because the regression framework, which was described in Section 5.2.1, became over-constrained with the increase in the number of predictors, and, therefore, assisted in improving the estimation of the unknown affine parameters. There was little change in the re-acquisition rates beyond 20 predictors, which appears to be an optimal choice.

5.4.5 Re-acquisition in a Crowd Video

The next set of experiments was performed on a video containing high density crowds. Again occlusion was simulated by introducing a synthetic dynamics occlusion. The qualitative results of this experiment are summarized in plots of trajectories in Figure 5.16, which shows that our algorithm can accurately predict the potential path of the individuals in the crowded scene. The predication was performed along the curved sections of the tracks.

5.5 Summary

In this chapter, a method was presented to re-acquire objects in moving aerial cameras. A novel concept of motion context (MC) was used to predict the position of target objects during the period that they are occluded. The MC consisted of a collection of trajectories that were representative of the motion of the occluded or unobserved targets. The MC was learned using a clustering scheme based on the Lyapunov Characteristic Exponent. The locations were predicted using the MC in a

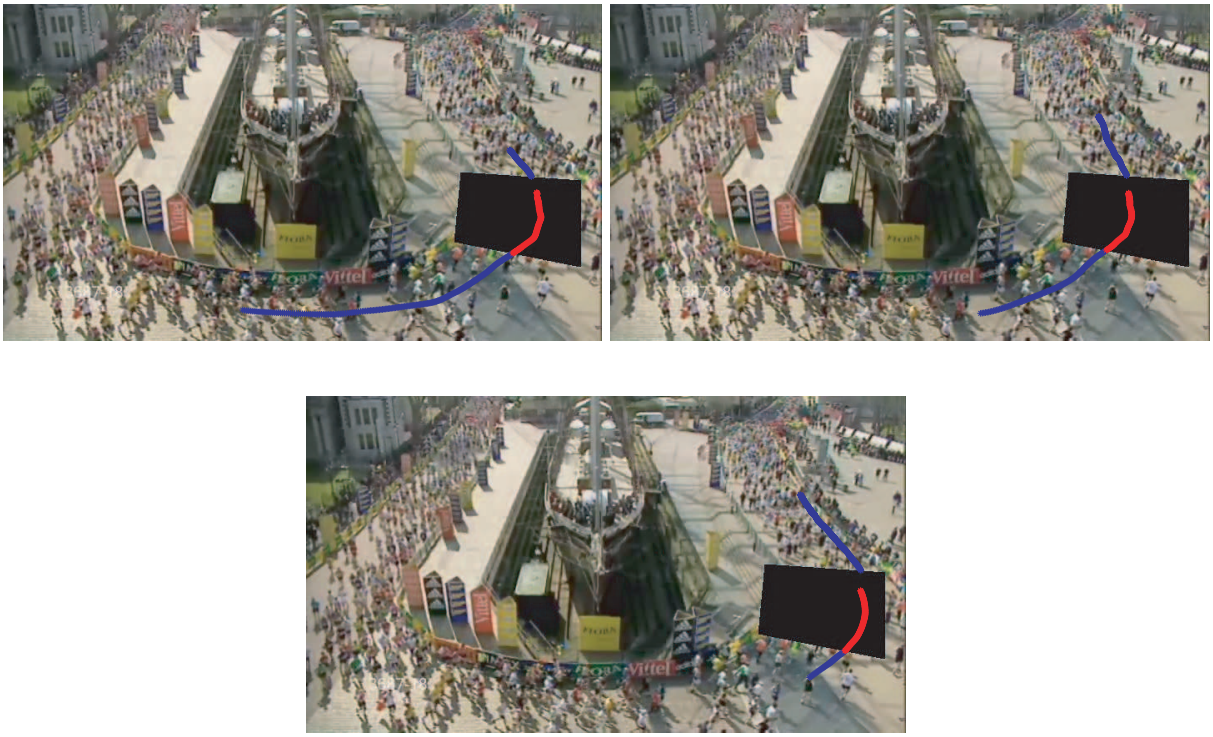

Figure 5.16: Qualitative results of re-acquisition in a crowded scenes. The black portion represents the synthetic occlusion. The red trajectories are the predicted parts of the blue trajectories.

regression framework. In addition, appearance context (AC) was used to differentiate targets when there was uncertainty is the MC-based prediction. The AC was encoded in terms of inter- and intra-class variations computed by using all previous observations of target objects.

The competitive performance of the proposed algorithm was demonstrated on challenging data sets, which included aerial videos and videos taken from a high-rise building. I compared the performance of our algorithm against the prediction model currently used in the literature ([91]) for re-acquisition in aerial sequences. The main advantages of our approach is the model-free

prediction and its ability to adapt to the motions present in the video by learning the MC. Therefore, it does not assume a fixed motion model, e.g., linear, quadratic or polynomial.

In the next chapter, I summarize the main contribution of this thesis and also discuss some possible future research directions.

CHAPTER 6
SUMMARY AND FUTURE WORK

The main theme of this dissertation has been the analysis of videos depicting high-density crowds. Typical examples of such scenes include sporting events, religious festivals, malls and subway stations. To that end, I investigated a global-level approach to generate a representation of the scene that captures both the dynamics of the crowd and the structure of the scene. This global-level analysis eliminated the need for low-level change detection algorithms and individual object localization/tracking.

In particular, this was achieved by developing a crowd-flow segmentation framework, which employed Lagrangian particle dynamics to uncover the spatial organization of the crowd. This segmentation information was then used to detect any temporal changes in crowd behavior, thus enabling the localization of abnormal events or behaviors within the crowd. Next, crowd segmentation information was used in conjunction with the scene structure to develop a tracking algorithm that was used to track an individual object of interest within the crowd. For this purpose, the structure of the scene was encoded in terms of "floor fields" that were used to constrain likely locations an object might pursue while moving in the scene. Finally, I proposed a target reacquisition algorithm that was employed to reduce the incidence of broken trajectories resulting from frequent occlusions in crowded scenes. The proposed re-acquisition algorithm used the contextual information in the form of appearance and motion context of the target object.

We summarize the primary contributions of this thesis in following section.

6.1 Summary of Contributions

- Crowd-Flow Segmentation
  - Representation of crowd motion in terms of Lagrangian Particle Dynamics.
  - Introduction of the concept of Lagrangian Coherent Structure for the analysis of crowded scenes.
  - Representation of crowd dynamics in terms of FTLE fields.
  - Application of the crowd-flow segmentation framework to the problem of abnormal event detection in crowds.
- Target Tracking in High Density Crowds
  - Algorithm for tracking individuals in high density crowded scenes containing large number of people.
  - Integration of high-level knowledge represented by floor fields into the tracking algorithm.
  - Introduction of the novel concept of floor fields to the vision community.
- Motion and Appearance Contexts for Re-Acquiring Targets
  - An algorithm for the persistent tracking of objects in complex scenes.
  - A regression-based framework to predict the locations of occluded objects.

– Introduction of the novel concept of motion and appearance context.

6.2 Future Directions

The methods developed in this work can be improved and extended along a number of lines. Some of these ideas are described in the following text.

6.2.1 Detailed Crowd Behavior Analysis

A more refined analysis of crowd behavior can be performed by labeling different crowd segments with one of several commonly observed crowd behaviors. The common types of crowd behaviors that can be detected include: lane formation, bottlenecks, intersections, freezing by heating, clogging and the faster-is-slower effect. This can be achieved by using as a basic building block the trajectories of particles belonging to the detected crowd-flow segments. By specifically labeling regions of crowd-flow segments with these behaviors, one will be able to generate a representation of the crowded scene that is easier for human operators to understand and interpret.

6.2.2 Directly Approximating the Crowd Groupings

Rather than attempting to extract Lagrangian Coherent Structures that are invariant manifolds and relate to boundaries between different crowd regions, one can also think of directly approximating the crowd grouping. In this context dynamically distinct regions can be identified by using the notion of *almost invariant sets*. These sets are the regions in the phase space that are almost invariant in the sense that, with a high probability, a trajectory starting in a particular set will

stay in this set for an extended period of time. This parallel paradigm is analogous to directly approximating the pixels of an image segment rather than the boundaries of the segment.

6.2.3 Multi-Modality

The results of the segmentation are reported on videos where only a single segmentation map can explain the dynamics of the underlying crowd. However, in many real world situations, the same spatial location may support multi-modal crowd dynamics. For example, one is confronted at intersections with various alternating and collective patterns of motion, which are often very short-lived and unstable. In order to manage scenes with these types of multi-modalities, a number of segmentation maps must be generated, each explaining the different modality.

6.2.4 Multi-Target Tracking in High Density Crowds

The individual target-tracking algorithm proposed in this thesis can be extended to multi-target tracking in high-density crowds. For that purpose, the multi-target configuration can be treated as a multi-particle system. This will add an interaction term in the tracking framework that will further constrain the likely locations taken by the individual targets.

LIST OF REFERENCES

[1] G. Haller, *Finding Finite-Time Invariant Manifolds in Two Dimensional Velocity Data,* Chaos, Vol. 10, No. 1, pp. 99-108, 2000.

[2] F. Lekien and N. Leonard, *Dynamically Consistent Lagrangian Coherent Structures,* American Inst. of Physics: 8th Experimental Chaos Conference, Vol. 742, p. 132-139, 2004.

[3] A. C. Poje and G. Haller, *The Geometry and Statistics of Mixing in Aperiodic Flows*, Physics of Fluids, Vol. 11, No. 10, 1999.

[4] G. Haller, *Lagrangian Strcutures and the Rate of Strain in Partition of Two Dimensional Turbulence*, Physics of Fluids, Vol. 13, No. 11, 2001.

[5] G. Haller, *Distinguished Material Surfaces and Coherent Strcutures in Three Dimensional Fluid Flows*, Physics of Fluids, Vol. 149, Issue 4, p. 248-277, 2000.

[6] G. Haller and G. Yuan, *Lagrangian Coherent Structures and Mixing in Two Dimensional Turbulence*, Physica D, Vol. 147, Issure 3-4, pp. 352-370, 2000.

[7] G. Lapeyre et. al, *Characterization of Finite-time Lyapunov Exponents and Vectors in Two-Dimensional Turbulence*, Chaos, Vol. 12, No. 3, p. 688-698, 2002.

[8] S. C. Shadden, F. Lekien and, J. E. Marsden, *Definition and Properties of Lagrangian Coherent Structures from Finite Time Lyapunov Exponents in Two Dimensional Aperiodic Flows,* Physica D, 212, Issue 3-4, p. 271-304, 2005.

[9] A. Chrisohoides and F. Sotiropoulos, *Experimental Visualization of Lagrangian Coherent Structures in Aperiodic Flows*, Physics of Fluids, Vol. 15, Issue 3, p. 25-28, 2003.

[10] N. Malhotra and S. Wiggins, *Geometric Structures, Lobe Dynamics and Lagrangian Transport in Flows with a Aperiodic Time Dependence, with Applications to Rossby Wave Flow*, Journal of Non-Linear Science, Vol. 8, p. 401-456, 1998.

[11] F. Lekien and J.E. Marsden, *Tricubic Interpolation in Three Dimensions,* Journal of Numerical Methods and Engineering, Vol. 63, No. 3, p. 455-471, 2005.

[12] T. Zhao and R. Nevatia, *Bayesian Human Segmentation in Crowded Situations,* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2003.

[13] B. Wu and R. Nevatia, *Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors*, IEEE International Conference on Computer Vision (ICCV), 2005.

[14] P. Tu and J. Rittscher, *Crowd Segmentation Through Emergent Labeling*, In ECCV Workshop SMVP, 2004.

[15] G. Brostow and R. Cipolla, *Unsupervised Bayesian Detetcion of Independent Motion in Crowds*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

[16] X. Huang, L. Li, and T. Si, *Stereo-based Human Head Detection from Crowd Scenes*, International Conference on Image Processing (ICIP), 2004.

[17] D. Faulhaber, H. Niemann, and P. Weierich, *Detection of Crowds of People by use of Wavelet Features and Parameter Free Statistical Models*, IAPR Workshop on Machine Vision Applications, 1998.

[18] L. Dong, V. Parameswaran, V. Ramesh, and I. Zoghlami, *Fast Crowd Segmentation Using Shape Indexing*, IEEE International Conference on Computer Vision (ICCV), 2007.

[19] A. Fod, A. Howard, and M. J. Mataric, *A Laser-based People Tracker*, IEEE International Conference on Robotics and Automation (ICRA), 2002.

[20] H. Zhao and R. Shibassaki, *A Novel System for Tracking Pedestrians Using Multiple Single Row Laser Range Scanners*, IEEE Transactions Systems, Man and Cybernetics, Vol. 35, No. 2, 2005.

[21] J. Cui, H. Zha, H. Zhao and R. Shibassaki, *Laser based Detection and Tracking of Multiple People in Crowds*, Computer Vision and Image Understanding (CVIU), Vol. 106 , Issue 2-3, 2007.

[22] P. Reisman, *Crowd Detection in Video Sequences*, IEEE Intelligent Vehicles Symposium, 2004.

[23] A. B. Chan and N. Vasconcelos, *Mixtures of Dynamic Textures*, IEEE International Conference on Computer Vision (ICCV), 2005.

[24] A. B. Chan and N. Vasconcelos, *Modeling, Clustering, and Segmenting Video with Mixtures of Dynamic Textures*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Vol. 30, No. 5, 2008.

[25] A. C. Davies, J. H. Yin, and S. A. Velastin, *Crowd Monitoring Using Image Processing*, Electronics and Communication Engineering Journal, 1995.

[26] T. Coianiz, M. Boninsegna, and B. Caprile, *A Fuzzy Classifier for Visual Crowding Estimates*, International Conference on Neural Networks, 1996.

[27] S. Y. Cho, T. W. S. Chow, and C. T. Leung, *A Neural-based Crowd Estimation by Hybrid Global Learning Algorithm*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 29, No. 4, 1999.

[28] A. J. Schofield, P. A. Mehta, T. John Stonham, *A System for Counting People in Video Images Using Neural Networks to Identify the Background Scene*, Pattern Recognition, Vol. 29, No. 8, 1996.

[29] F. Cravino, M. Dellucca, and A. Tesei, *DEKF System for Crowding Estimation by a Multiple-Model Approach*, Electronics Letters, Vol. 30, No. 5, 1994.

[30] A. Marana, L. da Costa, R. Lotufo, and S. Velastin, *On the Efficacy of Texture Analysis for Crowd Monitoring*, SIBGRAPHI 98: Proceedings of the International Symposium on Computer Graphics, 1998.

[31] A. Marana, L. da Costa, R. Lotufo, and S. Velastin, *Estimating Crowd Density with Minkowski Fractal Dimension Marana*, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 1999.

[32] Sheng-Fuu Lin, Jaw-Yeh Chen, and Hung-Xin Chao, *Estimation of Number of People in Crowded Scenes using Perspective Transformation*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 31, No. 6, 2001.

[33] L. Xiaohua, S. Lansun, and L. Huanqin, *Crowd Density Based on Wavelet and Support Vector Machine*, Transactions of the Institute of Measurement and Control, Vol. 28, Issue 3, 2006.

[34] V. Rabaud and S. Belongie, *Counting Crowded Moving Objects*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

[35] G. Antonini and J. P. Thiran, *Counting Pedestrians in Video Sequences Using Trajectory Clustering*, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 16, No. 8, 2006.

[36] D. Kong, D. Gray, and H. Tao, *Counting Pedestrians in Crowds Using Viewpoint Invariant Training*, Bristish Machine Vision Conference (BMVC), 2005.

[37] D. Kong, D. Gray, and H. Tao, *A Viewpoint Invariant Approach for Crowd Counting*, International Conference on Pattern Recognition (ICPR), 2006.

[38] N. Paragons and V. Ramesh, *A MRF-based Approach for Real-Time Subway Monitoring*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2001.

[39] D. B. Yang, H. H. Gonzalez-Banos, and L. J. Guibas, *Counting People in Crowds with a Real-time Network of Simple Image Sensors*, IEEE International Conference on Computer Vision (ICCV), 2003.

[40] A. Albiol, I. Mora, and V. Naranjo, *Real Time High Density People Counter Using Morphological Tools*, IEEE Transactions on Intelligent Transportation Systems, Vol. 2, No. 4, 2001.

[41] X. Zhang and G. Sexton, *AutomaticHuman Head Location for Pedestrian Counting*, IEE 6th International Conference on Image Processing and its Applications, 1997.

[42] K. Terada, D. Yoshida, S. Oe, and J. Yamaguchi, *A Method of Counting the Passing People by Using Stereo Images*, IEEE International Conference Image Processing (ICIP), 1999.

[43] J. W. Kim, K. S. Choi, B. D. Choi, and S. J. Ko, *Real-time Vision-based People Counting System for the Security Door*, In Proceedings of International Technical Conference On Circuits, Systems, Computers and Communications, 2002.

[44] T. H. Chen, T. Y. Chen, and Z. X. Chen, *An Intelligent People Flow Counting Method for Passing through a Gate*, In Proceedings of IEEE Conference on Robotics, Automation and Mechatronics, 2006.

[45] S. Harasse, L. Bonnaud, and M. Desvignes, *People Counting in Transport Vehicles*, In Transactions on Engineering, Computing and Technology, 2005.

[46] M. Bozzoli and L. Cinque, *A Statistical Method for People Counting in Crowded Environments*, International Conference on Image Analysis and Processing (ICIAP), 2007.

[47] A. Yilmaz, O. Javed, and M. Shah, *Object Tracking: A Survey*, ACM Journal of Computing Surveys, Vol. 38, No. 4, 2006.

[48] Z.Khan, T. R. Balch, and F. Dellaert, *An MCMC-based Particle Filter for Tracking Multiple Interacting Targets*, European Conference on Computer Vision (ECCV), 2004.

[49] Y.Cai, N. de Freitas, and J. Little, *Robust Visual Tracking of Multiple Targets*, European Conference on Computer Vision (ECCV), 2006.

[50] G. Gennari and G. D. Hager, *Probabilistic Data Association Methods in Visual Tracking of Groups*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2004.

[51] W. Lin and Y. Liu, *Tracking Dynamic Near-regular Textures under Occlusion and Rapid Movements*, European Conference on Computer Vision (ECCV), 2006.

[52] W. Lin and Y. Liu, *A Lattice-based MRF Model for Dynamic Near-regular Texture Tracking*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Vol. 29, No. 5, 2007.

[53] M. Betk, D. E. Hirsh, A. Bagchi, N. I. Hristov, N. C. Makris, and Thomas H. Kunz, *Tracking Large Variable Numbers of Objects in Clutter*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

[54] K. Li and T. Kanade, *Cell Population Tracking and Lineage Construction Using Multiple-Model Dynamics Filters and Spatiotemporal Optimization*, International Workshop on Microscopic Image Analysis with Applications in Biology, 2007.

[55] M. Yang, J. Yuan, and Y. Wu, *Spatial Selection for Attentional Visual Tracking*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

[56] 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 (IJCV), Vol. 69, No. 2, 2006.

[57] M. Ben-Akiva and M. Bierlaire, *Discrete Choice Methods and Their Applications to Short-term Travel Decisions*, In Handbook of Transportation Science, pp.534, R.Hall(ed.), Kluwer, 1999.

[58] S. M. Khan, M. Shah, *A Multi-View Approach to Tracking People in Crowded Scenes Using a Planar Homography Constraint*, European Conference on Computer Vision (ECCV), 2006.

[59] A. Mittal and L. S. Davis, *M2Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene*, International Journal of Computer Vision (IJCV), Vol. 51, No. 3, 2002.

[60] I. Haritaoglu, D. Harwood, and L. S. Davis, *Who, When, Where, What: A Real Time System for Detecting and Tracking People*, In Proceedings of the Third Face and Gesture Recognition Conference, p. 222227, 1998.

[61] I. K. Sethi and R. Jain, *Finding Trajectories of Feature Points in a Monocular Image Sequence*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Vol. 9, No. 1, 1987.

[62] A. Yilmaz, X. Li, and M. Shah, *Contour-Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Vol. 26, No.11, p. 1531-1536, 2004.

[63] J. MacCormick and A. Blake, *A Probabilistic Exclusion Principle for Tracking Multiple Objects*, International Journal of Computer Vision (IJCV), Vol. 39, No. 1, p. 57-71, 2000.

[64] Y. Wu, T. Yu, and G. Hua, *Tracking Appearances with Occlusions*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2003.

[65] Y. Huang and I. Essa, *Tracking Multiple Objects Through Occlusions*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

[66] S.A. Velastin et al., *Analysis of Crowd Movements and Densities in Built-up Environments Using Image Processing*, IEE Colloquium on Image Processing for Transport Applications, 1993.

[67] S. Bouchafa, D. Aubert, and S. Bouzar, *Crowd Motion Estimation and Motionless Detection in Subway Corridors by Image Processing*, IEEE Conference on Intelligent Transportation System, 1997.

[68] D.J. Fleet and A.D. Jepson, *Computation of Component Image Velocity From Local Phase Information*, International Journal of Computer Vision (IJCV), Vol. 5, No. 1, 1990.

[69] B. K. P. Horn and B.G. Schunck, *Determining Flow*, Artificial intelligence, 1981.

[70] S. Bouchafa, D. Aubert, L. Beheim, and A. Sadji, *Automatic Counter Flow Detection in Subway Corridors by Image Processing*, Journal of Intelligent Transportation Systems, Vol. 6, Issue 2 2001.

[71] J. H. Yin, *Automation of Crowd Data Acquisition and Monitoring in Confined Areas Using Image Processing*, Ph.D. Dissertation, Department of Electronic Engineering, Kings College London, University of London, 1996.

[72] E. L. Andrade, S. Blunsden, and R. B. Fisher, *Modeling Crowd Scenes for Event Detection*, IEEE International Conference on Pattern Recognition, 2006.

[73] L. Andrade, S. Blunsden, and R. B. Fisher, *Hidden Markov Models for Optical Flow Analysis in Crowds*, IEEE International Conference on Pattern Recognition, 2006.

[74] B. A. Boghossian and S. A. Velastin, *Motion-based Machine Vision Techniques for the Management of Large Crowds*, The 6th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 1999.

[75] B. P. L. Lo and S. A. Velastin, *Automatic Congestion Detection System for Underground Platforms*, Proceedings of International Symposium on Intelligent Multimedia, Video and Speech Processing, 2001.

[76] M. Boninsegna, T. Coianiz, and E. Trentin, *Estimating the Crowding Level with a Neuro-Fuzzy Classifier*, Joural of Electronic Imaging, Vol. 6, No.3, 1997.

[77] Y. Ke, R. Sukthankar, and M. Hebert, *Events detection in Crowded Videos*, IEEE International Conference on Computer Vision (ICCV), 2007.

[78] Q. C. Pham, L. Gond, J. Begard, N. Allezard, and P. Sayd, *Real-Time Posture Analysis in a Crowd Using Thermal Imaging*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

[79] D. Helbing, I. J. Farkas, P. Molnar, and T. Vicsek, *Simulation of Pedestrian Crowds in Normal and Evacuaion Simulations*, In Pedestrian and Evacuation Dynamcs, M.Schreckenberg and S. D. Sharma (eds.), Springer, pp.2158, 2002.

[80] D. Helbing and P. Molnar, *Social Force Model for Pedestrian Dynamics*, Physical Review E, Vol. 51, pp. 4282, 1995.

[81] V. J. Blue, and J. L. Adler, *Cellular Automata Micro Ssimulation for Modeling Bi-directional Pedesrian Walkways*, Transportation Research, Vol. 35, No. 3, 2001.

[82] A. Schadschneider, *Cellular Automaton Approach to Pedestrian Dynamics Theory*, In Pedestrian and Evacuation Dynamics, M.Schreckenberg and S.D. Sharma(eds.), Springer, pp.7586, 2002.

[83] A. Penn and A. Turner, *Space Syntax-based Agent Simulation*, In Pedestrian and Evacuation Dynamics, M.Schreckenberg and S.D.Sharma (eds.), Springer, 2002.

[84] M.Bierlaire, G.Antonini, and M.Weber, *Behavioral Dynamics for Pedestrians, In Moving Through Nets: The Physical and Social Dimensions of Travel*, K.Axhausen(ed.), Elsevier, pp.118., 2003.

[85] T. I. Lakoba, D. J. Kaup, N. M. Finkelstein, *Modificaions of the Helbing-Molnr-Farkas-Vicsek Social Force Model for Pedestrian Evolution*, Simulation, Vol.81, No.5, p. 339-352, 2005.

[86] R. Huges, *The Flow of Human Crowds*, Annual Review of Fluid Mechanics, Vol. 35, Issue 35, p. 169-182, 2003.

[87] R. Hughes, *A Continuum Theory for the Flow of Pedestrians*, Transportation Research, Vol. 36, No. 6, 2002.

[88] L. F. Henderson, *On the Fluid Mechanics of Human Crowd Motion*, Transportation Research, Vol. 8, 1974.

[89] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz, *Simulation of Pedestrian Dynamics Using a Two-dimensional Cellular Automaton*, Physica A, Vol. 295, No. 3, 2001.

[90] A. Kirchner and S. Andreas, *Simulation of Evacuation Processes Using a Bionics-Inspired Cellular Automaton Model for Pedestrian Dynamics*, Vol. 312, No. 1-2, 2002.

[91] A.G.A. Perera, C. Srinivas, A. Hoogs, G. Brooksby and W. Hu *Multi-Object Tracking Through Simultaneous Long Occlusions and Split and Merge Conditions*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

[92] R. Kaucic, A.G.A. Perera, G. Brooksby, J. Kaufhold and A. Hoogs, *A Unified Framework for Tracking through Occlusions and across Sensor Gaps*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

[93] Yaser Sheikh and Mubarak Shah, *Object Tracking Across Multiple Independently Moving Aerial Cameras*, IEEE International Conference on Computer Vision (ICCV), 2005.

[94] Yaser Sheikh, Xin Li and Mubarak Shah, *Trajectory Association across Non-overlapping Moving Cameras in Planar Scenes*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

[95] A. Nakazawa, H. Kato, and S. Inokuchi, *Human Tracking Using Distributed Vision Systems*, Proceedings of the International Conference on Pattern Recognition (ICPR), 1998.

[96] Q. Cai and J. K. Aggarwal, *Tracking Human Motion in Structured Environments using a Distributed Camera System*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Vol. 21, No. 1, 1999.

[97] T. H. Chang and S. Gong, *Tracking Multiple People with a Multi-Camera System*, IEEE Workshop on Multi-Object Tracking, 2001.

[98] S. Dockstader and A.Tekalp, *Multiple Camera Fusion for Multi-Object Tracking*, IEEE International Workshop on Multi- Object Tracking, 2001.

[99] S. Khan and M. Shah, *Consistent Labeling of Tracked Objects in Multiple Cameras with Overlapping Fields of View*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Vol. 25, No. 10, 2003.

[100] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou and S. Maybank, *Principal Axis-Based Correspondence between Multiple Cameras for People Tracking*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Vol. 28, No. 4, 2006.

[101] T. Huang and S. J. Russell, *Object Identification: A Bayesian Analysis with Application to Traffic Surveillance*, Artificial Intelligence, Vol. 103, 1998.

[102] V. Kettnaker and R. Zabih, *Bayesian Multi Camera Surveillance*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1999.

[103] R. Collins, O. Amidi, and T. Kanade, *An Active Camera System for Acquiring Multi-View Video*, IEEE International Conference on Image Processing (ICIP), 2002.

[104] O. Javed, Z. Rasheed, K. Shafique and Mubarak Shah, *Tracking in Multiple Cameras with Disjoint Views*, IEEE International Conference on Computer Vision (ICCV), 2003.

[105] C. Stauffer and K. Tieu, *Automated Multi-Camera Planar Tracking Correspondence Modeling*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2003.

[106] A. Gilbert and R. Bowden, *Tracking Objects Across Cameras by Incrementally Learning Inter-Camera Colour Calibration and Patterns of Activity*, European Conference on Computer Vision (ECCV), 2006.

[107] Y. Shan, H.S. Sawhney and R. Kumar, *Vehicle Identification between Non-Overlapping Cameras without Direct Feature Matching*, IEEE International Conference on Computer Vision (ICCV), 2005.

[108] Y. Shan, H.S. Sawhney and R. Kumar, *Unsupervised Learning of Discriminative Edge Measures for Vehicle Matching between Non-Overlapping Cameras*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005

[109] Y. Guo, S. Hsu, Y. Shan, H.S. Sawhney and R. Kumar, *Vehicle Fingerprinting for Reacquisition and Tracking in Videos*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

[110] O. C. Ozcanli, A. Tamrakar, B.B. Kimia and J.L. Mundy, *Augmenting Shape with Appearance in Vehicle Category Recognition*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

[111] O. Javed, K. Shafique, and Mubarak Shah, *Appearance Modeling for Tracking in Multiple Non-overlapping Cameras*, In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

[112] R. Collins and Y. Liu, *On-Line Selection of Discriminative Tracking Features*, IEEE International Conference on Computer Vision (ICCV), 2003.

[113] H. T. Nguyen and A. W. M. Smeulders, *Tracking Aspects of the Foreground against the Background*, European Conference on Computer Vision (ECCV), 2004.

[114] H. T. Nguyen, Q. Ji, and A. Smeulders, *Spatio-Temproal Context for Robust Multi-Traget Tracking*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Vol. 29, No. 1, 2007.

[115] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, *High Accuracy Optical Flow Estimation based on a Theory for Warping*, European Conference on Computer Vision (ECCV), 2004.

[116] C. Stauffer, *Estimating Tracking Sources and Sinks*, IEEE Workshop on Event Mining, 2003.

[117] D. Helbing et al., *Active Walker Model for the Formation of Human and Animal Trails*, Phys. Rev. E, Vol. 56, No. 3, 1997.

[118] L. Lama, *Active Walker Models for Complex Systems*, Chaos, Solitons & Fractals, Vol.e 6, Complex Systems in Computational Physics, 1995.

[119] F. Meyer, *Topographic Distance and Watershed Lines*, Signal Processing , Vol. 38, p. 113-125, 1994.

[120] S. Ali and M. Shah, *COCOA - Tracking in Aerial Imagery*, SPIE Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications, 2006

[121] J. Shi and J. Malik, *Normalized Cuts and Image Segmentation*, IEEE Transactions on Pattern Analysis and Machine Intellegience (TPAMI), Vol. 22, No. 8, p. 888-905, 2000.

[122] L. I. Piterbarg and T. M. Özgökmen, *A Simple Prediction Algorithm for the Lagrangian Motion in Two-Dimensional Turbulent Flows*, SIAM Journal on Applied Mathematics, Vol. 63, Issue 1, p. 116-148, 2002.

[123] L. I. Piterbarg, *Short-Term Prediction of Lagrangian Trajectories*, Journal of Atmospheric and Ocean Technology, Vol. 18, Issue 8, p. 1398-1410, 2001.

[124] T. M. Özgökmen, L. I. Piterbarg, A. J. Mariano, and E. H. Ryan, *Predictability of Drifter Trajectories in the Tropical Pacific Ocean*, Journal of Physical Oceanography, Vol. 31, Issue 9, p. 2691-2720, 2001.

[125] T. M. Özgökmen, A. Griffa, and A. J. Mariano, *On the Predictability of Lagrangian Trajectories in the Ocean*, Journal of Atmospheric and Ocean Technology, Vol. 17, Issue 3, p. 366-383, 2000.

[126] S. Lipster and A. N. Shiryaev, "*Statistics of Random Processes*",Springer Verlag, 1978.

[127] W. Kinser, *Characterizing Chaos Through Lyapunov Metrics*, IEEE International Conference on Cognitive Informatics, 2003.

[128] G. Tancredi, A. Sanchez and F. Roig, *A Comparison Between Methods To Compute Lyapunov Exponents*, Astronimical Journal, Vol. 121, No. 2, p. 1171-1179, 2001.

[129] R. Gurka et al., *Computation of Pressure Distribution Using PIV Velocity Data*, In Proc. of the 3rd International Workshop on Particle Image Velocimetry, 1999.

[130] S. Mann and R.W. Picard, *Video Orbits of the Projective Group: A Simple Approach to Featureless Estimation of Parameters*, IEEE Transactions on Image Processing, Vol. 6, No. 9, p. 1281-1295, 1997.

[131] Z. Yin and R. Collins, *Moving Object Localization in Thermal Imagery by Forward-backward MHI*, In IEEE International Workshop on Object Tracking and Classification Beyond the Vision Spectrum, 2006.

[132] D. Comaniciu and P. Meer, *Mean Shift: A Robust Approach Toward Feature Space Analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Vol. 24, No. 5, p. 603-619, 2002.

[133] A. Wolf, J. B. Swift, H. L. Swinney, and J. A.. Vastano, *Determining Lyapunov Exponent from a Time Series*, Physica D, Vol. 16D, p. 285-317, 1985.

[134] C. Stauffer and W. Eric L. Grimson, *Learning Patterns of Activity Using Real-Time Tracking*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Vol. 22, No. 8, p. 747-757, 2000.

[135] http://www.video.google.com

[136] http://www.gettyimages.com

[137] http://www.ngsim.fhwa.dot.gov/