ANOMALY DETECTION FOR VIDEO SURVEILLANCE IN CROWDED ENVIRONMENTS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

CİHAN ÖNGÜN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2014

Approval of the thesis:

ANOMALY DETECTION FOR VIDEO SURVEILLANCE IN CROWDED ENVIRONMENTS

Submitted by **Cihan Öngün** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering, Middle East Technical University** by,

Prof. Dr. Canan Özgen,
Dean, **Graduate School of Natural and Applied Sciences** ______________

Prof. Dr. Gönül Turhan Sayan
Head of Department, **Electrical and Electronics Engineering** ______________

Assoc. Prof. Dr. İlkay Ulusoy
Supervisor, **Electrical and Electronics Eng. Dept., METU** ______________

Assoc. Prof. Dr. Alptekin Temizel,
Co-supervisor, **Graduate School of Informatics, METU** ______________

Examining Committee Members:

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Engineering Dept., METU ______________

Assoc. Prof. Dr. İlkay Ulusoy
Electrical and Electronics Engineering Dept., METU ______________

Assoc. Prof. Dr. Alptekin Temizel
Graduate School of Informatics, METU ______________

Assoc. Prof. Dr. Afşar Saranlı
Electrical and Electronics Engineering Dept., METU ______________

Assist. Prof. Dr. Fatih Kamışlı
Electrical and Electronics Engineering Dept., **METU** ______________

Date: ______________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name, Last Name:	Cihan Öngün
Signature:	

ABSTRACT

ANOMALY DETECTION FOR VIDEO SURVEILLANCE IN CROWDED ENVIRONMENTS

Öngün , Cihan
M.S., Department of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. İlkay Ulusoy
Co-Supervisor: Assoc. Prof. Dr. Alptekin Temizel

August 2014, 61 Pages

Crowd behavior analysis and anomaly detection in crowded environments have become more important in recent years. In the literature there are two main approaches for crowd behavior analysis based on the density of the crowd. While individual analysis is more efficient for low and medium density crowds, holistic approaches which consider the crowd as a whole are more efficient for high density crowds. Crowd behavior analysis studies can be examined in 3 categories: group behavior analysis, crowd behavior analysis and anomaly detection. While group behavior analysis is based on detection and tracking of human groups, crowd behavior analysis studies considered the whole crowd in the video. These steps are generally followed by anomaly detection which is the task of detecting the events which are normally not expected in a scene. In this work, the aim is to detect behavioral anomalies in high density crowds where detection and tracking of individuals are difficult. Video scene is considered as a whole and a heat map is generated using Finite-Time Lyapunov Exponents (FTLE) based on motion changes and this heat map is divided into behavioral clusters using hierarchical clustering.

Then considering the distribution of these clusters existence of anomaly is determined and abnormal cluster are detected using an adaptive threshold.

Keywords: Computer Vision, Video Surveillance, Anomaly Detection, Crowd Behavior Analysis

ÖZ

KALABALIK ORTAM VİDEO GÖRÜNTÜLERİNDE ANOMALİ TESPİTİ

Öngün , Cihan
Yüksek Lisans, Elektrik Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. İlkay Ulusoy
Ortak Tez Yöneticisi: Doç. Dr. Alptekin Temizel

Ağustos 2014, 61 sayfa

Kalabalık ortamların analizi ve aykırı davranışların tespiti her geçen gün daha fazla önem kazanmaktadır. Kalabalık ortam analizi kalabalığın yoğunluğuna göre değişim göstermektedir. Düşük ve orta yoğunluklu kalabalıklarda bireysel analizler daha etkili olurken, yüksek yoğunluklu kalabalıklarda kalabalığı bir bütün gibi gören bütüncül yaklaşımlar daha etkili olmaktadır. Kalabalık ortam analizi üzerine yapılan çalışmalar 3 gruba ayrılabilir: grup davranışı analizi, kalabalık davranışı analizi ve aykırılık tespiti. Grup analizi bir grubun tespiti ve takibi ile yapılırken kalabalık analizi videodaki bütün kalabalık dikkate alınarak yapılır. Bu basamakları genelde aykırılık tespiti takip eder. Aykırılık tespiti video sahnesi üzerinde normalde olması beklenmeyen durumların tespitini amaçlar. Bu çalışmada amaç kişilerin ayrı ayrı tespit ve takibinin güç olduğu yüksek yoğunluklu kalabalıklarda davranışsal aykırılık tespitidir. Video sahnesi bir bütün olarak ele alınıp Sonlu-Zamanlı Lyapunov Üsleri (FTLE) kullanılarak hareket değişimine göre bir yoğunluk haritası elde edilmiş ve bu harita hiyerarşik sınıflandırma kullanılarak davranışsal kümelere bölünmüştür. Daha sonra bu kümelerin dağılımına bakılıp videoda aykırılık olup olmadığı belirlenmiş ve uyarlanabilir bir eşik değeri ile aykırı kümeler belirlenmiştir.

Anahtar Kelimeler: Bilgisayarlı görü, video gözetleme, aykırılık tespiti, kalabalık davranış analizi

To My Parents

ACKNOWLEDGEMENTS

I wish to express my thanks and gratitude to Assoc. Prof. Dr. Alptekin Temizel and Assist. Prof. Dr. Tuğba Taşkaya Temizel. During my graduate education, they were always eager to help me and they are the main inspiration for me to be an academician. I consider them as a role model. Also the assistance and reviews of Assoc. Prof. Dr. İlkay Ulusoy are gratefully acknowledged.

I am also thankful to my dear friend Ayşe Elvan Gündüz who is not just a colleague but a close friend. She has helped me during studies and supported me in personal life.

I would like to thank my "brothers" Cem Keser, Erdinç Yasin Dinç and Yusuf Türkyılmaz. They were always with me in all cases and I am sure they will be forever. I also want to thank Ufuk Irmak for his friendship, help and brilliant ideas during higher education.

Finally, my biggest gratitude is to my parents (Ayten Öngün, Nadir Öngün). They supported me for all the decisions I made and inspired me with their education mentality and determination. I am proud to be their son.

This research was funded by The Scientific and Technological Research Council of Turkey with the grant number 112E141.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

2D : Two Dimensional

3D : Three Dimensional

EWMA : Exponentially Weighted Moving Average

EWT : Equal Width Thresholding

FTLE : Finite-Time Lyapunov Exponents

GIS : Geographic Information System

GMM : Gaussian Mixture Model

GPU : Graphics Processing Unit

HMM : Hidden Markov Model

i=1 : integration length 1

i=10 : integration length 10

KL : Kullback-Leibler

KLT : Kanade–Lucas–Tomasi

LDA : Latent Dirichlet Allocation

MDT : Mixture of Dynamic Textures

METU : Middle East Technical University

MM : Mean-Maximum

MRF : Markov Random Field

PCA : Principal Component Analysis

PDF : Probability Density Function

SIFT : Scale Invariant Feature Transform

SVM : Support Vector Machines

Th : Threshold

UCSD : University of California, San Diego

UMN : University of Minnesota

CHAPTER 1

INTRODUCTION

1.1. Overview

Automated surveillance systems are gaining importance day by day due to security reasons and many applications for surveillance cameras have been developed in the recent years. Most of these applications aim to detect and analyze individuals. Especially face recognition systems are well developed and have high success rates. Combining face recognition and tracking systems, cameras are able to detect and track specific people. However, these systems fail to provide satisfactory performance in crowded areas where it is difficult to detect and track individuals. Another concern that has been raised regarding face recognition and person tracking systems is that of privacy. Due to the privacy concerns, privacy preserving systems have been gaining popularity.

Security in crowded areas is highly important since an undesired event may have catastrophic results. For instance, about 300 people lost their lives in a stampede in Mandher Devi temple on 25 January 2005 [1]. Jamarat Bridge in Mecca is one of the most crowded pedestrian bridges and many stampedes occurred in the history. 270 people died in May 1994 and 251 people died in February 2004 in the stampedes. After 2004 incident, a major construction work was carried out on the bridge but these precautions have not prevented death of 345 pilgrims in January 2006 in another stampede [2]. Occurrence of such incidents reveals the need of surveillance systems to enable taking proactive actions to prevent potential disasters.

Even though there is a high number of surveillance cameras installed on various sites, security staff are required to monitor the captures. However, there is a potential for human observers to miss the important events. Human observers cannot be fully focused at all times so it makes them unreliable. Computer vision systems are required to assist the observers to increase the effectiveness of surveillance systems. Computer vision systems can analyze all the captured streams simultaneously which is not possible by human observers.

Crowd dynamics varies with the density of the crowd. Individual analysis may be suitable for low density crowds but it is not effective for high density crowds since it is not possible to capture every individual clearly due to occlusions and resolution. Due to these reasons, high density crowds need different approaches and techniques to analyze.

The aim of this thesis is to analyze the high density crowd behavior and detect anomalies. First velocity and direction data are extracted from the video using optical flow without any preprocessing. Then a heat map is generated using Finite-Time Lyapunov Exponents (FTLE). This motion data is then clustered using agglomerative clustering where clusters represent different behaviors. Even anomaly is not detected, the motion is clustered into behavioral groups and this is helpful for analyzing different behaviors in the crowd. Anomaly localization process starts if anomaly is detected in these clusters. Incoherent behavioral clusters are labeled as abnormal at the end of the process. The method is unsupervised so predetermined rules or training are not necessary. Also there are no privacy concerns since the method does not track, identify or collect information about individuals.

Most of the anomaly detection studies aim to detect the starting frame of the anomaly, however localization of the anomaly is also important for high density crowds. One or two people may cause anomaly in hundreds of people and localization of these people by the operators may take time resulting in a delay in intervening with the event.

Definitions of the terms “crowd” and “anomaly” depend on the content and they have different in the context of different systems. Hence it is important to define these terms in the context of this work.

Crowd is formally defined as “a large number of people gathered together in a disorganized or unruly way” or “the mass or multitude of ordinary people” [3]. While these definitions are generally agreed upon, the term “density of the crowd” vary in meaning. Scope of this thesis is *high density crowds* such that people cannot move freely without touching anybody and it is not possible to observe each individual without occlusions in the scene.

Definition of *anomaly* is a significant part of this work to understand the methodology and results. Anomaly is formally defined as “something that deviates from what is standard, normal, or expected” [3]. Throughout this work, we use the definition “behaviors that are inconsistent with the general behavior in the scene”. Since this definition covers not only dangerous but also all inconsistent behaviors, these behaviors should be seen as potential anomalies for crowds. This method aims to aid human operators by advising the potential anomalies and localizing them in the frame.

1.2. Contributions

In this thesis, I aim

- Automated clustering of behaviors in high density crowds. This is the first step since we should see the behavioral clusters even there is not any anomaly.
- Detection and localization of abnormal or incoherent behaviors if there is any. After detecting if there is anomaly, localization is important to specify the position of anomaly to aid to operator.

- Using FTLE for global motion analysis instead of individual analysis. Since individual analysis is not efficient in high density crowds, FTLE is used to analyze global motion in the scene.

1.3. Limitations and Assumptions

As mentioned before, this thesis aims to analyze high density crowds followed by detection and localization of anomalies if there are any. The limitations and assumptions of this work are as follows:

- Low density crowds are considered to be out of scope of this work. Dynamics of low density crowds are highly different than high density crowds so object based approaches and individual behavior analysis and tracking systems are generally more effective for such environments.
- Since the method does not use any preprocessing for moving background detection, camera must be stationary otherwise background is possibly detected wrong. Camera position and angle should be positioned appropriately to capture the crowd. Moving or blocking objects near the camera may cause false alarms so cameras positioned at an elevated position and having wide field of view need to be used for effective detection.
- Inconsistencies with very low speed motion (or even motionless) can not be detected since these are probably clustered as background. Method analyzes the behaviors using motion speed and direction so very low speed motions give no information about behaviors.

1.4. Organization of the Thesis

The organization of the thesis is as follows: Chapter 2 reviews the literature and existing studies. This chapter is divided into 4 subsections as group behavior analysis, crowd behavior analysis, anomaly detection and crowd synthesis since each of these areas have different characteristics. Chapter 3 presents the methodology. A general flowchart of the proposed method is given and individual steps are explained in detail. Chapter 4 starts with the explanation of datasets and other resources. Generation of the simulation videos, captures and publicly available datasets are explained and test results are provided. A general walkthrough of all implementations and input/outputs are presented from the first step until the anomaly detection results on a sample test video. Concluding remarks and future work directions are provided in Chapter 5.

CHAPTER 2

LITERATURE REVIEW

2.1. Overview

Automated crowd analysis approaches are grouped into different categories in different studies. For example, in [4] approaches are analyzed in following 3 categories:

- Counting and density estimation: Mostly used for estimating the number or density of the crowd to avoid dangerous situations. Also people counting is used to estimate the number of participants in mass organizations nowadays with the increasing accuracy of these methods.
- Tracking: Can be individual tracking and crowd tracking. For video applications, tracking is basically detecting the position of the target in each frame. Individual tracking is used for tracking a specific person such as criminals, patients etc. Crowd or group tracking is used for detecting main flows and anomalies in the scene or estimating the future flow of the crowd.
- Behavior understanding: Estimating the behavior of a crowd in a given environment. There are two main approaches for crowd behavior analysis. In object-based approaches, the individuals in a crowd are detected and the behaviors of the individuals are analyzed. In holistic approaches, the crowd is treated as a single entity and analyzed as a whole.

In another study [5], approaches are divided into 2:

- Crowd information extraction: Contains the methods aiming to extract and analyze crowd information. This category is also divided into subcategories as crowd density measurements, recognition, and tracking.
- Crowd modelling and events: Model based approaches and methods attempting to infer events without construction of models are presented in this category. Also some models from non vision approach (physics inspired, agent based, cellular automation, nature based) are presented.

In this thesis, the literature is grouped into 3 categories based on the aim of the study: group behavior analysis, crowd behavior analysis and anomaly detection. Also crowd synthesis is in the scope of this thesis as some crowd simulations have been generated.

2.2. Group Behavior Analysis

These methods aim to analyze group behaviors by detecting and tracking the groups. Although it is not in the scope of this thesis, these methods are helpful to understand and analyze crowds. The study proposed in [6] focuses on group detection and classification. They use template matching to track individuals and generate a Voronoi diagram at each frame. Using Voronoi diagrams, they determine the temporal evolution of some sociological and psychological parameters, such as distance to neighbors and personal spaces. These individual characteristics are analyzed to detect the formation of groups and their classification as voluntary or involuntary. This study mostly focuses on analyzing psychological concepts about groups and individuals. In [7], a two layer method for tracking a group of objects is proposed. The first layer produces a set of spatio temporal strokes based on low level operations to track the active regions. The second layer performs a consistent labeling of the detected segments using a statistical model based on Bayesian networks.

In [8], authors aim group-level event recognition. The method is based on extraction of feature histograms for each frame to represent their group connectivity and motion features. These features are group connectivity, moving direction, connectivity change and moving speed. To extract these features, person detection and tracking is used. Features are represented using a bag-of-words approach and an SVM is trained to classify video segment into different categories for event recognition.

2.3. Crowd Behavior Analysis

Crowd behavior analysis is the initial step for anomaly detection studies since anomaly detection cannot be done without behavior analysis. Some studies just aim to analyze, categorize or detect different group behaviors without any anomaly detection. In one of the early studies [9], they aim to estimate the paths and main directions of a crowd. Anomaly detection algorithm of this study is based on the flow paths. They define three abnormal situations: circular flow paths which may cause bottlenecks for large crowds are detected by Hough voting, diverging flows indicating local threats (fights, fire etc.) are detected manually, and obstacles in the flow paths are detected by a region-growing segmentation to group the motion free regions in the scene. Algorithm used for anomaly detection is not fully automated and circular flow path detection method doesn’t work for elliptical paths. Study [10] aims to detect stationary crowds and estimate crowd density. To distinguish stationary crowds from moving crowds, Fourier Transform is used (moving objects are expected to cause more higher frequency components than stationary objects). Density estimation is based on the foreground such that number of foreground pixels is used to estimate the density. However this assumption is not efficient when there are large moving objects in the scene. In [11], they present a real-time system that detects moving crowd in a video sequence. Crowd detection algorithm is based on spatio-temporal analysis of the video sequence, where optic-flow patterns generated by a forward moving vehicle are analyzed. They assume that a forward moving vehicle generates outward flow, hence inward flow must be generated by independently moving objects. When a group of people is moving in opposite directions they form a collection of intersecting lines in the spatio-temporal domain. The number of intersections is used to distinguish between an inward moving vehicle or pedestrian and a crowd. In [12], a work based on motion estimation method to detect objects staying motionless for a period is provided. For motion estimation, block matching, optical flow and Gabor filter techniques are used and a modified Horn & Schunk method is implemented. System raises an alarm when both the stop duration computed is higher than a threshold and the location is not a place where a stop is normally expected (signals, phone box or authorized seller's boxes). In a later work of the same authors [13], they propose a variant of the method for counterflow detection in subway stations. The obtained optical flow motion vectors are filtered to reduce the noise which are subsequently used for the construction of motion trajectories. The trajectories are then used to detect counterflows. In [14], a method is proposed to detect dominant motions and anomalies in dense crowds. First they extract low level features and trajectories are generated by tracking of these features using optical flow. The trajectories which are spatially close to each other and have a similar direction of motion are clustered. The dominant motions in the video are calculated by following the cluster centers in each frame. However anomaly detection requires a human interpretation to analyze the motions and detect the abnormal ones. Study [15] uses feature tracking to count objects in moving crowds. KLT algorithm is used to track the features and then a connectivity graph is used for merging these features. Merged features are counted to find the number of objects. They have a high variance of error rates with the optimal parameters because of the difficulty of tracking features in different datasets. Study [16] is an unsupervised method to detect independent motions in crowd. They use a feature tracker to generate trajectories. Assuming that the features moving together should be part of the same independent motion, trajectories are clustered with an unsupervised Bayesian clustering method such that each cluster corresponds to a different independent motion in crowd. They aim to detect individual movements in the low density crowds. People carrying items, two bodies moving near each other may cause false results. Study [17] proposes a framework to identify behaviors in crowded scenes. Particle trajectories which are generated with a group of particles using optical flow are used to locate a region of interest in the scene. Using eigenvalues and angles of the particles at the accumulation points, behavior type, which is one of the five behaviors (blocking, lane, bottleneck, ring/arch, fountainhead), is identified. In [18], sources and sinks are modeled in crowded scenes. Trajectories are generated by forward and backward advection of particles. Start and end points of the particle trajectories are clustered to model sources and sinks respectively. The study also proposes a method to follow stranded particles to avoid occlusions. In [19], they proposed a Bayesian approach to segment individuals in crowds. They used 3D human shaped models to interpret the foreground. A probabilistic model based on Markov chain Monte Carlo integrates features including human shape, human height, camera model, head candidates, foreground objects etc. in a Bayesian framework. However, using these features is not efficient since the full body representation is usually not possible in high-density crows. In [20] they evaluate a crowd counting and event detection system based on holistic features like area, perimeter, internal edge and texture. For crowd counting, the scene is segmented into crowds moving in different directions, features are extracted from each segment, and the number of people in each segment is estimated with Gaussian regression. For event detection (evacuation, dispersion, merging, walking and running), an SVM classifier is trained using the KL (Kullback-Leibler) kernel.

There are also some works which were used in surveillance systems. In [21], they constructed a practical real-time system for estimating crowd density of certain areas by surveillance system of a city. They estimate low densities using feature extraction and high densities using Gray Level Dependence Matrix. After estimation, densities are marked on a city map using geographic information system (GIS). The system is aimed to provide early warning information and scientific basis for safety and security decision making. In [22], they developed a monitoring system to estimate crowding level for Dinegro underground station in Genoa, Italy. After background subtraction with a Kalman filtering based algorithm, neural networks are trained with the foreground features. Integration of the neural networks within the fuzzy decision rule results in an overall neuro-fuzzy classifier for crowding level.

There are some methods in the literature based on the models like Hidden Markov Models. In [23], a Mixture of Gaussians is used for background subtraction and optical flow is used for foreground objects in preprocessing part. To extract features, PCA is used on optical flow field. Hidden Markov Models are trained using spectral clustering. Anomaly detection is based on the comparison of the likelihood of the observation given by HMM and the detection threshold.

FTLE based approaches have recently gained popularity and there limited number of studies using this method. In [24], they propose a framework in which Lagrangian Particle Dynamics is used for the segmentation of high density crowd flows and detection of flow instabilities. After generating an FTLE field from a flow map over a grid of particles, they use the normalized cuts algorithm to segment the FTLE field. The boundary particles of the segments are used to measure the divergence between clusters and if the divergence is less than some threshold, segments are merged. This process is repeated for each block of frames and if a new segment appears, it is labeled as flow instability. Instability detection is supervised to learn the stable flow and only experienced on synthetic instabilities.

2.4. Anomaly Detection

Anomaly detection studies have been becoming increasingly popular and there are variety of methods. Most of the studies aims to detect the starting frame of the anomaly, however there are also recent studies aiming localization of the anomaly.

In [25], authors proposed an approach to estimate sudden changes and abnormal motion variations. First they generate a motion heat map to use as a foreground in the video. Then they detect features and track them using optical flow. A set of statistical measures are calculated and decision is based on these measures with a predetermined threshold. The method decides at which frame anomaly starts and needs a configured threshold since it varies depending on the camera position and angle. In [26], Social Force model is used to detect and localize abnormal behaviors in crowd videos. After generating particle trajectories on a group of particles using optical flow, Force Flow is obtained using social force model for every pixel. A code book is formed using K-means clustering and an LDA model is learned using only normal videos. Based on a fixed threshold on the estimated likelihood, frames are labeled as normal and abnormal using a bag of words approach. Localization is performed assuming that anomalies occur in active regions or the regions with higher social interactions so they localize abnormalities in the abnormal frame by locating the regions of high force flow. The anomaly detection method proposed in [27] consists of 4 steps: motion, size and texture feature extraction of foreground after dividing the image into cells, model estimation for each feature type, classifications which are likelihood of the magnitude of motion of foreground objects for speed control and likelihood of size of foreground objects, spatio-temporal post-processing to minimize isolated random noise present in the generated anomaly masks. They detect texture based abnormalities such as detecting skateboarders or bikers in the crowd and there are false results if anomalies have similar textures and when there are stationary foreground objects in the training. In [28], authors used two energy methods for anomaly detection. First method is based on the basic principle that where there is motion there is energy and they use variance of video divided into blocks. Second method uses the motion of the edges to evaluate the energy. Corners are detected using Harris corner detector and tracked using Lucas-Kanade optical flow. The approach is based on tracking corners of a group of three successive frames to determine the parameters needed for energy calculations. There are two anomaly definitions in the paper: The first is called static abnormality where the value energy at a given time greatly exceeds its supposed mean value at the time. The second is called dynamic abnormality when there is a sudden change in energy for a persistent period of time. In [29], a motion based method is used for anomaly detection. Optical flows are first estimated and then adjacency-matrix based clustering is used to cluster human crowds into groups. Cluster behaviors with attributes, orientation, position and crowd size, are characterized by a model of force field. Sudden appearance of a new group orientation is considered an unusual event. In [30], the authors introduced a new concept of contextual anomaly. The contextual anomaly is defined as ordinary behavior which is considered anomalous in a specific context. An example is a pedestrian walking in a different direction to all the other pedestrians in his neighborhood. After computing motion features, which are represented by spatio-temporal patches, all patches are classified and grouped into blobs to describe the position and size of each pedestrian. Then with the help of a patch histogram, the method detects contextual anomalies. In [31], they propose a technique for event detection in cluttered scenes in the presence of partial occlusions. An event model is constructed from a single training example. Video is segmented into spatio-temporal regions using mean shift and shape matching is used to match the model and video segments. Method needs training videos for each specific event.

There are also some studies based on different types of imaging techniques. For example, in [32] thermal imaging is used to detect pedestrian postures in a crowd. After background subtraction and head detection, posture is detected using a combination of several weak classifiers with the help of a 2D human shaped model. The aim of the study is to detect lying people which are defined as anomaly.

Model based approaches are popular for anomaly detection because of their high success rates. However these methods mostly need a learning step and complex models are used. In [33], anomaly detection in extremely crowded scenes is done based on spatio-temporal motion pattern behaviors. Video is divided into spatio-temporal cuboids and each cuboid is represented by a Gaussian distribution. Temporal relationships are encoded in distribution-based HMMs and spatial relationships in a coupled HMM. Results have false positives for slightly irregular motion patterns that may not have been captured in the training data. Also size of the cuboids is a limitation because of varying camera position and angles. In [34], authors used high-frequency and spatio-temporal features to detect the abnormal crowd behaviors in videos. To estimate these features, wavelet transform is applied to the video divided into cuboids and bag of words method is used to identify the likely patterns in the cuboids. For global anomaly detection LDA is used to model the normal scenes and for local anomaly detection Multiple HMMs are used. In [35], Gaussian Mixture Model is used for detection of abnormal sequences. After object detection, GMM is used for probability estimation of a new object which is determined by its position. Then with Exponentially Weighted Moving Average (EWMA) control charts, anomaly detection is performed using a probability threshold. However, anomaly detection is limited with the position of the object. In [36], authors proposed a two-step Markov Random Field (MRF)-based approach for density change detection. The first step is change detection that distinguishes background and foreground using a discontinuity preserving MRF-based approach where the information from different sources (background subtraction, intensity modeling) are combined with spatial constraints to provide a smooth motion detection map. In the second step, this map is combined with a geometry module to perform a soft auto-calibration to estimate a measure of congestion of the observed area. Study [37] uses HMM with optical flow features for emergency event detection. After background subtraction, optical flow features are generated. Then, Mixture of Gaussian Hidden Markov Models is trained in global and local scale. Emergency events are detected using a threshold. In [38], authors present an automatic method to detect abnormal crowd density by using texture analysis and learning. Image cells are generated by using the perspective projection model and the gray-level dependence matrix method is used to extract textural information within these cells. An SVM is trained to relate the textural features with the actual density of the scene. Anomaly detection is based on the assumption that density changes may indicate a potential danger or emergency so they developed a method to detect density changes. In [39], authors proposed a framework for temporal and spatial anomaly detection. For temporal anomaly detection, scene is divided into cells and each region of the scene, a Mixture of Dynamic Textures (MDT) is learned during training. At test time, the negative log-likelihood of the spatiotemporal cell is computed using the MDT. For spatial anomaly detection, saliency (locations with some attributes that makes them stand-out from their surroundings) is computed and locations whose saliencies are above some threshold are labeled as abnormal. Study [40] uses an unsupervised learning approach for anomaly detection. Video is divided into blocks and optical flow is computed for each pixel. A sparse vector machine based model is built by selecting only relevant samples for approximating the non-parametric likelihood. Finally, a classification rule is deduced from this likelihood to determine anomalies. There are some false results because of deficiencies in learning and noisy optical flow results due to light and shadow. Study [41] uses scale-invariant feature transform (SIFT) for feature detection and tracking. After background subtraction and normalization of the data, a Gaussian Mixture Model is trained. Anomaly detection is based on the motion vector change rates. Study aims to detect starting frame of the anomaly and localization of the anomaly is not in the scope of this work.

As mentioned before, there are only a few FTLE based approaches since this approach is very recent. In [42], the authors proposed a Lagrangian framework to cluster and detect abnormalities. After generating trajectories from optical flow field, they considered three types of Lagrangian measures: arc length, direction and separation. With these 3 measures, they implement some computer vision tasks, including crowd movement segmentation and abnormal behavior detection. However in anomaly detection part, a Gaussian model is used to learn the initial part of normal behavior and is updated in each frame. This method is a supervised method as it requires learning of normal behaviors and aims to detect the starting frame of the anomaly and not the location of the anomaly. In [43], Lyapunov exponents are used to capture both local and global characteristics. The method is based on the assumption that flow is incompressible and irrotational according to Helmholtz decomposition theorem. A Bayesian LDA model was fitted to build the models for the three different crowd scenes: rush, scatter and herding. While this method also uses Lyapunov exponents, different to the method proposed in this paper, it works in a supervised fashion and it is not able to localize the detected anomaly.

2.5. Crowd synthesis

While crowd synthesis is not the main focus of thesis, it is nevertheless a relevant topic for this work since simulation videos were created and tested during the study. Simulation videos are created to represent situations such as people fighting or a crowd is going through a door since it is difficult to obtain these kind of video captures with stationary cameras and proper angles. To measure the similarity between simulations and real-world data, in [44] they use Bayesian inference to estimate the simulation states and then Maximum Likelihood estimator to approximate the prediction errors. This process is iterated using Entropy Metric algorithm which is the distribution of errors between the evolution of a crowd predicted by a simulator and observed data. For testing, three simulation methods are selected in [44]. Steering methods use a set of rules to determine the motions of agents. [45] is a classical steering method which agents follow three rules: steer towards the goal, steer away from the nearest obstacle and steer away from the nearest person. Social force simulation methods determine the motion of agents based on Helbing's Social Force Model [46]. The motion of the agents is computed by applying a series of forces to each agent that depend on the relative positions and velocities of nearby agents. Predictive Planning based simulation methods attempt to anticipate collisions based on neighboring agents' positions and velocities and determine new paths which avoid these collisions. In [47], each agent navigates by constraining its velocity to those which will avoid collisions with nearby neighbors and obstacles. There are also some studies for dangerous situations. In [48], authors generate simulations of dangerous situations like blocked exit, collapse of a person in the crowd, and escape panic. They use Social Force model for pedestrian motion and a series of control points for pedestrian path.

CHAPTER 3

METHODOLOGY

3.1. Overview

In this work, we aim to develop an unsupervised method to analyze high density crowds and localize anomalies if there is any.

The method consists of four main steps:

1. Extracting the velocity and direction data from video using optical flow,

2. Creating heat maps from this velocity and direction data using Finite-Time Lyapunov Exponents (FTLE),

3. Defining behavior clusters in the video from heat maps by using agglomerative clustering,

4. Detecting abnormal clusters (if exists) with unsupervised thresholding.

The general flowchart of the method is shown in Figure *1*.

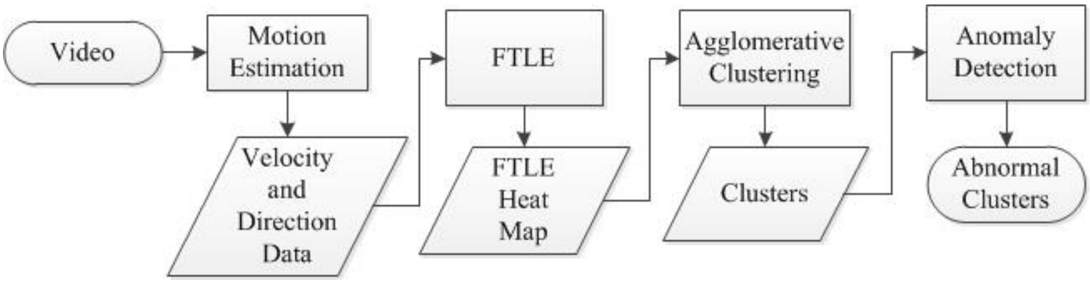


Figure 1. Flowchart of the proposed method

3.2. Step 1: Motion Estimation

Motion estimation is the process of determining motion speed and direction of objects in a scene. For video applications, this process is employed for consecutive frames. In this thesis, optical flow is used for motion estimation.

Optical flow is a well-known algorithm for image and video processing purposes. Basically, it is used for finding motion magnitude and direction between two consecutive frames. Optical flow algorithms can be divided into two groups: sparse and dense. Sparse methods process only some pixels of the whole image. Sparse methods are mostly used for tracking or getting motion data for interesting features in an image. Dense methods process and calculate optical flow values for all pixels in the image. Dense methods are used if optical flow values for all of the pixels in the video are needed. While sparse methods are faster, dense methods are more accurate.

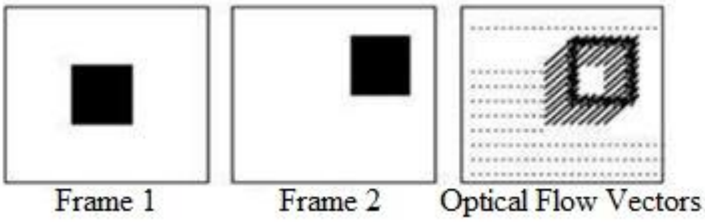


Figure 2. Optical Flow illustration

In this work, we use dense optical flow since all the motion in the crowd is significant and we need optical flow values for all the pixels. For implementation, Farneback algorithm is used [49]. Farneback proposes a two-frame motion estimation method based on polynomial expansion. For software implementation, OpenCV's built-in function for Farneback algorithm is used [50].

Since we are looking for a general movement in crowds, we used subsampling to decrease the resolution and make the process faster. Tests have been made with decreasing resolution to 4-to-1 so an optical flow value is calculated for a 4x4 area. As a result for a video of size MxN, the result of optical flow data has a size of $\frac{M}{4}x\frac{N}{4}$.

3.3. Step 2: Finite-Time Lyapunov Exponents (FTLE)

In this work, FTLE is used for motion analysis. FTLE is a tool for analysis of flows. It is mostly used for computation of liquid or gas flows. The idea behind using FTLE for crowds is the similarity between high density crowd dynamics and liquid dynamics. FTLE generates a heat map for global flow and this information is useful for analyzing global motion of high density crowds, since using individual motions is not efficient and costly.

FTLE is the value of stretching of a trajectory of a point for time interval [t,t+T] [51]. It is basically a measure of separation between infinitely-close particles for a time interval.

FTLE is calculated for all particles from the first positions at t_0 to the last positions at t_0+T. This calculation gives the information about particle trajectories and flow of particles for the time interval T. As a result, FTLE values are calculated for all particles for all frames. FTLE map is a heat map generated using all the particles in the flow.

We can define the flow of a point x from t_0 to t_0+T as (1). Since we need separation value between infinitely-close particles, we can define y, which is an infinitely close particle to x, as in (2) at t_0.

$$x \rightarrow \varphi_{t_0}^{t_0+T} \tag{1}$$

$$y = x + \delta x(t_0) \tag{2}$$

As time evolves from t_0 to t_0+T, the distance between x and y will change and this change can be defined as (3).

$$\delta x(t_0 + T) = \varphi_{t_0}^{t_0+T}(y) - \varphi_{t_0}^{t_0+T}(x) \tag{3}$$

Taking the Taylor series expansion of the flow point about *x*, second equality in (3) can be rewritten as (4).

$$\frac{d\varphi_{t_0}^{t_0+T}(x)}{dx}\delta x(t_0) + O(||\delta x(t_0)||^2) \tag{4}$$

Since $\delta x(t_0)$ is infinitesimal, $O(||\delta x(t_0)||^2)$ term can be ignored. The magnitude of perturbation is given in (5) where * denotes transpose operation.

$$\begin{aligned} ||\delta x(t_0+T)|| &= \sqrt{\langle \frac{d\emptyset_{t_0}^{t_0+T}(x)}{dx}\delta x(t_0), \frac{d\emptyset_{t_0}^{t_0+T}(x)}{dx}\delta x(t_0)\rangle} \\ &= \sqrt{\langle \delta x(t_0), \frac{d\emptyset_{t_0}^{t_0+T}(x)*}{dx}\frac{d\emptyset_{t_0}^{t_0+T}(x)}{dx}\delta x(t_0)\rangle} \end{aligned} \tag{5}$$

We can calculate the magnitude of deformation with the formula (6) which is a finite-time version of the Cauchy-Green deformation tensor.

$$\Delta = \frac{d\emptyset_{t_0}^{t_0+T}(x)*}{dx}\frac{d\emptyset_{t_0}^{t_0+T}(x)}{dx} \tag{6}$$

Since we are interested in the maximum stretching between x and y, maximum distance occurs when eigenvalue (λ) of Δ has the largest value (7).

$$\max_{\delta x(t_0)} ||\delta x(t_0+T)|| = \sqrt{\langle \overline{\delta x}(t_0), \lambda_{max}(\Delta)\, \overline{\delta x}(t_0)\rangle} = \sqrt{\lambda_{max}(\Delta)}\, ||\overline{\delta x}(t_0)|| \tag{7}$$

As a result; FTLE value at the point *x* from *t₀* to *T* can be defined as (8) [52].

$$\sigma_{t_0}^{T}(x) = \frac{1}{|T|} ln\sqrt{\lambda_{max}(\Delta)} \tag{8}$$

For implementation of FTLE, LCS Matlab Kit [53], which is a software package for Matlab, is used. The software takes optical flow values as input, computes particle trajectories for a given time period and produces an FTLE heat map for each frame as output. We modified the software to save FTLE map as a matrix so we can use values for each frame. Resulting matrix has the same resolution as the optical flow result. High speed movements or incoherent particles with neighbors have the highest values while coherent particles or low speed movements have low values. Stationary areas are “0” or close to “0” as expected. The visualization of the process can be seen in Figure 3.

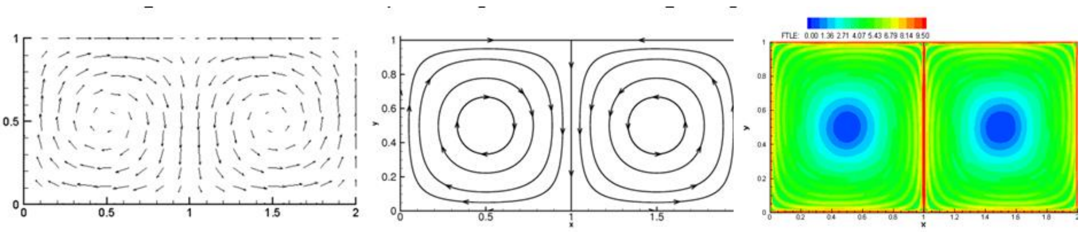


Figure 3. Visualization of FTLE calculation process [52]

In the software, the most important parameter is “integration length” which is denoted with *T* in the formulas above. Integration length determines the frame count for tracking particles to generate trajectories. The importance of this parameter will be explained later in the test results with different experiments. Other parameters are standard for all tests.

Using raw data for global motion information may cause false results since it is very noisy. FTLE process results a value for each point using motion vectors in two axes so it can be seen as a dimension reduction technique. All dimension reduction techniques cause information loss, however FTLE generates an incoherency value for each point so this technique gives the most useful information for detecting behavioral inconsistencies.

3.4. Step 3: Agglomerative Clustering

FTLE process generates a matrix for each frame. We need to cluster this data to determine behavior clusters. For this purpose, agglomerative clustering is used.

Agglomerative clustering is a type of hierarchical clustering technique [54]. In hierarchical clustering techniques, clusters are created by merging statistically nearest clusters in each iteration and a hierarchical representation is produced.

At first, each data is a cluster by itself. At each iteration, pairwise distances between clusters are calculated using a distance metric and nearest clusters are merged to form a new cluster. Iterations stop when it reaches a predetermined threshold for certain number of clusters. In produced hierarchical representation which is defined as “dendrogram tree”, at the bottom each cluster is a single data point and at the top there is only one cluster containing all the data. A visual representation of the algorithm for 6 data points can be seen in Figure 4.

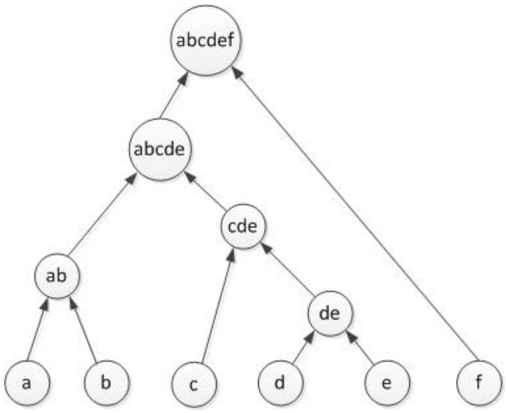


Figure 4. Visual representation of agglomerative clustering

There are 3 parameters for agglomerative clustering. The first is the distance method used to calculate the distances between clusters. There are many methods for computing distances between clusters, some of which are: weighted average, unweighted average, centroid, furthest, weighted center of mass, shortest, Ward’s method. After experimenting with all these methods, Ward’s method has been selected since it gives the best clustering result [55]. Many of them cannot cluster the data properly since the data is not appropriate for that algorithm and some of them give incoherent results.

Originally, Ward’s method is a criterion for hierarchical clustering suggesting that any objective function that is appropriate to calculate cluster distance can be selected to merge clusters. The most used function for this algorithm is error sum of squares function and this algorithm known as Ward’s minimum variance method. In this method, squared Euclidean distances are calculated between each cluster (9) and the clusters having the lowest value are merged. Basically the two clusters which cause the minimum variance are combined in each iteration, so statistically the nearest two clusters are combined

$$d_{ij} = d(\{X_i\},\{X_j\}) = \left\|X_i - X_j\right\|^2 \tag{9}$$

The second parameter is distance metric. Although there are many different distance metrics like Minkowski, Mahalanobis, Hamming etc. , we have to choose Euclidean distance since Ward’s minimum variance method needs Euclidean distance values to calculate inner squared distances.

The third parameter is the number of clusters. To determine number of cluster we need a threshold value to stop the iterations. This value can be either given manually or calculated automatically by giving number of clusters. After clustering process ends, a dendrogram tree is generated. An example of dendrogram tree can be seen in Figure *5*. x axis is cluster numbers, y axis is distances between clusters and the graph shows the merging points of clusters. From this dendrogram tree, a distance threshold can be set to stop merging. The threshold value can be extracted from the graph or iterations can be intercepted when number of clusters is reached.

In the implementation, FTLE data is used for clustering. FTLE data generates a value for all points in all frames. The output matrix of FTLE is input matrix of clustering with particles as variables and frames as observations. Result of the clustering process is a matrix with a cluster label per particle. Results for different number of clusters are analyzed in the Results section.

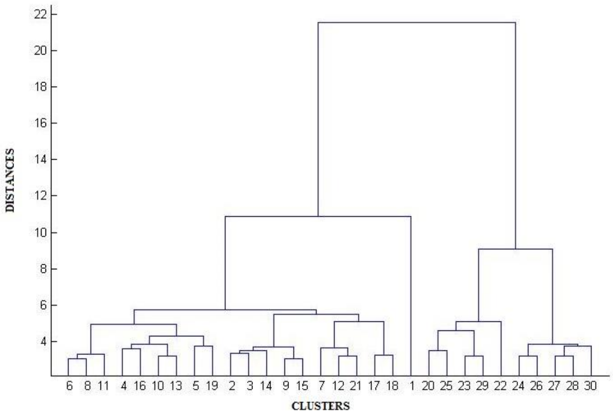


Figure 5. An example of dendrogram tree

Clusters should be divided into subclusters for higher number of cluster count since behaviors are combinations of other behaviors. Other clustering techniques (like K-Means) do not divide clusters into subclusters, instead they compute the clustering process from the beginning and give different results for each different cluster count. Agglomerative clustering produces a dendrogram tree (which is also useful for analyzing cluster distances and merge points) and divide clusters into subclusters for higher number of clusters so clustering process is computed for just one time before dividing clusters. By this way, time cost is decreased significantly and it is easier to analyze subclusters.

3.5. Step 4: Detecting Anomalies

This is the decision step to detect if a particular cluster is abnormal or not. The flowchart for this step is given in Figure 6.

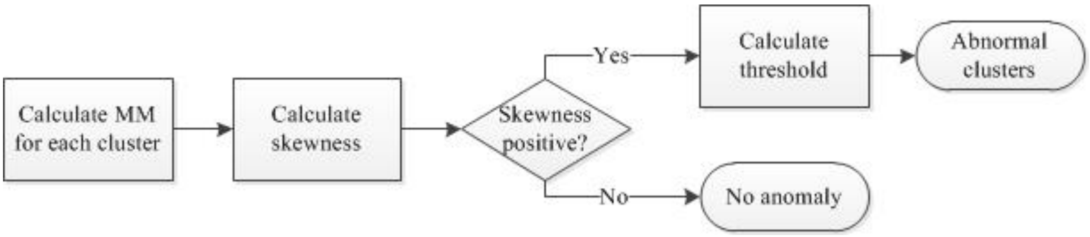


Figure 6. Flowchart of anomaly detection process

Since we have cluster labels, we can analyze FTLE values for each cluster. First, maximum FTLE values for each point are found and a matrix containing maximum FTLE values for each point is generated. Then mean of these maximum values are calculated for each cluster with the help of cluster labels. Resulting in a Mean-Maximum value (MM) for each cluster. The cluster having the minimum MM value is assumed to belong to the background and this cluster is excluded from the further processing steps.

Before detecting abnormal clusters, we should determine if the video has any anomaly. If there is no anomaly in the video, all of the clusters are normal clusters. To determine if there is an anomaly, we used skewness test on all MM values of clusters excluding the cluster having the minimum MM value which is labeled as background.

Skewness is the asymmetry value of a distribution [56]. The skewness value can be negative, zero, positive or even undefined. It is calculated with the help of mean and standard deviation as in the formula (10). Negative skewness value means the left tail of distribution is longer and distribution is skewed to the left. Positive skewness value means the right tail of distribution is longer and distribution is skewed to the right. Illustration of negative and positive skew is given in Figure 7.

$$\gamma_1 = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] = \frac{\mu_3}{\sigma^3} = \frac{E[(X-\mu)^3]}{(E[(X-\mu)^2])^{3/2}} = \frac{K_3}{K_2^{3/2}} \tag{10}$$

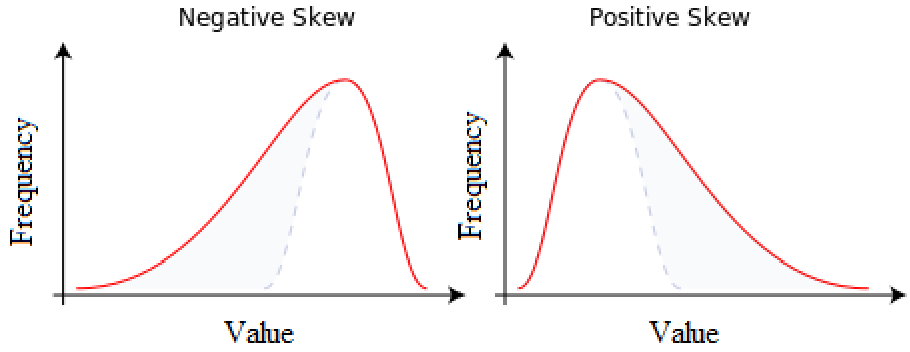


Figure 7. Illustration of negative and positive skewness [56]

In our case, negative skewness means MM values of clusters are normally distributed and there is no outlier or abnormal cluster so video does not contain anomaly. Positive skewness means there are one or more clusters not coherent with the distribution and the video is likely to have an anomaly.

If there is positive skewness, we need to detect abnormal clusters so we need a threshold for MM values. First of all, since all videos have different MM values, this value needs to be video specific. Secondly, as it is not desirable to manually determine a threshold for each video, we need an automatic method. For this purpose, we use Equal Width Thresholding (EWT). Equal Width Thresholding is a modified version of Equal Width Interval Binning [57] which is an unsupervised discretization method to discretize data into equal width bins. In highly positive skewed data sets, the anomaly points increase the range of data set and lie in distant from other data points. So, if we discretize the MM values into 2 bins equally, the mid-point can be a threshold value where abnormal clusters are expected to fall into the right tail of the distribution (second bin). Threshold value is calculated using (11) where max and min are the maximum and minimum of MM values, respectively. The clusters with higher MM values than this threshold are labeled as abnormal clusters.

$$Th = \frac{max - min}{2} + min \tag{11}$$

CHAPTER 4

EXPERIMENTAL RESULTS AND COMPARISONS

4.1. Overview

In this part, detailed information about datasets is given, visualization of the outputs of intermediary steps of the method are provided, and test results for different videos are demonstrated.

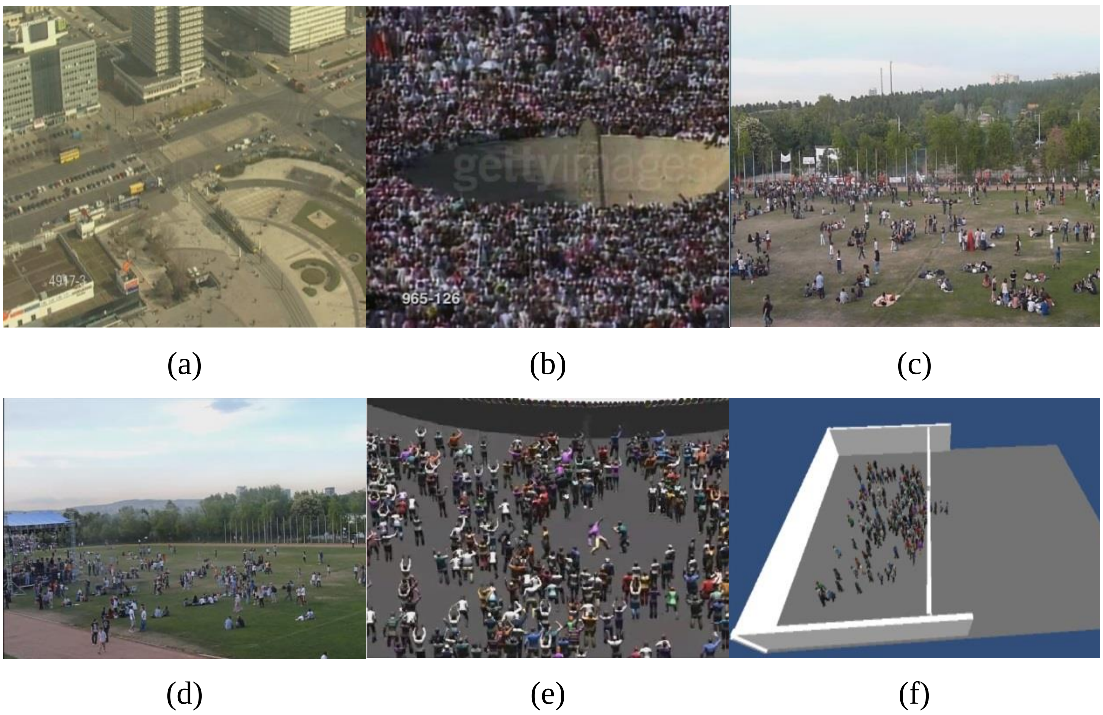
4.2. Datasets

As mentioned before, the method in this paper aims analysis of high density crowd scenes captured using stationary cameras. Low density crowds or narrow angle captures are out of scope and for such scenes detection and tracking based approaches are expected to be more suitable. Also moving or unstable camera captures are not considered since this type of videos need preprocessing. Background detection and localization of clusters are important steps of this method and it is not possible to implement these steps without preprocessing the video for these type of captures.

With these limitations, since there are very few publicly available videos, we had to create our own dataset to use for this method. To test the method we use

- Publicly available crowd datasets on the internet,
- Our video captures during the spring festival of the university,
- Our simulation videos which are hard to capture properly in real life.

Example images from datasets are given in Figure 8. In Figure 8, sample images (a) and (b) are from publicly available datasets [24], (c) and (d) are from our captures in METU Stadium with different angles, (e) and (f) are from some simulation videos generated in the context of this thesis.



(a) (b) (c)

(d) (e) (f)

Figure 8. Example frames from the test videos.

4.2.1. Available datasets

We use Crowd Flow Segmentation Dataset to test and compare the proposed method with other methods which also used the same videos [24]. This dataset contains videos of crowds and other high density moving objects. The videos were collected mainly from the BBC Motion Gallery and Getty Images website.

UMN dataset contains videos mostly having panic situations with ground truth labels for normal and abnormal frames [58]. However, there is no high density crowd so this dataset is not suitable to test our method.

UCSD Anomaly Detection Dataset [59] contains videos acquired with a stationary camera mounted at an elevation, overlooking pedestrian walkways and classified into normal training videos and test videos including anomalies of the circulation of non-pedestrian entities (bikers, skaters, small carts, and people walking across a walkway or in the grass that surrounds). Although the density of crowds in the videos varies, they do not contain high density crowds that we can use to test our method and the dataset is mostly used to detect non-pedestrian entities with supervised methods.

4.2.2. Captured Videos

During the spring festival 2013 of METU, we captured videos having a wide variety of scenes (both normal and having anomaly) to use for testing the method. These captures consist of crowd activity before and after concerts in the stadium, shopping areas and some demonstrations captured at wide, medium, narrow angles in scenes having low, medium and high density crowds. Video resolutions are 1280x720 and video durations vary from 5 seconds to 5 minutes.

4.2.3. Simulations

Simulations have been created using Unity 3D Software [60]. Unity is a cross-platform game and simulation development engine with necessary graphical, rendering and scripting tools. It has a realistic physics engine with tools for real-time computation and it eliminates the need for implementing and defining physics rules. For this work, pro version of this software has been used since the pro version contains the artificial intelligence tools for realistic simulations and technical support.

Unity development environment is based on objects and scripts. Objects in Unity are containers which can contain physical objects, characters, environments or a special effect. Objects do not do anything on their own, they need special properties or scripts to act. Objects can be invisible or even empty. Objects can be created, copied, edited or destroyed infinitely. Scripts can be written to describe the behaviors of objects such as movement, rotation, communication or all others mentioned above. Scripts also can be used to change parameters or use functions apart from objects such as resolution, overall speed, recording etc. Scripts can be programmed using C# script, JavaScript or Boo (Python-like syntax object oriented programming language). For this work C# script has been used which has the same basic syntax as C# but contains some extra syntax and functions defined by Unity. C# was selected since it is more common, well documented and easy to get community support.

For simulations, character packages [61] have been used which contains 6 human figures (3 male and 3 female) and 20 different textures for the figures resulting in a total of 120 different characters. Also simulation environments like library, concert stage etc. and other objects like umbrella, stone, backpack etc. are designed manually. While the characters and object creation were programmed to be random, in some situations, this randomness was biased to be have a more realistic simulation. For example, people are expected to carry umbrellas in rainy weathers.

For artificial intelligence and path finding, "Navigation Mesh" technique was used which is a feature in pro version. In this technique, after defining the areas where characters can walk, the real-time artificial intelligence engine handles the behavior of the characters in simulation. Walk speed, minimum distance between characters or character behaviors are programmed via some scripts. Once one character is programmed with artificial intelligence tools, other characters are copied from the original character randomly.

Simulations were recorded at 25 frames per second via Unity functions and videos were created with these frames containing specific behaviors and actions in different angles and densities for short durations. Example images from simulation videos are given in Figure 8 (e) and (f).

4.3. Implementation and Test Results

For illustration of the proposed method, we will use Stadium video captured during the METU Spring Festival throughout this section and explain the steps of the method using this video. Video resolution is 854x480 and it has a total of 700 frames. The video captures almost all of the pitch where a large crowd of people gather and have various activities. People are mostly in groups exhibiting different behavior. Most of the people are behaving normally and involve in activities such as picnicking, chatting, and walking. However, there are 2 groups behaving different than the expected crowd activities and we define their behavior as abnormal. One of the groups is playing Frisbee (throwing a disc shaped object). Although this behavior can be accepted as normal, since it is akin to people throwing foreign objects and incoherent with the rest of the video, it is considered as an anomaly. The other group is a protesting group which is running with red flags entering from the right side of the video and running to the left side. . An example frame from this video can be seen in Figure *9* where the group playing Frisbee is encircled in white and the protesting group (running with red flags) is encircled in red.

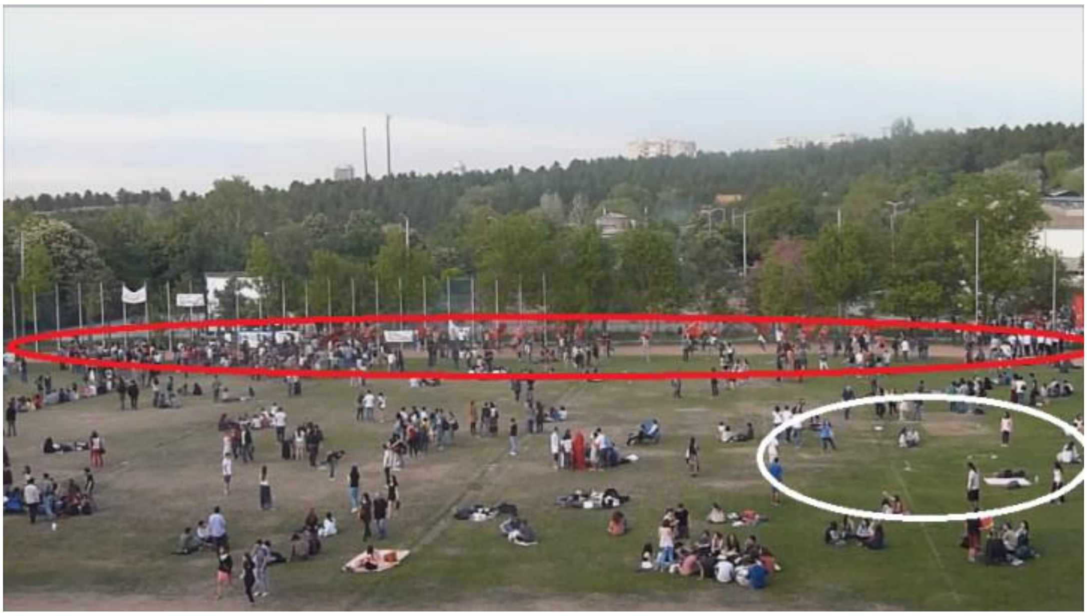


Figure 9. An example frame from test video

The first group plays Frisbee during the video and throws Frisbee at arbitrary intervals throughout the video. The second group starts entering the scene at the third second of the video and members of this group keep running until the end of the video. Sample frames from the beginning, middle and the end of the video can be seen in Figure 10. As we can see in the figure, the first group plays Frisbee at the same position throughout the video and the anomaly position does not change. On the other hand, the protesting group starts running from the right side to the left side until the end so anomaly area changes during the video.

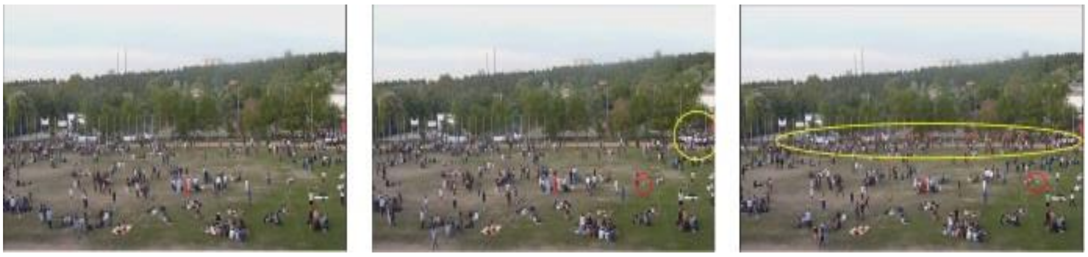


Figure 10. Sample frames from the beginning, middle and end of the test video

In optical flow step, video resolution is decreased to 213x120 and optical flow data is generated. Data is saved to text files with “X_coordinate Y_coordinate X_velocity Y_velocity” format. Each line in the file represents an optical flow output for a coordinate and there is one text file per frame. For this particular test video, 700 text files are generated in this format. An example of the optical flow output can be seen in Figure *11* where red lines indicate the motion direction and magnitude.

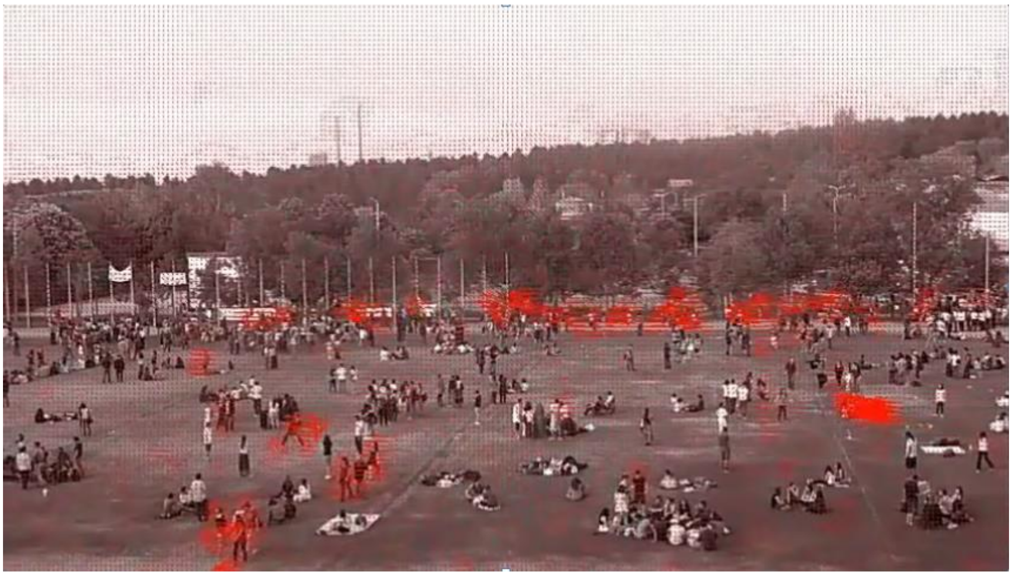
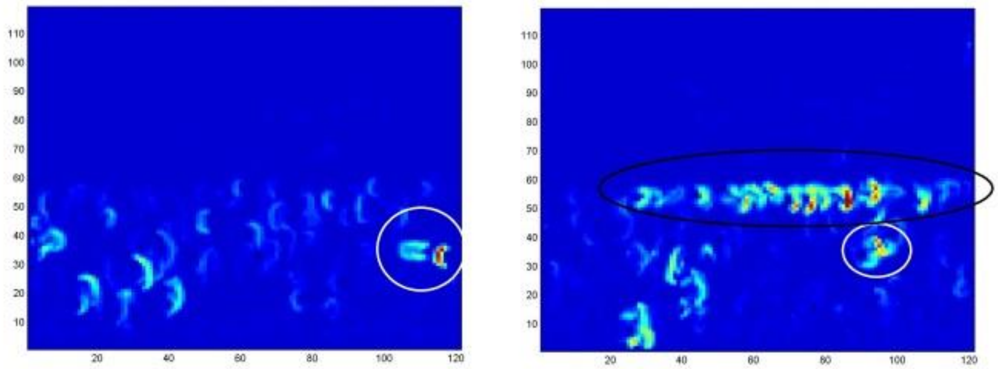


Figure 11. An optical flow output of test video

These text files are then fed into the FTLE implementation. During process, first these files are converted into Matlab files. Then trajectories are calculated by following the particles from the current frame for a number of frames given in integration length parameter. If integration length is set to “10”, particles are followed for 10 frames and a trajectory is generated indicating the path of particle during 10 fames. At the last step, FTLE values are computed from trajectories and an FTLE value for a particle is generated for each frame. As a result, a Matlab file containing a matrix has the same size as the input resolution is generated per input file.

Since the test video has 700 frames and resolution is subsampled to 213x120, the output of FTLE process is 700 matrices with 213x120 size. Two examples of outputs as heat map are given in Figure 12.Color range in heat map goes from blue to red representing the magnitude of FTLE value on that point.



(a) (b)

Figure 12. FTLE outputs of test video

Figure 12(a) is an output from the beginning of the test video. Frisbee playing group is encircled in yellow. Moving Frisbee has the red color in heat map indicating the highest value in the matrix as expected. There is no sign of protesting group since this example is from the beginning of the video and protesting group has not started running yet.

Figure 12(b) is an output from the middle of the test video. Frisbee playing group is encircled in yellow and protesting group is encircled in black. Red and yellow colors in the heat map can be seen in encircled areas since these areas have high speed or incoherent movement. Running people in protesting group and high speed Frisbee movement cause high values in the FTLE output.

The output of FTLE step is used for clustering as input. First, all of output matrices are merged together to generate a general matrix for clustering. Then agglomerative clustering process is started for this matrix with particles as variables and frames as observations. The result of process is a cluster label per particle.

Since FTLE output of test video contains 700 matrices with 213x120 size, merged general matrix has size of 213x120x700. Agglomerative clustering uses 2D matrix as input so this 3D matrix must be converted to 2D. For this purpose, raster ordering is implemented. Raster ordering is basically reading the matrix line-by-line. At the end of each line, the next value is first value of next line. The illustration of process is given in Figure *13*.

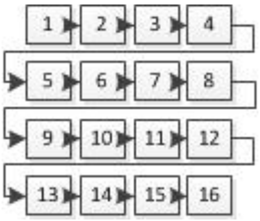


Figure 13. Illustration of raster ordering

After raster ordering, matrix is converted into a 2D matrix of size 25560x700. This matrix is the input matrix of agglomerative clustering with 25560 variables and 700 observations. The result of agglomerative clustering is a dendrogram tree with 25560 clusters at the bottom and 1 cluster at the top. This tree is a visualization of merging steps and distances between clusters. Dendrogram tree for the top 30 clusters is given in Figure *14*.

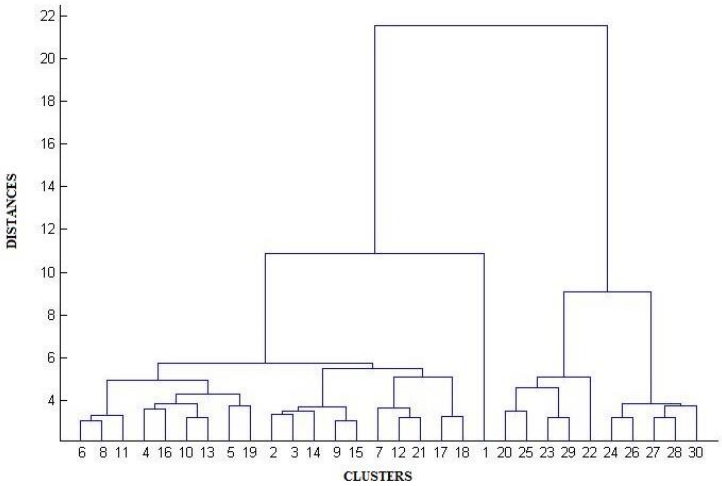
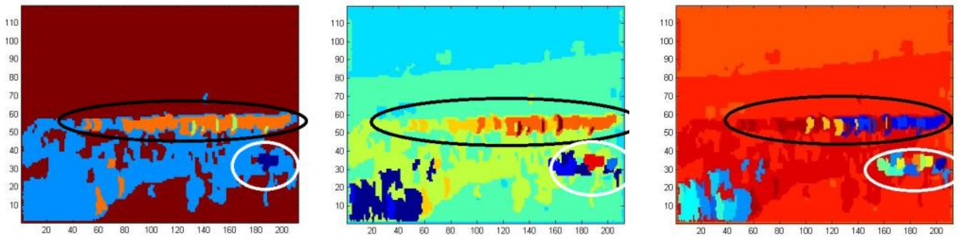


Figure 14. Dendrogram tree of test video

To segment the data into clusters, a distance threshold value is required to stop merging. Since determining a manual threshold for each different video is not desirable, a built-in Matlab function is used which stops merging when process reaches predetermined number of clusters.

Test video is clustered with 5, 10 and 20 cluster numbers. Integration length in FTLE process is set to 1 and 10 to examine different results for different cluster numbers. Integration length 1 is denoted as “i=1” and 10 is denoted as “i=10”.

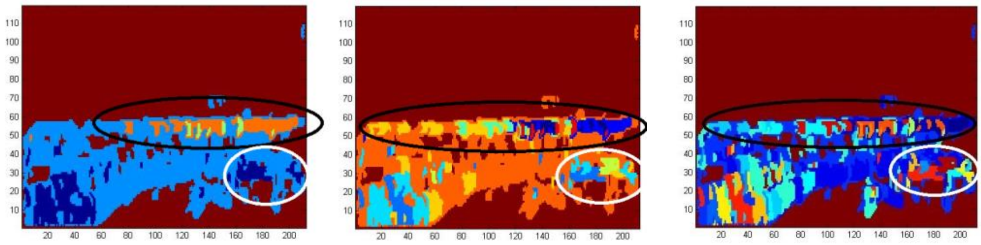
Results of clustering for i=1 are given in Figure *15*. To analyze results, we can look for different behavior areas. For all results, Frisbee playing group is encircled in white and protesting group is encircled in black. For 5 clusters (Figure *15*(a)), Frisbee is clustered as 1 blue cluster and protesting group is clustered as 1 orange cluster. However, since there are not enough clusters for all different behaviors, some other walking people are considered to have the same behavior as protesting group and clustered as orange too. For 10 clusters (Figure *15*(b)), both groups are divided into subclusters since there are different behaviors in both groups. There are people running with different speeds in protesting group and people throwing Frisbee in different styles in other group. There is also another blue cluster at the left-bottom of the image which contains some people going in and out of the video. For 20 clusters (Figure *15*(c)), there are many subclusters for all behaviors. Protesting group has dark blue cluster for highest speed of running people while other clusters contains lower speeds. Other group has many subclusters for different angles of Frisbee movement and different throwing styles. Also area at the left-bottom has subclusters for different walking groups in this area. For all three cluster numbers, protesting group is clustered from the right side to the middle of the image while protesting group reaches to the left side at the end of the video. The background is identified successfully. However for 10 and 20 clusters, background is divided into 2 clusters for areas having no motion and very little motion.



(a) (b) (c)

Figure 15. Clustering results of test video using i=1 for (a)5 (b)10 (c)20 clusters

Figure *16* shows the results for 5, 10 and 20 clusters with i=10. For 5 clusters (Figure *16*(a)), the Frisbee playing group is detected in the same cluster with the area at the left-bottom which means the group movement is not detected correctly. This is a significant point for analyzing integration length and cluster number parameters. The comparisons for these parameters will be discussed later. Protesting group is clustered with orange color until the midpoint of the video. For 10 clusters (Figure *16*(b)), Frisbee playing group is successfully detected with subclusters. Also protesting group is clustered from right side to the left side of the video for the first time. The left part of protesting group is clustered with dark yellow color like some other walking people since speed of the group is decreasing near to left side. For 20 clusters (Figure *16*(c)), there are different clusters for almost each group of people in the video. It may be confusing for perceptual analysis at first but it is useful for defining every type of behavior in the video specifically. For all cluster numbers, background is detected successfully as one big cluster.



(a) (b) (c)

Figure 16. Clustering results of test video for i=10 for (a)5 (b)10 (c)20 clusters

Since we have the same number of cluster results for integration lengths 1 and 10, we can make comparison and analyze the integration length parameter. To analyze the differences for different integration lengths, we should look for the differences between same cluster numbers with different integration lengths. For 5 clusters, Frisbee playing group is detected for 1 integration length while it is misclustered for 10 integration length. For 10 clusters, Frisbee playing group is clustered successfully for both integration lengths while protesting group is clustered until the midpoint of the video for 1 integration length and to the left side of the video for 10 integration length. For 20 clusters, the 3 main areas (Frisbee playing group, protesting group and the area at the left-bottom) is divided into subclusters for 1 integration length while almost each behavior is clustered specifically for 10 integration length.

With these observations, we can say that Frisbee playing group is clustered better with integration length set to 1 as expected since it only considers 2 consecutive frames for generating trajectories and the FTLE calculations are based on a single frame. Frisbee movement is a short time high speed movement and it causes high FTLE values for i=1. Especially for 5 clusters, this group is misclustered for 10 integration length while it is clustered successfully for 1 integration length. For 10 and 20 clusters, protesting group is clustered until the midpoint of the video for 1 integration length while it is clustered from right side to the left side of the video for 10 integration length. For 20 clusters, this group is divided more into subclusters for 10 integration length. Other normal behaviors like walking, picnicking, chatting etc.

are clustered specifically with 10 integration length while they are clustered as one big cluster with 1 integration length. With these observations we can say that, protesting group and other normal behaviors clustered better with 10 integration length as expected since with 10 integration length, trajectories are generated for duration of 10 frame by following the movement of a particle for 10 frames. Also background is clustered as 1 cluster with 10 integration length while it is divided into 2 clusters as no motion and very less motion areas with 1 integration length.

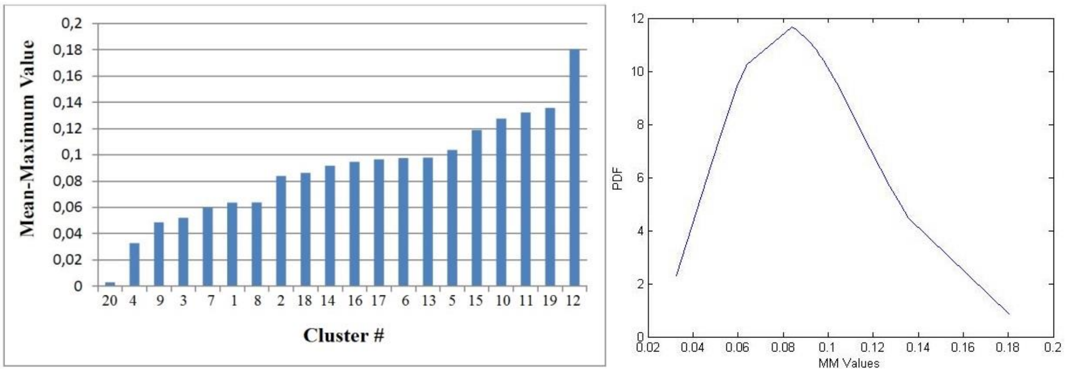
As a result, we can say that, 10 integration length is better for longer duration moderate movements while 1 integration length is better for shorter duration more intense movements. Also with the information of background clustering performance, 1 integration length is more noise sensitive than 10 integration length.

The last step is to detect abnormal clusters if there is anomaly in the video. First of all, mean-maximum (MM) values are calculated to have a characteristic value for each cluster. Since MM values are calculated with maximum speed of every particle in a cluster, these values are indications of characteristic movement speeds of the clusters. MM values for each cluster for the test video, with integration length parameter 10, are tabulated in the Table *1* (the results are sorted in ascending order of MM values).

Table 1.Cluster number, cluster size and MM values after clustering of test video for i=10.

Cluster #	Cluster Size (Pixels)	Mean-Maximum	Cluster #	Cluster Size (Pixels)	Mean-Maximum
20	16382	0.0029	16	146	0.0946
4	2974	0.0328	17	507	0.0965
9	1058	0.0485	6	131	0.0975
3	1548	0.0520	13	263	0.0978
7	507	0.0597	5	146	0.1037
1	276	0.0636	15	46	0.1188
8	492	0.0637	10	20	0.1275
2	66	0.0840	11	36	0.1323
18	184	0.0862	19	126	0.1358
14	317	0.0918	12	23	0.1805

As we can see in the table, cluster #20 has the minimum MM value so this cluster is assumed to be the background and excluded from the remaining operations. This cluster can be seen in the Figure *16* as dark red cluster which is the background cluster of the video. This is the correction of the assumption that the cluster with the minimum MM value is the background cluster.



(a) (b)

Figure 17.(a)Graph of ,(b)PDF of MM values of test video for i=10.

All MM values can be seen in the graph given in Figure 17(a). First of all, the background cluster #20 has an MM value near 0. In all videos, MM values of background clusters are near 0 since almost there isn't any movement in the background. This value is ignored in the calculations because this is an outlier as expected and may change the skewness value and threshold so it may harm the detection of the anomaly. Other outliers except background like cluster #12 cannot be ignored since these are abnormal clusters and significant parts of the detection process.

There are 2 jump points in the graph. First is between clusters 8 and 2. This jump separates the clusters having very little motion speed and normal motion speed. The second jump is between clusters 5 and 15. This jump separates the clusters having normal motion speed and high motion speed. Since high motion speed is incoherent to the general behavior of the scene, second jump is expected to become the threshold to detect abnormal clusters.

To determine if there is any anomaly in the video, skewness value is calculated using the remaining 19 clusters. The skewness value of all clusters excluding the cluster #20 is 0.5391. The value is positive indicating that there is anomaly in the video as expected. Positive skewness can be seen in Figure 17(b). Right tail of distribution is longer as mentioned before.

To detect abnormal clusters, an adaptive threshold is calculated using Formula 11. Maximum MM value is 0.1805 (cluster #12) and minimum MM value is 0.0328 (cluster #4). From the Formula 11;

$$Th = \frac{0.1805 - 0.0328}{2} + 0.0328 = 0.1066$$

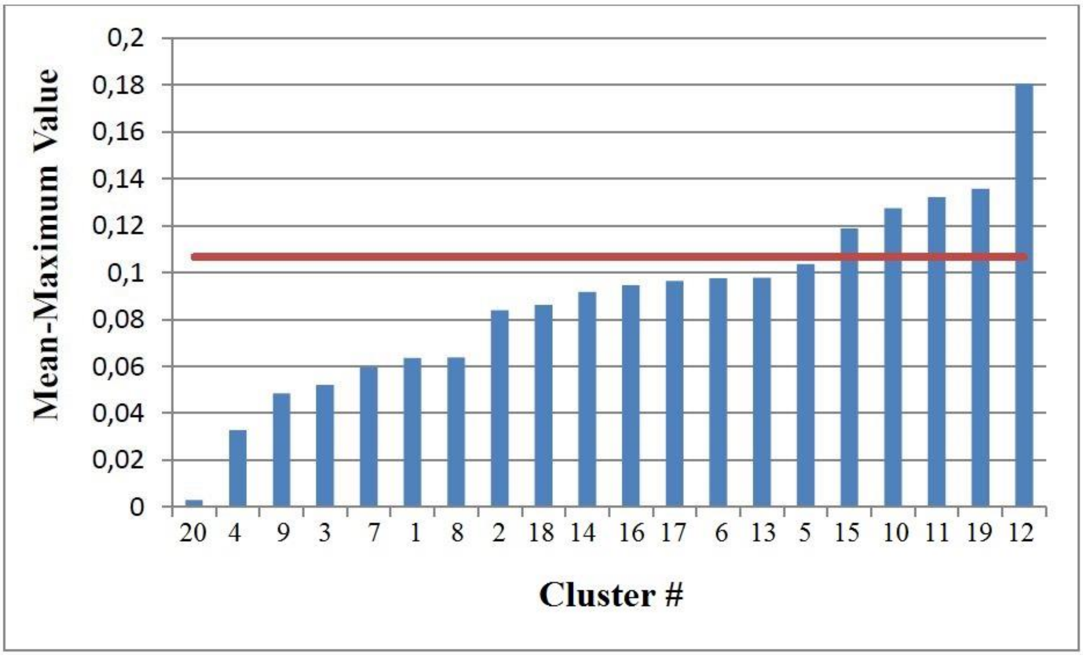


Figure 18. Graph of MM values and threshold line of test video for i=10.

A graph containing all MM values and the threshold value is given in Figure *18*. The threshold value is marked as a horizontal red line. As we can see in the graph, threshold value is on the second jump between clusters 5 and 15 as expected. Clusters with higher MM values than threshold value are abnormal clusters. Abnormal clusters can be seen in Figure *19*.

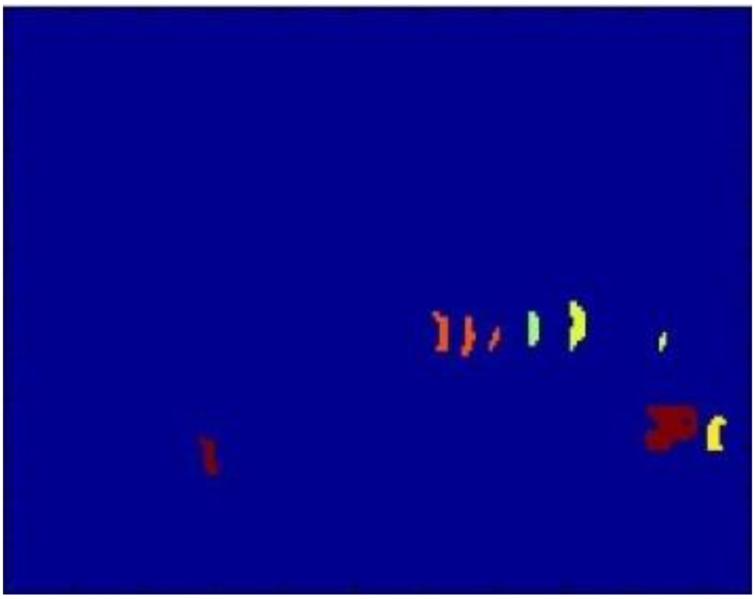
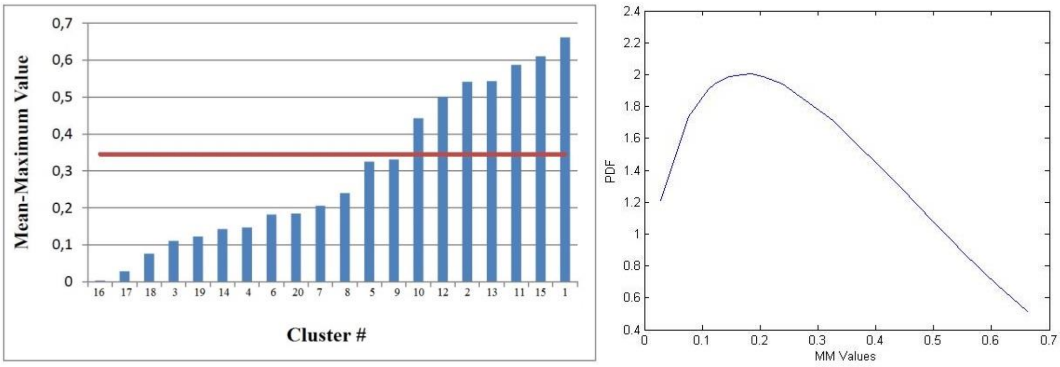


Figure 19. Abnormal clusters of test video for i=10.

Frisbee playing group is detected as abnormal successfully. Protesting group is divided into subclusters since there are some people watching the group and blocking the motion in some parts. The cluster at the bottom left belongs to another Frisbee playing group, however this group throws Frisbee only one time during the video. As a result, for test video with 20 clusters and 10 integration length, the anomaly detection and localization is achieved.
For the same video using the same process with 1 integration length, calculated skewness value is 0.3605. The skewness value is still positive indicating that there is anomaly in the video. MM values and for each cluster are tabulated in the
Table *2* with ascending order. Also a graphical representation of MM values and threshold value is given in Figure 20(a) and positive skewness can be seen in PDF graph given in Figure 20(b). As we can see in the graph, threshold value is on the jump between cluster 9 and 10.

Table 2. Cluster number, cluster size and MM values after clustering of test video for i=1.

Cluster #	Cluster Size (Pixels)	Mean-Maximum	Cluster #	Cluster Size (Pixels)	Mean-Maximum
16	7387	0.0033	8	234	0.2397
17	10167	0.0283	5	119	0.3253
18	4918	0.0759	9	164	0.3309
3	298	0.1100	10	70	0.4428
19	522	0.1220	12	33	0.5002
14	135	0.1426	2	31	0.5411
4	157	0.1468	13	34	0.5430
6	615	0.1821	11	37	0.5874
20	70	0.1849	15	42	0.6105
7	200	0.2054	1	15	0.6617



(a) (b)

Figure 20. (a)Graph and threshold line, (b)PDF of MM values of test video for i=1.

Clustering results can be seen for 20 clusters in Figure 15(c) and abnormal clusters are given in Figure *21*. The main difference is that, protesting group cannot be detected as abnormal. As mentioned before, since integration length is only 1, particle trajectories are created for 2 consecutive frames so long time movements, like running protesters, are ignored. As a result we can only see the Frisbee playing group as abnormal clusters.

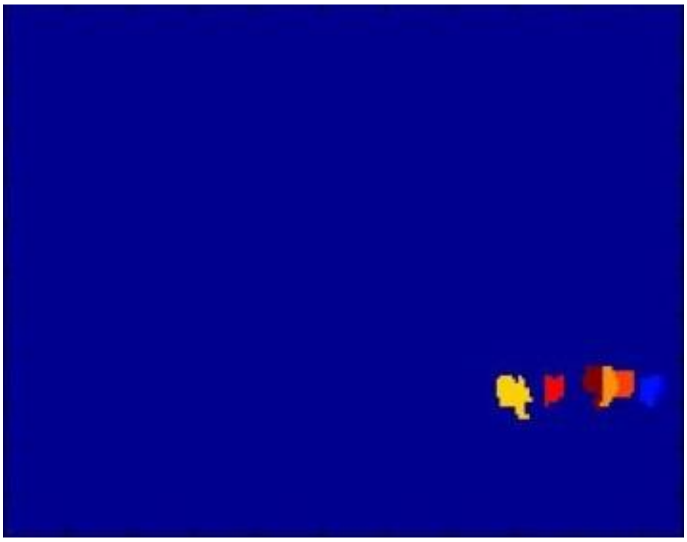


Figure 21. Abnormal clusters of test video for i=1.

As mentioned before, higher integration length is better for longer duration anomalies (such as people demonstrating), a lower integration length is better to detect shorter duration or high activity anomalies (such as fighting, fainting). Selection of this parameter is application dependent. Different integration lengths can be selected to detect different types of behaviors better, however more than one process with different integration lengths can be run concurrently to detect both longer and shorter term anomalies.

If the total cluster count is set to 10, skewness value for the test video is negative. This is because 10 clusters are not enough to characterize and cluster all different behaviors in the video and in this case abnormal clusters are combined with normal clusters. If total cluster count is set to 30, for both integration lengths 1 and 10, skewness value is positive and visual results are the same as the 20 cluster case. As the number of clusters is higher than the number of different behaviors in the scene, abnormal clusters are further divided into several clusters. As the adaptive threshold also changes the same clusters are labeled as abnormal. As a result, 20 clusters are sufficient for this process.

For all of the videos, 20 cluster number is used since the threshold is adaptive and 20 clusters are enough to cluster each behavior in the scene. Also the methods for determining cluster number or validation of clusters are not used since there are too many variables and observations such that these methods don't work or takes too much time. For example, one of the most common methods for cluster validation, silhouette values of all variables are negative indicating that all variables should be in one cluster which is meaningless. Silhouette is known as inefficient for high number of variables and not suitable in this case. The visualization of the clusters is useful to see different behaviors and a perceptual analysis of the clusters is enough to validate the clustering since there is not any method to validate clusters if it is coherent with the behaviors in the video. Experimental results are satisfactory for 20 cluster number however a method for automatic selection of number of clusters could be devised as a future work.

Cluster size is presented for detailed visual information but not used in any calculation since cluster size may vary with different factors. Although background cluster is mostly the biggest cluster, this assumption can not be used to determine background cluster just based on cluster size. Different types of behaviors and video resolutions may cause different cluster sizes so cluster size can not be used in calculations for now. However, analysis of cluster sizes and cluster evaluation may give information about behavior types which is considered as a future work.

To illustrate the method using another video with anomaly, we use a simulation video named "Fight". In this video, there is a high density crowd dancing with the music in a concert. After a while, 2 people start fighting near the center of the crowd. After the breakout of fighting, people near the fight form a circle and start watching the fight. Resolution of the video is 854x480 and it has a total of 450 frames. An example frame from this video can be seen in the Figure 8(e).

The same process is employed for this video with 20 clusters. For both 1 and 10 integration lengths, the skewness value is positive indicating there is anomaly in the video as expected. The visual results for abnormal clusters can be seen in Figure *22* (a) and (b) for integration lengths 1 and 10 respectively.

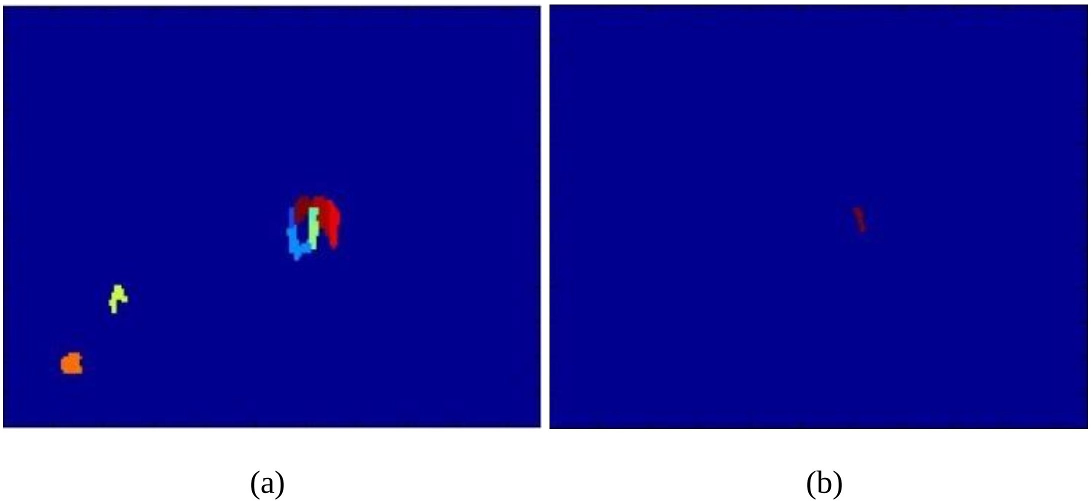
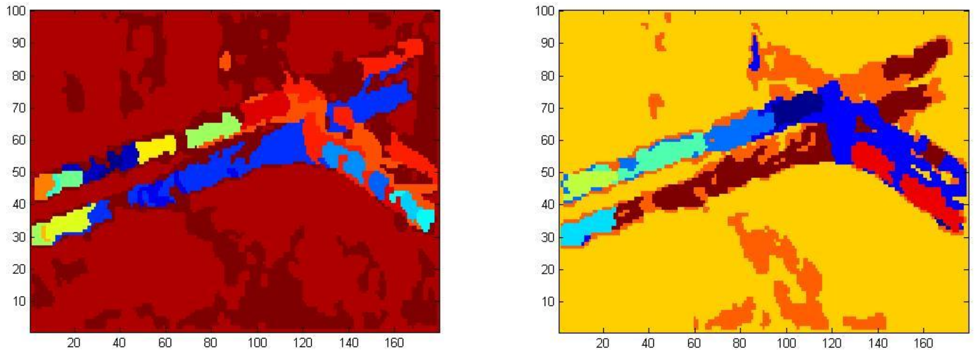


Figure 22. Abnormal clusters for the fight video (a) for i=1, (b) for i=10.

As seen in Figure *22*(a), for i=1, the fighting area in the center is marked as abnormal correctly, however 2 other clusters at the bottom-left corner are mislabeled as abnormal. 1 integration length considers movements for very short time (1 frame) so it is sensitive to noise. The 2 mislabeled clusters may be a high speed dance move or another short time high speed gesture such that this movement is incoherent with the scene and labeled as abnormal. For 10 integration length, the area of fight is labeled as abnormal correctly. In this case, the marked anomaly area is smaller and it corresponds to the location where the fighting is observed for a longer time.

To test a normal video, a surveillance camera capture from “Crowd Segmentation Dataset” is used. Video resolution is 854x480 and has 1750 frames. An example frame can be seen in the Figure 8(a). The video contains a normal crossroad traffic captured from a high angle. Using the same process with 20 clusters, for both 1 and

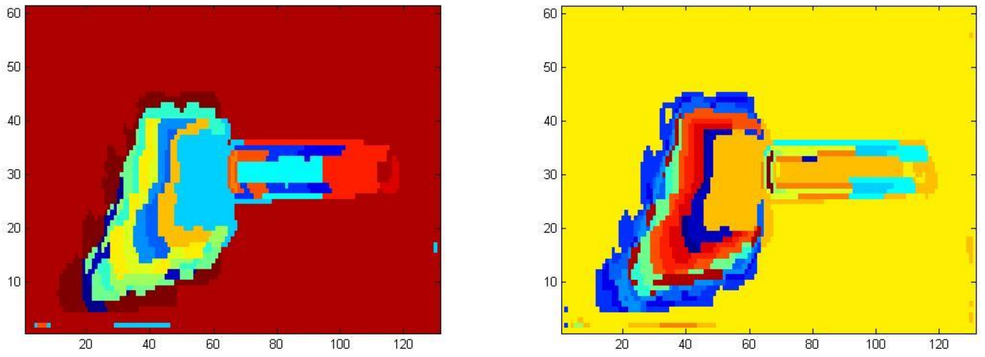
10 integration lengths, calculated skewness value is negative indicating there is no anomaly in the video. The clustering results can be seen in Figure 23 (a) and (b) for integration lengths 1 and 10 respectively.



(a) (b)

Figure 23.Clustering results of City Traffic video for 20 clusters using integration length (a) 1 and (b) 10.

For testing a normal simulation video, we created a normal scene where people try to go through a door one by one then keep walking on the right side. Video resolution is 528x250 and has 800 frames. An example frame can be seen in the Figure 8(f). Skewness value is negative for both 1 and 10 integration lengths and 20 clusters as expected. The clustering results can be seen in Figure 24 (a) and (b) for integration lengths 1 and 10 respectively.



(a) (b)

Figure 24. Clustering results of Door(S) video for 20 clusters using integration length (a) 1 and (b) 10.

4.4. All Test Results

We have used a total of 12 videos, 6 of which having anomalies in our tests. The remaining 6 videos have usual people activity. Skewness values for the test videos calculated for 20 clusters and for integration lengths of 1 and 10 (denoted as “i=1” and “i=10”) are tabulated in Table 3 along with their ID#. Simulation videos from our dataset are marked with “(S)” and Stadium videos (ID#5, 7 and 9) are from our capture dataset. The remaining videos are from the dataset presented in [24].

Table 3. Results of all test videos.

ID#	Name	Anomaly	Skewness	
			i=1	*i*=10
1	City Traffic	No	-0.6265	-1.0938
2	Door (S)	No	-0.6365	-0.3864
3	Marathon	No	-1.0310	-0.7314
4	Bazaar	No	-0.0121	-0.4151
5	Stadium 3	No	-0.4189	-0.0682
6	Station	No	-1.4031	-0.8049
7	Stadium	Yes	**0.3605**	**0.5391**
8	Fight (S)	Yes	**0.1427**	**1.6535**
9	Stadium 2	Yes	**1.2625**	-0.4148
10	Stoning	Yes	**1.7487**	**0.4046**
11	Hall	Yes	**0.5791**	**0.2223**
12	Crossroad	Yes	**0.7595**	**0.0882**

For videos having no anomaly (ID#1 to 6), the skewness values are calculated as negative for both i=1 and 10. The algorithm does not detect any abnormal clusters since skewness value is not positive and the videos are labeled, correctly, as normal.

The video Stadium 2 has similar characteristics to Stadium (ID#7) as this was captured at the same location with a different angle and only some people throwing flying disc can be seen for a short time (short bursts of intense activity) while demonstrating group is not visible. For this video the skewness value for i=1 is positive, correctly detecting this anomaly. On the other hand, skewness value for i=10 is negative as there is no longer duration anomaly.

Stoning video is a capture from a religious ritual in Mecca where people throw stones to a wall. Since people are throwing objects for a long time both skewness values are positive indicating an anomaly.

Hall video is also a capture from Mecca where pilgrims walk in a building, however some pilgrims in the center of the video are running so this area is clustered and labeled as anomaly.

Crossroad video is a capture of people crossing a road. In this video while most people are walking, a few people are running at a faster pace. Skewness values are calculated positive, indicating an anomaly. Examination of abnormal clusters shows that the detected anomaly locations are areas where people run. While this is unlikely to be of interest to a human operator, the system detects running people as anomaly as they exhibit different dynamics than the rest of the scene.

3 of the videos having anomaly (Stadium, Fight, Stadium 2) are from our dataset. The other 3 (Stoning, Hall, Crossroad) are from dataset [24]. Existing studies and experiments on this dataset is mostly about flow segmentation since there is no ground truth and labeling for anomaly detection. The other datasets which are used for anomaly detection do not contain high density crowds and not suitable for this work as explained in Chapter 4.2.1. The studies aiming anomaly localization mostly work on low/medium density crowds and aim to detect non pedestrian entities or people in undesired locations as mentioned in Literature Review. As a result, a comparison with existing methods is not possible for now.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

In this thesis, an unsupervised method for anomaly detection and localization in high density crowds using FTLE is presented. The method does not require any user defined rules and all the steps are unsupervised so no training is required. The method first extracts motion data and computes FTLE values for global motion information. Then this data is clustered to find behavioral clusters and inconsistent behaviors are detected as anomaly. The threshold is calculated using cluster data so it is video specific.

The integration time parameter can be configured to obtain anomalies with different characteristics. While shorter integration time allows detection of short bursts of intense activity (such as fighting, throwing objects), longer integration time allows detection of relatively less intense-more persistent activities (such as demonstrations). While it is possible to run the system with multiple integration times simultaneously to detect different type of activities, this brings more computational complexity.

The main contribution of the work is to cluster the crowd into behavioral groups. The promising results show that, this approach has a potential in crowd behavior analysis. The proposed method not only detects the anomaly but it also allows localization of anomalies.

In the future, the proposed method could be experimented with different type of anomalies. This requires videos having different type of anomalies or better simulations. Lack of videos and datasets for crowd analysis is commonly mentioned in the existing studies in this area, so this work involves capturing new data and generating realistic simulation videos.

Since we have the behavioral clusters, analyzing the cluster evaluation in time in terms of position and shape may result in better results and can lead to different approaches. Also newly formed and disappearing clusters may be analyzed in the future as this information may give clues for anomalies. To be able to analyze this, the method needs to be modified to have adaptive cluster count and an automated method for determining cluster count needs to be implemented.

Another potential future work is implementation of the system to work in real-time. Parallelization of the algorithms and utilization of Graphics Processing Units (GPU) may increase the speed of the method to allow real-time working.

REFERENCES

[1] The Guardian, "Hundreds die in pilgrimage crush," 26 Jan. 2005. [Online]. Available: http://www.theguardian.com/world/2005/jan/26/india.randeepramesh.

[2] The Sidney Morning Herald, "Deadly Mecca crush blamed on bridge bottleneck," 14 Jan. 2006. [Online]. Available: http://www.smh.com.au/news/world/deadly-mecca-crush-blamed-on-bridge-bottleneck/2006/01/13/1137118970154.html.

[3] "Oxford Dictionaries," Oxford University Press, [Online]. Available: http://www.oxforddictionaries.com/.

[4] J. C. S. J. Junior, S. R. Musse and C. R. Jung, "Crowd Analysis Using Computer Vision Techniques," *Signal Processing,* pp. 66-77, 2010.

[5] B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin and L.-Q. Xu, "Crowd analysis: A survey," *Machine Vision and Applications,* vol. 19, no. 2, pp. 345-357, 2008.

[6] J. C. S. J. Jr., A. Braun, J. Soldera, S. R. Musse and C. R. Jung, "Understanding people motion in video sequences using Voronoi diagrams," *Pattern Anal. Applic.,* pp. 321-332, 2007.

[7] J. S. Marques, P. M. Jorge, A. J. Abrantes and J. M. Lemos, "Tracking Groups of Pedestrians in Video Sequences," *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW '03. Conference on,* vol. 9, p. 101, 16-22 June 2003.

[8] Y. Zhang, W. Ge, M.-C. Chang and X. Liu, "Group context learning for event recognition," *Proceedings of the 2012 IEEE Workshop on the Applications of Computer Vision,* pp. 249-255, 2012.

[9] B. A. Boghossian and S. A. Velastin, "Motion-based machine vision techniques for the management of large crowds," *Electronics, Circuits and Systems, 1999. Proceedings of ICECS '99. The 6th IEEE International Conference on,* vol. 2, pp. 961-964, 1999.

[10] A. C. Davies, J. H. Yin and S. A. Velastin, "Crowd monitoring using image processing," *Electronics & Communication Engineering Journal,* vol. 7, no. 1, pp. 37-47, Feb. 1995.

[11] P. Reisman, O. Mano, S. Avidan and A. Shashua, "Crowd detection in video sequences," *IEEE Intelligent Vehicles Symposium,* pp. 66-71, 2004.

[12] S. Bouchafa, D. Aubert and S. Bouzar, "Crowd motion estimation and motionless detection in subway corridors by image processing," *IEEE Conference on Intelligent Transportation,* pp. 332-337, 1997.

[13] S. Bouchafa, D. Aubert, L. Beheim and A. Sadji, "Automatic Counterflow Detection in Subway Corridors by Image Processing," *Journal of Intelligent Transportation Systems,* vol. 6, no. 2, pp. 97-123, 2001.

[14] A. M. Cheriyadat and R. J. Radke, "Detecting Dominant Motions in Dense Crowds," *IEEE Journal of Selected Topics in Signal Processing,* vol. 2, no. 4, pp. 568-581, August 2008.

[15] V. Rabaud and S. Belongie, "Counting Crowded Moving Objects," in *IEEE Conference on Computer Vision and Pattern Recognition* , June 2006.

[16] G. J. Brostow and R. Cipolla, "Unsupervised Bayesian Detection of Independent Motion in Crowds," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* Washington DC, 2006.

[17] B. Solmaz, B. Moore and M. Shah, "Identifying Behaviors in Crowd Scenes Using Stability Analysis for Dynamical Systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI),* 2012.

[18] M. R. Planinc, "Modeling Sources and Sinks in Crowded Scenes by Clustering Trajectory Points Obtained by Video-based Particle Advection," Wien, 09 June 2010.

[19] T. Zhao and R. Nevatia, "Bayesian human segmentation in crowded situations," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* vol. 2, pp. 459-466, 18-20 June 2003.

[20] A. Chan, M. Morrow and a. N. Vasconcelos, "Analysis of crowded scenes using holistic properties," *Proceedings of the 11th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance,* June 2009.

[21] H. Song, X. Liu, X. Zhang and J. Hu, "Real-Time Monitoring for Crowd Counting Using Video Surveillance and GIS," *2nd International Conference on Remote Sensing, Environment and Transportation Engineering (RSETE),* pp. 1-4, 2012.

[22] M. Boninsegna, T. Coianiz and E. Trentin, "Estimating the Crowding Level with a Neuro-Fuzzy Classifier," *Journal of Electronic Imaging,* vol. 6, no. 3, 1997.

[23] E. L. Andrade, S. Blunsden and R. B. Fisher, "Modelling Crowd Scenes for Event Detection," *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on,* pp. 175-178.

[24] S. Ali and M. Shah, "A Lagrangian Particle Dynamics Approach for Crowd

Flow Segmentation and Stability Analysis," *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on,* pp. 1-6, 17-22 June 2007.

[25] N. Ihaddadene and C. Djeraba, "Real-time Crowd Motion Analysis," *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on,* pp. 1-4, 8-11 Dec. 2008.

[26] R. Mehran, A. Oyama and M. Shah, "Abnormal Crowd Behavior Detection using Social Force Model," *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* , pp. 935-942, 20-25 June 2009.

[27] V. Reddy, C. Sanderson and B. C. Lovell, "Improved Anomaly Detection in Crowded Scenes via Cell-based Analysis of Foreground Speed, Size and Texture," *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* pp. 55-61, 2011.

[28] Z. Zhong, M. Yang, S. Wang, W. Ye and Y. Xu, "Energy Methods for Crowd Surveillance," *Proceedings of the 2007 International Conference on Information Acquisition,* pp. 504-510, 9-11 July 2007.

[29] D.-Y. Chen and P.-C. Huang, "Motion-based unusual event detection in human crowds," *Journal of Visual Communication and Image Representation,* vol. 22, no. 2, pp. 178-186, Feb. 2011.

[30] F. Jiang, Y. Wu and A. K. Katsaggelos, "Detecting contextual anomalies of crowd motion in surveillance video," *16th IEEE International Conference on Image Processing (ICIP),* pp. 1117-1120, 7-10 Nov. 2009.

[31] Y. Ke, R. Sukthankar and M. Hebert, "Event Detection in Crowded Videos," *IEEE 11th International Conference on Computer Vision,* pp. 1-8, 2007.

[32] 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),* pp. 1-8, 2007.

[33] L. Kratz and K. Nishino, "Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models," *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on,* pp. 1446-1453, 20-25 June 2009.

[34] B. Wang, M. Ye, X. Li, F. Zhao and J. Ding, "Abnormal crowd behavior detection using high-frequency and spatio-temporal features," *Machine Vision and Applications,* vol. 3, no. 23, pp. 501-511, 2012.

[35] S. Hommes, R. State, A. Zinnen and T. Engel, "Detection of Abnormal Behaviour in a Surveillance Environment Using Control Charts," *8th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS),* pp. 113-118, 2011.

[36] N. Paragios and V. Ramesh, "A MRF-based approach for real-time subway monitoring," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR),* vol. 1, pp. 1034-1040, 2001.

[37] E. L. Andrade, S. Blunsden and R. B. Fisher, "Hidden Markov Models for Optical Flow Analysis in Crowds," *18th International Conference on Pattern Recognition (ICPR),* vol. 1, pp. 460-463, 2006.

[38] X. Wu, G. Liang, K. K. Lee and Y. Xu, "Crowd Density Estimation Using Texture Analysis and Learning," pp. 214-219, 17-20 Dec. 2006.

[39] V. Mahadevan, W. Li, V. Bhalodia and N. Vasconcelos, "Anomaly Detection in Crowded Scenes," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition,* pp. 1975-1981, 2010.

[40] B. Luvison, T. Chateau, P. Sayd, Q.-C. Pham and J.-T. Lapreste, "An Unsupervised Learning based Approach for Unexpected Event Detection," *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications,* pp. 509-513, 2009.

[41] P. Güler, "Automated Crowd Behavior Analysis For Video Surveillance Applications," in *MS. Thesis*, Middle East Technical University, Sept. 2012.

[42] A. Kuhn, T. Senst, I. Keller, T. Sikora and H. Theisel, "A Lagrangian Framework for Video Analytics," *Multimedia Signal Processing (MMSP), 2012 IEEE 14th International Workshop on,* pp. 387-392, 17-19 Sept 2012.

[43] O. P. Popoola and H. Ma, "Detecting Abnormal Behaviors in Crowded Scenes," *Research Journal of Applied Sciences, Engineering and Technology,* pp. 4171-4177, 2012.

[44] S. J. Guy, J. v. d. Berg, W. Liu, R. W. H. Lau, M. C. Lin and D. Manocha, "A statistical similarity measure for aggregate crowd dynamics," *ACM Trans. Graph.,* vol. 31, no. 6, p. 190, 2012.

[45] C. W. Reynolds, "Steering Behaviors For Autonomous Characters," Sony Computer Entertainment America, California, 2004.

[46] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E,* vol. 51, no. 5, p. 4282, May 1995.

[47] J. v. d. Berg, S. J. Guy, M. C. Lin and D. Manocha, "Reciprocal n-body Collision Avoidance," *Proc. of International Symposium on Robotics Research (ISRR),* pp. 3-19, 2009.

[48] E. L. Andrade and R. B. Fisher, "Simulation of Crowd Problems for Computer Vision," *First International Workshop On Crowd Simulation,* 2005.

[49] G. Farneback, "Two-Frame Motion Estimation Based on Polynomial

Expansion," in *13th Scandinavian Conference on Image Analysis*, 2003.

[50] "OpenCV v2.4.2," Intel, 04 July 2012. [Online]. Available: http://opencv.org/.

[51] 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: Nonlinear Phenomena*, vol. 212, no. 3-4, pp. 271-304, 2005.

[52] F. P. Group, "Lagrangian Coherent Structures," Illinois Institute of Technology, 15 04 2005. [Online]. Available: http://mmae.iit.edu/shadden/LCS-tutorial/FTLE-derivation.html. [Accessed 13 07 2014].

[53] B. P. Laboratory, "LCS MATLAB Kit Version 2.3," California Institute of Technology, 5 Nov. 2012. [Online]. Available: http://dabiri.caltech.edu/software.html. [Accessed 13 07 2014].

[54] T. Hastie, R. Tibshirani and J. Friedman, "Hierarchical Clustering," in *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (Second Edition)*, Springer, Feb. 2009, p. 523.

[55] J. H. Ward, "Hierarchical Grouping to Optimize an Objective Function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236-244, 1963.

[56] Wikipedia, "Skewness," 21 July 2014. [Online]. Available: http://en.wikipedia.org/wiki/Skewness.

[57] J. Dougherty, R. Kohavi and M. Sahami, "Supervised and Unsupervised Discretization of Continuous Features," *Machine Learning: Proceedings of the Twelfth International Conference*, pp. 194-202, 1995.

[58] R. a. V. L. Artifical Intelligence, "UMN Dataset," University of Minnesota ITS Institute, [Online]. Available: http://mha.cs.umn.edu/Movies/. [Accessed 13 07 2013].

[59] S. V. C. Laboratory, "UCSD Anomaly Detection Dataset," University California, San Diego, [Online]. Available: http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm. [Accessed 13 07 2014].

[60] U. Technologies, "Unity 3D v4.0.0," [Online]. Available: http://unity3d.com/. [Accessed Nov 2012].

[61] 3DRT, "Male/Female Character Packs," Dec. 2012. [Online]. Available: http://3drt.com/store/characters/realpeople-males.html - http://3drt.com/store/characters/realpeople-females.html.

TEZ FOTOKOPİSİ İZİN FORMU

ENSTİTÜ

Fen Bilimleri Enstitüsü ☒

Sosyal Bilimler Enstitüsü □

Uygulamalı Matematik Enstitüsü □

Enformatik Enstitüsü □

Deniz Bilimleri Enstitüsü □

YAZARIN

Soyadı : Öngün
Adı : Cihan
Bölümü : Elektrik Elektronik Mühendisliği

TEZİN ADI (İngilizce) : ANOMALY DETECTION FOR CROWDED ENVIRONMENT VIDEO SURVEILLANCE APPLICATIONS

TEZİN TÜRÜ : Yüksek Lisans ☒ Doktora □

1. Tezimin tamamından kaynak gösterilmek şartıyla fotokopi alınabilir. ☒
2. Tezimin içindekiler sayfası, özet, indeks sayfalarından ve/veya bir bölümünden kaynak gösterilmek şartıyla fotokopi alınabilir. □
3. Tezimden bir bir (1) yıl süreyle fotokopi alınamaz. □

TEZİN KÜTÜPHANEYE TESLİM TARİHİ: