ANOMALY DETECTION USING SPARSE FEATURES AND SPATIO-TEMPORAL HIDDEN MARKOV MODEL FOR PEDESTRIAN ZONE VIDEO SURVEILLANCE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

AYŞE ELVAN GÜNDÜZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

AUGUST 2014

ANOMALY DETECTION USING SPARSE FEATURES AND SPATIO-TEMPORAL HIDDEN MARKOV MODEL FOR PEDESTRIAN ZONE VIDEO SURVEILLANCE

Submitted by **Ayşe Elvan Gündüz** in partial fulfillment of the requirements for the degree of **Master of Science in Information Systems, Middle East Technical University** by,

Prof. Dr. Nazife Baykal,
Director, **Graduate School of Informatics** _____

Prof. Dr. Yasemin Yardımcı Çetin
Head of Department, **Information Systems** _____

Assist. Prof. Dr. Tuğba Taşkaya Temizel
Supervisor, **Information Systems, METU** _____

Assoc. Prof. Dr. Alptekin Temizel,
Co-supervisor, **Work Based Learning, METU** _____

Examining Committee Members:

Prof. Dr. Yasemin Yardımcı Çetin
Information Systems, METU _____

Assist. Prof. Dr. Tuğba Taşkaya Temizel
Information Systems, METU _____

Assoc. Prof. Dr. Aysu Betin Can,
Information Systems, METU _____

Assist. Prof. Dr. Erhan Eren
Information Systems, METU _____

Assist. Prof. Dr. Sinan Kalkan
Computer Engineering, 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: **Ayşe Elvan Gündüz**

Signature: _____

ABSTRACT

ANOMALY DETECTION USING SPARSE FEATURES AND SPATIO-TEMPORAL HIDDEN MARKOV MODEL FOR PEDESTRIAN ZONE VIDEO SURVEILLANCE

GÜNDÜZ, Ayşe Elvan

M.S., Department of Information Systems

Supervisor: Assist. Prof. Dr. Tuğba Taşkaya Temizel

Co-Supervisor: Assoc. Prof. Dr. Alptekin Temizel

August 2014, 45 Pages

Automated analysis of crowd behavior for anomaly detection has become an important issue to ensure the safety and security of the public spaces. Public spaces have varying people density and as such, algorithms are required to work robustly in low to high density crowds. Mainly, there are two different approaches for analyzing the crowd behavior: methods based on object tracking where individuals in a crowd are tracked and holistic methods where the crowd is analyzed as a whole. In this work, the aim is to detect anomalies in pedestrian zone videos using a holistic approach. The pedestrian zone videos are automatically grouped according to crowd density. The pedestrian motion is modeled as a whole without detecting and tracking the individuals using the features obtained Oriented Fast and Rotated Brief (ORB) feature detector and thus the model is privacy preserving. These features are then represented using Binary Robust Independent Elementary Features (BRIEF) descriptor and a spatiotemporal Hidden Markov Model is used for anomaly detection.

Keywords: Computer Vision, Video Surveillance, Pedestrian Zone Analysis, Anomaly Detection.

ÖZ

SEYREK ÖZNİTELİKLER VE UZAY-ZAMANSAL GİZLİ MARKOV MODELLERİ KULLANILARAK YAYA BÖLGELERİNDE VIDEO GÖZETLEME İÇİN AYKIRILIK TESPİTİ

GÜNDÜZ, Ayşe Elvan

Yüksek Lisans, Bilişim Sistemleri Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Tuğba Taşkaya Temizel

Eş Tez Yöneticisi: Doç. Dr. Alptekin Temizel

Ağustos 2014, 45 sayfa

Halka açık alanların güvenliği için gözetim araçlarının kullanımıyla kalabalık davranışının otomatik analizi önem kazanmıştır. Halka açık alanlarda insan yoğunluğu değişiklik göstermektedir. Bu yüzden ilgili algoritmaların hem düşük hem de yüksek yoğunluklu kalabalıklarda birden düzgün çalışması gerekmektedir. Temelde, kalabalık davranışını inceleyen iki metot mevcuttur: Objelerin takibini temel alan ve kalabalıktaki bireyleri tek tek takip eden yöntemler ve kalabalığı bir bütün olarak analiz eden tümcül yöntemler. Bu çalışmada amaç yaya bölgelerine ait videolardaki aykırılıkların tümcül bir yaklaşım kullanılarak tespitidir. Yaya bölgelerine ait videolar, insan yoğunluğuna göre otomatik olarak gruplanarak analiz edilir. Yaya hareketi bireylerin tespit ve takibi yapılmaksızın, "Oriented Fast and Rotated Brief (ORB)" öznitelik tespit edicisi kullanılarak bulunmuş özniteliklerle modellenmiştir ve bu sayede gizlilik koruyucu özelliğe sahiptir. Bu öznitelikler, daha sonra "Binary Robust Independent Elementary Features (BRIEF)" öznitelik tanımlayıcısı kullanılarak temsil edilmiş ve aykırılık tespiti için uzay-zamansal Gizli Markov Modeli kullanılmıştır.

Anahtar Kelimeler: Bilgisayarlı görü, video gözetleme, yaya alanı analizi, aykırılık tespiti.

ACKNOWLEDGEMENTS

I wish to express my thanks and gratitude to Assist. Prof. Dr. Tuğba Taşkaya Temizel and Assoc. Prof. Dr. Alptekin Temizel. Throughout the past two years, they have been there whenever I needed help and assisted me throughout my graduate studies. I have had the chance to observe two excellent researchers at work and thanks to them, now I know how to perform research and that I can be an excellent researcher. None of this would have been possible without their guidance.

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

I am also thankful to my dear friend Cihan Öngün who has helped with any question I might have for him. He has helped me tremendously throughout my studies and supported me in all cases.

Furthermore, I am grateful to my family who have been supportive of me no matter what happens. There are no words that can express my gratefulness and my love towards them. I could not ask for a better family and I deem myself the luckiest person on earth for having them as my family.

Last but not least, I'd like to thank my friends Emine Çimen Öztürk and Samet Nargül who made the late-night assignments not only sufferable but also fun. I would not have been able to finish this degree in this semester if it were not for you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

2D	: Two Dimensional
3D	: Three Dimensional
AUC	: Area Under Curve
BRIEF	: Binary Robust Independent Elementary Features
CPU	: Central Processing Unit
DOG	: Difference of Gaussians
EER	: Equal Error Rate
EM	: Expectation Maximization
FAST	: Features From Accelerated Segment Test
FPS	: Frame Per Second
GLDM	: Gray Level Dependency Matrix
HMM	: Hidden Markov Model
IOCM	: Invariant Orthonormal Chebyshev Moments
kNN	: K Nearest Neighbors
LDA	: Latent Dirichlet Allocation
MCMC	: Markov-Chain Monte-Carlo
MDT	: Mixtures of Dynamic Textures
MRF	: Markov Random Field
ORB	: Oriented FAST and Rotated BRIEF
RANSAC	: Random Sample Consensus
RJMCMC	: Reversible Jump Markov-Chain Monte-Carlo
ROC	: Receiving Operational Characteristics
ROI	: Region of Interest
SIFT	: Scale-Invariant Feature Transform
SOM	: Self Organizing Map
SURF	: Speeded-up Robust Features
UCSD	: University of California, San Diego

CHAPTER 1

INTRODUCTION

Computer vision assisted surveillance systems are gaining importance for security reasons. Security is a big concern especially where there are large groups of people. In 2010, during a Love Parade in Duisburg, a disaster occurred due to overcrowding in which 20 people died and more than 500 people were injured. In 1989, during a Liverpool football game in Hillsborough Stadium, there was a stampede, also due to overcrowding, which resulted in 96 deaths and numerous injuries. In both of these events, if the crowd was monitored using computer vision systems, the disasters could have been prevented [1] [2].

When crowded areas are monitored by human observers, there is always a possibility of missing an important event due to exhaustion or lack of focus. The goal of the computer vision surveillance systems is to minimize the risk of missing important alarms while also minimizing the false alarm rates.

Correctly analyzing the crowded scenes requires having contextual information regarding that scene. An event that is anomalous in one context can be considered normal in another. For instance, in an area where there is a bike lane, presence of a bike is considered normal while in an area which is strictly pedestrian, the existence of a bike is considered an anomaly.

Also, to be able to make a correct analysis, the correct tools which are appropriate for the task should be selected. There are many computer vision techniques that are used in crowd analysis. Most of these, such as optical flow based people tracking or flow analysis techniques are computationally expensive and thus they are generally not suitable for real-time applications. Also, as previously pointed out by Mehran et al. [3], optical flow can result in noisy data. Other methods employ feature based analysis technique using methods such as SIFT [4] or SURF [5]. The detailed literature review of crowd anomaly detection methods using these techniques are given in Section 2.

Nevertheless, crowd anomaly detection is a challenging task. The aim of this work is to perform anomaly detection in pedestrian zones. In this work, the aim is to detect velocity based anomalies such as bikes or skaters occupying the scene. The anomalous entities are assumed to be banned vehicles in a strictly pedestrian zones. These entities usually have a higher velocity than the pedestrians so the purpose of this thesis is to detect velocity based anomalies which are on the higher end of the velocity spectrum.

For this purpose, a pixel based approach is adopted. Firstly, the video is partitioned into spatial regions to alleviate problems that might arise due to the perspective. Then, ORB

[6] features are obtained and by calculating the matching features in consecutive frames, velocities of the detected features are calculated. Using these velocities, representative statistics are obtained. The probabilistic graphs are constructed using these statistics and appropriate graph is selected using the density switch functionality. Finally, the anomaly detection is performed.

Pedestrian zone anomaly is defined as a behavior that is not expected to occur in the strictly pedestrian scene. In Figure 1.1, a sample of normal and abnormal scenes are provided. In the abnormal frame there is a bike, which is not allowed in the strictly pedestrian area.

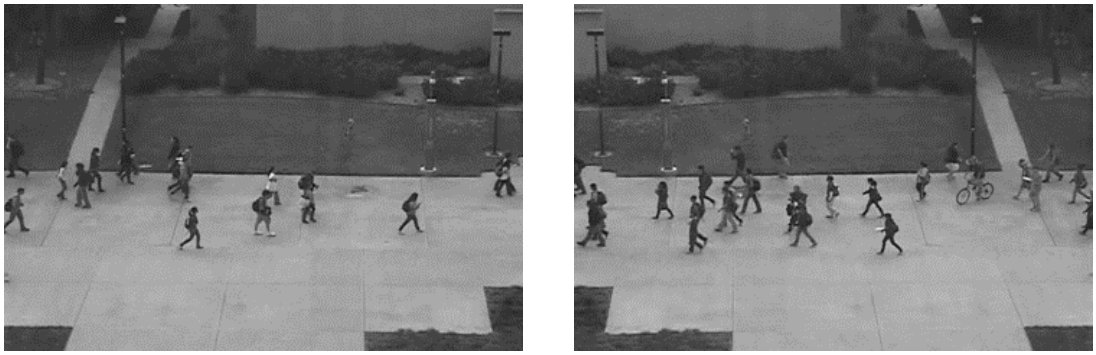

Figure 1.1 Normal frame (left) and abnormal frame (right)

As can be seen from Figure 1.1, there is a bike entity in the frame on the right. Bikes, carts, skateboards etc. are banned in pedestrian zones. This is why these are considered as anomalies.

The contributions of this work is as follows:

- Real-time detection of anomalies.
- Density switch method that allows accurate model selection. Since the characteristics of pedestrian motion change based on density (for example, when the crowd is dense, the pace of people is slower) it is important to select an appropriate model that represents this information. This idea is explained in Section 4 in detail.
- Usage of ORB instead of more computationally complex methods such as SIFT or SURF. In the literature, SIFT is the de facto pixel based method in crowd analysis systems. However ORB has a comparable performance and it has much less computational complexity.
- A simple model using frame statistics instead of raw velocity values. This work shows that statistics derived from scene can represent the frame accurately instead of using the raw velocity data.
- The training requires only normal frames (i.e., videos having usual behavior). This alleviates the need for acquisition of scenes having abnormal events -which are more difficult to obtain than normal scenes-.
- The users do not need to define any anomalies and the method does not need a threshold to detect anomalies.

The rest of the thesis is structured as follows. In Chapter 2, a literature review of crowd analysis techniques is provided. This chapter is divided into two. Since this thesis includes both crowd density information and crowd anomaly detection, literature review for both of these techniques are done separately. In Chapter 3, the proposed anomaly detection method is explained in detail. After giving a general overview of the proposed method, each component of the method is explained. First, the partitioning methodology and feature extraction are explained. Later the details of HMM modelling are provided. In Chapter 4, the density switch methodology, which is a major part in this work, is explained. Chapter 5 provides the details of another approach that was developed during the studies for this thesis. Chapter 6 provides the experimental results of the method and comparisons with existing methods are provided. Finally, the thesis is finished with the conclusion and directions for future work.

CHAPTER 2

AN OVERVIEW OF RELATED WORKS

2.1. Overview

While there are numerous studies regarding the pedestrian behavior in crowded environments, most of these focus on two major topics; namely, crowd density estimation and crowd anomaly detection. Crowd density estimation is essential for making emergency evacuation plans. It is also important as a preliminary information source in anomaly detection tasks. Crowd density estimation methods are typically presented along with crowd counting problems. Crowd anomaly detection is an equally important, albeit a challenging task. Detection of anomalies allow taking timely precautions to prevent potential incidents. The anomaly detection problem in crowd videos cover a wide range of anomalies such as anomalies in outdoor pedestrian movement, indoor pedestrian movement and even traffic flow.

In this chapter, existing approaches to crowd density estimation and crowd anomaly detection are explained and discussed.

2.2. Crowd Density Estimation

Crowd density estimation is an important factor in anomaly detection since it is related to the context where the anomaly is present. As mentioned in Section 1, the motion context changes with the increase or decrease of the crowd density. In dense crowds, pedestrians are restricted by the motion of the crowd. However in sparse crowds, they have more space to move. In this case, pedestrians may choose to move more freely (i.e. they can speed up or slow down or change directions.). In [7], the authors used Minkowski Fractal Dimensions [8], Fourier Spectrum [9] and Gray Level Dependency Matrix [10] as inputs to three different classifiers to classify the crowd density as very low (VL), low (L), medium(M), high (H) or very high (VH). The classifiers they used are a Bayesian classifier, a self-organizing map (SOM) [11] and a fitting function. The best performing model was reported to be the Bayesian classifier with GLDM as the input with 85% correct classification rate. This method is very simple and it uses supervised learning for density estimation purposes. While it works well in terms of classification rate, it cannot distinguish between high and very high crowd densities [7]. Also it is not tested in outdoor environments. This method is tested on images captured in Liverpool Street Railway Station, London, UK.

Another technique that uses both GLDM and SOM are presented in [12], however in this paper two SOMs are used. First, the initial SOM classifies the textural features (GLDM) according to their finery. Later, histograms of textural features are calculated. Finally, using these histograms the crowd density classification is performed with another SOM. The authors provided a confusion percentage matrix instead of correct classification rate.

The average of the correctly classified percentages is 77.4%. This method is also tested on images captured in Liverpool Street Railway Station, London, UK.

In [13], the authors use GLDM and MDT. They also propose a new method called Invariant Orthonormal Chebychev Moments (IOCM) where the centroid location is subtracted from all orthonormal Chebychev moments [14] to achieve position independence. After obtaining GLDM, MDT and IOCM features, the authors perform classification using SOM. The authors use an outdoor dataset where images are captured in the morning and afternoon. The combined dataset contains images from both scenes. For the morning scenes, they obtained 65%, 48% and 70% correct classification rate for GLDM, MDT and IOCM respectively. In the afternoon scene, they achieved 70%, 60% and 95% correct classification rate for GLDM, MDT and IOCM respectively. In the combined data, they achieved 80%, 35% and 85% correct classification rate for GLDM, MDT and IOCM respectively.

GLDM is quite sensitive to any outlier objects in terms of color. For example, when a white object enters into a scene which is otherwise in various shades of gray, the features obtained by GLDM vary significantly. So either image needs to be smoothed beforehand or the GLDM needs to be smoothed. In either case, there is a good chance the real outliers or information will be smoothed along with any problems data might have. GLDM is a very simple approach that is not very robust to noise. In real-life scenarios, there is no guarantee that the texture only will give information about the actual crowd density because the texture only represents the textural properties of the scene and not the number of people in the crowd.

Along with texture based methods, other techniques have also been proposed which aim to count pedestrians by first detecting them in the scene. This type of methods have privacy issues since they require pedestrians to be detected before counting. In [15], the authors propose a head detection algorithm to find heads in a scene and then count them. They make use of gradients in a ROI where they assume the heads would normally be (i.e. in the upper section of the image). First they detect the gradient-orientation based interest points in a scene and then they apply background subtraction. They select LUV channels, gradient intensity channels and six oriented gradient channels as initial features. They generate final features using these initial features. They also generate a large set of random features and use AdaBoost [16] algorithm to select and train the features. It is stated in the paper that the method requires a large number of positive and negative samples to train. This results in a large proportion of testing data to be used as negative samples in the training test. When using so much data for training, there is always a possibility to overfit to the data. Also, there may not always be abnormal frames in the training data which makes it difficult for the model to be trained.

In [17], the authors utilize a Reversible Jump Markov-Chain Monte-Carlo (RJMCMC) [18] algorithm to explore configurations of different numbers of people in crowd. RJMCMC algorithm is an extension of Markov-Chain Monte-Carlo (MCMC) algorithm which was originally developed to generate samples from complex distributions. RJMCMC algorithm extends it by exploring dimensionalities that are different than the initial configuration. The authors of this paper try to generate samples which have

different numbers of people and try to find the sample that best explains the image. The problem with MCMC based methods relies in the selection of appropriate distribution to sample from. The distribution needs to represent the underlying data while remaining simple. This is easy when the data amount is small, however in video processing, the amount of data is expected to be very large and MCMC algorithms may not scale very well, certainly not in terms of speed.

In [19], a different approach is used. The authors propose a pixel counting method for density estimation. First the authors perform a foreground detection using either determining a region of interest (ROI) manually or detecting the foreground using Eq. 2.1. In this equation I_{mask} represents the binary mask to be applied to the scene. I_{FG} represents the binary foreground pixels (i.e. either foreground pixel or not). And t is the threshold value which is the median of all foreground pixels. Initially, all of the pixels are deemed as foreground. The median of the pixels is calculated and the pixels with lower values than the median are eliminated. The binary mask consists of the remaining pixels.

$$I_{mask} = \sum I_{FG} > t\,, t = median(\sum I_{FG}) \tag{2.1}$$

After obtaining the foreground pixels, a geometric correction is applied to them. They learn a "usual crowdedness" model and detect unusual crowdedness based on this model. The issue with this approach is that the threshold is very crude. Some foreground pixels may be eliminated due to the crude and static threshold while some background pixels are retained. This results in a noisy data.

In our earlier work [20], we proposed a feature point count based crowd density classification algorithm. In this paper, the feature counts are obtained from the training videos and a kNN algorithm is trained using the feature counts and the density class of the training videos. Later, the frame densities are classified based on this model. It is a supervised approach and requires labelling of the training frames according to the crowd density.

The aim of this thesis is to detect pedestrian anomalies in the scene. To select the appropriate model for a specific crowd density, the proposed method uses the calculated density information as a model selection tool. Our method is different to the aforementioned methods in the sense that it takes velocity information and characteristic motion behavior of the crowd. This is explained in detail in Chapter 4.

2.3 Crowd Anomaly Detection

Crowd anomaly is defined as any unusual event occurring in a crowd (i.e., deviation from the common behavior observed in the particular scene). There are several studies related to crowd anomaly detection.

In [21], the authors apply a spatiotemporal grid to the video frame and obtain optical flow values in each grid. Using these values, atomic motion patterns are learnt using Mixtures of Probabilistic Principal Component Analysis [22]. Later, these patterns are

modeled using Markov Random Field (MRF) using online learning technique (i.e. parameters are updated with each new observation). Optical flow is a computationally complex algorithm unless it is accelerated by an accelerator such as GPU or a sparse optical flow is used. Also, optical flow results in very noisy features in some cases [3].

In [3], the authors compute trajectories with particle advection. They compute the social force [23] between the particles and use Latent Dirichlet Allocation (LDA) [24] to detect anomalies. Particle advection builds on optical flow which has high computational cost. Social force is also a computationally expensive process and combining these steps results in a slow algorithm as demonstrated in Chapter 5.

In [25], the authors propose to learn a normal behavior model using low level features of the scene and then compute the likelihood of a new observation. If the likelihood is below a predetermined threshold, then the new observation is signaled as an anomaly. This method is very simple. However, crowd behavior may not always be represented by simple low level features. While the simplicity of the algorithm has computational benefits, it may not always be adequate in explaining some events in the scene.

In [26], the authors compute GLDM data in 2D spatial dimensions and in time dimension, resulting in a 3D GLDM output. They also compute the optical flow in the scene. They adopt a mixture model approach to train the obtained data and statistical outliers are labeled as anomalies. Problems with GLDM are explained in Section 2.2. Also, as mentioned previously, optical flow is a computationally expensive method.

In [27], the authors propose a method based on social force model [23]. At each iteration, particles are initialized at random locations and social forces are obtained. Then, a minimization procedure is applied to the data which eventually results in optimal particle initialization which allows to calculate optimal flows. After calculating optimal flows, the authors perform anomaly detection using Random Sample Consensus (RANSAC) [28] algorithm.

In [29], the authors adopt a method that uses both textural data and the motion based data to model the scene. They extract spatiotemporal cuboids from the data and estimate both appearance and the motion of the scene. With each new frame, the authors perform a state search in the training data using a predetermined radius, and if the search turns empty, the new frame is classified as anomaly. The weakness of the approach is that there is no generally accepted way of determining the radius.

A method that makes use of SIFT feature detection and tracking is presented in [30]. However, in case of large number of interest points, SIFT can also be slow. There are faster algorithms which can compete with SIFT in terms of accuracy. In our earlier work we have shown that ORB works significantly faster while producing satisfactory results in crowded scenes compared to the other feature description methods and a feasible alternative in crowd behavior analysis applications [31].

2.4 Feature Detection, Description and Matching

In this work, a feature detection based method is utilized. Feature detection, in the broadest of terms, performs interest point detection or corner detection. The most widely

used feature detection and description method is SIFT [4]. While computing SIFT interest points, first the image is described by a scale and location invariant feature vector using Difference of Gaussians (DOG) [32] in several octaves of image pyramids. Later, these DOG values are compared for each pyramid and neighbor pixels. If pixel *p* is either maximum or a minimum in this comparison, it is a potential interest point. To filter the results, a threshold is used. Pixels with intensity lower than the threshold *t* are discarded. In the paper, this threshold is given as 0.03. SIFT is also a feature descriptor. It describes the features using the histograms of the pre-computed DOG values.

ORB [6] Feature detector uses a modified features from Accelerated Segment Test (FAST) feature detector [33] to perform orientation independent feature detection. To determine if a pixel *px* is an interest point or not, FAST selects that pixel as the center of a circle with the radius of 3. This circle's border passes through 16 pixels. In order to determine if the pixel *px* is an interest point, 12 of the 16 border pixels need to have an intensity above or below a threshold *T* determined by the user. The authors used a step-wise approach to speed up the algorithm. First, they check if 3 of the pixels with numbers 1, 5, 9, 13 (which are at angles 90^o, 0^o, 270^o and 180^o in the circle) are above or below *T*. If not, pixel *px* is not an interest point. If 3 of those 4 pixels are either above or below the threshold, then the algorithm checks if 12 of 16 pixels of the circle are either above or below the threshold. If this criterion holds, then *px* is classified as an interest point. However this approach is not orientation independent. ORB feature detector improves FAST by modifying it to be so. ORB uses an approach called *intensity centroid*, initially proposed by Rosin [34]. Intensity centroid assumes, the intensity of a pixel offsets from its center and this vector can be used to impute the orientation of the pixel. Moment of the patch (the circular area of FAST) is calculated as in Eq. 2.2 and denoted as m_{pq}.

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \tag{2.2}$$

Where *i* and *j* are the coordinates of pixel *pix*, *p* and *q* are degrees of moment. And the *I(i,j)* denotes the intensity of *pix*. Using this moment, the intensity centroid (denoted as *C*) can be calculated as in Eq. 2.3.

$$C = (\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}) \tag{2.3}$$

Finally, the orientation θ of the patch is calculated as in Eq. 2.4.

$$\theta = atan2(m_{01}, m_{10}) \tag{2.4}$$

The feature descriptors generate a representation to *describe* the detected feature points by using different approaches. For instance, Scale Invariant Feature Transform (SIFT) [4] describes the feature point with histograms of image gradients. BRIEF [35] on the other hand, uses a binary description approach. It describes the patch (*p*) of size *S*x*S* with

an *n*-dimensional (n_d) bit string using the function given in Eq. 3.5 with the help of test (τ) function described in Eq. 2.5.

$$\tau(p; x, y) = \begin{cases} 1 \; if \; p(x) < p(y) \\ 0 \quad otherwise \end{cases} \tag{2.5}$$

Where $p(x)$ is the smoothed pixel intensity at $x = (u, v)^T$.

$$f_{n_d}(p) = \sum_{1 \le i \le n_d} 2^{i-1} \tau(p; x_i, y_i) \tag{2.6}$$

Where τ is the test, p is the patch, x_i and y_i are randomly selected location pairs in the patch. The dimension n_d can take values of 128, 256, or 512.

For the purpose of this work, the detected and described features need to be matched. There are two different matching algorithms. FLANN Based Matcher [36] and Brute Force Matcher [37]. FLANN Based Matcher is used for matching 32-bit features while Brute Force Matcher is used to match binary features.

FLANN based matcher makes a nearest neighbor search amongst all described features. It uses two different algorithms to perform nearest neighbor search: Randomized KD-Tree algorithm [38] and Hierarchical K-means algorithm. The matching algorithm automatically switches between these two algorithms using a cost function as shown in Eq. 2.7.

$$cost = \frac{s + w_b b}{(s + w_b b)_{opt}} + w_m m \tag{2.7}$$

In this cost function, *s* represents the search time for the number of vectors in the sample dataset, *b* represents the tree build time, w_b represents the weight for tree build time, w_m represents the memory weight and $m = m_t/m_d$ represents the ratio of memory used for tree to the memory used to store data.

Binary descriptors are matched using the Brute Force Matcher. For binary descriptors, Hamming distance is used while calculating the similarities between two features. Hamming distance calculates the percentage of bits that differ in binary vectors. To find the match for the feature point *fp*, Brute Force Matcher [37] calculates the Hamming distance between *fp* and all possible feature points.

For instance, let's say there are *N* number of feature points detected in frame *t* and *M* feature points detected in frame *t+1*. To match the feature points between two consecutive frames, Brute Force Matcher calculates *NxM* Hamming distance values and selects the most similar feature point in the frame *t+1* for each feature in frame *t*.

CHAPTER 3

HMM BASED PEDESTRIAN MOTION MODEL

Modelling the pedestrian motion in the pathways is a challenging task. Because pedestrian pathways do not have strict rules regarding the behavior of their occupants, there are many actions that a pedestrian can take while walking such as changing direction, slowing down, pacing up, taking sharp turns and stopping instantly. Those actions can occur at any arbitrary order making it not feasible to model the pedestrian motion with deterministic models. To overcome this issue, a probabilistic framework, namely a Hidden Markov Model (HMM), is proposed to be used in this thesis.

3.1. Overview

In this thesis, the main task is to infer if the pedestrian motion is normal or abnormal, given the velocity observations. To perform this task, an HMM based approach is proposed. First, partitioning is carried out if necessary. After partitioning, ORB [6] features are extracted and described using BRIEF [35]. Feature matching is performed using Brute Force Matcher and raw velocities are obtained. Using these raw velocities, representative statistics are computed. To learn the normal state, the representative statistics of the training set are used. Since the training set has no abnormal frames, noise injection is performed to artificially create anomalous data. Using both the normal representative statistics and noisy representative statistics, HMM is trained. For test data, model selection with density switch is performed and anomaly detection is carried out using the selected model. This process is explained in Figure 3.1 and it is detailed further in the following paragraphs of this section.

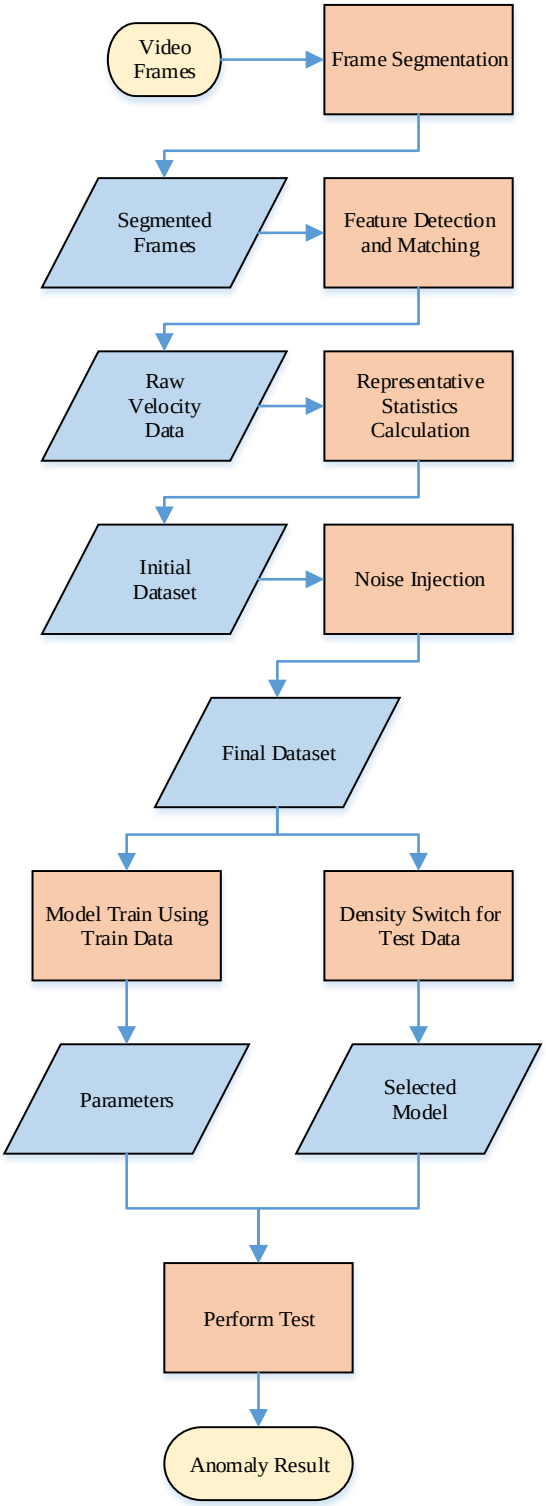

Figure 3.1 Process flow of the proposed system

In certain cases, depending on the camera position, perspective might be a problem while computing the representative velocity statistics. Because the further away the pedestrian is, the lower the computed pixel-wise velocity will be. To overcome this issue, the scene is partitioned into smaller overlapping spatial zones in which the calculated velocities for similar real world velocities are expected to be similar. This partitioning is done manually.

After the partitioning, feature detection and matching between consecutive frames is done for each separate segment. Feature detection and partitioning are explained in Section 3.1.2

Having obtained the matched features, pixel-wise Euclidean distances are calculated between these points. These velocity values are stored in velocity vectors based on their directions. In this work, the velocities are separated into eight different directions based on their motion angles. These angles are named as follows: {[0º-45º):D1, [45º -90º):D2, [90º-135º):D3, [135º-180º):D4, [180º-225º):D5, [225º-270º):D6, [275º-315º):D7, [315º-360º):D8}. This stage is applied to each area resulting in M x (N-1) x 8 velocity vectors for each video where M is the number of areas and N is the total number of frames. As the velocity calculation is based on feature matching between two consecutive frames, there is no data for the last frame hence there exist data for N-1 frames. Each vector has a size equivalent to the number of matched features.

Then, to reduce the dimensionality of the dataset, representative statistics are calculated for each of the velocity vectors obtained in the previous step. Those statistics are mean of the velocity vector, standard deviation of the velocity vector, feature count in the velocity vector, skewness of the velocity vector. These statistics are stored in a statistics vector. For each direction in each frame and segment, four representative statistics are calculated. This results in $M\ x\ (N-1)$ x 8 x 4 scalar values for each video instead of $M\ x\ (N-1)$ x 8 vectors of size S, which is the number of matched features.

The feature count statistic is only used for density switch for model selection before the testing phase. This switching step is explained in Chapter 4 in detail. Other 3 statistics are used in constructing the HMMs using the training data. The HMMs are trained for each video in the training set.

In some datasets, the training set only consists of normal frames. This data is sufficient for learning the normal state of the pedestrian behavior. In principle, it is possible to learn the normal behavior only and classify frames as normal and abnormal based on the deviations of the new data from the model. However, this technique requires a thresholding approach. Thresholds are manually determined by domain knowledge and susceptible to biased selection. Instead of thresholding, artificial noise is injected to the normal data to reflect the anomalies that might occur and an abnormal state is learned from this artificially generated noisy data.

Assume there are V_{tr} number of training videos in the training set, this means that V_{tr} x M x 2 number of HMMs are learned from the data for each video and each partition and for each pedestrian state.

Finally, having obtained the training videos and HMM parameters, density switch is performed and anomaly detection is carried out using the selected model. Model selection based on density switch is explained in detail in Chapter 4.

3.1.1. Scene Partitioning and Feature Matching

Depending on the camera view, there might be a perspective problem that results in inconsistency in calculated pedestrian velocities. Velocities of the pedestrians that are further away from the camera may be calculated slower due to the distance between them and the observation point. To overcome this issue, spatially overlapping binary perspective masks are applied to the scene and every mask is analyzed as a separate image. The masks are determined manually according to the size of the scene or the severity of the perspective problem. A sample image that has a perspective problem is given in Figure 3.2 and sample image of five masked areas is given in Figure 3.3.

This partitioning is done manually. It is done only once based on the positioning of the camera. After fixing the camera position, the binary masks are created and the binary masks are applied to each new frame in real-time.

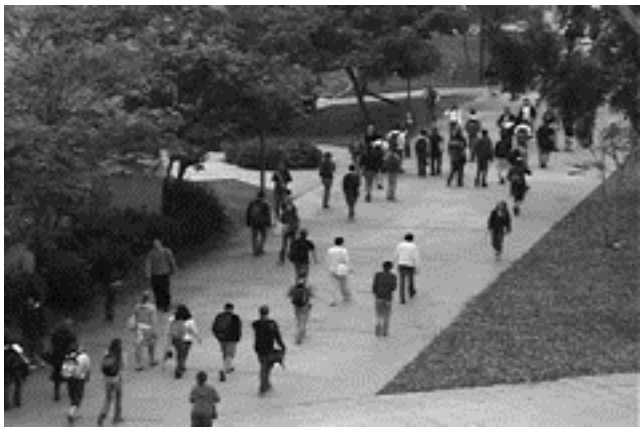

Figure 3.2 Sample image that shows the perspective problem

As apparent in Figure 3.2, the pedestrians further from the camera seem smaller. This results in relatively smaller velocities compared to those near the camera.

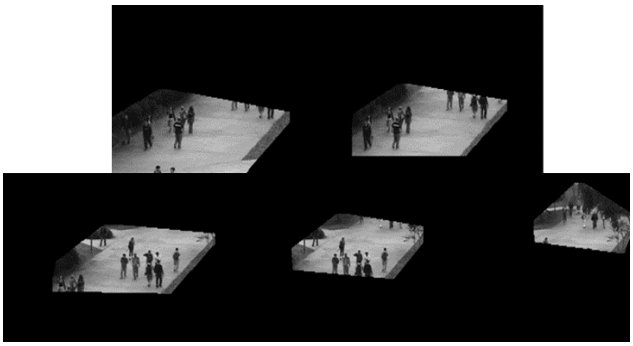

Figure 3.3 Sample masked image

After scene partitioning, feature detection and matching is performed ORB Feature Detector [6], BRIEF Feature Descriptor [35] and Brute Force Matcher algorithms in the OpenCV [37] computer vision library. A sample image for feature matching in partitioned areas is given in Figure 3.4.

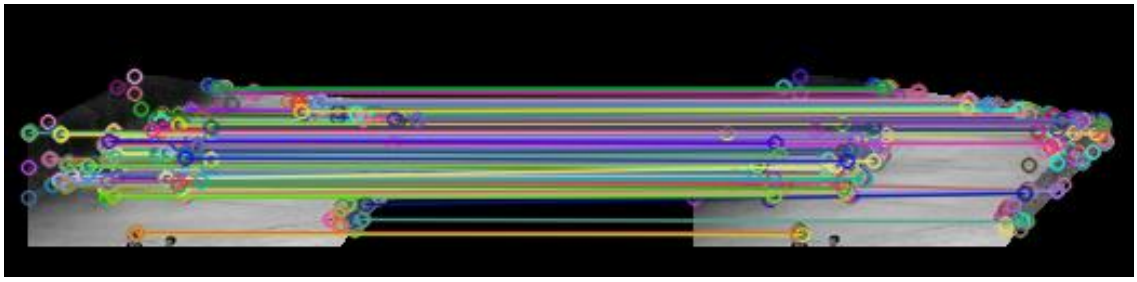

Figure 3.4 Sample image of feature matching

3.1.2. Noise Injection

To model the abnormal data, artificial noise is injected to the normal values obtained from the normal frames. The pedestrian motion consists of relatively low velocity values compared to other means of transport. Stopping and turning are considered as normal behavior on pedestrian pathways. However, high speed actions such as running are considered abnormal because usually, pedestrians are not expected to run unless there is a compelling reason to do so. Likewise, using bikes or other means of transport in pedestrian pathways are also considered as anomalies because in most cases, use of these types of transportation is not allowed in areas restricted for pedestrian use. However it is not easy to detect all types of anomalies by rule based methods or thresholding. In some cases, the velocities of the other means of transport may be very close to pedestrian velocities. Also, normally behaving pedestrians usually outnumber the entities with anomalous behavior. This results in a bias towards the normal velocity values in the calculation with some high valued entries. In other cases, due to noise in the captured frame, erroneous velocity values might be calculated in the matching phase. When this occurs, there is also a bias towards normal velocity values however in this

case, the high velocity values do not represent an anomaly. With rule based systems or simple thresholding methods, there is no way to separate erroneous noise from actual anomalies. Because of all these difficulties, a random noise generation method is adopted in this work to represent the effects of anomalous behavior in the statistics.

As mentioned above, the normally behaving pedestrians usually outnumber any other entity in the scene. The velocities of those pedestrians are low and thus they lower the average velocity in the scene. In case of high velocity anomalous entries, the average of the velocities in the scene might slightly increase depending on the number of entities present in the scene and the velocity of the anomalous entity. This is also the case with the noise that acts as a false anomaly. However, there are other statistics that can help distinguish the difference between noise and anomalies.

Usually, there are several feature points detected on any entity in the scene. The velocity values of the feature points detected on a single entity are similar to one another but they might be different from other entities. In case of an anomalous entity, the difference in the velocities will increase the standard deviation of the scene. The increase in standard deviation might also appear when there is noise in the data. However, unlike anomalous entities, noise usually does not have a coherent behavior. This means that noisy data in itself is expected to have a larger amount of deviation than an anomalous entity. In case of enough amount of noisy feature points in the scene, the standard deviation of noisy data is expected to be higher compared to the standard deviation of an actual anomalous entity.

Since the pedestrian velocities are relatively low and there is to be some amount of high velocity noise in the data, the data is expected to accumulate at the lower values. In case of anomalous entities, there will be a higher number of feature points with high velocity values. In this case, normal scene is expected to have a higher skewness value than the anomalous scene.

Another thing to consider while performing noise injection is that the motion of individuals is restricted by the motion of the crowd. Usually, pedestrian walkways have some directions that have more people than other directions. The anomalies in less used directions are easier to detect since it is less likely that there would be other pedestrians in those directions to outnumber the anomalous entity. This means that if noise injection is performed in those directions, it will not represent the general behavior of the direction and will result in a wrong model. This is why the noise injection should only be performed in major directions. Major directions are said to be the directions in which the majority of the motion occur.

In this work, the major directions are determined by calculating the total number of features that move in each direction. The major directions are determined by examining the feature counts manually. An example of average features that move in each direction is illustrated in Figure 3.5. The reason for using only the major directions is that the higher the data dimensions, higher the computational cost. This study aims for real-time performance. In order to achieve that, only necessary features need to be used. Since there is very little movement in minor directions, those are deemed unnecessary.

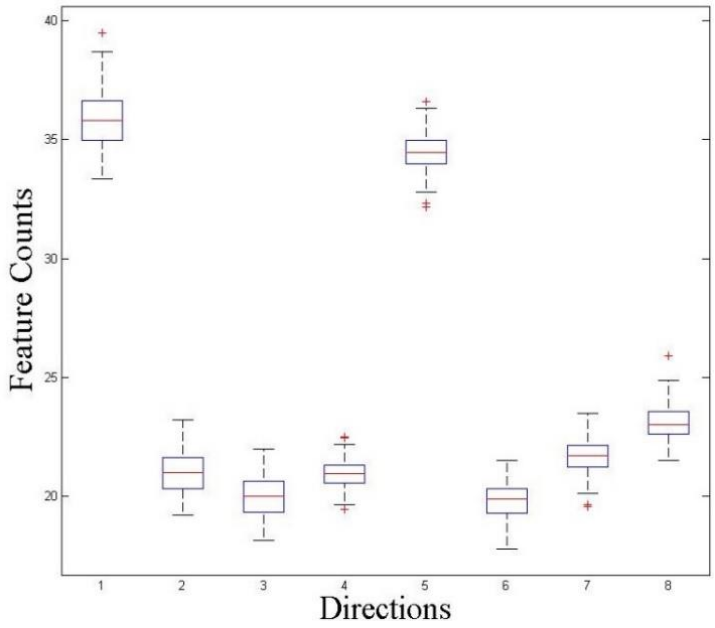

Figure 3.5 Feature counts per frame

Based on these assumptions, a new data set is generated to represent the anomalies that might be present in the scene. The data set is generated by random sampling using the statistics calculated from the training set. For each training video statistics matrix, a corresponding artificial anomalous data is generated. Let μ_m be the average of the mean velocity values in the statistics matrix and S_m be the standard deviation of the mean velocity values in the statistics matrix, to generate the average velocity values of the artificial mean data (denoted as *MA*) in major directions (MD), Eq. 3.1 is used.

$$MA_{Di} \sim N(c^* \mu_{mDi} , S_{mDi}) \qquad i \text{ in MD} \qquad (3.1)$$

Let μ_{std} be the average of the standard deviation of normal velocities and S_{std} be the standard deviation of the same values, then the artificial standard deviation is calculated using Eq. 3.2 and denoted as *StdA*.

$$StdA_{Di} \sim N(c^* \mu_{stdDi} , S_{stdDi}) \qquad i \text{ in MD} \qquad (3.2)$$

Let μ_s be the average of the skewness of normal velocities and S_s be the standard deviation of the same values, then the artificial skewness is calculated using Eq. 3.3 and denoted as *SA*.

$$SA_{Di} \sim N((1/c)^* \mu_{sDi} , S_{sDi}) \qquad \text{i in MD} \qquad (3.3)$$

The multiplier *c* is determined by the user. The choice for c is based on the anomaly that user wants to detect. If the user is looking for very subtle high velocity anomalies, then *c* needs to be slightly higher than 1 while if the user is looking for subtle low velocity anomalies, then *c* needs to be slightly lower than 1. More apparent anomalies can also be

detected by amplifying the magnitude of *c*. The effect of noise injection coefficient is given in Section 6.4.

Noise injection is done for each video. Assume we have *V* number of videos of *N* number of frames. The artificial noisy dataset consists of *V* number of artificial data matrices of *N* number of frames and each matrix corresponds to one video in the normal training set.

3.2. Hidden Markov Model

HMM is a probabilistic approach to model time-dependent and stochastic data. They are named after the hidden (latent) variables they have and the Markov assumption they employ. Markov assumption states that the state of a variable at time *t*, is only dependent on the state of that variable at time *t-1*. The latent variables of HMM are represented using the Markov assumption.

HMMs are mainly divided into two groups: Discrete HMMs and Continuous HMMs. Hidden variables of discrete HMMs are discrete valued while the hidden variables of continuous HMMs are continuous valued. In this work, a discrete HMM is adopted because the focus of this thesis is to detect if an anomaly is present or not. In other words, if the state of hidden variable is anomalous or not.

The observations for this problem are velocity statistics and thus are continuous valued. The hidden variables represent the state of the frame at time *t*, so have discrete values (1 for anomalous frame and 0 for normal). Because of this discrete nature of the state, Discrete HMM with Continuous Observations is adopted in this work. The HMM that is used is represented in Figure 3.6. In this figure, rectangular nodes represent the discrete hidden variables while the circular nodes represent the continuous observations.

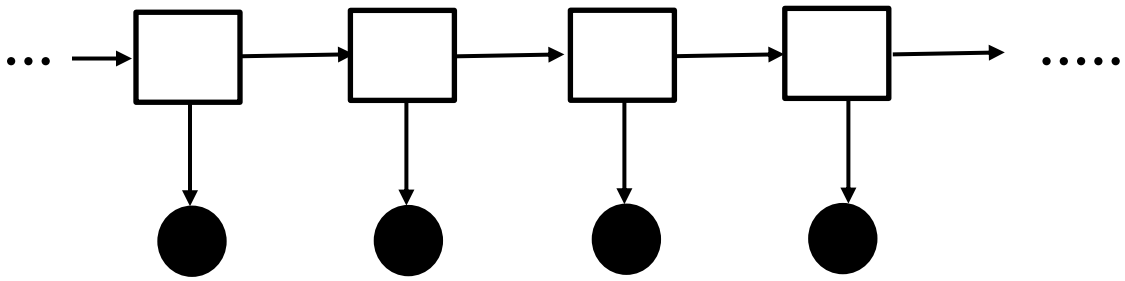

Figure 3.6 Representation of used model

This figure explains the general idea behind HMM. The time flows from left to right as denoted by the pointed arrows. The hidden nodes (the actual values) affect each other through time intervals. The idea behind is that an effect that is present on a hidden node at time *t*, is likely to prevail at time *t+1*. This is represented by a *transition matrix* of size *S*x*S* that contains probabilities of switching states or remaining in a state.

The arrows pointed towards the observation nodes represent the effect of hidden nodes on the observation nodes. Since the observed data is normally the readings obtained from a state, they are directly affected by the state of the hidden node. In our case, the observations are the representative statistics obtained from the scene. It is expected that

those statistics directly represent the underlying state of the scene. The main task in such a model is to infer the state of the hidden nodes given the observations.

3.2.1. Learning the HMM

The model that is used in this work is a Hidden Markov Model with Mixture of Gaussians outputs. The state parameter *s* represents the number of hidden states and the mixture parameter *m* represents the number of Gaussians. In this work, it is assumed that the HMM has 3 different states (*s*=3) and the number of Gaussians in the mixture is 3 (*m*=3). The reasoning for that selection is based on crowd density assumptions and the major direction assumptions. The crowd is assumed to have three different density levels, namely: low, medium and high. The crowd motion is initially assumed to be bidirectional which is usually the case unless the scene is of a crossroads. In this bidirectional movement, the crowd is assumed to have three possible motion behaviors. The first one is where majority of the movement is in direction *d*. The second one is where the majority of the movement is in direction *–d*. Finally, there is no major direction, meaning both of the directions have equal number of pedestrians walking in them.

The training is done using the Expectation Maximization algorithm. Expectation maximization algorithm estimates the parameters θ' which maximize the log probability $\log P(x|\theta')$ of model explaining the training data. In this log probability representation, x represents the training data and θ' represents the parameters. This formula reads as "Log probability of observing the data x, given the parameters are θ'. So by maximizing this probability, the EM algorithm automatically finds the parameter set θ' that best explains the observed data. In case of complete observed data, expectation maximization algorithm reaches a global optimum. However, in partially observed data, EM algorithm may converge to a local optima. To avoid this, it is recommended to do multiple initializations.

EM algorithm first separates the optimization problem into smaller sub problems which have a unique global optima. Optimization of these sub problems result in a local optima for the whole problem.

EM algorithm consists of two steps called E-step and M-step. In E-step, the algorithm finds a function that lower bounds the probability to be maximized everywhere. That function is represented in Eq. 3.4:

$$G(\theta'^{(t)}) = log(P(x|\theta'^{(t)}) \quad (3.4)$$

During the M-step, the algorithm estimates a new parameter set $\theta'^{(t+1)}$ that increases the log likelihood.

Because of the real time concern in this study, the HMM is trained with the data in the major directions instead of all of the directions.

The training is done with two different datasets for each video. One dataset is the dataset with original values and the second dataset is the dataset with the artificial values. The

first HMM represents the normal motion of pedestrians and the second HMM represents the abnormal motion of pedestrians. The classification in testing phase is done via calculating the log likelihood of each model for the new data. This step is explained in detail in Chapter 5.

CHAPTER 4

DENSITY SWITCH FOR MODEL SELECTION

In the pedestrian pathways, usually the crowd density is not consistent throughout the time intervals. This results in different behavior characteristics in crowds with different levels of densities such as pedestrians slowing down in a denser crowd while they move faster when they have more space to move in crowds with lower density. In the former case, pedestrians are usually restricted by the general behavior of the crowd. However, in the latter pedestrians are not restricted so they can change their movements in more drastic ways such as taking sharp turns or stopping instantly. Due to this different behavior patterns in different crowd densities, we claim that the crowd density is an important factor in analyzing the pedestrian behavior.

To the best of our knowledge, there are no studies in the literature that include the effect of changes in crowd density while trying to detect anomalies. In this chapter, a simple yet effective method for selecting the appropriate model based on the density of the scene is proposed. This method is applied to each test video frame individually.

4.1. Overview

Using the representative statistics (feature count moving in each direction, mean velocity, skewness of the velocity values and variance of the velocity values) which are computed as explained in Section 3.1, density switching is performed and appropriate model is selected.

4.1.1. Density Switch Function

In order to perform an appropriate model selection, a penalty function is developed. Using this penalty function, the models which have high degree of noise or unwanted deviations are eliminated while retaining the characteristics of pedestrian motion.

As described in Chapter 3, pedestrian motion has certain characteristics. Based on these characteristics, model selection is performed.

With each new frame input, the characteristics of the new frame and all training sets are compared. The training dataset and the test frame are expected to have some degree of similarity as the test frame and training dataset are expected to have a similar density. The difference between the feature counts of the training set and the test frame should be low. Let FC_{TRid} denote the feature counts of training video i in direction d and FC_{TEjdf}

denote the feature count in frame *f* of test video *j* in direction *d*, then the feature count similarity of train video *i* (denoted as FCD_i) is calculated as in Eq. 4.1.

$$FCD_{id} = Mean(|(\mathrm{FC_{TRid}} - \mathrm{FC_{TEjdf}})| \big/ norm(|(\mathrm{FC_{TRid}} - \mathrm{FC_{TEjdf}})|)) \quad (4.1)$$

This function calculates the absolute difference between the feature count of the test frame and the feature count of the training set values and normalizes the obtained distance vector.

Furthermore, even though test frame might be anomalous, we expect it to have entities with normal behavior too. This is why we also expect the mean velocities to be similar to each other. Let MD_{i} be the mean velocity distance between the test frame and the training video *i*, M_{TRid} denote the mean velocities of train video *i* in direction *d* and M_{TEjdf} denote the mean velocity in frame *f* of test video *j* in direction *d*, then MD_{id} is calculated as in Eq. 4.2.

$$MD_{id} = Mean(|(\mathrm{M_{TRid}} - \mathrm{M_{TEjdf}})| \big/ norm(|(\mathrm{M_{TRid}} - \mathrm{M_{TEjdf}})|)) \quad (4.2)$$

This function calculates the absolute difference between the mean velocity of the test frame and the mean velocity of the training set values and normalizes the obtained distance vector.

Also, with the assumption that pedestrian velocities are generally low, the training videos with high mean velocity values are also penalized as they might have some noise in the data. Let MV_i denote the penalty for high mean values of the training data *i*, then MV_{id} is calculated as in Eq. 4.3.

$$MV_{id} = \mathrm{Mean}(M_{TRid} \big/ \mathrm{Norm}(M_{TRid})) \quad (4.3)$$

As mentioned before, deviation in training sets are undesirable. The training videos with high standard deviations are also penalized. Let SV_{id} denote the penalty for high standard deviation of the training data *i* in direction *d*, and S_{TRid} denote the standard deviations of video *i* in direction *d*, then SV_{id} is calculated as in Eq. 4.4.

$$SV_{id} = \mathrm{Mean}(S_{TRid} \big/ \mathrm{Norm}(S_{TRid})) \quad (4.4)$$

Finally, the skewness difference between the training video *i* and test video frame are also taken into account. Let SK_i be the skewness of the distance values computed between the test frame and the training video *i*, SK_{TRid} denote the skewness values of train video *i* in direction *d* and SK_{TEjdf} denote the skewness in frame *f* of test video *j* in direction *d*, then SK_i is calculated as in Eq. 4.5.

$$SK_{id} = SK_{TEjdf} - \mathrm{Mean}(SK_{TRid}) \quad (4.5)$$

After calculating the penalty components depending on the directionality, all of those components are normalized with their vector norms using Eq. 4.6.

$$Vector = \mathrm{Vector/Norm}(Vector) \quad (4.6)$$

Where Norm is computed as in Eq. 4.7 and shown as ||A|| where A is a vector or a matrix and $\mathrm{A^H}$ is the conjugate transpose of A.

$$||A|| = \sqrt{\max eigenvalue\ of\ A^H A} \tag{4.7}$$

After normalization as above, single values for each of the components are obtained as follows: *FCD* denotes the feature count distance penalty component, *MD* denotes the mean distance penalty component, *MV* denotes the mean value penalty component, *SV* denotes the standard deviation penalty component and finally *SK* denotes the skewness penalty component. Given all these values the penalty function *P* for training video *i* is defined as in Eq. 4.8.

$$P_i = FCD_i + MD_i + MV_i + SV_i + \mathrm{SK}_i \tag{4.8}$$

Please note that unlike other distance components, the skewness distance is not calculated using the absolute distance and the training skewness values are subtracted from the test frame skewness value. This is because unlike other distance values, it is desired for the skewness of the training video to be higher than the testing frame.

Based on the obtained penalty values, the model is selected as in Eq. 4.9.

$$Model = \mathrm{argmin}(P_i, i) \tag{4.9}$$

The effect of density switch methodology on classification results is analyzed in Section 2.2.

CHAPTER 5

CHMM BASED PEDESTRIAN MOTION MODEL

During the development of this thesis, several approaches were tested. The most successful of these was a CHMM approach using the mean velocities of the data.

In this study, the scene is partitioned into smaller overlapping spatial zones as described in Section 3.1.1. Using these smaller zones, feature detection and matching are performed. With the extracted features, velocities are calculated in eight directions. Under the assumption that the anomalies would be on the higher end of the velocity spectrum, a quartile analysis is performed. Four quartiles are calculated and only the data points in the fourth quartile are used for mean velocity calculation. The number of features in the final quartile C_{Di} for each direction *Di* are calculated using Eq. 5.1. $\mu_{FC(i)}$ is the average feature count moving in direction *i*.

$$C_{Di} = \frac{\mu_{FC(i)}}{4} \tag{5.1}$$

After obtaining feature counts in the final quartile, the average velocity $\mu_{V(f,m,i)}$ in final quartile for each frame *f*, mask *m* and direction *i* are calculated using Eq. 5.2 where $V_j(f,m,i)$ denotes the velocities in frame *f*, mask *m*, direction *I* and quartile *j*.

$$\mu_{V(f,m,i)} = \frac{\sum_{j \in Q4} V_j(f,m,i)}{C_{Di}} \tag{5.2}$$

Finally, to smooth out the noise present in the dataset, velocity average of all videos is taken using Eq. 5.3 obtaining single training data matrix.

$$TrD = \frac{\sum_{v \in V} \mu_{v(f,m,i)}}{V} \tag{5.3}$$

After obtaining this matrix, noise injection is performed to represent the abnormal activity in the scene using Eq. 5.4.

$$N_i = \alpha * \mu_{tr} + Z \tag{5.4}$$

Where μ_{tr} is the average velocity of TrD and Z is random noise with parameters N(0,1). The coefficient α is defined as in Eq. 5.5.

$$\alpha = \frac{\mu_{tr} + s * \sigma_{tr}}{\mu_{tr}} \tag{5.5}$$

Where σ_{tr} is the standard deviation of TrD and s is the shifting parameters. Under the Gaussian distribution assumption, anomalies normally would be present after the *Mean + 3* Std* threshold which is why s is selected as 3.

Using the normal training matrix and artificial noise, two coupled HMMs are trained. The adjacent nodes of the CHMMs represent the spatial zones and the slices represent

the time points. Later a classification is performed based on the likelihood of the model. The results of this work are given in Section 6.3.

CHAPTER 6

EXPERIMENTAL RESULTS AND COMPARISONS

In this chapter, the evaluation of the proposed method is provided. First, the dataset that is used to evaluate the method is explained in detail. Later, the results on the dataset and comparison with the existing literature are presented. Finally, strengths and weaknesses of the proposed method are discussed.

6.1. Dataset

To evaluate the robustness and performance of the proposed anomaly detection method (presented in Section 3), UCSD Peds dataset has been used. This dataset contains videos from two different camera angles. There are only naturally occurring anomalies present in this dataset (i.e. the anomalies are not staged). The anomalies include carts, bikes, motorbikes and skaters. The authors of this dataset also classified wheelchairs and baby carts as anomalies. Although those entities cannot be considered anomalies in pedestrian walkway context, they are taken as anomalies in this work for accurate comparison to the methods in the literature.

6.1.1. Peds1 Dataset

Peds1 Dataset contains 34 training videos which consist only of normal frames and 36 test videos which consist of both normal and abnormal frames. Each video contains 200 frames of size 238 x 158. This dataset has a perspective problem due to camera angle and to overcome this, a partitioning approach is used as mentioned in Section 3.1.2. A sample frame from this dataset is given in Figure 6.1. The test dataset of Peds1 is explained in Table 6.1. This table explains which anomalies are present in videos and what characteristics they have.

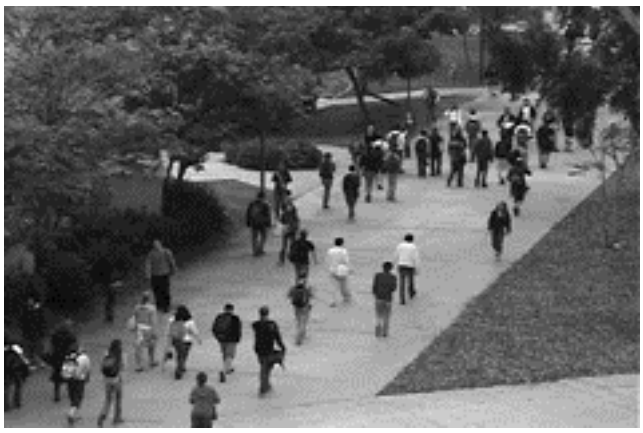

Figure 6.1 Sample frame of Peds1 dataset

Table 6.1 Details of Peds1 test dataset

Video	No. of Frames	Anomalous Frames	Anomaly Description
1	200	From 60 to 152	A fast paced motorbike
2	200	From 50 to 175	A Fast Paced Bike and A Fast Paced Skater
3	200	From 91 to 200	A normal pedestrian paced bike
4	200	From 31 to 168	A normal pedestrian paced skater
5	200	From 5 to 90 From 140 to 200	2 fast paced bikes and 1 normal pedestrian paced bike
6	200	From 1 to 100 From 110 to 200	2 fast paced bikes, 2 fast paced skaters and 1 normal pedestrian paced bike
7	200	From 1 to 175	2 moderately fast paced bike and 1 moderately fast paced skater
8	200	From 1 to 94	A fast paced skater
9	200	From 1 to 48	Pedestrian walking on grass
10	200	From 1 to 140	A normal pedestrian paced skater
11	200	From 70 to 176	A pedestrian walking in the wrong direction
12	200	From 130 to 200	A fast paced skater
13	200	From 1 to 156	A baby cart
14	200	From 1 to 200	2 fast paced bikes and 1 moderately fast paced cart
15	200	From 138 to 200	A fast paced bike
16	200	From 123 to 200	A fast paced bike
17	200	From 1 to 47	A fast paced bike
18	200	From 54 to 120	A fast paced skater
19	200	From 64 to 138	A fast paced cart
20	200	From 45 to 175	A fast paced cart
21	200	From 31 to 200	A wheelchair
22	200	From 16 to 107	A fast paced skater
23	200	From 8 to 165	A moderately fast paced wheelchair
24	200	From 50 to 171	A fast paced cart and a moderately fast paced skater
25	200	From 40 to 135	A fast paced skater
26	200	From 77 to 144	A fast paced bike and a normal pedestrian paced bike
27	200	From 10 to 122	A fast paced cart
28	200	From 105 to 200	A fast paced bike
29	200	From 1 to 15 From 45 to 113	2 fast paced bikes

30	200	From 175 to 200	A moderately fast paced bike
31	200	From 1 to 180	2 fast paced bikes
32	200	From 1 to 52 From 65 to 115	2 fast paced bikes
33	200	From 5 to 165	3 fast paced bikes
34	200	From 1 to 121	A fast paced skater
35	200	From 86 to 200	A fast paced skater
36	200	From 15 to 200	A fast paced cart and a normal pedestrian paced bike

6.1.2. Peds2 Dataset

Peds2 dataset contains 16 training videos which consist of normal frames and 12 test videos which consist of both normal and abnormal frames. Each video contains varying number of frames and the frame size is 360 x 240. This dataset does not contain a perspective problem, therefore no partitioning is used in this one. A sample image of this dataset is provided in Figure 6.2. The test dataset of Peds2 is explained in Table 6.2. This table explains which anomalies are present in videos and what characteristics do they have.

Figure 6.2 Sample frame of Peds2 dataset

Table 6.2 Details of Peds2 test dataset

Video	No. of Frames	Anomalous Frames	Anomaly Description
1	180	From 55 to 180	A moderately fast paced bike
2	180	From 90 to 180	A fast paced bike
3	150	From 1 to 146	2 moderately fast paced bike
4	180	From 27 to 180	A moderately fast paced cart and a

			moderately fast paced bike
5	150	From 1 to 130	A bike with normal pedestrian pace
6	180	From 1 to 160	Two fast paced bikes
7	180	From 1 to 160	One moderately fast paced bike, one bike with normal pedestrian pace and a fast skater
8	180	From 1 to 180	One fast paced bike and one fast paced skater
9	120	From 1 to 120	One fast paced bike and one fast paced skater
10	150	From 1 to 150	A moderately fast paced bike
11	180	From 1 to 180	A moderately fast paced bike
12	180	From 84 to 180	A skater with normal pedestrian pace

6.2. Results and Comparison

To evaluate the performance of the proposed system, a comparison with the techniques in the literature has been performed. The AUC values, ROC curves and computation times are reported and compared to the existing methods.

Also, to illustrate the performance of the density switch method, a sample of matched frames is provided.

6.2.1. Density Switch Results

In this section, the results for the proposed density switch method are illustrated with sample frames from the testing and training videos. The selected training video for the given test frame is also provided to further explain the result of the density switch methodology in Figure 6.3.

The frames on the left are taken from the test videos. The first frame on the left does not have any anomalies while the other frames have a biked entity. The frames on the right are all from the matched video for the frame on the left using the density switch methodology. As can be seen from the frames, the pedestrian density and the motion directionality are also very similar.

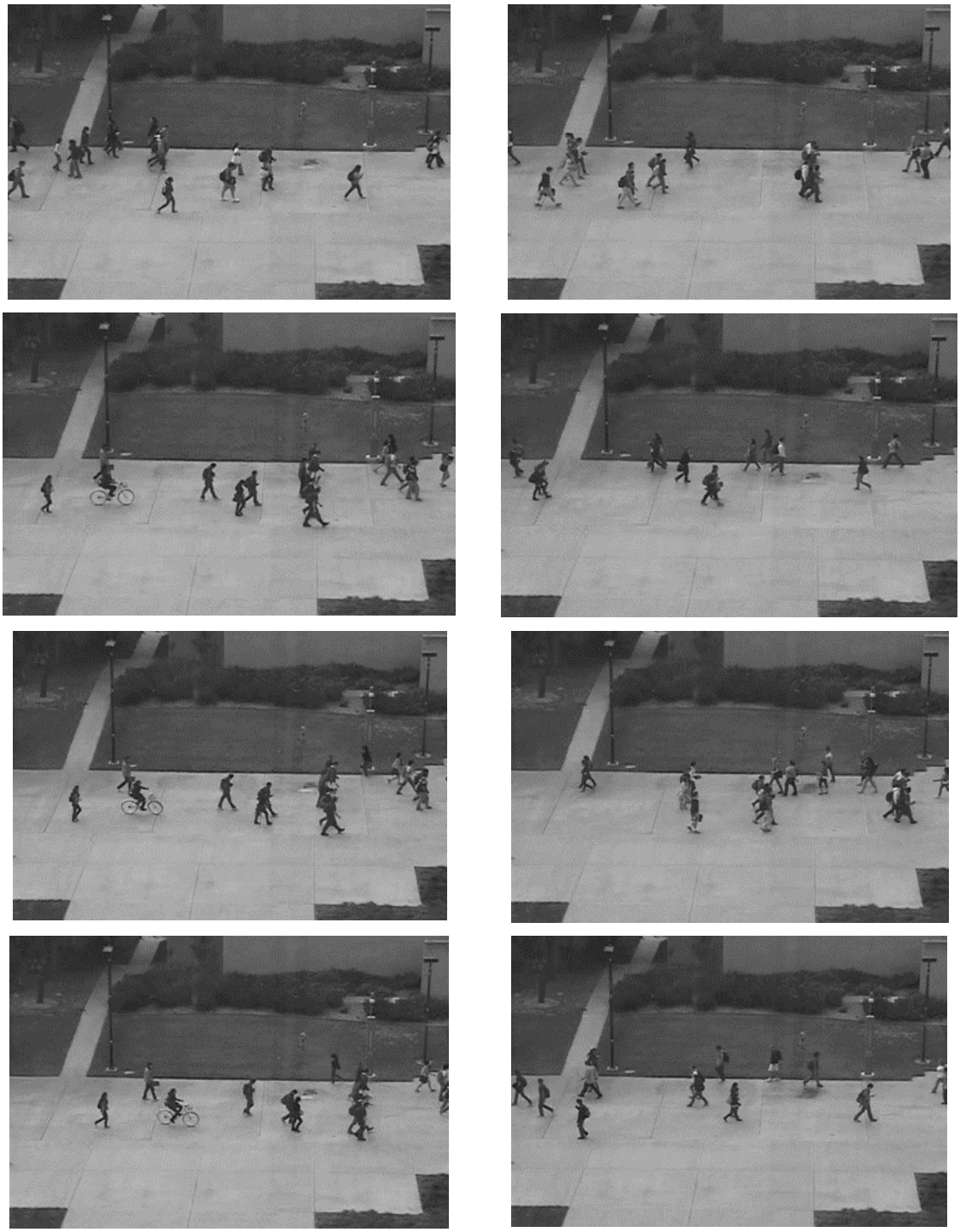

Figure 6.3 Matched training frames (right) for the given test frames (left) by density switch

As can be seen from the figure, the densities of the frames are quite close to each other. However they are not exactly the same. The reason for that is, the method proposed in this thesis also considers other aspects of the scene than the feature count as explained in Section 2.2.

The feature count difference has a mean of -88.7256 and a standard deviation of 198.1397. There are two reasons why the mean is different than zero. The first one is that this approach also considers other statistics while selecting a model. The second one is that, both training and testing videos have frames where there is very little activity while the other one has some activity. This increases the difference between feature counts. The plot for difference in feature counts is given in Figure 6.4. This figure is generated with 100 randomly selected frames from the dataset.

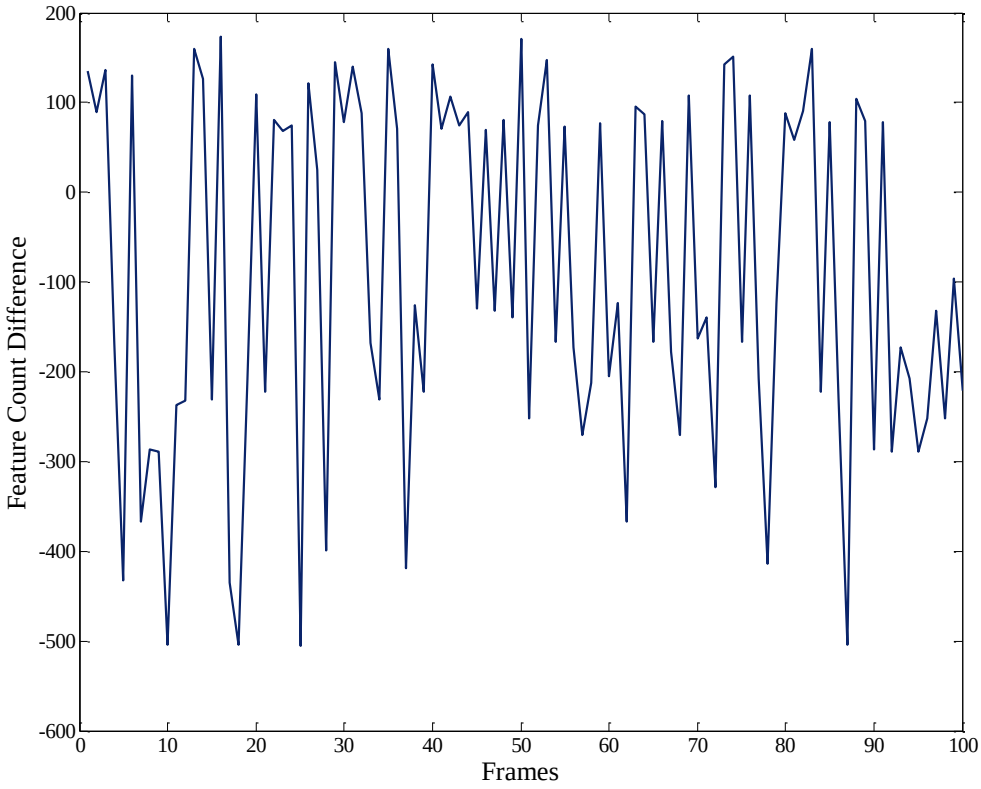

Figure 6.4 Feature Count Difference Between Training and Test Frames

6.2.2. Anomaly Detection Results

In this chapter, results for the anomaly detection task are provided. The section is divided into two subsections for each separate dataset. Although the data that we used is all grouped in "Peds" dataset, the sub-datasets in this large dataset employ different characteristics. To avoid any confusion, the results on those datasets are given in separate sections. The proposed method is referred to as "HMM" in all of the graphics provided.

6.2.2.1. Peds1 Dataset

This section explains the results obtained using the Peds1 dataset. This dataset is challenging due to the perspective problem it has.

The ROC curve for the anomaly detection is provided in Figure 6.5. In this ROC curve, a comparison between the state of the art techniques and the proposed system is provided. Also, in order to understand the impact of density switch methodology, the algorithm is run without the density switch methodology. Instead the trained model is selected randomly from the pool of training videos. This process is repeated 5 times and the average EER and AUC values of the proposed method without density switch are provided in Table 6.3. The standard deviations of the values are given in parentheses in the table.

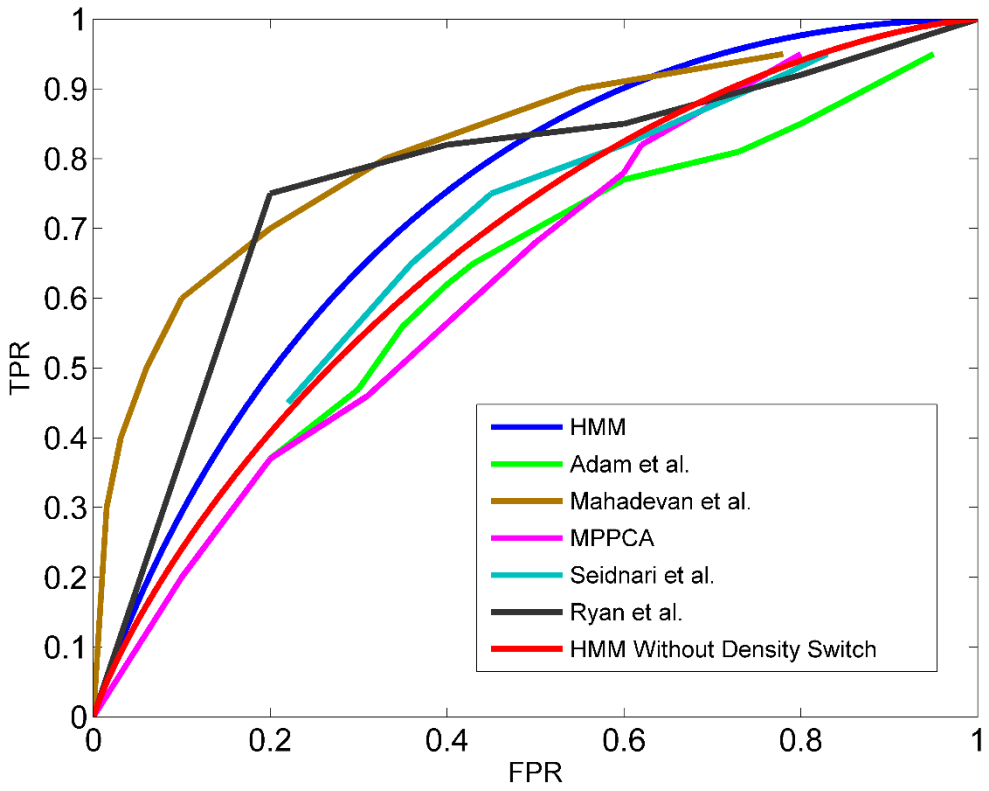

Figure 6.5. ROC curve comparison for Peds1 dataset

The AUC values and EER values are provided in Table 6.3.

Table 6.3 AUC and EER comparison of Peds1

	Peds1	
System	**EER**	**AUC**
Proposed Sys.	0.33	0.74
Proposed Sys. Without Density Switch	0.367 (Std=0.016)	0.68 (Std=0.014)
Ryan et Al. [26]	**0.23**	**0.838**
MDT [39]	0.25	-

Adam et Al. [25]	0.38	-
MPPCA [22]	0.40	-
Seidnari et Al. [29]	0.39	-

The proposed algorithm has an EER value of 0.33 and an AUC value of 0.74 which are better than the competing methods except MDT and Ryan et al [26]. The methods which did not report the AUC values are represented in the table with a dash. EER value is the intersection between a 135^{o} line and the ROC curve. Therefore the higher EER generally indicates a lower AUC.

6.2.2.2. Peds2 Dataset

This section explains the results obtained using the Peds2 dataset which is explained in Section 6.2.2.

The ROC curve for the anomaly detection is provided in Figure 6.6. In this ROC curve, a comparison between the state of the art techniques and the proposed system is provided. Also, in order to understand the impact of density switch methodology, the algorithm is run without the density switch methodology. Instead the model is selected randomly.

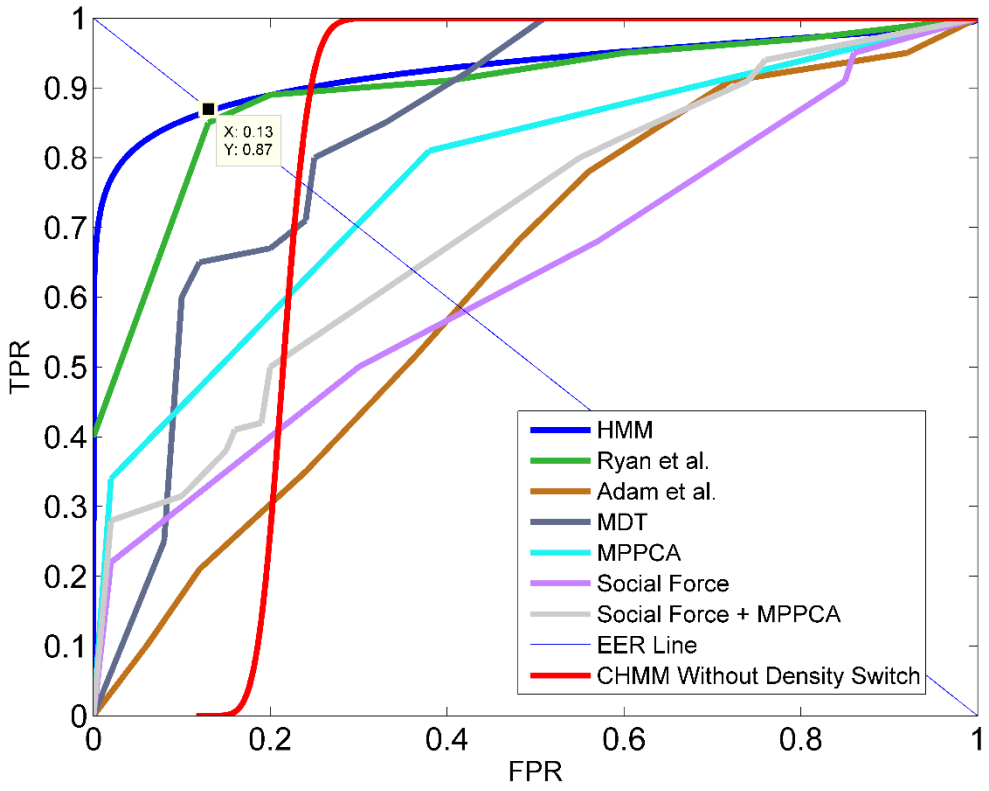

Figure 6.6 ROC curve comparison for Peds2 dataset

Table 6.4 EER and AUC comparison of Peds2 dataset

	Peds2	
System	**EER**	**AUC**
Proposed Sys.	**0.13**	**0.945**
Proposed Sys. Without Density Switch	0.21	0.77
Ryan et Al. [26]	0.13	0.934
MDT [39]	0.25	-
Adam et Al. [25]	0.42	-
SF-MPPCA [39]	0.36	-
MPPCA [22]	0.30	-
SF [3]	0.42	-

The proposed system has an EER of 0.13 and an AUC of 0.945. These are the best results in the literature. When there is no perspective problem, the proposed method outperforms other methods in the literature. The method which is closest to our method in terms of accuracy is Ryan et al. [26].

In both cases, the density switch methodology improves the results. However, it improves the results obtained in Peds2 data set more than Peds1. The reason for that is the underperformance of the solution to the perspective problem.

6.2.3. The Performance of the Algorithm

The speed of the algorithm depends on the size of the training set. Because, during density switch, each new testing frame is compared to each training frame for similarity. Since this is the case, the larger the training set to perform density switch with the longer the algorithm takes. A comparison between these values and the existing literature is provided in Table 5.5.

Table 6.5 Speed Comparison for Peds dataset of HMM Approach

Method	**Peds1 FPS**	**Peds2 FPS**
Proposed Method	6.4	**94.34**
Ryan et al. [26]	**9.4**	9.4
MDT [39]	0.04	0.04

The comparison with other methods is not given in this table because the authors did not provide the runtime measurements. In this table, it is visible that when the training set is not large the proposed method works very fast, in fact faster than other existing methods. When training set gets larger, the density switch needs to analyze more frames in order to compute similarity between the testing frame and the training frame. When training set is large, the speed difference between proposed method and existing faster methods is not very large.

The computational complexity of the algorithm is O(*N**M) where *N* is the number of frames and *M* is the number of overlapping spatial masks. The number of masks increases the passes over the training data. Assume there are *M* number of masks, in this case, *M* number of passes are done over the entire training data, resulting in the O(*N**M) complexity.

As mentioned in Section 6.3.2.1, MDT outperforms our method in terms of accuracy, however due to its high computational complexity; this method is not suitable for real-time applications.

The experiments in this thesis are conducted using a personal computer that has Windows 7 OS, Intel i7-3630QM 2.4 GHz CPU and 8 GB RAM. The feature detection and matching phase are carried out using OpenCV [37] in Visual Studio 2010 with C++ language. The model construction, density switch and anomaly detection are done using MATLAB. The model construction phase is carried out via Bayes Net Toolbox [40]. The setup of Ryan et al. [26] is not provided in the paper. The setup for MDT [39] is a Pentium machine with 3GHz CPU and 2GB RAM.

The limitations of this algorithm are based on the fact that this algorithm makes a velocity based anomaly detection. The biggest limitation is not being able to detect textural anomalies along with velocity based anomalies. Also, in case of an anomalous entity with velocity that is consistent with the normal behavior, the false negatives occur. The solution to that problem might be utilizing a feature count based anomaly detection or textural anomaly detection.

6.2.4 Effect of Noise Coefficient

The effect of noise coefficient is provided in Figure 6.7. This figure depicts the effect of noise coefficient on accuracy and AUC.

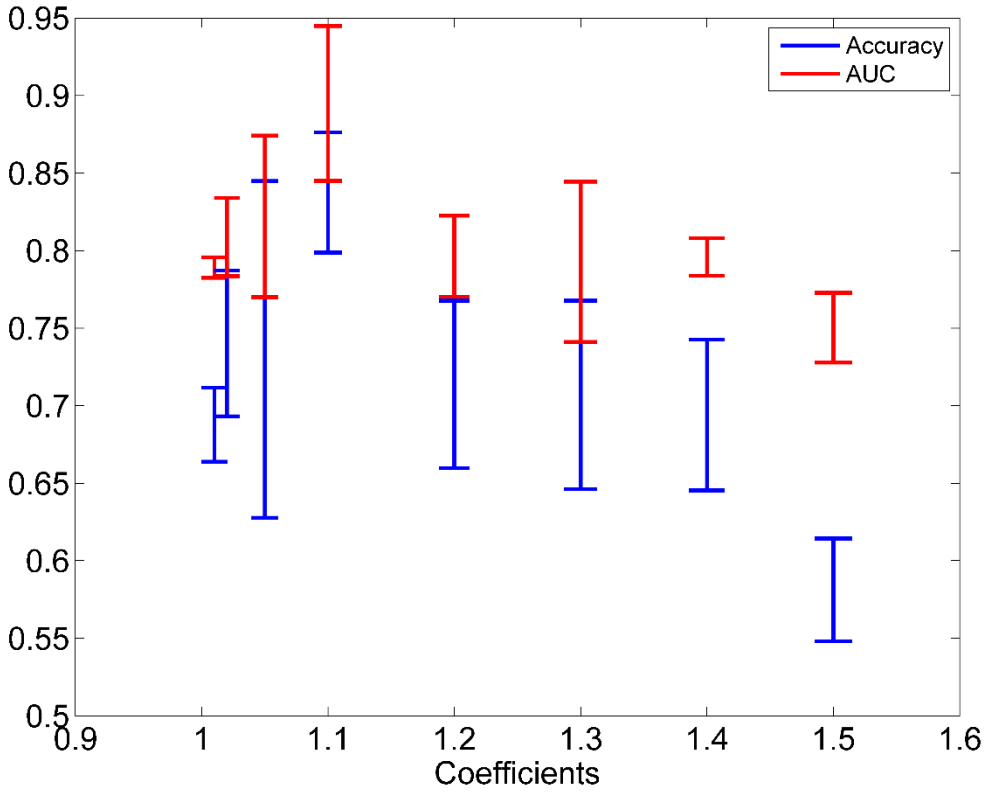

Figure 6.7. Effect of Coefficients on Accuracy and AUC

Initially, the coefficients start with 1.01. When the noise is generated with that coefficient, a lot of the normal values which are slightly higher than the general speed of the pedestrians are detected as anomalies, which is why both accuracy and AUC are low for this coefficient. The same idea applies to coefficient 1.02 which is the second coefficient in this plot however, we can see some improvement in both accuracy and AUC. Later, the coefficient is increased to 1.05 and with that increase the performance improves. The AUC is around 0.82 and accuracy is around 0.77. The coefficient is increased further to 1.1, and the plot shows that this is where the method performs best. If the coefficient is increased further than 1.1, the performance drops significantly. This is due to the fact that the method is no longer able to detect subtle anomalies. "Subtle anomalies" defines the anomalies which have velocities slightly higher than the pedestrian velocities.

As can be seen from the plot, there is some variability in the results represented using "error bars". This is due to the fact that the noise generation is done completely randomly. However, even if this is the case, the performance pattern based on the coefficients is visible in the plot.

6.3. CHMM Based Pedestrian Motion Model Results

The results of the CHMM Based Pedestrian Motion Model study mentioned in Section 2.5 are given in this section. The results and the comparison with the methods existing in the literature when the study was conducted are given in Figure 6.8. Figure 6.8 depicts the ROC Curves obtained.

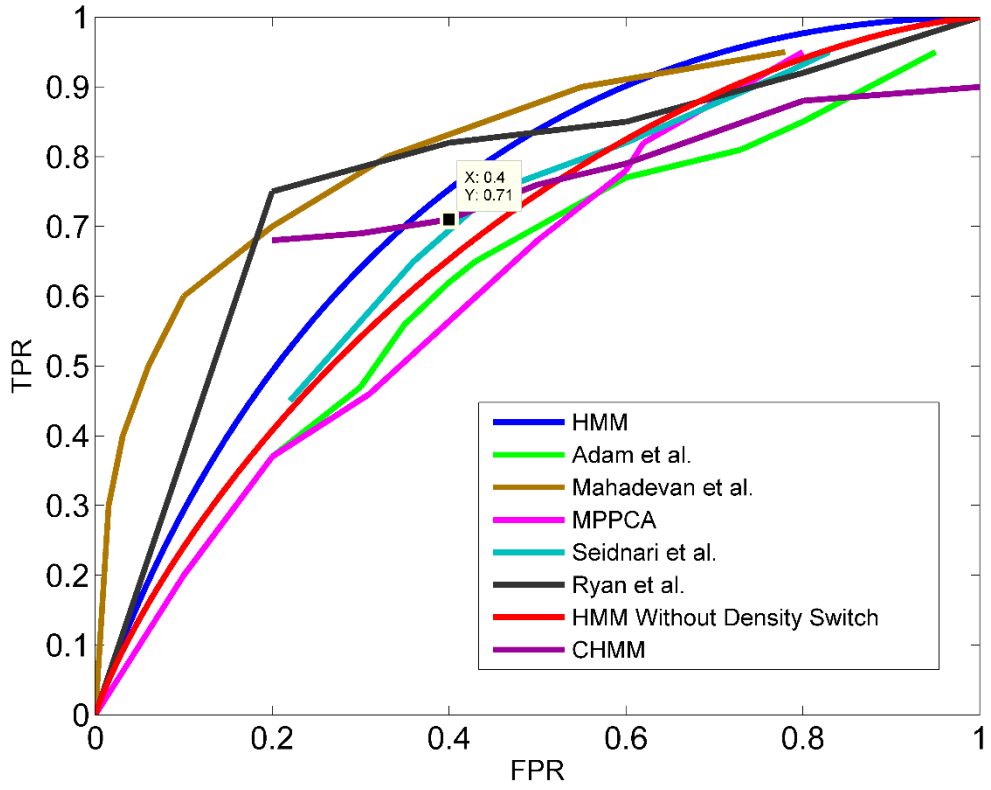

Figure 6.8. ROC curve comparison for CHMM Method on Peds1 dataset

In comparison with the method proposed in this thesis based on the ROC curves, there is an improvement from the CHMM method. The accuracy of the CHMM algorithm is given in Table 6.6 and performance of the CHMM algorithm is given in Table 6.7.

Table 6.6 AUC and EER comparison of Peds1 (CHMM included)

	Peds1	
System	**EER**	**AUC**
Proposed Sys.	0.33	0.74
Proposed Sys. Without Density Switch	0.367 (Std=0.016)	0.68 (Std=0.014)
CHMM Approach	0.39	0.67
Ryan et Al. [26]	**0.23**	**0.838**
MDT [39]	0.25	-
Adam et Al. [25]	0.38	-

MPPCA [22]	0.40	-
Seidnari et Al. [29]	0.39	-

Table 6.7 Speed Comparison for Peds1 dataset of CHMM Approach

Method	**Peds1 FPS**
Proposed Method (HMM)	6.4
CHMM	**34**
Ryan et al. [26]	9.4
MDT [39]	0.04

As can be seen from the table CHMM clearly outperforms all the methods (including the proposed method) in terms of runtime performance. However, its accuracy is also lower than the proposed method. CHMM method was not tested on Peds2 Dataset.

CHAPTER 7

CONCLUSION AND FUTURE WORK

In this work, a pedestrian zone anomaly detection method that can compete with the existing methods is presented. An important contribution of this work is considering the underlying crowd density and velocity behavior information in the density switch stage. The contributions of this work can be summarized as follows:

- Two different methods have been proposed for anomaly detection, namely CHMM and HMM approach. HMM approach has higher accuracy while CHMM approach is faster.
- A framework that takes the crowd density and velocity behavior information into account while performing anomaly detection.
- Representing the pedestrian behavior with statistics obtained from the raw velocity data that is obtained by feature matching and velocity calculation for each detected feature point.
- Real-time application for HMM approach in cases where the training data set size is not very large and near real-time application in large datasets. Real-time application for CHMM approach without any constraints.
- Mimicking the anomalous behavior using the noise injection. This enables the user to use only normal data for training both the normal and abnormal cases. If the scene has different characteristics, injected noise can be adapted to this abnormal case.
- The method uses an artificial noise injection to learn the anomalous events which allows the users to remove any threshold for anomaly detection. It also enables users to mimic other anomalies they may want to detect.
- The proposed framework mostly works without any user defined parameters. The only parameter that needs to be manually selected is the noise injection parameter.
- Using the domain knowledge regarding the pedestrian motion such as the skewness and standard deviation of the data. This shows how having domain knowledge affects the modelling simplicity. Unlike other methods in the literature, we used the domain knowledge to extract statistics from the pedestrian velocity calculations and this resulted in a very simple model that is able to work well with only representative statistics of the raw data. This improves the runtime performance as well as the accuracy of the system. According to the results provided above, it can be said that the proposed algorithm can compete with the

existing algorithms in the literature in terms of both accuracy and speed. However, the performance of the proposed method degrades when there is a perspective problem. Also, as the training set gets larger, the algorithm works slower.

In Peds2 dataset, none of these problems is present. Even though the proposed method is able to overcome those issues to some extent, it still performs the best where the aforementioned problems do not exist. The algorithm is able to work in real-time when the training set is not very large or number of spatial masks are small.

In Peds1 dataset, due to perspective problem some anomalies cannot be detected. For instance when the anomaly is further from the camera, the false negative rate increases. Also for both of the datasets, if the anomaly has the same velocity with the pedestrians it cannot be detected.

In terms of accuracy, there are some missed alerts where the velocity of the anomalous entity is not different from the pedestrian velocities. However the work proposed is able to distinguish between pedestrians and anomalous entities even the difference is very small. The number of missed alerts can be reduced by including a textural anomaly detection system.

Generally, both of the proposed approaches work at least as well as the existing methods in the literature while outperforming them in some cases. HMM method outperforms CHMM method in terms of accuracy however CHMM method is faster than HMM. This is due to the fact that CHMM does not utilize a switching mechanism that searches the entire training data space for an appropriate training set. However, using the average mean velocity vector as the training resulted in lower accuracy.

It is shown in the work that density switch improves the anomaly detection results. This is also a major contribution of this work.

For future work, the possible improvements could be summarized as follows:

- Texture information for detecting the abnormal entities with normal velocity.
- The major direction selection could be automatized. Currently, it depends on the manual analysis of the training set.
- The scene partitioning could be automatized or perspective normalization could be adopted. Or, instead of partitioning, scene normalization can be used. Currently, the proposed method works best when there is no perspective problem.

REFERENCES

[1] Taylor, "Lord Taylor's final report on the Hillsborough stadium disaster," 1990.

[2] D. Helbing and P. Mukerji, "Crowd disasters as systemic failures: analysis of the Love Parade disaster," *EPJ Data Science,* 2012.

[3] R. Mehran, A. Oyama and M. Shah, "Abnormal crowd behavior detection using social force model," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision,* vol. 60, no. 2, pp. 91-110, 2004.

[5] H. Bay, T. Tuytelaars and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV* , 2006.

[6] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 IEEE International Conference on Computer Vision (ICCV)*, 2011.

[7] A. Marana, L. da Fontoura Costa, R. Lotufo and S. Velastin, "Estimating crowd density with Minkowski fractal dimension," *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on,* vol. 6, pp. 3521-3524, 1999.

[8] J. C. Russ, Fractal Surfaces, New York: Plenum Press, 1994.

[9] R. Haralick, "Statistical and structural approaches to texture," in *Proceedings of the IEEE*, 1979.

[10] R. Haralick, K. Shanmugam and I. Dinstein, "Textural Features for Image Classification," *Systems, Man and Cybernetics, IEEE Transactions on* , Vols. SMC-3, no. 6, pp. 610-621, 1973.

[11] K. T., "The Self-Organizing Map," in *Proceedings of the IEEE,* 1990.

[12] A. Marana, S. Velastin, L. Costa and R. Lotufo, "Automatic estimation of crowd density using texture," *Safety Science,* vol. 28, no. 3, pp. 165-175, 1998.

[13] H. Rahmalan, M. Nixon and J. Carter, "On Crowd Density Estimation for Surveillance," in *Crime and Security, 2006. The Institution of Engineering and Technology Conference on*, 2006.

[14] R. Mukundan, "Discrete Orthognal Moment Features," in *Proceedings of International Conference on Image and Visual Computing*, 2000.

[15] V. Subburaman, A. Descamps and C. Carincotte, "Counting People in the Crowd Using a Generic Head Detector," in *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on* , 2012.

[16] R. E. S. Yoav Freund, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119-139, 1997.

[17] W. Ge and R. Collins, "Evaluation of sampling-based pedestrian detection for crowd counting," in *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on* , 2009.

[18] M. Harkness and P. Green., "Parallel Chains, Delayed Rejection and Reversible Jump MCMC for Object Recognition," in *BMVC*, 2000.

[19] R. Ma, L. Li, W. Huang and Q. Tian, "On pixel count based crowd density estimation for visual surveillance," in *Cybernetics and Intelligent Systems, 2004 IEEE Conference on*, 2004.

[20] A. E. Gunduz, T. T. Temizel and A. Temizel, "Density estimation in crowd videos," in *Signal Processing and Communications Applications Conference (SIU)*, 2014.

[21] J. Kim and K. Grauman, "Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009.

[22] C. B. M. Tipping, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, no. 2, pp. 443-482, 1990.

[23] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, no. 5, pp. 4282-4286, 1995.

[24] D. M. Blei, A. Y. Ng and M. I. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, pp. 993-1022, 2033.

[25] A. Adam, E. Rivlin, I. Shimshoni and D. Reinitz, "Robust Real-Time Unusual Event Detection using Multiple Fixed-Location Monitors," in *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2008.

[26] D. Ryan, S. Denman, C. Fookes and S. Sridharan, "Textures of optical flow for real-time anomaly detection in crowds," in *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, 2011.

[27] R. Raghavendra, A. Bue, M. Cristani and V. Murino, "Abnormal Crowd Behavior Detection by Social Force Optimization," in *Human Behavior Understanding Lecture Notes in Computer Science*, Amsterdam, Springer, 2011, pp. 134-145.

[28] M. A. Fischler and R. c. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381-395, 1981.

[29] M. B. D. B. Lorenzo Seidenari, "Dense spatio-temporal features for non-parametric anomaly detection and localization," in *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams*, 2010.

[30] P. Guler, A. Temizel and T. T. Temizel, "Real-Time Global Anomaly Detection for Crowd Video Surveillance Using SIFT," in *International Conference on Imaging for Crime Detection and Prevention*, 2013.

[31] A. Gunduz, A. Temizel and T. Taskaya Temizel, "Feature detection and tracking for extraction of crowd dynamics," in *Signal Processing and Communications Applications Conference (SIU), 2013 21st*, April 2013.

[32] D. Marr and E. Hildreth, "Theory of Edge Detection," *Proceedings of the royal society*, vol. 207, no. 1167, pp. 187-217, 1980.

[33] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision*, 2006.

[34] P. L. Rosin, "Measuring corner properties," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 291-307, 1999.

[35] K. Daniilidis, P. Maragos and N. Paragios, "BRIEF: Binary Robust Independent Elementary Features," in *Computer Vision – ECCV 2010 Lecture Notes in Computer Science*, Springer, 2010, pp. 778-792.

[36] M. Muja and D. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Applications*, 2009.

[37] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[38] C. Silpa-Anan and R. Hartley, "Optimised KD-trees algorithm for fast image descriptor matching," in *CVPR*, 2008.

[39] V. Mahadevan, W. Li, V. Bhalodia and V. Nuno, "Anomaly Detection in Crowded Scenes," in *CVPR*, 2010.

[40] K. Murphy, "BayesNet Toolbox," [Online]. Available: https://github.com/bayesnet/bnt.

TEZ FOTOKOPİSİ İZİN FORMU

ENSTİTÜ

Fen Bilimleri Enstitüsü	
Sosyal Bilimler Enstitüsü	
Uygulamalı Matematik Enstitüsü	
Enformatik Enstitüsü	X
Deniz Bilimleri Enstitüsü	

YAZARIN

Soyadı : Gündüz
Adı : Ayşe Elvan
Bölümü : Bilişim Sistemleri

TEZİN ADI (İngilizce) : ANOMALY DETECTION USING SPARSE FEATURES AND SPATIO-TEMPORAL HIDDEN MARKOV MODEL FOR PEDESTRIAN ZONE VIDEO SURVEILLANCE

TEZİN TÜRÜ : Yüksek Lisans [X] Doktora [ ]

1. Tezimin tamamından kaynak gösterilmek şartıyla fotokopi alınabilir. [X]
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: