Automatic Human Detection and Tracking in Crowded Scenes using Histogram of Oriented Gradients (HOG) and Particle Filter

Doctoral Course Thesis

BHUYAIN MOBAROK HOSSAIN
March 2019

Automatic Human Detection and Tracking in Crowded Scenes using Histogram of Oriented Gradients (HOG) and Particle Filter

By
BHUYAIN MOBAROK HOSSAIN

Doctoral course Thesis approved by the Graduated School of Advanced Technology and Science for the Degree of

Doctor of Engineering
In
System Innovation Engineering

March 2019
Department of Information Science
and
Intelligent Systems
Tokushima University, Tokushima 770-8506, Japan

Automatic Human Detection and Tracking in Crowded Scenes using Histogram of Oriented Gradients (HOG) and Particle Filter

Curriculum Vitae

Bhuyain Mobarok Hossain was born on February 2, 1976 in Bangladesh. After he finished higher Secondary School, he enrolled in Department of Economics at National University in 1994, and received a B.S.S. degree in Economics in 1997. He entered department of Civil and Environment at the Tokushima University, Japan for Master degree in October 2009 and graduated in September 2012. After, he enrolled for the doctoral degree at the department of Civil and Environment at Tokushima University, Japan in October 2012 and researched until 2015 September. After that, he enrolled to the department of Information Science and Intelligent system at Tokushima University in October 2015. His main research field is human detection and tracking using particle filter and histogram oriented gradients.

Abstract

Recently, attention has been focused on monitoring crowded areas using video surveillance systems to provide security, safety, monitor human activity, etc. In video surveillance systems human detection and tracking is a very important obligation. Although it is a well-known subject, many challenges are yet to be resolved for real world applications. These include, changes in illumination, camera motion, independent and random human movements, etc. On the other hand, interesting information such as interaction between people or between them and vehicles can also be obtained. Crowded scenes include public places such as airports, bus station, concerts, subway, religious festivals, football matches, railway stations, shopping malls, etc., where many people gather is a challenge for those interested in safety and security systems. In such cases, for effective video surveillance, we may need to monitor a specific place or area. One application of such a system is in crowd control to maintain the general security in public places.

However, the main limitation of video surveillance system is the required continuous manual monitoring especially in crime deterrence. In order to assist the security personnel monitoring live surveillance systems, target detection and tracking techniques can be used to send a warning signal automatically. To realize such a system, in this thesis, we propose an innovative method to detect and track a target person in crowded area using an individual's features.

In our thesis, we realize automatic detection and tracking by combining Histogram of Oriented Gradients (HOG) feature detection with a particle filter. The HOG feature is applied for the description of contour detection in humans, while the particle filter is used for tracking the targets using skin and clothes color based on a feature of target person. In this thesis we use HSV (Hue, Saturation, and Value) color space for preprocessing and personal feature extraction because RGB (red, green, blue) color is not stable as it is affected by the environmental conditions especially when lighting is changed. We have developed the evaluation system implementing our thesis, and have achieved high accuracy detection rate and tracked the specific targets precisely. Moreover, another challenge of research in crowded scenes is frequent and instantaneous occlusion of the target by other objects. The visual occlusions and ambiguities in crowded scenes are complex, making

scene semantics difficult to analyze. In this thesis, we have focused on the overlapping issues in the case of a target person wearing similar clothes aiming to overcome the occlusion problem.

In experimental results, our system achieved more than 90% accuracy in human detection and particle filter tracking specific person accuracy of around 99% in all dataset.

Acknowledgments

First of all, I am grateful to Almighty Allah, the most merciful, the most beneficent, for giving me the opportunity to carry out this research work in Tokushima University, Japan and most importantly provide me with patience and strong determination to successfully complete this work.

It is my privilege to express my sincere sense of gratitude and whole-hearted indebtedness to my guide, my supervisor Professor Stephen Karungaru, Department of Information Science and Intelligent Systems, Tokushima University, for his supervision, all kind support, persistent encouragement and assistance during the course of the research work. I very much appreciate his tireless effort, continuous cooperation, and advice and deep insight to complete my study and always his confidant face encouraged me to solve this critical problem. I cannot forget his patience and kind behaviors when he deals with me throughout the study period. I also thank him for introducing me with the cutting edge research of image processing fields of information science and intelligent systems. My thanks and appreciation goes to all of my colleague and friends in the department for their education support and unconditional friendship.

I also would like to express my deepest appreciation to my Professor Kenji Terada and Dr. Akinori Tsuji for their sincere and cooperative supervision, recommendations and support throughout the research work, which made it easy to carry out my research work and to prepare this research paper.

I would like to express my thanks to my beloved wife Begum Nadira, my beloved sons Abir Mahafuz Rahman and my beloved daughter Begum Mahibah. It would not have been possible to complete my work without their unconditional love, support, patience, prayer and supplication. My deep gratitude towards them for understanding my business and for sacrificing their joy for me. Especially when it was close to finish my work I had to work throughout day and night, and they accept all of my limitation and provide their full support without which I could not complete my study. I also wish to express my deeply felt gratitude to my previous supervisor professor Hitoshi Imai and professor Sakaguchi Hideo. I also express my deep appreciation to all of my friends in Japan especially in Tokushima for their continuous support and encouragement.

Moreover, I extend my sincere gratitude and appreciation to Prof. Tetsushi Ueta, Prof. Norihide Kitaoka and Prof. Masami Shishibori for their tireless effort to review and evaluate this thesis. Their details checks, comments and advice was invaluable. I will forever appreciate their input.

In the end, I am thankful to the almighty for blessing me to complete my work peacefully and successfully.

Bhuyain Mobarok Hossain

Tokushima University

Table of Contents

List of Figures....12

List of Tables....14

Publications....15

Journal papers....15

Conference papers....15

Chapter 1....16

Introduction....16

1.0 Background....16

1.1 Motivation....17

1.2 Challenges....18

1.3 Research Approach....18

1.3.1 Contribution of this work....18

1.4 Computer Vision Introduction....20

1.4.1 Moving Object Tracking....20

1.4.2 Measurement information in video surveillance....21

1.5 Chapter information....21

Chapter 2....23

Related Work....23

2.1 Introduction....23

2.2 Visual observation details....23

2.3 Literature Review....23

2.4 Human detection and Tracking....24

2.4.1 Single scene based detection approaches....24

2.4.2 Multi- scene detection methods....26

2.4.3 Real time people tracking system for security....27

Chapter 3....28

Theoretical Background....28

3. 1 Histogram of Oriented gradients (HOG)....28

3.1.1 HOG Original Work....28

3.1.2 Detection window....28

3.1.3 Step 1: Calculation of the image gradient for each pixel....29

3.1.4 Step 2: Calculate Gradient Histogram in 8x8 cells....30

3.15 Step 3: Block Normalization....32

3.16 Step 4: Calculate the HOG feature vector....33

3.2 Support Vector Machines 33

3.3 Sequential Monte Carlo (SMC) 34

3.3.1 Recursive State Estimation 35

3.3.2 Particle filter..37

Chapter 4 39

Proposed Human Detection and Tracking 39

4.1 Introduction..39

4.2 Human Detector Design...41

4.2.1 HOG Feature 41

4.2.2 The HOG-SVM object detector 45

4.2.3 HOG-SVM Detector 47

4.3 Human Tracking Design..48

4.3.1 Particle filter..48

4.4 Pre-processing 50

4.5 Image Pyramids 51

5.4.1.1 Image pyramid parameters...52

Chapter 5 53

Experiments and results 53

5.1 Introduction..53

5.2 Experiments and Results...53

5.2.1 HOG Feature extraction...53

5.2.2 Basic HOG-SVM detector Training and Testing..55

5.2.2.1 Datasets 55

5.2.2.2 Training Data...55

5.2.2.3 Testing Data 55

5.2.2.4 SVM training results..56

5.2.2.5 Results 58

5.2.2.6 Basic System and Skin color detection (Speed up)...59

5.2.2.7 Basic System, Skin color detection (SD) and Image pyramids (Size invariance) 59

5.2.2.8 Results 59

5.3 System Application on Datasets 60

5.3.1 Experimental Procedure..60

5.3.2 Experimental Results..61

5.4 Comparative experiments 64

5.4.1 HOG-SVM detector 64

5.4.2 Particle Filter...65

5.5 Discussions..65

5.5.1 Possible issues 66
Chapter 6 68
Conclusion and Future work 68
6.1 Achieved goals 68
6.2 Future Work 69
References 70

List of Figures

Figure	Title	Page
3.1	Example training windows	29
3.2	Gradient Kernels	29
3.3	Pedestrian example from [INRIA]. Gradients computed for an image section.	30
3.4	Gradient Histogram	31
3.5	Gradient allocation 1	31
3.6	Gradient allocation 2	32
3.7	Overlapping blocks	32
3.8	SVM for (a) linearly separable case and (b) linearly non-separable case [2]	34
4.1	Proposed Method flowchart	40
4.2	Image and HOG window sizes	41
4.3	Feature compression from 128 to 9 per cell	43
4.4	Feature Normalization	44
4.5	HOG features virtualization	45
4.6	Simple SVM illustration	46
4.7	RGB skin color image.	51
4.8	HSV skin color image.	51
4.9	Visual representation of an image pyramid with 5 levels (Source: wiki)	52
5.1	Test Image	54
5.2	Number of training samples	56
5.3	Number of training samples, gamma = 0.001	57
5.4	Number of training samples, gamma = 0.01	57

5.5	Effects of parameter C	58
5.6	Detecting and tracking specified person running on structured crowd scene experimental	62
5.7	Detecting and tracking a specified running person in a structured crowd scene	62
5.8	Detecting and tracking a specified running person in an unstructured crowd scene	63
5.9	Detecting and tracking a specified walking person in an unstructured crowd scene	63

List of Tables

Table	Title	Page
1.0	Contributions at a glance	19
4.1	HOG descriptor parameters	44
4.2	SVM parameters	47
4.3	Particle filter parameters	49
4.4	Image pyramid parameters	52
5.1	Specification of experiment environment.	53
5.2	Testing the HOG feature extraction speed	54
5.3	Test Video details	56
5.4	Results of human detector testing	58
5.5	Speed improvement	60
5.6	Environment specification for experiment	60
5.7	Evaluation of proposed method.	64

Publications

Journal papers

(1) B. M. Hossain, S. G. Karungaru, K. Terada and A. Tsuji, “Human Detection and Tracking Using HOG Feature and Particle Filter in Video Surveillance System,” International Journal of Advanced Intelligence, Vol. 9, no 3, pp.397-407, 2018.

Conference papers

(1) B. M. Hossain, S. Karungaru, K. Terada and A. Tsuji, “Robust detection and tracking of moving objects using particle filter,” ACEAT International conference, pp. 27-29, Osaka, Japan, 2018.

(2) B. M. Hossain, S. Karungaru, K. Terada and A. Tsuji, “Real time specified person tracking in a crowded Scene using particle filter,” ICEE 2017 International Conference on Electrical Engineering No.SS1-3, PP. 2087-2091, Weihai, China, 2017.

Chapter 1

Introduction

1.0 Background

In the early 1940s, the emergence of the first computer eased research in computing perspective, growing exponentially with the ever increasing computational power and decreasing cost. Initially, the purpose was to create computers that help solve many complex computing problems that existed. A very quick calculator, so to speak. A few years later when this motivation was successful, a new way of using the power of chips processors emerged. In it, one of the most difficult and most researched field applications is computer vision.

Computer vision philosophy is an emerging engineering discipline in computer science and electronics, which can be seen in the same environment as the human ability to see. The field is wide, encompassing machine learning, image and signal processing, video processing and control engineering. This invaluable power machine allows more intelligent work and thus provides a better service to their human operators. Computer vision philosophy is easily available in commercial applications, such as health imaging, and medical robotics.

Computer vision uses computer software and hardware to report real-time image information about an object. It enables the computer to identify, define or interpret objects from image processing, which then aims to correct an image to be seen or interpreted by humans. Actually, image processing is a part of the computer's perspective. In computer science, image processing uses an image as input. Image processing commonly refers to digital image processing, but optical and analog image processing are also implied. In image processing part our research handles people detecting and tracking using histogram oriented gradient and particle filter in crowded area. In crowded area all people do not behave or move the same.

Usually suspicious activity is detected using a predefined set of behavioral and motion constraints that are different from other untracked activity. Our research work track and detect people based on specific features.

1.1 Motivation

The motivation for the development of computer theory of vision is based on human being sense of sight. Though it may seem easy or second nature to the average person, in reality we are loading 60 images per second with millions of pixels (pixels) in each picture. In fact, about half the human brain is involved in the process of visual data and it seems a good indication that it is a very complex task. On top of it, there is a big blind spot in the retina that has photo-captors in the middle of the eye, sensitive to the color, and the optic nerve.

Computing vision imitates people's vision but this goal is still a very long way. The computer's perspective is visually associated with problems related to interfacing between computers with their surrounding environment. Interaction with people and the intensity of their understanding is the root of many problems within the intelligent system, such as human-computer interaction and robotics. An algorithm for human identification binds high-bandwidth video into a compact story of human description in that scene. These high level details can then be used by other applications.

Unfortunately, the recognition of objects is still not optimal. The main reason is that embedded computers do not provide sufficient hardware to ensure the proper functioning of built-in algorithms and smooth storage. This embedded system simply cannot guarantee that the required information comes in time and is indeed reliable. Some examples of applications that can be provided with reliable human speed are detection and tracking: Human detection in crowd is one of the major challenges over the years owing to variable faces and environment. Moreover, tracking of human is very important research subject in the field of computer science [1]. During last few years a lot of researchers proposed their algorithms focusing on normal problem of tracking in crowded scenes [2]. Identification and tracking people using video surveillance activity and analysis, intelligent control and interaction of human computer, has been a focus for the last few years. [3]

Crowd monitoring in crowded public places such as airports, bus stations, concerts, subways, football matches, shopping markets, etc. is a challenge for those interested in safety and security systems. In these cases, we need to watch only a specific place or area like entrance and exit. A crowded scene can be divided into either structured or unstructured. Structured crowded scene have people moving in a fixed direction that does not change much with time. On the other hand, in case of unstructured crowded scene, the direction of movement is random and overlapping in all directions [2].

With the spread of applications of video systems surveillance, the interest in the field of research in image processing and computer vision in crowded scenes is growing because of the demand for security and safety systems in public places [3].

1.2 Challenges

Solving the main tasks mentioned above, such as human tracking and detecting, are not trivial due to various reasons. The identification and tracking of these subjects is important for the task, as well as ways to search and decide which must be resolved. One type of challenge is the large variation of the human appearance. These includes the underlying variations of the individual physiology, clothing and accessories as well as dimensions. In addition, movements are considered uncontroversial, since they are done freely in the scene and the environment in which people take their activities are generally cluttered. Moreover, many objects in the environment can cause a total or partial view of the people.

1.3 Research Approach

To overcome the above-mentioned challenges, in this thesis we approach it using the Histogram of Oriented Gradients (HOG) algorithm for detection of human features in the crowded area and the particle filter algorithm for tracking a specific human in the crowded area.

In our work, Histogram of Oriented Gradients features are learned using the linear Support Vector Machines (SVM) algorithms. Skin color detection is used to reduce the human search area for fast processing. Moreover, image pyramids are used to enable the detection of persons larger than the selected detection window. This window is 32x64 pixels.

1.3.1 Contribution of this work

The paper's main objective is to continuously track a specified human in a crowded scene. Two processes are necessary to accomplish this: Human detection and tracking. The paper proposes use of HOG-SVM system to detect and particle filter to track, since they are well established in this field. The main contributions of this work are as follows (Table 1.0):

1. To improve the speed of the HOG feature calculation, a smaller feature window is used. The original HOG window 64x128 pixels for 3,780 features. The proposed HOG is just 32x64 pixels for 756 features. The size is four times smaller.
2. The linear SVM is used to learn the HOG feature to detect human regions. No improvements have been added but fewer training samples are used, 2,000 humans and 3,000 non-humans.
3. When searching, the conventional method is to calculate the HOG feature for the all image pixel locations. This is very time consuming. The proposed method suggests an initial step to detect skin color regions before, and apply the HOG only on these color pixels. Experiments show that, on average skin color regions are less than a quarter of the pixels, improving the speed by least 4 times.
4. The pyramid method is used to detect regions larger than the selected window. The image size is reduced by 20% for 2 times.
5. Once the human regions are detected, a target person is selected for tracking. The selection is based on the person's clothes color, etc. The color is captured using a histogram and later used as the particle filter's feature. Initially, the conventional particle filter is applied using about 200 particles.
6. To track during occlusion, the last know particle weights, location and speed before occlusion are saved. The weights are updated based on the last known speed and location for about 2 seconds. After that, the search area is doubled and searched again. If the target cannot be found, tracking is considered to have failed.
7. To ensure continuous tracking, the human detector is applied once every 5 second to reconfirm the human regions (followed by the particle filter). Tracking accuracy of about 99% was achieved.

Table 1.0: Contributions at a glance

	Proposed method	Related work	Comments
HOG feature total	756	3,780	4 times smaller
Search Speed	On skin pixels	All pixels	4 times faster
SVM samples	5,000	Over 10,000	Faster to train
Particle filter	Some occlusion	none	Better accuracy

1.4 Computer Vision Introduction

Computer vision is a science that understands visual patterns using a computer. It is related to the automatic data analysis and useful information from a single image or sequence of pictures. In computer science, image processing is a method that analyzes images, for enhancement or to extract some useful information from it. Nowadays, image processing is among the rapidly growing technologies. It forms core research area within engineering and computer science disciplines.

1.4.1 Moving Object Tracking

Tracking moving objects, measuring motion parameters and obtaining a visual record of a moving object is an important area of application in image processing [4]. In general, there are two different ways to track an object:

1. Recognition-based tracking
2. Motion-based tracking.

Systems for rapid target tracking (military aircraft, rocket, etc.) have been developed based on motion-based predictive techniques such as Kalman filtering, extended Kalman filtering, particle filtration, etc. In object tracking systems based on automated image processing, target objects that enter the sensor view are automatically obtained without human intervention. In the recognition-based tracking, the pattern of the object is recognized in the frames of successive images and the trace is performed using the location information.

The discovery of animated objects in video streams is the first step in extracting information into many computer vision applications; including video surveillance, as well as tracking people, monitoring traffic and semantic commentary on videos. In these applications, strong tracking of objects in the scene calls for the detection of a reliable and efficient moving object, which must be equipped with certain tasks.

In this task, assume that the objects are modeled and moved randomly, in order to achieve maximum application independence. In case of absence from any prior knowledge of the target environment, the most widely dependent approach is based on moving the detection object with a static camera on the background subtraction [5]. Animated objects in the scene are detected by the difference between the current frame and the current background model. If the background model is inaccurate or inactive, the background subtraction causes the discovery of pseudo-objects, often referred to as "ghosts" [5, 6]. In addition, fragmentation of the moving object with background suppression is affected by the problem of shadows [7].

1.4.2 Measurement information in video surveillance

Automated security in public areas is of interest to government, institutions and the general public. Various computer technologies can be very useful in meeting this demand. Common closed-off circuit television networks are well-developed shelf products [7, 8]. However, such monitoring can generate large amount of information causing management problems.

Security personnel monitor the video to determine whether a response activity is needed. Given that such incidents can be irregular, identification and silence events require user focused monitoring for an extended period of time. Commercially available video surveillance systems are trying to impede the burden by using video motion detectors to analyze a particular scene change [9]. Video motion detection software can be programmed into alarm programs for a variety of different reasonable conditions, but overall the wrong warning rates for most systems in the environment are still incredible.

Basically, user video surveillance should be used to explain the video in a simple and intuitive way. In many cases, the information generated by the system is valuable so that a timely fashion explanation is needed. Therefore, the challenge is hands-free, commercially available sensors and jamming devices with strong, real-time, easy-to-use video surveillance system. In addition, it may be possible to achieve continuous activity for long periods, focusing on observations.

1.5 Chapter information

In chapter 1, we introduce the topic investigated in this work including the background, motivation and the research approach. At the end of the chapter some theoretical background information about computer science is presented.

Chapter 2, discusses the related works of all the algorithms used in our work.

Chapter 3, presents the detailed explanations of the methods used in this work. It is the textbook version of the methods.

Chapter 4, presents the proposed algorithm implementation details. It explains how the methods discussed in chapter 3 are improved and applied to our work.

Chapter 5, discusses the chapter experimental environment, results and discussions.

Chapter 6, presents the conclusions and future works

Chapter 2

Related Work

2.1 Introduction

Nowadays, real time human detection and tracking are very important research area in the video Surveillance system for security. Many algorithms have been proposed to detect and track a targets in a crowded scene in the field of computer vision.

2.2 Visual observation details

Traditional visual surveillance systems have a number of modules; each one performs a unique function. A very common visual surveillance system is displayed. This surveillance network consists of a single camera and a sequence of processing steps that are broadly classified into the following categories:

1. Environmental modeling.
2. Extract feature and / or object discovery.
3. Understanding the event and high level behavior.
4. Merge information from single cameras

The intelligence received from each camera is combined to get practical meaning data. The terminology used here is very common: a surveillance system can be designed to track one person and detect multiple person locations in a scene.

2.3 Literature Review

Video surveillance system of human activities utility is required to slow down people and track the resolution conditions, so tracking object is a very significant role in video surveillance system.

Myo Thida et al. presented a macroscopic model, microscopic model, crowd event detection [10]. Li Teng, et al. provided survey of crowded scene analysis, which considered crowd behavior and their activities [11]. Aggarwal and Cai reviewed motion analysis, tracking and recognizing human activities [12]. Wang et al. presented human detection, tracking and behavioral understanding [13]. Hu et al. also presented motion detection, tracking and behavior understanding [14]. Zhan et al. investigated crowd information extraction and crowd modelling [15]. On the other hand, using particle filter aiming for tracking people has presented with the image classification techniques to enhance the results and solve the problems of contour information in video surveillance. SIFT, Harris-SIFT [16], Histogram of Oriented Gradients (HOG) [17], AdaBoost, and Cascaded AdaBoost are used to extract the human area from video [18]. Dalal and Triggs presented a human detection algorithm with excellent results. Their method used a dense grid of HOG feature, computed over blocks of size 16x16 pixels to represent a detection window. This representation is enough to classify human using a linear SVM [19]. Ruiyue Xu have applied multiple human detections and tracking based on head detection for real-time video surveillance [20].

In contrast of existing methods, our proposed method has focused on the target detection and tracking based on a feature of individual person using the combination HOG feature detection and particle filter as we mention in previous section.

2.4 Human detection and Tracking

Human identification has gained huge interest in computer viewing community over the past few years [21]. Many techniques, models and general architecture have been suggested [22]. According to the number of cameras employed in the detection mode, these approaches can be divided into two main categories: Single and Multi- scene detection methods [23].

2.4.1 Single scene based detection approaches

The single aspect is to perform a human identification based on one single camera input. To identify people, these methods generally try to determine if all input images are relevant positions and scales, and whether they contain humans have or not.

For model-based approaches, to match objects, the shape is advantageous because it can distinguish a strong object that is relatively stable for environmental lighting changes. There are two methods of representation to model the shape space: discrete and continuous approaches [24]. For discrete approaches, the shape manifold can be represented by a set of model shapes [25]. Typical model-based matching techniques based on distance conversions are combined with pre-measured hierarchical structures, allowing real-time matching on the Internet with thousands of models [26].

This technique gives good results for setting goals without prior knowledge in a crowded scene. The effectiveness of this method is proved using about 4,500 models to match infantry in images in [25]. The basic idea uses the distance scale, so that the template matches the DT image to a similarity scale. At the same time, this approach can use an electronic search algorithm. However, if the pixel location is only calculated by the edges without looking at its direction when converting computing distance, it leads to numerous false alarms in the presence of clutter. Another notable work [25] is the use of the template hierarchy, which is automatically generated from available examples, and is con Figured through a bottom-up approach. They are only looking at places where the scale is below a certain threshold, so the acceleration is displayed three times the size, compared to the overall search.

Features selections play an important role in the performance of the classics, therefore, it is vital to choose the most fitting features according to the application environment, in order to discriminate from one to the other, following we will have a brief discussion on several features and class ER presentations.

Haar Wavelet features are proposed by [22], and more adapted by [27, 28]. They introduce a thicker representation using waveforms, which reflect local scales at different locations, scale and forecasts. However, due to spatial migrations, many types of transit representations require the process to select the most suitable subsets from the potentially significant features [24].

Histograms oriented Gradient descriptor (HOG) were proposed in [19] to detect people, and obtained good results on multiple data sets. HOG features the spatial distribution of edge orientation. The HOG representation depends on the graphs of the local gradient, and it contributes to the intensity of the image at each pixel in within its neighboring cells' graphs by triple interpolation. Because representation depends on local distributions of gradient positions and directions, it seems to be more effective for the appearance of the modeling object than for the Haar features, while being strong for noise and change within the layer. Preventing normalization makes the HOG descriptor strong for lighting changes. However, while excellent performance appears, HOG approaches depend on dealing with people in the presence of blockages and in situations where people are particularly prone to fluctuations. As the most representative in the various grades of the class, Support Vector Machines (SVM) has proven to be a powerful tool for solving processor-style problems, combining data in two categories by clicking the super-limit of the marginal super-boundary separating one layer from the other. Linear SVM has been successfully used by combining various features [19], while nonlinear SVM gives further performance improvements, yet computational costs are much higher [22, 28,29, 30]. Adaboost is an iterative method of mixing a strong classifier with a moderately accurate one. However, the methods of supervision are generally required to expand relevant training information from each class object. They are not only annoying but often very bad, because it is not clear that any part of the distribution of the class is represented and there is still specimen of any part of the distribution [31].

2.4.2 Multi- scene detection methods

Using many cameras provides a better solution for discovering and mapping multiple locations. People can be closed, where multiple camera views can be used to retrieve 3D structure information and close closure in crowded environments, as well as calculate accurate 3D locations for targets in complex scenarios.

Most multi-view detection procedures depend on the zoning scheme, which collects counted masks in many observations [32]. One of the main aspects of these methods is called a document in front of the mask. Some works align the mask with the convictions of monotheism [33]. The work is available in each scene on the map of the probability of foreground, and then it converts from all the other opinions into reference view [34]. Thus, this forward regression increases the probability of creating a bipartite network of possible maps for the possibilities of occupation [35, 36].

All camera scenes are set using a reference bar design and a density link is used to identify the users' head. [37], it is used in a uniform framework, in which the particle filters identify a person's ground.

The homograph constraint based methods can be explained by localization of public as a visible body intersection [38]. They have a reasonable accounting facility, where the decision to grab the ground plane is directly taken from monitoring the project in front of each point of view. However, depending on the entire body part resolution, not the complete silhouette, its methods can cause many false positive flaws due to shades, repetition on the ground, the reason for crowd density. The housing map is a representation of the field of interest and information collected from different perspectives, usually the presence of a person. The probability feasibility map (POM) [39] has been proposed, and it is assumed that objects are monitored by multiple cameras at the head level. In this work, the ground level is estimated for each grid cell using the results from the projection behind the distilled and proactive feasibility of a regular grid. Specifically, a simple rectangle model is used in the direction of the width below the assumption that the probability of the map to achieve all possible feasibility tension is obtained through frequent optimization, so that the difference between projection and input binary image is decreased. This method is heavily influenced by defective forums found in cylinders, similarly, [40].

The official map collection of frontend images by creating limited inverse issues representing multimedia the basic framework for identification and identification of simultaneous identities from multiple cameras to non-sequential numbers. It encodes many of the weak characteristics, such as distinctive maps, which make the concept of a mass map common, in each cell replacing the license probability [41]. Strengthen the nozzle objects and silo sets used to calculate the level projection of the visual hall of the scene. This dam is used for projection number and place possible of humans [42, 43]. Compared to exclusive detection, multi-spectral detection is able to determine the accuracy of three-dimensional space accuracy. Therefore, it is used for many other high-level vision procedures, such as tracking multiple objects, counting human, visual understanding etc.

2.4.3 Real time people tracking system for security

Real-time human moving information is essential security application for human life such as, pedestrian traffic management. Detecting and tracking moving human is very important for our daily security. There have been many researchers proposing this method for detecting and tracking people for human security. Bhuyain et. al. proposed an inactive method for detecting and tracking specific person in video scenes [44]. Segan and Pingali, they introduced a system that is derived and tracked in a hierarchical silhouette. The system runs real-time, but the algorithm cannot work very well with very hard and temporary landscape to track too many people [45]. Haritaoglu proposed method is inspired by the consideration of a ground-based surveillance system to increase the area of supervised personnel activities. The surveillance system should detect objects and identify as identifiable as their people, animals, and vehicles. When one or more people are detected, their movements should be analyzed to identify how they are connected [46]. Frejlichowski et al presented a smart Monitor as "a pure security system that relies on image analysis that combines the benefits of alarms, video surveillance and home automation systems.

Chapter 3

Theoretical Background

In this section, an introduction of the method used in this work is presented. This theoretical information is widely available elsewhere but is included in this thesis for completeness and clarity.

3. 1 Histogram of Oriented gradients (HOG)

Histograms of Oriented Gradients (HOG) is a global feature descriptor. A feature descriptor generalizes an object so that a similar one produces features that are as close as possible to the original one in a different environment.

The HOG uses a single detection window to describe an object rather than a component features. Therefore, the object is entirely represented by a single feature vector, as compared to many feature vectors for smaller parts of the person.

The HOG object detector uses a sliding detection window. A HOG descriptor is computed for the detection window at each location on the image.

3.1.1 HOG Original Work

The HOG person detector was introduced by Dalal and Triggs [19] in 2005. This description is based on that work.

3.1.2 Detection window

The first part of the HOG algorithm is to decide the size of the feature descriptor window. This window size is chosen to be a 64x128 pixel window. Figure 3.1 shows examples of the original images used to train the detector, resized to 64x128 pixels.

Figure 3.1: Example training windows

The feature descriptor is calculated as follows.

3.1.3 Step 1: Calculation of the image gradient for each pixel

To calculate a HOG descriptor, the horizontal and vertical gradients for each pixel in the selected window are calculated first. For HOG feature extraction we compute the gradients using the centered horizontal and vertical kernels. In the aggregate, we want to estimate the gradient's histogram. The calculation of the gradients is done using equations below for the *x* and *y* directions

$$\text{For } x\text{-direction: } \frac{d_I}{d_x} = f(x+1, y) - f(x-1, y) \tag{3.1}$$

$$\text{For } y\text{-direction: } \frac{d_I}{d_y} = f(x, y+1) - f(x, y-y) \tag{3.2}$$

In an image, this can be achieved using the following kernels, Fig. 3.2.

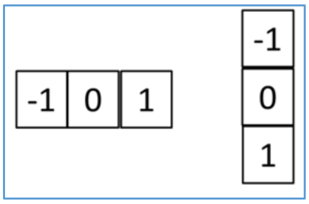


Figure 3.2: Gradient Kernels: left: *x*-kernel, right: *y*-kernel

After that, we can find the level of gradient (g) and direction (θ) using the following formula.

$$g = \sqrt{g_x^2 + g_y^2} \tag{3.3}$$

$$\theta = \arctan \frac{g_y}{g_x} \tag{3.4}$$

x-kernel detects the in vertical features and the *y*-kernel detects the horizontal features.

3.1.4 Step 2: Calculate Gradient Histogram in 8x8 cells

In the next step, the window is divided into 8x8 cells and histogram gradients is calculated for each 8x8 cells. One of the important reasons to use a feature descriptor to describe an image patch is that it provides a compact presentation.

An 8x8 image patch contains 8x8x2=128 features. The idea is to represent the 128 features using a 9-bin histogram. The gradients orientations calculated are in the range 0 to 360 degrees. However, to fit them in the histogram we convert them to between 0 to 180 degrees. An example of the gradient calculation if shown in Fig. 3.3.

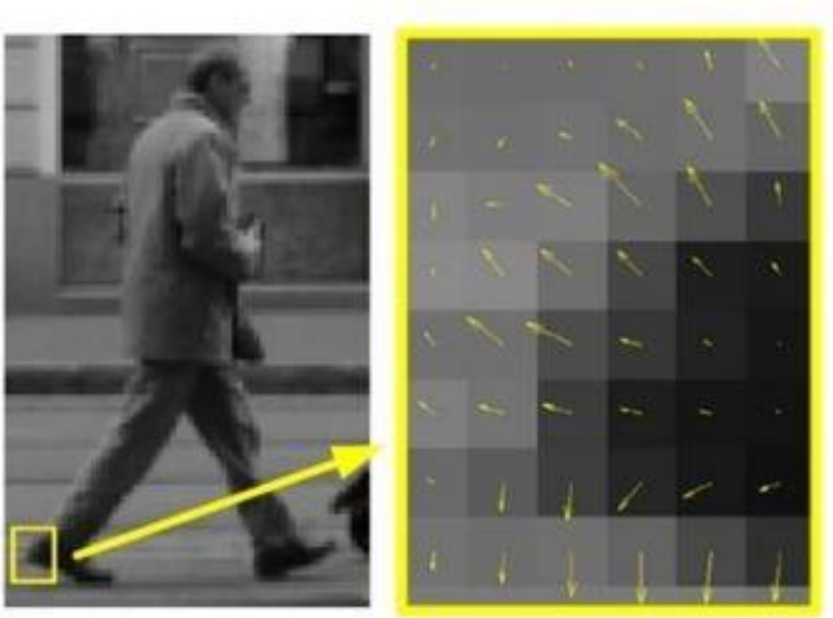


Figure 3.3. Pedestrian example from [INRIA]. Gradients computed for an image section.

The Histogram ranges from 0 to 180 degrees, therefore there are 20 degrees per bin, Fig 3.4.

For each gradient vector, its contribution to the histogram is based on its magnitude.

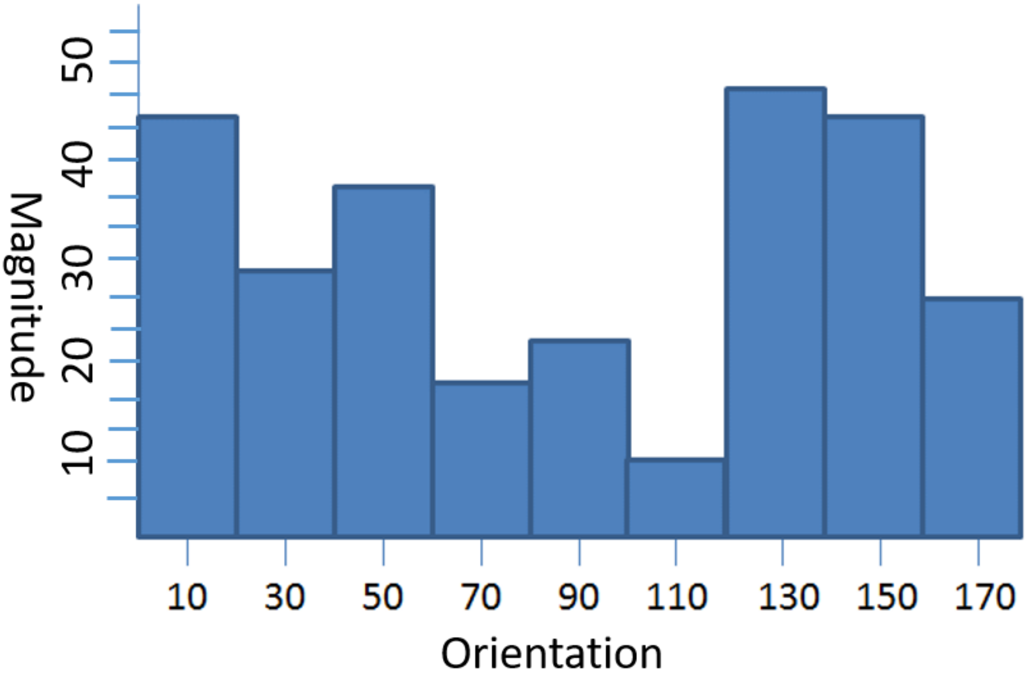
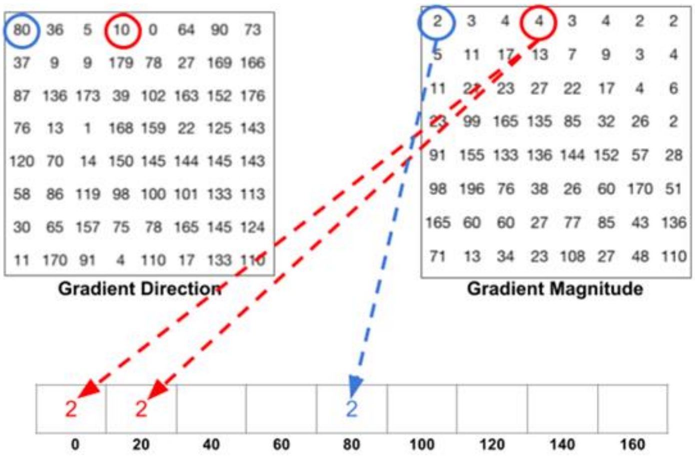


Figure 3.4. Gradient Histogram

Contribution between the two closest bins are split. For example, if a gradient vector angle is 85 degrees, add 0.25 of magnitude to the bin centered at 70 degrees, and 0.75 of the magnitude to the bin centered at 90, Figs 3.5 and 3.6 below.



80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

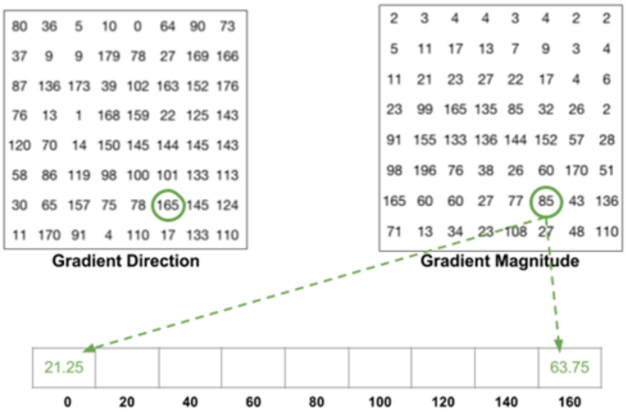
Gradient Direction

2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

2	2			2				
0	20	40	60	80	100	120	140	160

Figure 3.5: Gradient allocation 1(*Adapted from: http://www.learnopencv.com/histogram-of-oriented-gradients*)



80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction

2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

21.25								63.75
0	20	40	60	80	100	120	140	160

Figure 3.6: Gradient allocation 2 (*Adapted from: http://www.learnopencv.com/histogram-of-oriented-gradients*)

The splitting minimizes the issues of gradients right on the boundary between two bins. Otherwise, a slight angle change for a strong gradient on the edge of a bin, could have a strong impact on the histogram.

The gradient histogram reduces 64 vectors with 2 components each down to a just 9 values. The feature descriptor compression is important for the performance of the classifier.

3.15 Step 3: Block Normalization

Histogram oriented gradient is based on image gradient. Image gradients are generally sensitive to lighting. Instead of normalizing each histogram individually, the cells blocks are normalized based on all histograms in the block.

The original algorithm [19] use blocks that consisted of 2 cells by 2 cells. The blocks overlap by 50%, as illustrated in Fig 3.7.

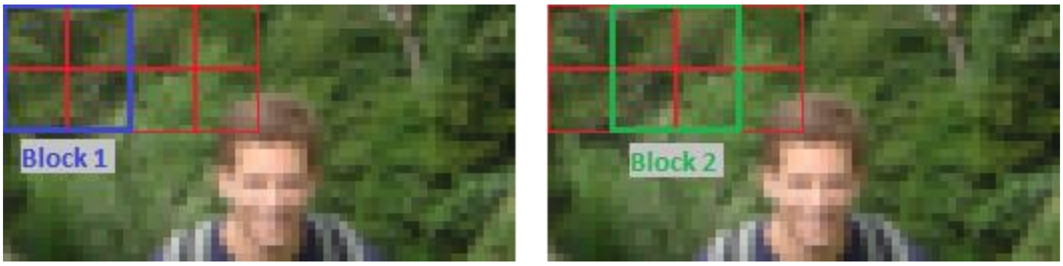


Figure 3.7: Overlapping blocks

This block normalization is produces 36 components (4 histograms x 9 bins per histogram). Each vector is divided by its magnitude for normalization. The normalization formula is

$$f = \frac{\nu}{\sqrt{\|\nu\|_2^2 + e^2}} \tag{3.5}$$

3.16 Step 4: Calculate the HOG feature vector

The 64x128 pixels detection window has 7 blocks horizontally and 15 blocks vertically (105 blocks total). Each block has 4 cells and 9 features per cell, for a total of 36 values per block.

Therefore, final vector size is 105 blocks by 36 features = 3,780 values.

3.2 Support Vector Machines

Support Vector Machines (SVM) is a supervised learning algorithm introduced by Vapnik [47]. SVM is used in many applications and has high performance results in image classification, text categorization and bioinformatics. SVM outputs a map of the sorted data with margins between the two as far apart as possible. It is usually used in classification problems [48]. It is based on a dataset. If the dataset has high variance, we need to reduce the number of features and add more dataset.

The theory and concept of SVM will be mentioned in this section. The simplest SVM formula for input data *x* is:

$$f(x) = wx + b \tag{3.6}$$

In this equation, SVM finds a hyperplane in a space different from that of the input data *x*.

This space condition is:

$$\|f\|_k^2 < \infty \tag{3.7}$$

Where k is the kernel and $\|f\|_k^2$ is the Reproducing Kernel Hilbert Space (RKHS).

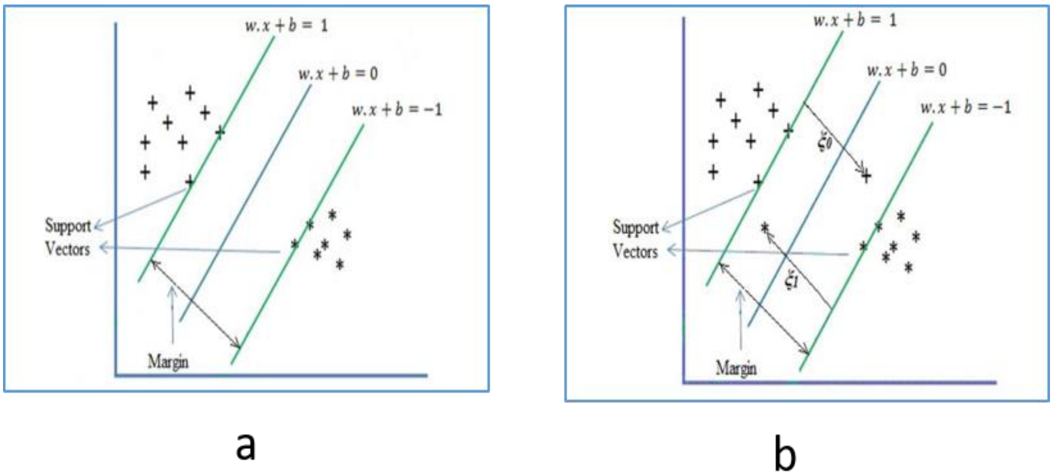


Figure 3.8: SVM for (a) linearly separable case and (b) linearly non-separable case [2]. Where $\xi_k,\ k=0,1,..$ is a positive slack variable that enables minimization of the misclassification error.

3.3 Sequential Monte Carlo (SMC)

Particle filters or Sequential Monte Carlo (SMC) methods are a set of Monte Carlo algorithms used to solve filtering problems. The problem consists of estimating the internal states in dynamical systems based on partial observations made, and random perturbations present in the sensors and in the dynamical system. The aim is to calculate the posterior distributions of the states of some Markov process, given some noisy and partial observations.

Particle filtering uses a set of particles to represent the posterior distribution of some stochastic process given noisy and/or partial observations. The state-space model can be nonlinear and the initial state and noise distributions can take any form required. Particle filter techniques provide a well-established methodology for generating samples from the required distribution without requiring assumptions about the state-space model or the state distributions. However, these methods do not perform well when applied to very high-dimensional systems [49].

3.3.1 Recursive State Estimation

The measure of the state is a process of estimating the amount that cannot be directly monitored, but it can be measured using other quantity data of measurement. State feasibility groups, which belong to particle filtration, deliver possible distribution in possible cases. The x_t status is the description, at t time, of the system (for example, Android site, laptop battery level, weather), which cannot get direct information about them. The measurement of z_t is data that can be obtained directly from the environment (sonar range determination device, voltage in battery, humidity in the air). The state of x_t is generated from the state of x_{t-1}. The evolution of the case can be expressed using the probability distribution [49]

$$p\left(x_t | x_{0:t-1,} z_{0:t-1}\right) \tag{3.8}$$

This is called the state of transit feasibility. One common idea is that state x is full, i.e., it contains enough information on previous states, or more accurately, if the state is given t -1, the state is uniquely measurement

$$p\left(x_t | x_{0:t-1,} z_{0:t-1}\right) = p(x_t | x_{t-1}) \tag{3.9}$$

Another process that can be modeled using a probability distribution is the generation of measurements

$$p\left(x_t | x_{0:t-1,} z_{0:t-1}\right) = p(x_t | x_t) \tag{3.10}$$

This is called the probability of measurement; the equation holds if x is a complete case.

This sample is known as the Dynamic system. Alternatively, the dynamic system can be represented using equations: the state transition equation

$$x_t = f(x_{t-1,} \mu_{t-1}) \tag{3.11}$$

Where f is the evolution function and $\mu_t - 1$ is add noise called state noise called state the measurement equation.

$$z_t = g(x_t, \varepsilon_t) \tag{3.12}$$

Where ε_t is the added noise known as the measurement noise.
Recursive filters are state estimation techniques using a repeat strategy, consisting of two phases:

Predict: In the prediction stage the next state is predicted

$$p(x_{t-1}|z_{0:t-1}) \rightarrow p(x_t|z_{0:t-1}) \tag{3.13}$$

Update: In the update phase the new measurements are merged

$$p(x_t|z_{0:t-1}) \rightarrow p(x_t|z_{0:t}) \tag{3.14}$$

Bays Filter

The most common recursive filter is the Bayes filter. Bayes Filter calculates the status estimate directly from the measurements and the previous case. The stages of their prediction and modernization are as follows:

Prediction

$$p(x_t|z_{0:t-1}) = \int p\,(x_t|x_{t-1})p(x_{t-1}|z_{0:t-1})dx_{t-1} \tag{3.15}$$

Update: In the update step, the expected background probability density function (PDF) is calculated from the expected PDF and the new measurement

$$\begin{aligned} p(x_t|z_{0:t}) &= \frac{p(z_{0:t}|x_t)p(x_t)}{p(z_{0:t})} \\ &= \frac{p(z_{t,}z_{0:t-1}|x_t)p(x_t)}{p(z_{t,}z_{0:t-1})} \\ &= \frac{p(z_t|z_{0:t-1,x_t})p(z_{0:t-1}|x_t)p(x_t)}{p(z_t|z_{0:t-1})p(z_{0:t-1})} \\ &= \frac{p(z_t|z_{0:t-1,x_t})p(x_t|z_{0:t-1})p(z_{0:t-1})p(x_t)}{p(x_t|z_{0:t-1})p(z_{0:t-1})p(x_t)} \\ &= \frac{(z_t|x_t)p(x_t|z_{0:t-1})}{\mathrm{p}(z_t|z_{0:t-1})} \end{aligned} \tag{3.16}$$

The update equation structure can be parsed

$$posterior = \frac{likelihood \cdot prior}{evidence} \tag{3.17}$$

Where probability is given by the measurement model, formerly known as the prediction phase and the evidence is a normalization constant that can be calculated by

$$p(z_t|z_{0:t-1}) = \int p\,(z_t|x_t)p(x_t|z_{0:t-1})dx_t \qquad (3.18)$$

The Bayes filter can provide an ideal solution in the meaning of computing the posterior PDF, but consolidation is not generally non-identifiable, so in all cases, the simplest, the base filter is incomplete. The particle filtering algorithm that uses the sample method to approximate the best optimal Bayesian solution.

3.3.2 Particle filter

The particle filtering sensor is a condition for estimating the condition of the x_t system for speedy, dynamic time. The slight change of notation from x_t to X_t. The *X* Particle Filter is reserved for particle representation. The particle filtering *X* is stored for representing a particle. The objective of the particle filter is to guess the next post using a range of potential particle (particle) samples on state territory, x_t^i for the situation. An estimate particle filter is the real situation *t*. Every sample indicates the importance of the sample, which is associated with a weight x_t^i this value or gravity for the sample. The particle is described as set

$$X_t = \{\langle x_t^i, \omega_t^i\rangle | i = 1, \dots, N\} \qquad (3.19)$$

Posterior concentration by insight particle set is approximate. Ideally the possibility of being x_t^i to be part of particle set will likely be proportional to its Bayes filter subordinate. The greater the density of the sub-region of the state space by the particles, the greater the probability of the real state to be in that area. The particle filter adds a third step, re-sampling, to the usual prediction steps for recursive filters:

Prediction: Forecasting prediction particle is used based on current state estimation by correcting the $x_t$particles using transcription function f, adding random words to simulate the sound effects in real-time.

$$q_{t-1} = f(x_t)+\mu_t \qquad (3.20)$$

Where μ_t is the noise.

Update: The weight of each added particle is evaluated using new measurements $z_{t.}$

$$\omega_{t+1} = g(q_{t+1}, z_t) + \varepsilon_t \qquad (3.21)$$

The update function g measures similarities between z_t measurements and particle value q_{t+1} . The filter can only consist of these steps (with normalization weights $\sum_{i-1}^{N} \omega_i = 1$)

The problem with this strategy is that after a few repetitions, most particles will have little weight (most of the weight will not focus on only a few particles). In fact, these low weight particles are not used, although they were ideal for representing a larger area near the real state. This technique is achieved by adding a third step.

Re-sample: A new set of X_{t+1} particles is generated by sampling N times, with the frequency, of the expected particle range q_{t+1} according to ω_{t+1}^{i} weights. The re-sampling step transformed the seized particles into highly probable areas. Large-weight particles are expected to produce more copies of themselves than low-weight molecules in the re-sampling process, thus focusing all arithmetic resources in areas with higher probability of being the correct state.

Chapter 4

Proposed Human Detection and Tracking

4.1 Introduction

Tracking and detecting target objects in crowded area is essential especially in assisting the monitoring officers in video surveillance system. In crowded scenes each person might have different characteristics and features like hair color, heights, weights, clothes color, etc.

The proposed system consists of a human detection system using HOG-SVM and a human tracking system using a particle filter.

Human detection systems can be divided into two main categories:

a. Component-based methods
b. Single detection window analysis.

Component-based techniques are used to detect object parts separately in a normal geometric configuration. It uses a hierarchical systems detection window. Body parts are detected according to priority; if one part is missing, the other parts are not searched. It is slow and it doesn't deal with multi-view and multi-pose cases. In contrast, the single detection window method is based on sequence-based detection window labeled at all possible sub-windows. The most important feature is its speed. However, partial occlusion handling flaw is its limitation. In our system, we choose the single detection window method because of its speed.

This paper's main objective is to continuously track a specified human in a crowded scene using a particle filter. To accomplish this objective, two main processes are necessary: Human detection and tracking. The paper proposes the use of HOG-SVM to detect and particle filter to track since they are well-established in this field. To support these methods, skin color detection, image pyramids and other methods are used. The proposed method's flowchart is shown in Figure 4.1

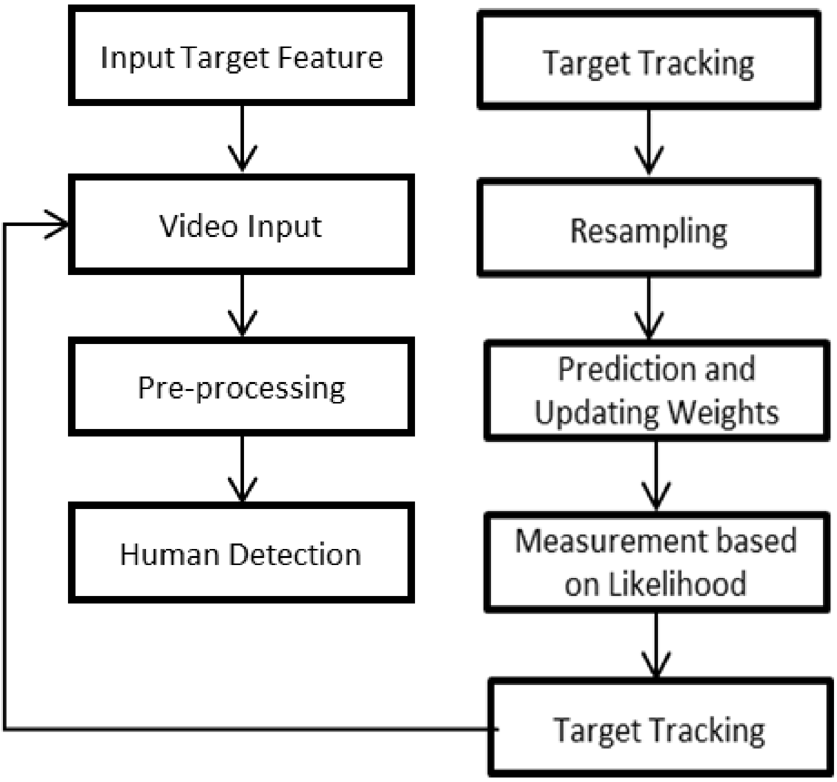


Figure 4.1 Proposed Method flowchart

In a nutshell, the proposed method consists of the following steps:

1. Human detector design
   a. Basic HOG feature design
   b. Feature extraction speed improvement
   c. Feature training using SVM
   d. Final testing on special test data
2. Human tracking design
   a. Basic particle filter design
   b. Defining color feature
   c. Optimizing particle number
3. System application
   a. Optimizing human detection: application step
   b. Improving speed: skin color detection
   c. Handling occlusions

Each of the above topics we be describe in details in the following sections.

4.2 Human Detector Design

The human detector proposed is the combination of HOG features and SVM learning. The design formulation will be discussed in this section.

4.2.1 HOG Feature

HOG features are widely used for the detection of objects. HOG feature is calculated on a single fixed window based on the orientation of the gradient in localized regions, called cells. It enables the rough shape of an object to be represented and is robust to variations in geometry and illumination changes because of normalization. However, rotation and scale changes are not supported.

4.2.1.1 Window size determination

The size of the window, determines the size of the final HOG feature. On one hand, a small window will be faster to process but is incapable of detecting larger objects. On the other hand, a large window is slow but might be better for larger objects. The best choice depends on the size of objects in the image, which in turn depends on the capture device location. In this work,

a) The captured image size is 640x480 pixels
b) Based on the capture location, manual observation reveal that the smallest person size in the image is about 30x60 pixels, Fig. 4.2.

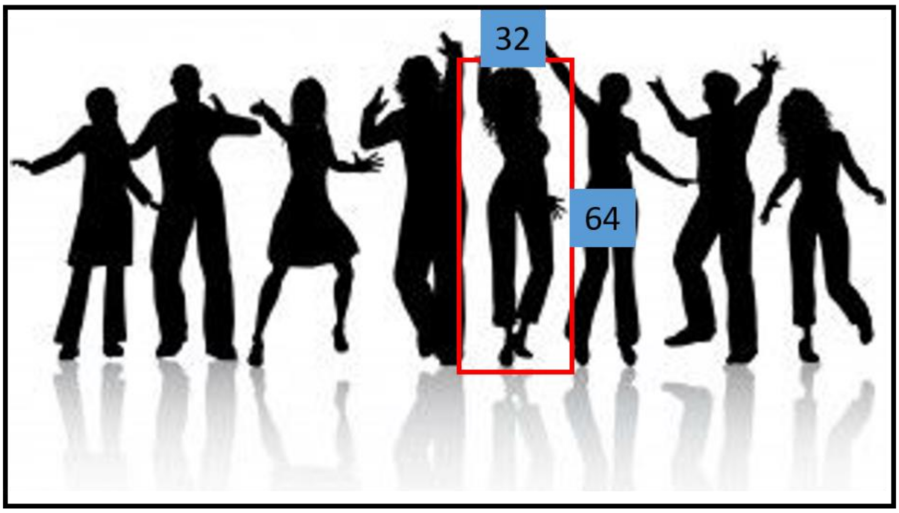


Figure 4.2: Image and HOG window sizes

c) Therefore, the Window size is set as 32x64 pixels. This size allows for easier calculation of the HOG features.

4.2.1.2 Magnitude and Orientation computation

To calculate HOG descriptors, we first need to calculate the horizontal and vertical edges at all pixels in the window. Several methods are available. In this work we chose the following kernels (Robert's operator).

For an image location (i, j).

1. For x-direction: Operator: $[-1, 0, 1]$

   Derivative calculation: $g_x = f(i + 1, j) - f(i - 1, j)$

2. For y-direction: Operator: $[-1, 0, 1]^T$

   Derivative calculation: $g_y = f(j, i + 1) - f(j, i - 1)$

The gradients and orientations are then calculated as follows:

Gradient Magnitude: $M(\mathrm{i}, \mathrm{j}) = \sqrt{g_x^2 + g_y^2}$

Gradient Orientation: $O(i, j) = \arctan(g_y / g_x)$

4.2.1.3 Feature calculation cell size

The detection window is subdivided into 8 x 8 pixel regions called cells. Each cell has 64 magnitudes and 64 orientation features. The 128 features are then compressed into 9 using histogram binning, Fig 4.3.

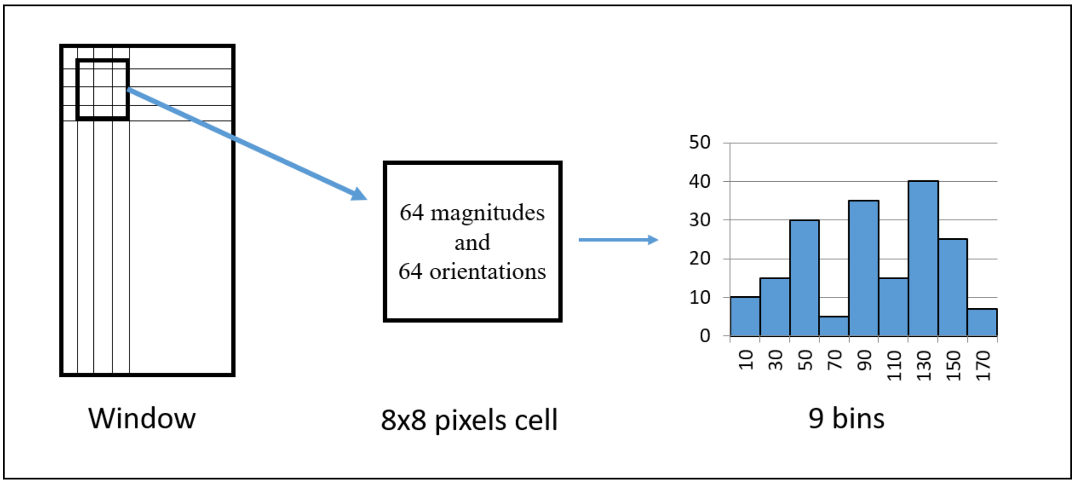


Figure 4.3: Feature compression from 128 to 9 per cell

4.2.1.4 HOG Feature size reduction

After all the cells in the window have been compressed in to 9, the total feature size for the window can be evaluated as follows:

a. The original magnitudes and orientations feature size per window is 32x64x2 = 4,092 features
b. After cell reduction:
    a. The *x*-direction, width 32 pixels, we have 4 cells
    b. The *y*-direction, height 64 pixels, we have 8 cells
    c. There are 9 features per cell
    d. Therefore, total features are: 4x8x9 =288 features
c. Therefore, the features are compressed from 4,092 to 288.
d. The basic descriptor has 288 features

4.2.1.5 HOG Feature Normalization

To improve the stability of the descriptor to fluctuations in brightness and intensity, normalization is required. There is a wide range of normalization functions. In this work, the L2 normalization is used after the cells are grouped into nested blocks. For each block, the histogram of the cells binds together to form a vector.

Each block is set as a 2x2 cell region. Each block is then normalized. To allow better normalization block overlapping is introduced at 50% overlap, Fig. 4.4.

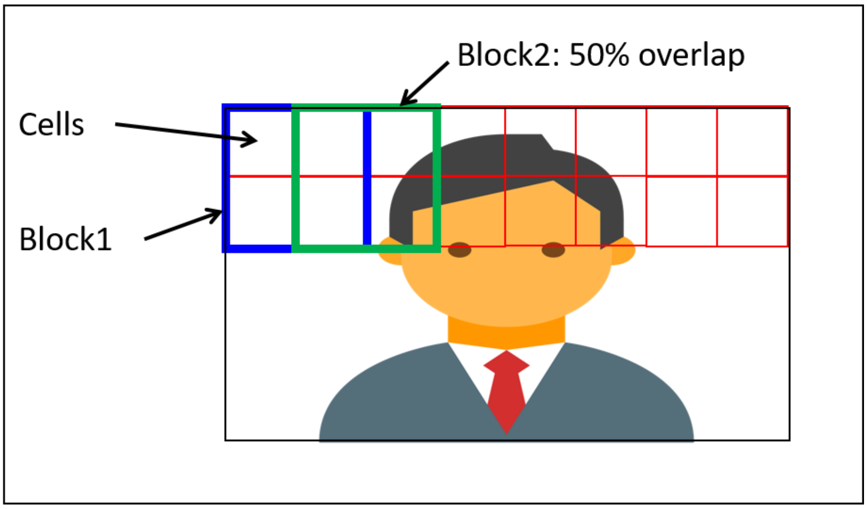


Figure 4.4: Feature normalization

Once all the cluster vectors are created, they are all serialized to form the final descriptor.

The size of the final descriptor calculated as follows:

a. Each block has 4 cells and a total of 36 features
b. There are 3 blocks horizontally
c. There are 7 blocks vertically

Final feature size for a 32x64 window is therefore: 36x3x7 = 756 features

4.2.1.6 Propose HOG descriptor parameters

The proposed HOG descriptor parameters are summarized in table 4.1.

Table 4.1: HOG descriptor parameters

Window Size	**Cell Size**	**Block Size**	**Feature Size**
32x64	8x8	2x2	756

4.2.1.7 Example output HOG feature

The following Figure shows an example of HOG features computed using the algorithm, Fig. 4.5.

4.2.2 The HOG-SVM object detector

Histogram oriented Gradient features describes an object. It can be used to detect the object using a learning algorithm. While many learning methods exist, it has been proven that it works particularly well when combined with a linear SVM classifier. This combination was proposed by Dalal and Triggs [19].

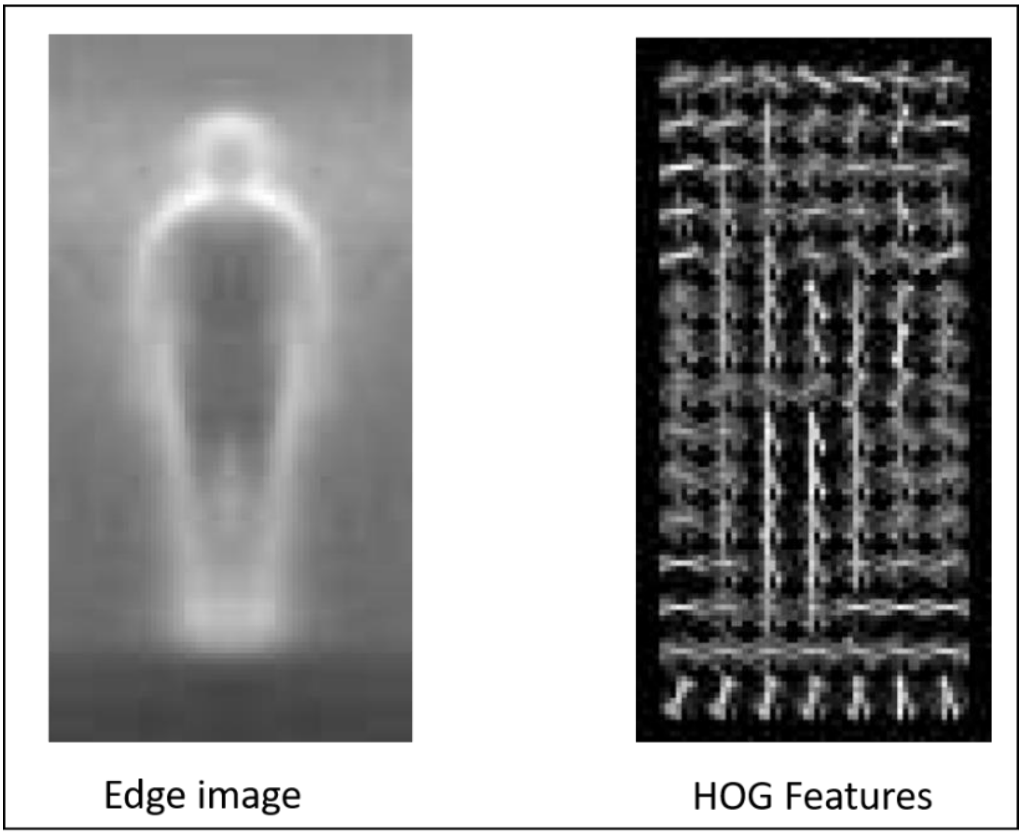


Figure 4.5: HOG features virtualization

4.2.2.1 SVM

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. When given labeled training data, the algorithm outputs an optimal hyperplane which classifies new examples. In two dimensional space this hyperplane is a line dividing a plane in two parts with each in either side, Fig. 4.6.

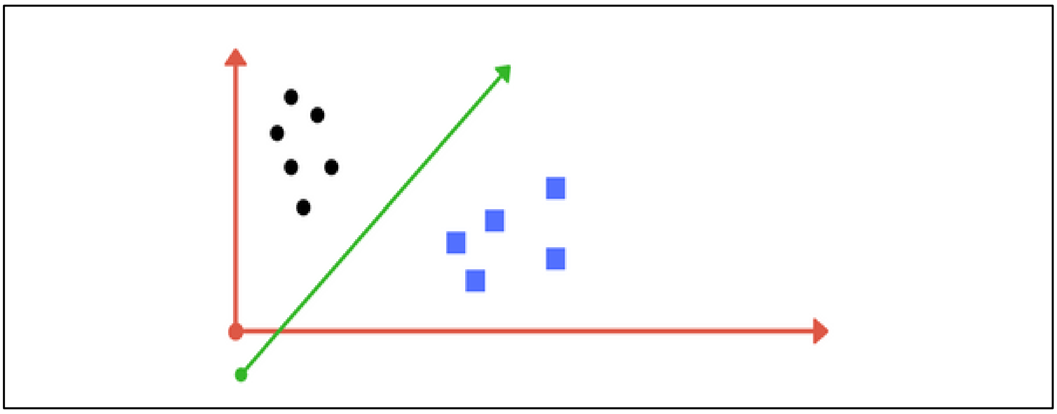


Figure 4.6: Simple SVM illustration

In this work, the linear SVM is chosen because our object it to determine if an image patch is a human or not. The conventional SVM is selected, but we set the parameters to suit our application. These include, kernel, regularization, gamma and the margin.

4.2.2.1.1 Kernel

The learning of the hyperplane in linear SVM is performed by transforming the problem using linear algebra. There are linear, exponential and polynomial kernels. In this work, the linear kernel is selected.

For linear kernel, the prediction for a new input can be represented using the dot product of the input (x) and each support vector (x_i). It is calculated as follows:

$$f(x) = B(0) + \sum(a_i * \ (x, x_i)) \qquad (4.1)$$

The coefficients B(0) and a_i must be estimated from the training data by the learning algorithm.

4.2.2.1.2 Regularization

The Regularization parameter informs the SVM optimization how to avoid misclassifying the training samples. For large C, the optimization chooses a smaller-margin hyperplane if it produces better results.

4.2.2.1.3 Gamma

The gamma parameter defines how far the influence of a single training example. Low values mean “far” and high values mean “close”. Hence, with low gamma, points far away from plausible separation line are considered in calculation for the separation line.

4.2.2.1.4 Margin

A margin is a separation of line to the closest class points. The best margin represents where this separation is larger for both classes. It enables the points to be in their assigned classes without crossing to other class.

4.2.2.1.5 Propose SVM parameters

The proposed SVM parameters are summarized in table 4.2.

Table 4.2: SVM parameters

Kernel	Regularization	Gamma	Margin
Linear	2	1/features	5

4.2.3 HOG-SVM Detector

The final step in the design of the classifier is to learn the HOG descriptors using the SVM to create the final HOG-SVM detector.

4.2.3.1. Training data

To train the HOG-SVM detector, training data is required. We use 2,000 and 3,000 positive and negative samples respectively.

Our dataset is a combination of random samples from the INRIA person dataset and others manually created using our own images. All the training sample images are resized to 32x64 pixels.

4.3 Human Tracking Design

The human tracker proposed is the particle filter. The design formulation will be discussed in this section.

4.3.1 Particle filter

The Particle filter is a technique for implementing recursive Bayesian filter by Monte Carlo sampling. The concept is represented by a set of concentric densities, weight associated with the random particles; and compute estimates based on these samples and weights. The particle filter uses multi-modal probability function and can disseminate distribution of multi-modular posters that will occur in temporary events and background clusters.

4.3.1.1 Speed up particle filter

Normally, particle filter execution time on a single processor machine can be improved, by making smart design options, such as optimization, choice of parameters, etc.

4.3.1.2 Tracking specific target human

The particle filter achieves Bayesian recursion through Monte Carlo simulation. This method is suitable for any non-linear system that can be represented by the case model. However, particle filter accuracy is mainly based on two main factors: effective particle sampling and a reasonable measurement of molecular weight.

We apply the particle filter to the image after the human regions are extracted by the HOG-SVM. The processing by particle filter has several steps, resampling, prediction and updating the weights, and measurement based on likelihood function.

Our particle filter is implementing recursive Bayesian filter by Monte Carlo sampling; it represents the intensity of the rear set of random particles with associated weights and also calculates estimates. In the case of our method, the Bayesian filter approximating the prior and posterior density functions are used with a set of the target features. Therefore, the generated particles move to the specific target regarding to the feature set of individual person.

4.3.1.3 Resampling

Resampling step is a very important and computationally valuable part of a particle filter. Resampling provides a way to give an approximate delivering targets by resampling *N* particles. The probability of the weight associated with it is sorted with each particle. Resampling means that if a particle is removed with high probability, their copied weight is produced in multiple copies of particles with low and high weight; this future can be searched independently. It can be seen that proven immediately future stability in the growth cost of the evolution of Monte Carlo.

4.3.1.4 Prediction and Updating Weights

Weights of our particles are updated according to likelihood. Prediction of location assumes that the next state is almost the same position. The variance is set at 8.0 experimentally.

4.3.1.5 Measurement based likelihood

This time the likelihood is set at 30x30 centered on particles. The likelihood was the existence rate of the feature in the specified persons.

We also defined the parameter of target features as skin color and red, blue and black color shirts using histograms.

4.3.1.6 Particle filter parameters

The proposed particle filter parameters are summarized in Table 4.3.

Table 4.3: Particle filter parameters

Particle number	Variance	Likelihood range
200	8.0	30x30 pixels

4.3.1.6 Handling Occlusions

To track during occlusion, the last know particle weights, location and speed before occlusion are saved. The weights are updated based on the last known speed and location for about 2 seconds. After that, the search area is doubled and searched again. If the target cannot be found, tracking is considered to have failed.

4.4 Pre-processing

To speed up the human detection process, we use color processing to detect skin color regions. There are a lot of color spaces like, RGB, HSV, YIQ, YUV, CMYK, YCbCr, etc. For skin color detection HSV and YCbCr color spaces are better compared to the RGB color space.

In our method we use the HSV color space is used in order to detect skin color regions. First we transformed RGB color space to HSV color space, because the RGB color space is not robust for skin color detection as it is affected by the environmental condition where illumination changes.

$$H=\begin{cases} 60^{\circ} \times \left(\frac{G'-B'}{\Delta} \operatorname{mod} 6\right), & C_{\max} = R' \\ 60^{\circ} \times \left(\frac{B'-R'}{\Delta} + 2\right), & C_{\max} = G' \\ 60^{\circ} \times \left(\frac{R'-G'}{\Delta} + 4\right), & C_{\max} = B' \end{cases} \tag{4.2}$$

$$S = \begin{cases} 0 & if\ C_{\max} = 0 \\ \frac{\Delta}{C_{\max}} & otherwise \end{cases} \tag{4.3}$$

$$V= C_{\max} \tag{4.4}$$

Where:

$$C_{\max}= \max(R', G', B')$$

$$C_{\min}= \min(R', G', B')$$

$$\Delta=C_{\max} - C_{\min}$$

The comparison result is shown below. Figure 4.7 skin color detection using RGB. Figure 4.8 show the results using HSV. It can be observed that better results are obtained using HSV.

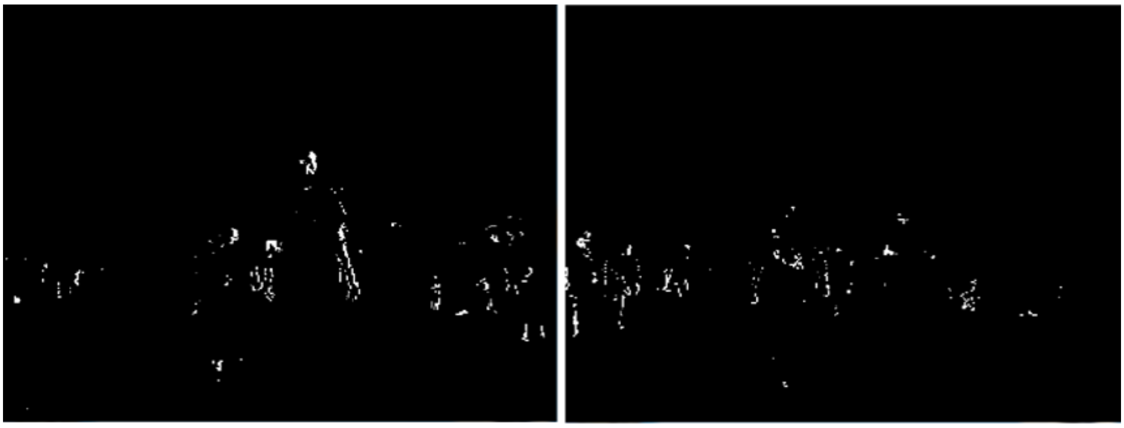


Figure 4.7: RGB skin color image.

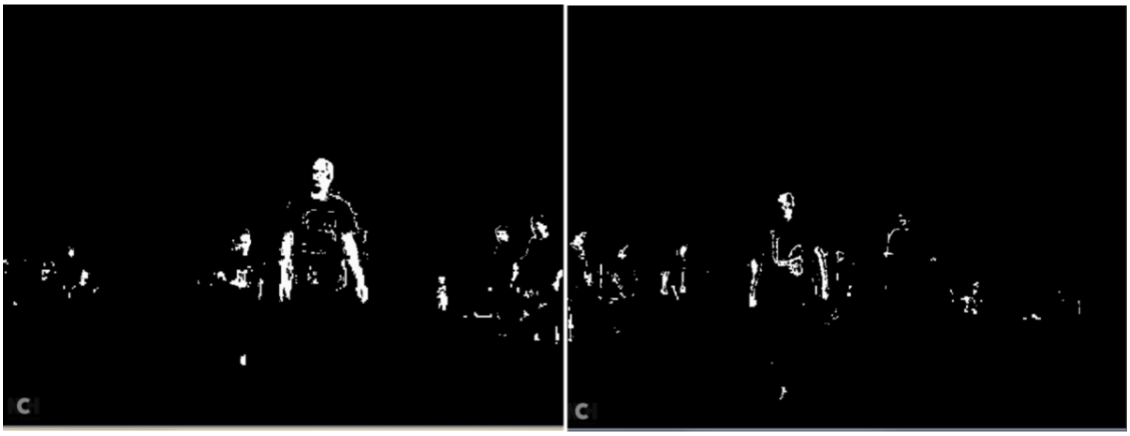


Figure 4.8: HSV skin color image.

4.5 Image Pyramids

Our HOG-SVM detector can only detect humans for size 32x64 pixels. Since we assume that this is the smallest human in the image, the ability to detect larger human sizes is necessary. However, since it is not feasible to train several HOG-SVM detectors for all sizes, image pyramids are used. pyramid representation, is a type of multi-scale signal representation, in which an image is subject to repeated smoothing and subsampling.

Therefore, we will continuously resample the image and apply the detector at each scale. The final result is a combination of all the detections at every scale. We chose to resample the image 2 times at 20% scale. Figure 4.9 shows this process. The Gaussian kernel is used during resampling.

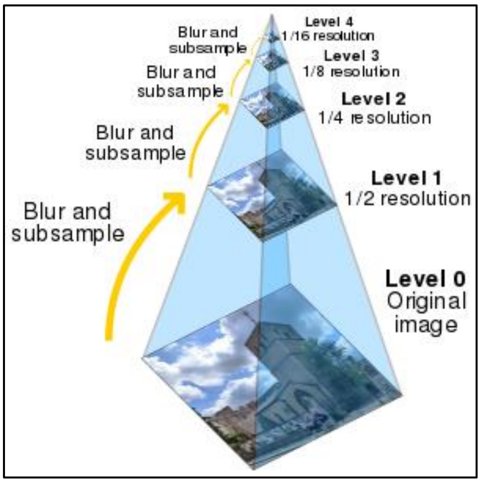


Figure 4.9 Visual representation of an image pyramid with 5 levels (Source: wiki)

5.4.1.1 Image pyramid parameters

The proposed image pyramid parameters are summarized in table 4.4.

Table 4.4: Image pyramid parameters

Number of Resampling	Scale	Kernel
2	-20%	Gaussian

Chapter 5

Experiments and results

5.1 Introduction

To prove the effectiveness of the proposed method, we conducted several evaluation experiments. The aim of experiments is to detect and track a specific target person. We assume a crowded situation with random movement and different person characteristics. It should be noted that occlusion is a significant and difficult problem in crowded scenes.

We conduct the experiments in out of laboratory using datasets to evaluate our method. The specification of the experiment environment is shown in Table 5.1.

Table 5.1: Specification of experiment environment.

Computer	**Intel Core i7, Memory 8GB**
Camera	**Logicool USB Camera**
Resolution	**640 x 340, 30fps**
Software	**Visual C++, OpenCV**

5.2 Experiments and Results

In this section, we will discuss the experiments conducted and the results obtained. The HOG feature extraction experiment was conducted first. Then, the human detector training and testing is performed. After that the particle filter is tested and the combined system evaluated using the databases. The results will be presented at the end of each section.

5.2.1 HOG Feature extraction

Based on the proposed design the HOG extracted is implemented. The main extractor is a single window 32x64 pixels. It is compared for speed with the conventional 64x128 window and also a 48x96 window. The test image size is set at 640x480, Fig. 5.1.

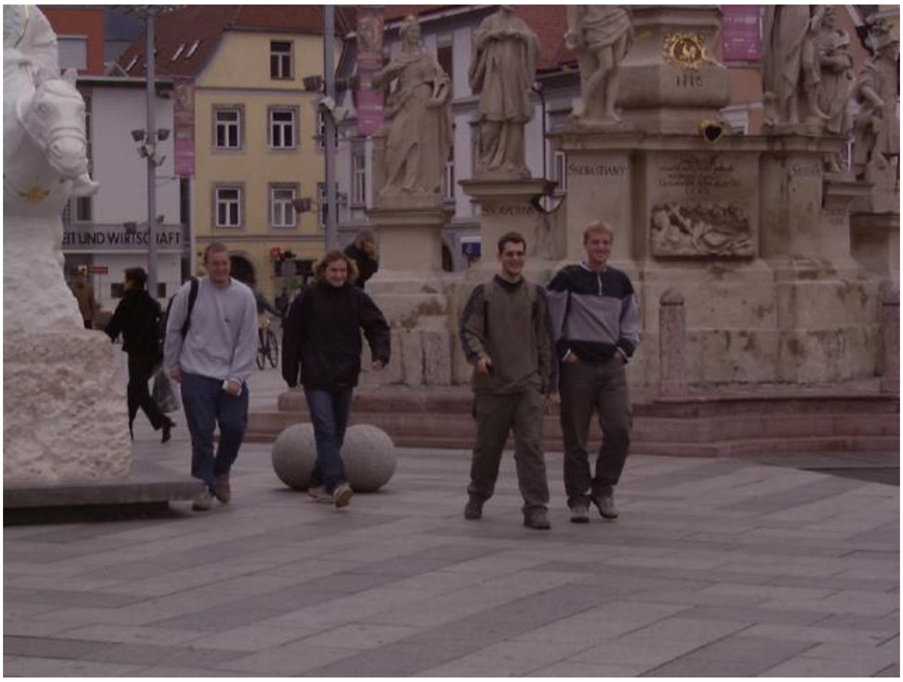


Figure 5.1. Test Image

Table 5.2 shows the results of the test experiments. As can be observed different window size produce different results.

Table 5.2: Testing the HOG feature extraction speed

Window size	**HOG Feature size**	**Number of windows**	**Total Time taken**	**Calculation Speed Per window**
32x64	756	253,953	114s	0.00044890s
48x96	2,800	228,305	215s	0.00094172s
64x128	3,780	203,681	328s	0.00161036s

Comments:
The stride was set to 1, size of picture is 640x480 pixels and the HOG parameters are an 8x8pixels cell and 2x2 cell normalization blocks. The average extraction speed improved from 328 seconds for the 64x128 window to 114 seconds for our proposed 32x64 window.

5.2.2 Basic HOG-SVM detector Training and Testing

After confirming the effectiveness of the HOG feature extraction, the Human detector (HOG-SVM) is trained and tested. The linear SVM used is as described in the proposed method section. The parameters remain unchanged.

5.2.2.1 Datasets

The training data used is obtained from 2 sources.

1. INRIA pedestrian dataset: The dataset is divided in two formats: (a) original images with corresponding annotation files, and (b) positive images in normalized 64x128 pixel format with original negative images. The images are resized to 32x64 pixels
2. Our database: The images are extracted from a videos captured randomly in our laboratory and around the campus. Most of the humans in the images are about 32x64 pixels.
3. Test Scene: After the initial testing using single images, the system is applied to a test video scene.

5.2.2.2 Training Data

The training (2,000 positives, 3,000 negatives) data samples are collected as follows:

Training data:

a. Positive samples: 1,500 from the INRIA database + 500 from our database = 2,000 in total
b. Negative samples: Negative samples: 2,300 from the INRIA database + 700 from our database =3,000 in total

5.2.2.3 Testing Data

The testing (600 positives, 600 negatives) data samples are likewise selected as follows:

Testing data (Still images):

a. Positive samples: 500 from the INRIA database + 100 from our database = 600 in total.

b. Negative samples: Negative samples: 400 from the INRIA database + 200 from our database =600 in total.

Testing data (Video):

The details of the video used to test the system are shown in the table below. The video contains numerous movies objects including humans, vehicle. Bicycles, etc. The human size is about 32x64 pixels. However, larger humans than this size are also included.

Table 5.3: Test Video details

Dataset	Total No, of frames	Time duration
Scene T	4,830 frames	161 sec

5.2.2.4 SVM training results

The Figures below show the effect (Learning curves) of training the SVM for 32x64 window size with different SVM parameters. Cell size is (8x8) with (2x2) blocks

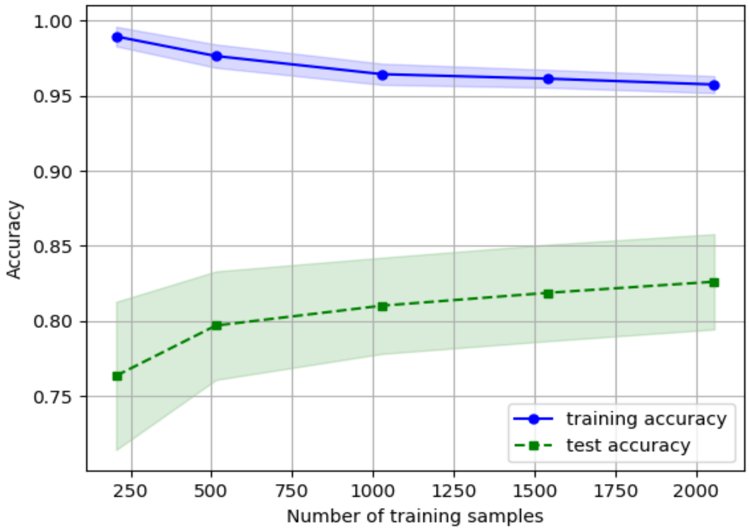


Figure 5.2: Number of training samples

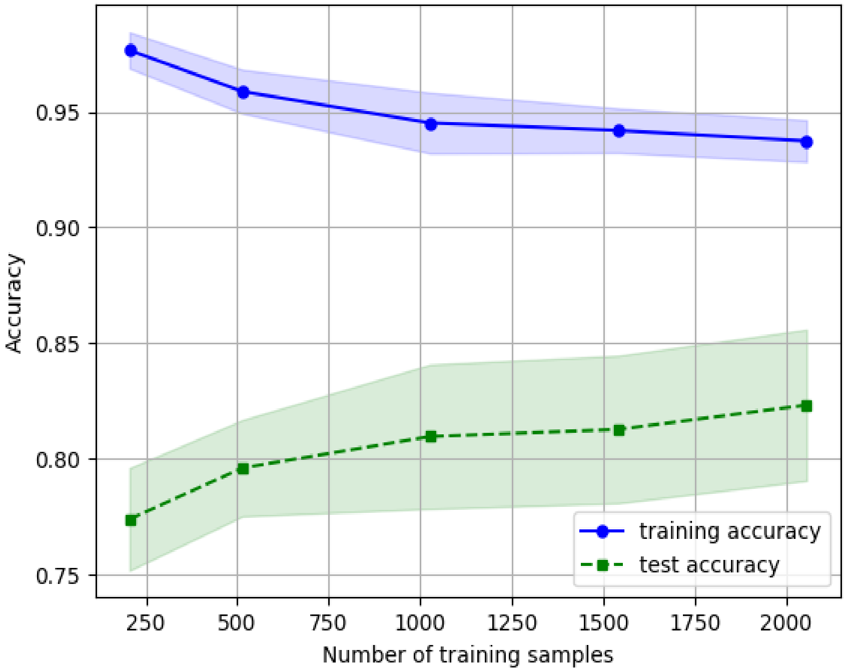


Figure 5.3: Number of training samples, gamma = 0.001

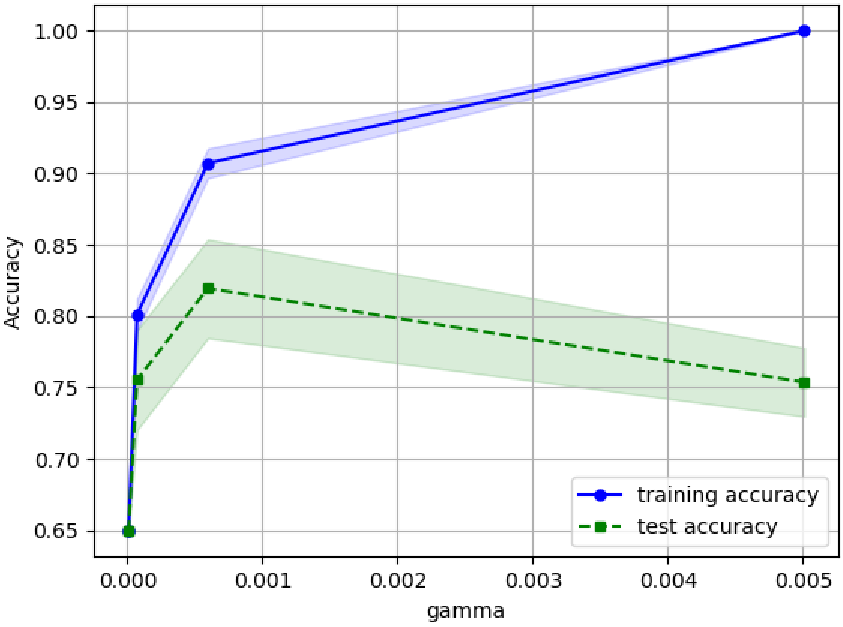


Figure 5.4: Number of training samples, gamma = 0.01

- Validation curve (different 'C' parameter)

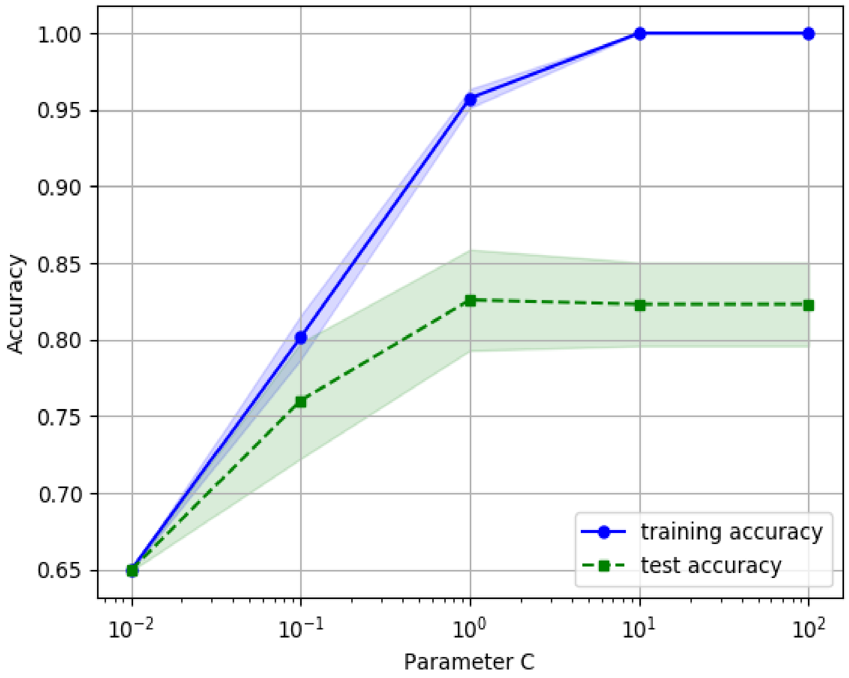


Figure 5.5: Effects of parameter C

It can be concluded that the training progressed successfully using the SVM parameters and the training data.

5.2.2.5 Results

The results of training are shown in Table 5.4. The final accuracy is calculated based on the number of correct detection divided all the humans in all the frames. This correct total number of humans is evaluated manually.

Table 5.4: Results of human detector testing

Widow Size (pixels)	**Still database images**	**Scene T**
32x64	92.2%	90.5%
48x96	90.1%	88.2%
64x28	95.3%	90.2%

5.2.2.6 Basic System and Skin color detection (Speed up)

Searching for human in all locations in the image can be computationally expensive and redundant. To speed up the process, we assumed that humans are highly likely to be near skin color regions. Therefore, we created a skin color detector which is run on every frame before the HOG-SVM detector. Since the skin color regions are less than the total image pixels, the processing speed improved. On average, skin color regions were less than 30% of the total pixels.

The skin color detector is based on the HSV color space. In the end, the detection results remain the same but at a faster processing speed.

In our datasets, the skin color of most humans is detectable due to the capture camera position and distance.

5.2.2.7 Basic System, Skin color detection (SD) and Image pyramids (Size invariance)

The skin color detection improves the speed while maintaining the basic accuracy. However, for human regions larger than 32x64 pixels, the human detection will fail. Our dataset is designed to only capture humans at about 32x64 pixels. However, in scene D especially, subjects are moving away from the camera. This changes the size of the humans.

To solve this problem, image pyramids are applied. The image size is reduced by about 20% scale. We create 2 such images. Therefore, for each frame the HOG-SVM is applied, the 3 images are searched for humans. The final result is a combination of all the humans detected at different scales.

The advantage of this step is to improve accuracy but it comes at an increased computation cost. Luckily, since the HOG-SVM is applied only once every 5 seconds, the effect of this increase is small.

5.2.2.8 Results

The results of training are shown in Table 5.5.

Table 5.5: Speed improvement

Window size	Number of windows (before SD)	Number of windows (After SD) (Approx.)	Total Time taken (Secs)
32x64	253,953	38,000	17.06
48x96	228,305	38,000	35.79
64x128	203,681	38,000	61.19

Note: The number 38,000 represents the approximate number of skin color locations. The different windows are extracted per location.

5.3 System Application on Datasets

After confirming the viability of the system, it is tested on 4 crowded scenes prepared by our group. The main algorithm is divided into two parts, human detection and tracking the specific target.

Before target detection and tracking, we set the features of the target, which are already known to the operator, e.g. black shirts, wearing masks, and glasses. These features are used for particle filter tracking.

In this thesis, our datasets are used for human detection and tracking specific person in video surveillance system. We used four structure and unstructured crowded scene for testing our method. Each scene is different; Scene1, 2, 3, and 4 of recorded video as shown in Table 6.5. All datasets are crowded scenes. The background and illumination of the video is natural condition.

Table 5.6: Environment specification for experiment

Dataset	Total No, of frames	Time duration
Scene A	**4,830 frames**	**161 sec**
Scene B	**1,199 frames**	**35 sec**
Scene C	**1,962 frames**	**65 sec**
Scene D	**2,197 frames**	**73 sec**

5.3.1 Experimental Procedure

The final experiments were conducted using the following procedure. The same procedure is used on all the four datasets.

1. Input video
2. Apply the skin color detector
3. Run HOG-SVM including the image pyramids on skin color regions.
4. Specify the features of the target person
5. Track the target person

5.3.2 Experimental Results

We validate our method on four different data sets. In this section, we discuss the experimental results using the datasets. It involves detecting and tracking specified person walking on structured and unstructured crowd scenes.

In the results, the people who were successfully detected by the HOG feature are surrounded by a blue rectangle in the image. In the processed images, the green dots indicate the particles used for tracking the target.

5.3.2.1 Detecting and tracking a specified walking person in a structure crowd scene

The first experiment is conducted using scene A which is a structured crowd scene. The persons in the scene have different feature (clothes) such as black, red, blue white, sky, black, etc. In this, part the target is wearing black pants and blue shirt. He is mostly occluded by other persons. Result example captured frames are shown in Figure 5.6.

5.3.2.2. Detecting and tracking specified person running on structured crowd scene.

In our second experimental data (Scene B) people are moving away from the camera. This scene involves, human detecting and tracking specified person running on structured crowd scene. Figure 5.7 shows example frames.

5.3.2.3. Detecting and tracking specified person running on unstructured crowd scene.

In the unstructured crowded scenes, seven people are featured for detection and tracking specified person. In this scenario (Scene C), a person is running far away from the camera. This target person wears black pants and red shirt. Figure 5.8 shows some example frame results.

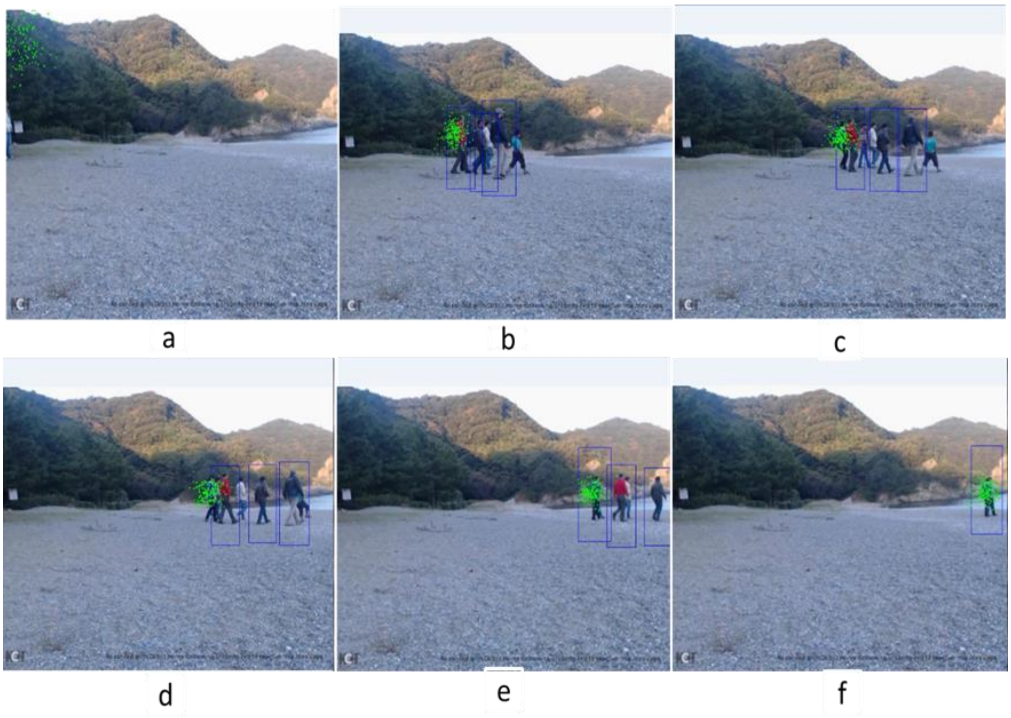


Figure 5.6: Detecting and tracking specified person running on structured crowd scene experimental (a) Initial particle filter, (b) Particle filter tracking (c~f) Particle filter tracking specific person.

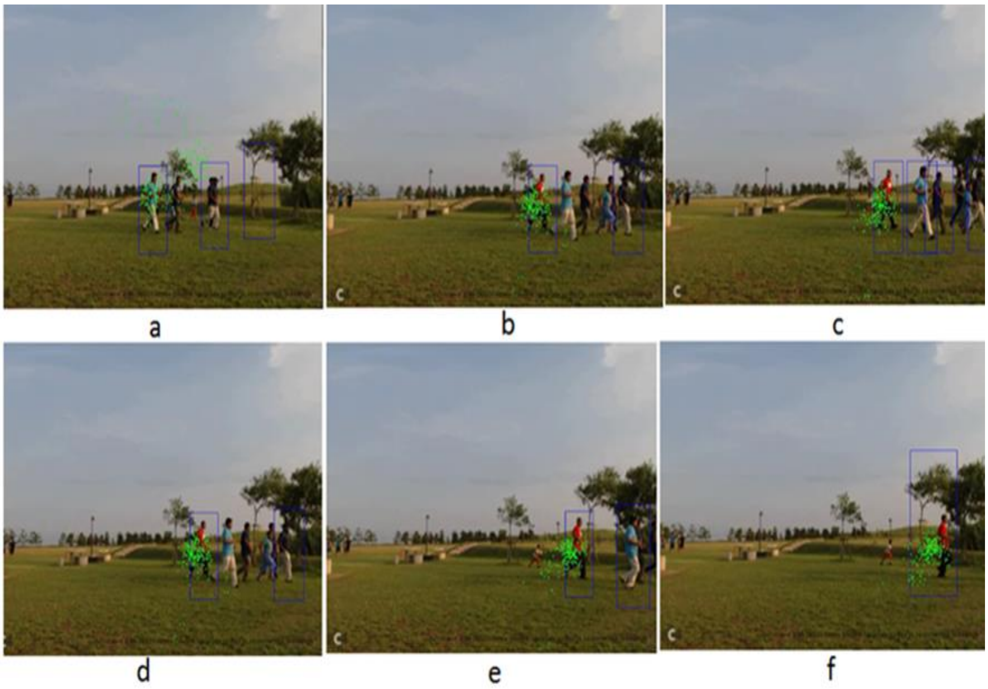


Figure 5.7. Detecting and tracking a specified running person in a structured crowd scene (a) Initial particle filter (b ~f) Particle filter tracking specific person.

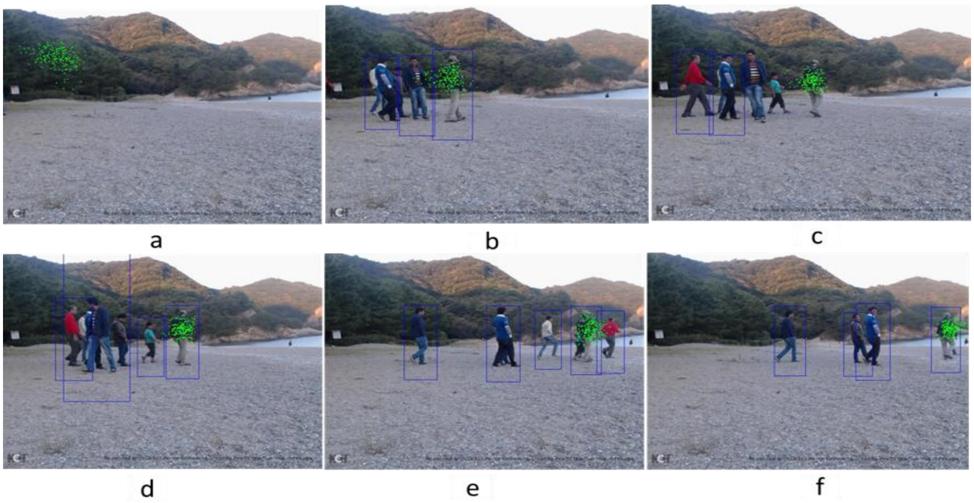


Figure 5.8: Detecting and tracking a specified running person in an unstructured crowd scene (a) Initial particle filter (b~f) Particle filter tracking specific person.

5.3.2.4 Detecting and tracking specified person walking on unstructured crowd scene.

In scene D, the people are walking away from the camera. In this unstructured crowd scene, we track a person wearing a red shirt and black pants. The particle filter works well even under occlusion.

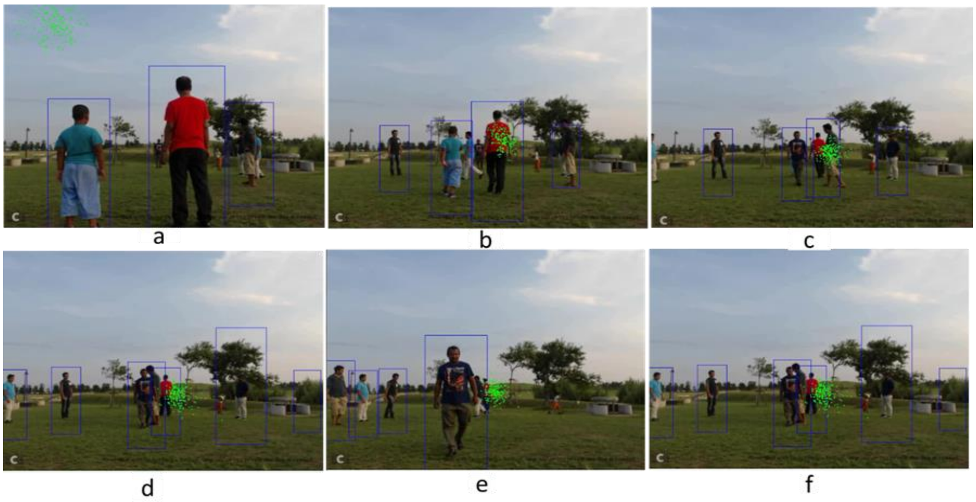


Figure 5.9: Detecting and tracking a specified walking person in an unstructured crowd scene (a) Initial particle filter (b~f) Particle filter tracking specific person

Table 5.7. Shows the experimental results of success and failure rate for detection and tracking the target person in crowded areas.

Table 5.7. Evaluation of proposed method.

Dataset	**HOG feature detection [%]**		**Particle Filter [%]**	
	Success	Failure	**Success**	Failure
Scenario A	**81.0**	19.0	**99.5**	0.5
Scenario B	**92.0**	8.0	**99.7**	0.3
Scenario C	**93.0**	7.0	**98.7**	1.3
Scenario D	**90.0**	10.0	**99.9**	0.1

The experimental results show that our system achieved more than 90% of human detection and for particle filter tracking specific person around 99% in all dataset.

5.4 Comparative experiments

In this section, we will compare the proposed method to existing ones. For practical reasons, we will compare separate systems; HOG-SVM detector and particle filter.

5.4.1 HOG-SVM detector

As can be shown on table 5.2 and 5.5, the results of our detector compares well to the original HOG method. However, our method performs better in terms of processing speed because of two factors:

1. Smaller window size
2. Search area reduction using skin color detection

It should however be noted that our test videos capture the humans at about the proposed window size.

5.4.2 Particle Filter

The conventional particle filter is used in this experiment. Therefore, it should perform like other researcher’s methods. However, the settings of the number of particles, variance and likelihood could lead to different results.

5.5 Discussions

In this work first we transformed RGB color to HSV color for skin color detection. RGB color is used for displays like television, computer monitors, etc. But HSV color space is used for human and skin color detection. When we finished transforming the color space from RGB to HSV, we used the Histogram Oriented of Gradients (HOG) feature descriptor for object detection. In object detection there are two main processes component-based and single-window based techniques.

In our research, human detection we use single window object detection technique. The HOG-SVM detector is used to detect humans in images. We have two categories SVM linear and non-linear. If the dataset has high variance, that time we need reduce the number of features then we apply non-linear algorithm for classification technique. If the datasets have low variance, then we apply linear SVM. In this work, we apply linear SVM classifier technique. When we apply SVM classifier we can find humans inside the datasets.

Our particle filter is the implementation of the recursive Bayesian filter by Monte Carlo method. It represents the intensity of the background group of random particles with the associated weights and estimates. In our method, the Bayesian filter is used by approximating the previous and background density functions with a set of target features. Hence, the resulting particles move to the target set according to the set of individual evidence. To prepare a particle filter, we use the number of particles variance to randomize particle and scanning range for likelihood function as the parameters. When we are successfully tracking the specific target person, the particles gather around the specific target area. We have also defined the target feature parameter as skin colors, red, blue, and black shirts. For particle filters, the particle number is set to 200 the particle is defined as 8.0 and the possibility to randomize the scanning range as 30 pixels in 30 seconds. Setting different values could produce different results.

For experimental results we apply our datasets in crowded area. Crowded area scene have two probabilities: (1) structure crowded scene and (2) Unstructured crowded scene. In our experiment we discuss these four scenes as follow, Scenario (A): People walking in a structure crowd scene, Scenario (B): People running in a structure crowd scene, Scenario (C): People walking in an unstructured crowd scene and Scenario (D): People running in an unstructured crowd scene. After experiment we find that the scene when people are walking Scenario (A) human detecting is 81% and the particle filter is 99.5%. In the running Scenario (B) human detection 90% and the particle filter is 99.7%. In the unstructured scene when people are walking in Scenario (C) human detection is 93% and the particle filter success is 98.7%. In the unstructured scene when people are running Scenario (D) human detecting success is 93% and the particle filter success 98.7%. From these results, we confirmed that our proposed method that combines HOG-SVM and particle filter with the target feature is strong towards the occlusion problem.

5.5.1 Possible issues

Accuracy:

1. **Training data size**: It could be affected by the window size and training of the HOG-SVM detector. In our work, we use only 2000 positive and 3,000 negative samples. So the training accuracy is only about 85%. However, this is not a big issue because the detector is applied only once every 5 seconds.
2. **Skin color detector**: By applying HOG-SVM on the skin color detector results only we risk the possibility of miss-detections due to skin color failure.
3. **Image pyramids**: Image resampling reduces the details in the image and this could lead to poor HOG features for detection. We limit the resampling to 2 times only. Only down sampling is considered.
4. **Occlusions**: Although the particle filter performance is good, continuous occlusion can affect the accuracy. Long occlusions are defined as periods beyond 2 seconds. If such situation arises, the tracking is stopped and the process from the human detection step is restarted.

Processing Speed:

1. **Skin color detector**: It reduced the number of location to apply the

HOG-SVM. This resulted in improved processing speed as opposed to the conventional system.

2. **Image pyramids**: Although this method allows us to detect humans at different sizes, the detector must be applied at every resolution. This reduces the operating speed of this work.

Chapter 6

Conclusion and Future work

In this chapter, we summarize the main contributions of this thesis and thus conclude the work. According to these observations, possible improvements and some interesting trends are referred to for future research.

6.1 Achieved goals

This dissertation is the result of three years of doctorate studies. The main objective was to study, analyze, propose and develop structures, models and algorithms for video surveillance of people in the crowded area. During these three years, all units were considered for a real video surveillance system.

Outdoor videos are more challenging than indoor videos due to lighting changes, complex backgrounds and applicability. High amount of noise and the uncertainty observed in the external sequences makes tracking in these sequences a religious problem to solve. We have designed a tracking system that can detect and track moving objects in the external video. After creating a platform, we were able to make important tracking improvements using new algorithms. Thus, we have successfully achieved our goals in establishing a system that could serve as a platform for future automated video surveillance research.

The paper's main objective is to continuously track a specified human in a crowded scene. Two processes are necessary to accomplish this: Human detection and tracking. The paper proposes use of HOG to detect and particle filter to track, since they are well established in this field. The main contributions of this work are as follows:

1. To improve the speed of the HOG feature calculation, a smaller feature window is used. The original HOG window 64x128 pixels for 3,780 features. The proposed HOG is just 32x64 pixels for 756 features. The size is four times smaller.
2. The linear SVM is used to learn the HOG feature to detect human regions. No improvements have been added but fewer training samples are used, 2,000 humans and 3,000 non-humans.

3. When searching, the conventional method is to calculate the HOG feature for the all image pixel locations. This is very time consuming. The proposed method suggests an initial step to detect skin color regions before, and apply the HOG only on these color pixels. Experiments show that, on average skin color regions are less than a quarter of the pixels, improving the speed by least 4 times.
4. The pyramid method is used to detect regions larger than the selected window. The image size is reduced by 20% for 2 times.
5. Once the human regions are detected, a target person is selected for tracking. The selection is based on the person's clothes color, etc. The color is captured using a histogram and later used as the particle filter's feature. Initially, the conventional particle filter is applied using about 200 particles.
6. To track during occlusion, the last know particle weights, location and speed before occlusion are saved. The weights are updated based on the last known speed and location for about 2 seconds. After that, the search area is doubled and searched again. If the target cannot be found, tracking is considered to have failed.
7. To ensure continuous tracking, the human detector is applied once every 5 second to reconfirm the human regions (followed by the particle filter). Tracking accuracy of about 99% was achieved.

Therefore, as a result of evaluation using the dataset including occlusion problem, our proposed method can be applied to various scenes where several people are crossing.

6.2 Future Work

Now we proposed two guidelines for the future expansion of work in this thesis. Improvements to the detection and tracking algorithms of the current system, for protecting the elderly and the socially vulnerable people and extension system to create smart applications high-level robust motion tracking in crowded scene.

In terms of improving the current system, there are many opportunities for future work in object detection and object tracking, Monte Carlos and Bayesian tracking algorithm sections.

References

[1] S. Ali, M. Shah, "*Floor Fields for Tracking in High Density Crowd Scenes," The 10th European Conference on Computer Vision (ECCV), 2008,* Vol. 5303. Springer, Berlin, Heidelberg.

[2] S. Ali, M. Shah, "*A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis," IEEE International Conference on Computer Vision and Pattern Recognition,* 2007, pp. 1–6.

[3] M. Rodriguez, S. Ali; T. Kanade, "*Tracking in unstructured crowded scenes," 2009 IEEE 12th International Conference on Computer Vision, ISSN: 1550-5499, ISBN: 978-1-4244-4420-5,* 2009, pp.1389-1396.

[4] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. on Automation and Control*, Vol. AC-24, pp. 84-90, 1979.

[5] M. Gelgon and P. Bouthemy, "A region-level motion-based graph representation and labeling for tracking a spatial image partition," *Pattern Recognition*, Vol. 33, pp. 725–740, 2000.

[6] I. Haritaoglu, D. Harwood, and L.S. Davis, "W4: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, no. 8, pp. 809–830, 2000.

[7] N. Amamoto and A. Fujii, "Detecting obstructions and tracking moving objects by image processing technique," *Electronics and Communications in Japan*, Part 3, Vol. 82, no. 11, pp. 28–37, 1999.

[8] Collins, Lipton, Fujiyoshi, and Kanade, "Algorithms for cooperative multisensory surveillance, *Proc. IEEE*, Vol. 89, pp. 10, 2001.

[9] C. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, no. 7, pp. 780–785, J, 1997

[10] S. Munder, C. Schnoerr, and D. M. Gavrila, "Pedestrian Detection and Tracking Using a Mixture of View-Based Shape-Texture Models," *IEEE Transactions Intelligent Transportation Systems*, Vol. 9, No, 2, pp. 333-343, 2008.

[11] L. Teng, et al, "Crowded scene analysis: A survey," *IEEE transactions on circuits and systems for video technology*, Vol 25, no, 3, pp. 367-386, 2015.

[12] J. Aggarwal, Cai, Q, "Human motion analysis: a review," *Computer Vision and Image Under-standing*, Vol 73, Issue, 3, pp. 428–440, 1999.

[13] L. Wang, Hu, W., Tan, T, "Recent developments in human motion analysis," Pattern Recognition. Vol, 36, no, 3, pp. 585–601, 2003.

[14] W. Hu, Tan, T., Wang, L., Maybank, S, "A survey on visual surveillance

of object motion and behaviors," *IEEE Trans Systems, Man and Cybernetics*. Vol, 34, no, 3. pp 334–352, 2004.

[15] B. Zhan., et al," Crowd analysis: a survey," *Machine Vision Applications*. Vol, 19(5–6), pp 345–357, 2008.

[16] Z. Qi, Ting, R., Husheng, F., Jinlin, Z, "Particle filter object tracking based on Harris-SIFT feature matching," *Procedia Engineering,* Vol. 29, pp 924–929, 2012.

[17] H. Liao, "*Human detection based on Histograms Oriented Gradients and SVM*," Pattern Recognition EECE-7313, SPRING, 2013.

[18] M. Wojnarski, "Absolute contrasts in face detection with AdaBoost cascade," https://www.mimuw.edu.pl/~mwojnars/papers/07jrsadaboost.pdf.

[19] N. Dalal and B. Triggs, "*Histograms of oriented gradients for human detection," Conference on Computer Vision and Pattern Recognition*, Vol, 1, pp. 886-893, 2005.

[20] Ruiyue et al, "Multiple human detection and tracking based on head detection for real-time video surveillance," *Journal multimedia Tools and Applications,* Vol, 74 Issue- 3, pp. 729-742, 2015.

[21] X. Wang, X. Han, and S. Yan, "*An HOG-LBP Human Detector with Partial Occlusion Handling," In IEEE International conference on Computer Vision*, ISSN: 1550-5499. ISBN: 978-1-4244-4420-5, pp. 32-39, 2009.

[22] C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection," *International Journal of Computer Vision*, Vol 38, no, 1, pp.15-33, 2000.

[23] J. Berclaz, F. Fleuret, and P. Fua, "*Robust People Tracking with Global Trajectory Optimization," In IEEE Conference on Computer Vision and Pattern Recognition*, Vol, 1, pp. 744-750, 2006.

[24] M. Enzweiler and D. M. Gavrila, "Monocular Pedestrian Detection: Survey and Experiments," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol, 3, no, 12, pp.2179-2195, 2009.

[25] D. M. Gavrila, "*Pedestrian Detection from a Moving Vehicle," In Proc. of European Conference on Computer Vision*, pp. 37-49, 2000.

[26] D. M. Gavrila, "A Bayesian, Exemplar-based Approach to Hierarchical Shape Matching," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol, 29, no, 8, pp.1408-1421, 2007.

[27] P. Viola, M. Jones, and D. Snow, "*Detecting Pedestrians using Patterns of Motion and Appearance," In International Conference on Computer Vision*, pp. 734-741, 2003.

[28] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-Based Object

Detection in Images by Components," *IEEE Transaction on Pattern*. Vol, 23, no, 4, pp. 349-361, 2001

[29] S. Kang, H. Byun, and S. W. Lee, "*Real-Time Pedestrian Detection Using Support Vector Machines," In Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines*, pp. 268-277, 2002.

[30] I. P. Alonso, D. F. Llorca, M. A. Sotelo, L. M. Bergasa, P. Revenga deToro, J. Nuevo, M. Ocana, and M. A. G. Garrido, "Combination of Feature Extraction Methods for SVM Pedestrian Detection," *IEEE Transactions on Intelligent Transportation Systems*, Vol, 8, no,2, pp.292-307, 2007.

[31] L. Pishchulin, A. Jain, C. Wojek, T. Thormaehlen, and B. Schiele, "*In Good Shape: Robust People Detection based on Appearance and Shape," In Proceedings of the British Machine Vision Conference*, pp. 5.1-5.12, 2011.

[32] C. Wojek, G. Dorko, A. Shulz, and B. Schiele, "*Sliding-Windows for Rapid Object Class Localization: A Parallel Technique," In DAGM-Symposium*, pp. 71-81, 2008.

[33] S. M. Khan and M. Shah, "*A Multiview Approach to Tracking People in Crowded Scenes Using a Planar Homography Constraint," In European Conference on Computer Vision*, pp. 133-146, 2006.

[34] M. C. Liem and D. M. Gavrila, "*A Comparative Study on Multi-Person Tracking Using Overlapping Cameras," In the 9th International Conference on Computer Vision Systems*, pp. 203-212, 2013.

[35] S. M. Khan and M. Shah, "Tracking Multiple Occluding by Localizing on Multiple Scene Planes," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol, 31, no, 3, pp. 505-519, 2009.

[36] D. Delannay, N. Danhier, and C. D. Vleeschouwer, "*Detection and Recognition of Sports (Wo) Men from Multiple Views," In Proceedings of ACM/IEE International Conference on Distributed Smart Cameras*, pp. 1-7, 2009.

[37] R. Eshel and Y. Moses, "*Homography Based Multiple Camera Detection and Tracking of People in a Dense Crowd," In International Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[38] K. Kim and L. Davis, "*Multi-Camera Tracking and Segmentation of Occluded People on Ground Plane using Search-Guided Particle Filtering," In European Conference on Computer Vision*, pp. 98-109, 2006.

[39] F. Feuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probalistic occupancy map," *IEEE Transactions on*

Pattern Analysis and Machine Intelligence, Vol, 30, no, pp. 267-282, 2008.

[40] A. Alahi, Y. Boursier, L. Jacques, and P. Vandergheynst, "*A Sparsity Constrained Inverse Problem to Locate People in a Network of Cameras," In Proceedings of the 16th International Conference on Digital Signal Processing*, pp. 22-28, 2009.

[41] D. B. Yang, H. H. Gonzalez-Ba~nos, and L. J. Guibas, "*Counting People in Crowds with a Real-Time Network of Simple Image Sensors," In International Conference on Computer Vision*, pp. 122-129, 2003.

[42] A. Utasi and C. Benedek, "*A 3-D Marked Point Process Model for Multi-View People Detection," In International Conference on Computer Vision and Pattern Recognition*, pp. 3385-3392, 2011.

[43] A. Utasi and C. Benedek, "A Bayesian Approach on People Localization in Multi-Camera Systems," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol, 23, no, 1, pp. 105-115, 2012.

[44] B. M. Hossain, S. Karungaru, K. Tereda, "Human Detection and Tracking Using HOG Feature and Particle Filter in Video Surveillance System", *International Journal of Advanced Intelligence,* Vol 9, no 3, pp.397-407, 2017.

[45] Segan and S. Pingali, "*A camera based-system for tracking people in real time," IEEE Proc. of Int. Conf. Pattern recognition,* Vol 3, Pp. 63-67, 1996.

[46] I. Haritaoglu, "*Detection and tracking shopping groups in stores," Proceedings of the IEEE computer society conference on computer vision and pattern Recognition*, ISSN: 1063-6919, pp. 0-7695-1272-0, 2001.

[47] P. Borah and D. Gupta "Review, "Support Vector Machines in Pattern Recognition", *International Journal of Engineering and Technology* (IJET), Vol, 9, no, 3S July 2017.

[48] A. Doucet, S. Godsill and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering, *Statistics and Computing*, Vol, 10, pp. 197–208, 1999.

[49] Y. M.-Attias, "*Persistent Particle Filters for Background Subtraction," The Hebrew University of Jerusalem, Faculty of Mathematics and Sciences School of Computer Science and Engineering*, Jerusalem, 2010.