UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

HUMAN DETECTION AND TRACKING ENHANCING SECURITY SYSTEMS AT PORTS OF ENTRY

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

MOUHAMMAD AL AKKOUMI
Norman, Oklahoma
2011

HUMAN DETECTION AND TRACKING ENHANCING SECURITY SYSTEMS AT PORTS OF ENTRY

A DISSERTATION APPROVED FOR THE
DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

BY

Dr. James J. Sluss, Jr., Chair

Dr. Samuel Cheng

Dr. Pramode K. Verma

Dr. Monte P. Tull

Dr. William O. Ray

Dedication

I dedicate this work to Allah for helping and blessing me throughout my life.

Acknowledgments

I would like to express my sincerest gratitude to my advisor, Dr. James J. Sluss, Jr. for his excellent guidance, patience and support. And to Dr. Pramode Verma for his trust, mentoring, and providing me the opportunities to assist in teaching telecommunications classes and managing the telecommunications lab, thank you.

Many thanks to my PhD committee members for all the great input and recommendations they have given me. I would not have been able to complete this work without them. I would also like to acknowledge all of those who took part in the testing process.

My never-ending thanks go to my mom and dad for their continuous giving and for being the biggest fans in my life. A special thanks to Dr. Robert C. Huck for supporting and motivating me in my educational career. Dr. Huck influenced me greatly both as a researcher and as a student of life. Also, I would like to thank my best friend, Wael El Wali, for being there for me and encouraging me to achieve my goals.

I would like to thank the TCOM family and the OU-Tulsa family for taking care of me in my academic journey and creating a second home for me in Oklahoma. Last but not least, I would like to thank my brothers and sisters for their encouragement and kind support.

Table of Contents

List of Figures

List of Tables

Abstract

The dissertation undertakes the critical application of establishing smarter surveillance systems to improve security measures in various environments. Human detection and tracking are two image processing methods that can contribute to the development of a smart surveillance system. These techniques are used to identify and detect moving humans in a surveyed area. The research enables the incorporation of personnel detection and tracking algorithms to enhance standard security measures that can be utilized at ports of entry where security is a major hurdle. This system allows authorized operators on any supported console to monitor and receive different alerts levels to indicate human presence.

The presented research focuses on two human detectors based on the histogram of oriented gradients detection approach and the Haar-like feature detection approach. According to the conducted experimental results, merging the two detectors, results in a human detector with a high detection rate and lower false positive rate. A novel approach to use both detectors is proposed. This approach is based on a feedback messaging system that inputs parameters from both detectors to output better detection decisions. An object tracker complements the detection step by providing real-time object tracking. An alert system is also proposed to automatically report potential threats occurring in the surveyed area.

Chapter 1

Introduction

Establishing exceptionally accurate pedestrian detection and tracking are two major hurdles facing computer vision today. Overcoming these challenges can result in providing more secure surveillance systems to monitor indoor and outdoor spaces. Some places where such systems are being utilized are airports, subways, banks, shipping ports, etc. Human detectors and trackers are limited by their accuracy and ability to minimize false detections. Achieving higher detection accuracy can improve the reliability of these image processing applications so that they can be used for security systems. In this research, a more robust and smart surveillance system is developed to detect and track pedestrians moving around a secure area. This surveillance system is an automated human tracking and detection system that sends alerts of human presence or object motion in a secured area to authorized personnel. The proposed system can increase the detection rate and decrease the errors produced by the detectors and trackers using a feedback messaging system. For clarification, the following definitions are used in this research:

- Human Detection: is the process of finding and classifying an object of interest as a human in a given area of investigation in an image.
- Object Tracking: is the process of locating a moving object and identifying it in a given set of frames.
- False Positive: is falsely detecting an object of interest in a given area of investigation in an image when an object does not exist.

- False Negative or Non Detection: is not detecting an object of interest in a given area of investigation in an image when an object does exist.

The motivation for the research presented in this dissertation was to find efficient ways to develop more secure surveillance systems at ports of entry. In a project sponsored by the Federal Highway Administration and administered by the Oklahoma Department of Transportation, a low cost approach to improve security at the Port of Catoosa in Oklahoma, Figure 1.1, was proposed. This system provides building blocks for a complete security system that aims at achieving overall port security [1, 2]. The system included off-the-shelf hardware that integrated cameras, wireless sensors, autonomous ground, and areal unmanned vehicles to secure containerized freight entering the United States. One of the next steps that can be used to establish a more secure port is the use of image processing techniques in a smart surveillance system.

Figure 1.1 – Aerial Snapshot of the Port of Catoosa, OK. [Photo provided by the Port of Catoosa]

There are many ways to detect humans in still images and video, but most of these can be categorized in one of two major groups: part-based or feature-based detection. In part-based detection, the application's aim is to find body limbs of a human in an image. Also, in some cases, part-based detection can be the recognition of holistic bodies that are matched with a given body template. On the other hand, feature-based detection focuses on extracting certain features that can be used in the decision-making process to determine whether a human is present in an image.

In video, the part-based detection approach can be initialized in three major steps [3]. The three steps are kinematic structure initialization, shape initialization, and appearance initialization. The first step involves knowing the initial kinematic structure of the person with a fixed number of joints and with specified degrees of freedom. The second, shape initialization, is performed using a humanoid model to fit the person in the video frame.

An example of the third step, appearance initialization, is clothing initialization. Using these three steps, pedestrian recognition can be achieved with relatively high accuracy in a given set of frames. In contrast to the part-based detection approach, feature-based detection does not require searching for body limbs, but rather for pixel information that is unique when used in classification methods. One common classification method is the Support Vector Machine (SVM) [4, 5]. Using a linear support vector machine, classifying the data outputs of the proposed detection system is made possible to improve the decision-making process.

After detecting potential targets of interest, tracking is the next essential step to stay on the detected target as it moves through the video stream. The overall system aims at enabling pedestrian tracking upon detection. Pedestrians present in any scene will be identified by a colored bounding box.

Several challenges face tracking applications that might trick these systems into making false decisions. Occlusion by other objects or pedestrians is a major concern for example, pedestrians move around in staging and dock areas that are filled with containers and other objects that can block their view by the camera. In [6], Aggarwal exploited a solution for performing tracking under occlusion. A simple object tracker proved to be efficient to perform tracking in outdoor scenes. However, the challenges that come with providing accurate human detection and tracking applications are many. Among them are two major hurdles: computational cost and average precision. Current state of the art techniques offer limited success in overcoming either hurdle efficiently.

Achieving high detection rates in real time increases the required computational power. Another factor that affects the system performance is the false positive rate. A false positive occurs when the detection system detects an object that is not of interest (i.e., not a human). To reduce this rate, a feedback system that makes use of two strong detection approaches, the histogram of oriented gradients (HOG) [21] and the Haar-like feature detection, [8], is designed. The feedback system provides informative messages that are sent between the two detection systems. This implementation decreases the overall error rate and can increase the detection rate. The proposed approach is a novel way to perform pedestrian detection using a feedback system based on two detection schemes.

The research objectives of this dissertation are:

- Perform accurate pedestrian detection in container staging areas under various image processing challenges.
- Improve overall port security by providing a means to detect unauthorized personnel in restricted areas and report them to port authorities.
- Establish a smart surveillance system that automatically alerts users of suspicious actions.
- Track detected pedestrians in captured video frames to determine the motion paths of those pedestrians present in the current scene.
- Detect potential threats and assess risks of harmful scenarios that include tampering with containers in staging areas.

Research Applications:

- Port security: inland and coastal ports are in great need of improving their current security systems and providing low cost techniques to strengthen the overall security system.
- Smart surveillance systems: can be used at airports, train stations, road intersections, and other areas where surveillance is required.
- Intelligent transportation systems: road side assistance initiatives where vehicles are equipped with cameras to detect pedestrians and alert drivers of their locations.

Research Contributions:

- Integrated an object tracker and two pedestrian detectors to achieve better human detection and tracking with lower error rates.
- Created an interoperable video surveillance system and demonstrated the feasibility of merging more than one detector to perform human detection.
- Managed the two detection techniques using a feedback messaging system that improves the overall performance of the system.
- Built a smarter surveillance system that can alert security personnel of suspicious activity in a secured area.
- Built another 'building block' to the overall security system proposed in [1] that will enhance security at ports of entry.

The organization of the remaining text of this dissertation is as follows: Chapter 2 contains an overview of the Haar-like features detection scheme and shows experimental results of its application. Chapter 3 reviews the HOG detection approach and shows the experimental results when applying it. Chapter 4 explains the overall object tracking system. An object tracker is shown to be sufficient to provide pedestrian tracking for the goal of this dissertation. In Chapter 5, experimental results of the human detection and tracking system are presented. .The feedback messaging system used between the Haar-like features and the HOG detectors is analyzed and shown to overcome some of the challenges faced by each detector solely. Additionally, an automated alerting system based on the human detection and tracking methods is shown. The last Chapter includes a conclusion, closing remarks and areas for future research.

Chapter 2

Haar-like Feature Pedestrian Detector

2.1 Introduction

Pedestrian detection is a fast growing and promising technique used in various applications to find humans in given images. Researchers are trying to accomplish this type of detection using methods that result in high accuracy and fast computation. A notable approach to pedestrian detection was the one proposed by Viola et al. in [7] that shows the feasibility of detecting pedestrians by integrating image intensity information with motion information. The authors use a trained detector to detect pedestrians appearing as small as 20x15 pixels in a video with a frame rate of 4 frames per sec. There were two kinds of detectors used for this application: a dynamic detector and a static detector.

The dynamic detector is trained using consecutive frame pairs while the static detector used static patterns. During the training process, the two detectors go through thousands of classifiers to achieve sufficient training, thus producing acceptable results. Using the dynamic detector, a false detection rate of 1/400,000 was shown; whereas for a static detector the rate was 1/15,000. This approach is well suited to perform pedestrian tracking in outdoor scenes. Viola's approach is a unique study in which the detectors were trained based on motion and appearance simultaneously. With margins of one second, the system was able to detect pedestrians in very low resolution video frames. This capability makes this type of detection very appealing, especially for outdoor applications such as surveillance at a port facility.

To establish a low cost smart surveillance system, off-the-shelf network cameras are primarily used in this proposed research. Basically, any type of camera that can record a 640x480 pixel video can be used for image capturing. Though this resolution is not an essential requirement, other resolutions were used in the conducted experimental results. Choosing a higher resolution increases the computation time while choosing a much lower resolution can potentially increase the false positive rate.

The detection method adopted for fast pedestrian detection is the Haar-like feature pedestrian detection (HFPD). The main advantage for using Haar-like features in pedestrian detection is the fast detection capability. While this method is not very accurate on its own, it is generally used in near real-time to or real-time object detection. The fast detection factor makes up for the accuracy factor in applications where the quick capturing of objects of interest is of higher importance. As explained in later chapters, this factor fits well in the overall detection system presented in this dissertation.

The rest of the chapter is organized in the following manner: Section 2.2 gives a brief look at the Haar-like feature detection literature, with currently proposed approaches. Section 2.3 defines Haar wavelets and Haar-like features and describes some applications. Section 2.4 provides an overview of the Viola and Jones approach. Section 2.5 specifies the training process of the Haar-like feature detector and the parameters used for it. Experimental results of the HFPD are presented in Section 2.6. A summary of the chapter is stated in Section 2.7.

2.2 Background

Several researchers used Haar-like features for object detection. Viola and Jones were among the first to propose an approach for detecting objects in images based on Haar-like features in 2001 [8]. This approach has been used previously to perform face detection, upper and lower body detection, and full body detection with moderately good detection results [9-11]. While face detection was introduced first and showed very promising results, Haar-like feature detection has been used in many other human and object detection algorithms.

The Viola and Jones detector has been used in different applications to perform fast object recognition. One of the drawbacks of this detector is its detection inconsistency with object rotation in images. In [12], Kolsch and Turk proposed a Viola and Jones detector to do hand detection with a degree of rotation. The detector was trained using a dataset that contained images of hands with different angles of rotation. The results showed an increase of one order of magnitude in the detection rate of the hand in input image frames.

A more advanced version of the Viola and Jones approach was proposed in [13] by Mita et al. The authors introduce a new approach for face detection using joint Haar-like features. The joint features are located through the co-occurrence of face features in an image. The classifiers are then trained using these features under adaptive boosting (AdaBoost). The results showed a faster detection time, 2.6 times faster, with similar face detection accuracy. The joint Haar-like features also reduced the overall detection error by 37% compared to the traditional Viola and Jones approach.

While Haar-like features are mostly used for human feature detection, they can be used for other detection methods. One approach was the one proposed by Han et al in [14], using Haar-like features for vehicle detection. This method was divided into two main steps, a hypothesis generation step and a hypothesis verification step. Vehicle candidates were chosen during the first step and it was then determined whether there was sufficient evidence to classify each as a vehicle. The results showed a detection rate of 91.2% for vehicles moving in the same lane with a sensing rate of 15 frames per second. Changes in illumination due to road environment caused an increase in the error rate for the detector.

In [15], Cui et al demonstrated a 3D Haar-like feature pedestrian detector that was used with a SVM classifier. According to Cui, capturing motion and appearance in 3D is more effective for video based detection. The results showed a detection rate of 91% and a false positive rate of 5%. The training for the detector was done on 4000 positive and 4000 negative images, whereas the testing was performed on 2000 positive and 2000 negative images.

The approach introduced by Viola and Jones, as described above, used Haar-like features on grayscale images. A study performed by Chang and Cho, [16], investigated the use of color-based Haar-like features in the detection process. According to the authors, implementing one strong classifier based on the color-based Haar-like features can result in a better detection rate when compared to the traditional Haar-like feature detection approach. The test results were conducted on a vehicle to perform detection of pedestrians, vehicles and motorcycles. The proposed multi-class boosting algorithm showed better detection and better accuracy than the two-class algorithm.

2.3 Haar-like Features and Their Applications

Haar-like features are based on the Haar wavelets that portray a simple on-off status shifter. The Haar-like features are most suitable for the detection algorithm implemented for fast computation purposes. The Haar-like features were used in the first real-time face detection algorithms due to the computation reduction it can achieve using image intensities.

2.3.1 Haar-wavelets

A Haar wavelet is the simplest type of wavelet that can be denoted by a value of 1 in one interval and 0 elsewhere. It was created by Alfred Haar in 1909 as an example of countable orthonormal systems. Haar wavelet functions were first used for counting square-integrable functions. It was then used in machine failure monitoring, image compression, power systems analysis, and other applications. The Haar wavelet basis or mother function is given by the following equation:

$$\Psi(t) = \begin{cases} 1 & if\ 0 \le t < \frac{1}{2}, \\ -1 & if\ \frac{1}{2} \le t < 1, \\ 0 & otherwise \end{cases} \qquad \text{Eq. 1}$$

A basis function is a combination of transformed scaling functions. A scaling function can be defined as follows:

$$\varphi(t) = \begin{cases} 1 & if\ 0 \le t\ < 1, \\ 0 & otherwise \end{cases}$$

The simplicity of the Haar wavelet makes it a great candidate for use in simple classifiers. Figure 2.1 shows a Haar wavelet of magnitude 1 and step size 0.5. The magnitude of the function as well as the step size can be varied according to the system usage. To monitor a machine status over time, the machine can be idle, working or damaged. Each status can be described by a different level at a chosen period of time.

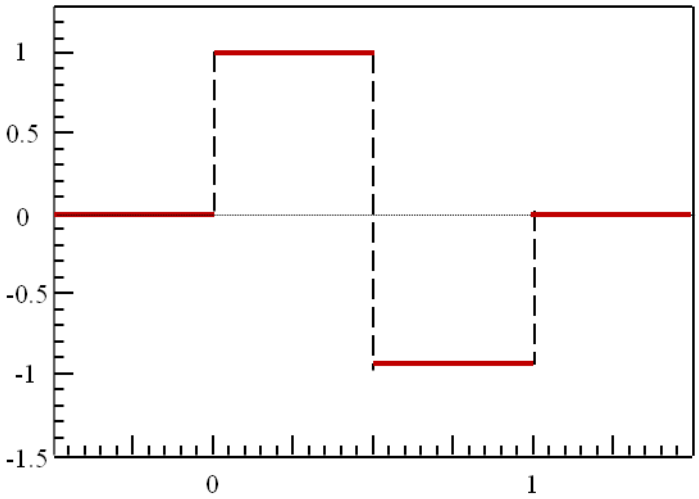


Figure 2.1 – Haar Wavelet Function.

2.3.2 Haar-like features

The use of HAAR-like algorithms simplifies locating all the desired features. A feature is selected if the difference between the average dark region pixel value and the average of the light region is higher than a preset threshold. An example of HAAR features is shown in Figure 2.2. As shown in the figure, the features can be used to detect different pixel orientations throughout a defined region of interest. A combination of a certain arrangement of edges can then be identified as the desired object or not. The features present in Figure 2.2 are either 2-rectangle or 3-rectangle features. 4-rectangle features are also used in other implementations of Haar-like features. The feature can be computed quickly using integral images which are defined as two-dimensional lookup tables and have the same size as the input image. A 2-rectangle Haar-like feature needs 6 lookups while a 3-rectangle feature needs 8 lookups and a 4-rectangle feature needs 9 lookups.

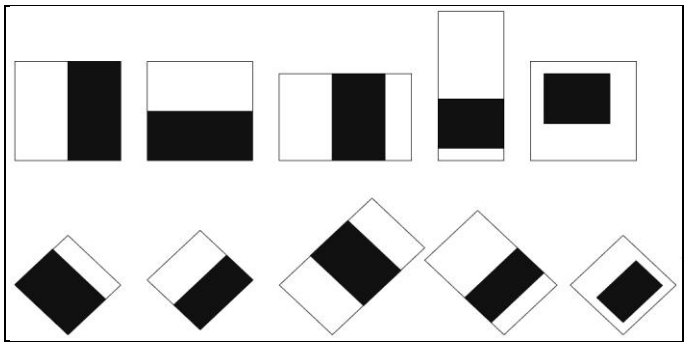


Figure 2.2 – Haar-like features.

2.4 Viola and Jones Human Detector

Feature detection establishes a more efficient recognition process by encoding information related to the detection class. This process is possible, using Haar-like features, to calculate oriented pixel intensity differences throughout all regions in an image. Pixel contrasts and their spatial relationships can be used to determine human presence in an image.

The classifier proposed by Viola is basically a cascade of boosted classifiers that make use of the Haar-like features in the detection process. The classifier is trained using several hundreds of positive and negative samples. A positive sample is an image that contains one or more captures of the desired object. A negative image is the one that does not contain the desired object. The positive samples are then scaled to the same size and used during the training process.

To determine whether an image contains the object of interest or not, the image is scanned using a window of detection that undergoes all stages of the classifier. Each stage is a boosted classifier that is applied to the detection window. If the detection window passes all the stages, then the desired object is detected. Otherwise, the detection window is moved to the next portion of the image. The classifiers are scale adaptive and can be easily resized rather than resizing the whole image.

Trying different scales for the classifier requires multiple scans of the image. The classifiers are made of decision trees with at least two leaves. A decision tree is a graph model used for decision making algorithms. Each leaf in the tree is usually given a predetermined weight. These weights help determine the possibilities of choosing one route versus the other.

Figure 2.3 shows a general overview of the Viola and Jones Detection scheme and each step an image goes through. As mentioned in Section 2.3.2, an integral image is needed to insure fast computation of the Haar-like features. An integral image serves as an image representation for computing rectangular two-dimensional image features. The integrating function looks at all the pixel values that are above the current pixel and to its left. Scanning through the whole image, an integral image representation can be accomplished with few integer operations per pixel. Viola and Jones used AdaBoost to combine several weak classifiers into one strong classifier. The method states assigning weights to each of the weak classifier and an acceptance threshold that is usually set low. These classifiers act as filters for the image region under inspection as it goes through every one of them. The order of these filters is related to the type of detection application being used.

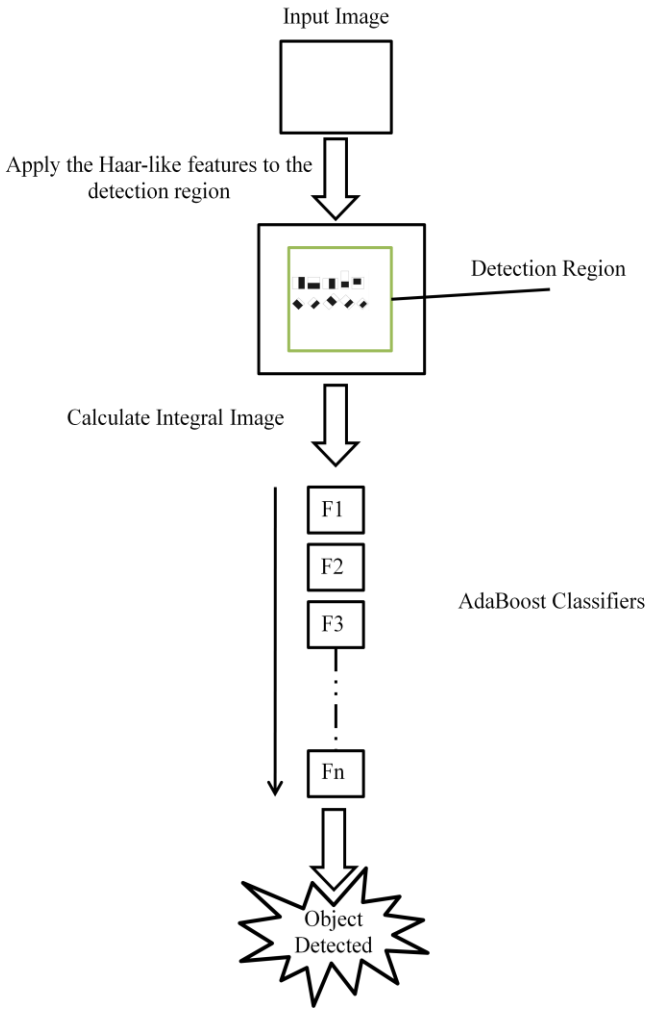


Figure 2.3 – Viola & Jones Object Detection Algorithm.

2.5 Haar Detector Training

The training process is essential for the classifiers used in the HFPD. A combination of training samples is used to formulate a cascade of classifiers to be used in the detection process. The complete process goes through four main stages: data preparation, object marking and creating object samples, training and testing. The detector was trained for detecting pedestrian legs in a given image. The choice for a leg detector was made due to the availability of detectors for the other body parts by the OpenCV library.

2.5.1 Data Preparation

To train the leg detector, a set of positive and negative samples is needed. The positive samples contain one or more human legs. The negative samples contain backgrounds with no human presence.

2.5.1.1 Negative Samples

The negative samples were obtained from the online dataset in [17]. The dataset includes 2977 negative samples of various grayscale backgrounds in the absence of any human or human like objects. Figure 2.4 shows a group of negative images from the given dataset.

Figure 2.4 – Negative Training Samples.

As shown in the images in Figure 2.4, a negative sample can be a picture of an empty classroom, a living room, or any outdoor scene with no trace of the object of interest. These images are used to train the detector to what is not an object for detection and to improve the overall detection rate. The wider the range of backgrounds used, the lower the false positive rates and the stronger the classifier. The file format of the negative images was JPG.

2.5.1.2 Positive Samples

The positive samples were taken in a lab environment with different backgrounds. Three detectors were trained using 890, 1890 and 2890 positive samples and thus three resulting cascades were produced. The choice of the number of positive samples was semi-random, such that the intention was to try various numbers of positive images and compare the results. One might think that increasing the number of positive samples would result in a stronger cascade of classifiers, but that was not the case. There are several factors that determine the strength of the cascade and these include, but not limited to: the type of object being detected, the backgrounds of the positive samples, and object rotation and scaling. Figure 2.5 shows a group of positive samples used in the training process. The leg samples were taken from different viewpoints and appear in different poses. The illumination was kept the same with minor differences. The positive samples were acquired using a high definition camcorder with a 1280x720 pixel resolution. The larger resolution for these images is not an issue as all the images are rescaled down during the training process. It is easier to take a video of the object of interest to create a large set of images to be used in training. Through software, the positive frames are extracted and saved to a directory. These positives will be used later to specify the precise location of the object of interest. The file format of the positive images is bitmap (.bmp). A list of the names of the positive samples are then stored in a text file to be used for creating the cascade of classifiers. Various poses of the legs are picked to strengthen the cascade to overcome the rotation drawback of Haar-like features. The images used in the training process are initially converted to grayscale, thus no color constraints are taken into consideration.

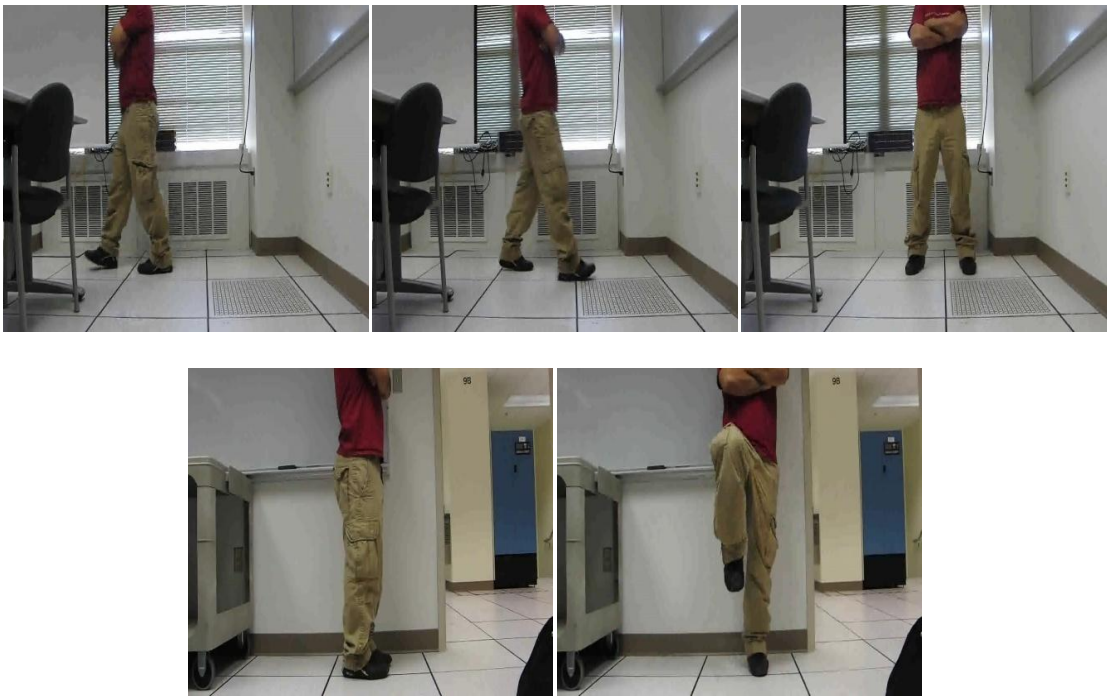


Figure 2.5 – Positive Training Samples.

2.5.2 Object Marking and Creating training Samples

The last step prior to starting to train the detector is to specify where the object of interest is located in a positive sample. This is done for all the positive samples, which is usually a large number, as mentioned in Section 2.5.1.2. To overcome such a time consuming mechanism, a program called objectmarker, [18], was used to manually mark the legs in a bounding box in every positive sample and save its coordinates in a text file. An example of the output of one marked image follows:

image1.bmp 1 547 127 286 589

The format is defined as follows: “image1.bmp” is the positive sample being worked on, the “1” following that is the number of times the object of interest appears in the image, and the number “547 127 286 589” represents the coordinates, or number of pixels, for the bounding box of that object from the upper left corner of the image.

Although it is acceptable to have multiple instances of the object in a positive sample, all the collected positive samples contain only one instance of the object for simplicity. The text file containing the rest of the image bounding box coordinates is then saved in the same folder location as the positive samples. Now, to continue the pre-training process, there is a need to create a vector file for the positive samples. This vector file is an output file that contains information regarding the generated samples. The OpenCV library provides a function called "opencv_createsample.exe" that is used to create the vector file [19]. The command used is as follows:

opencv_createsamples.exe -info positives/positives.txt

-vec data/positives.vec -num 2890 -w 20 -h 20

The "-info" specifies the location of the positive samples text files, the "-vec" determines where the output vector file will be stored, the "-num" is the number of positive samples used and the "-w" and "-h" are the width and height of the rescaled images which in this example are 20x20 pixels. The process takes a few minutes and creates a positives.vec file in the data folder where the cascade xml file will be saved.

2.5.3 Training Process

To this point, the following steps have been performed:

- Collected negative samples and saved them in a folder with a text file containing their names
- Collected positive samples and saved them in a folder with a text file containing their names and the coordinates of the marked object
- Created a positives vector file that contains information regarding the positive samples

The training process was performed on an i7, 2.8 GHz processor with 4 GB of memory. The training process time varies according to several factors, among these are: the number of training samples being used, the number of stages the cascade needs to cover, the memory allocation for the process, and the processor speed. On average, the training process took between 2 to 4 hours.

2.5.4 Testing

Three cascades were trained with 890, 1890 and 2890 positives, 2977 negative and 20 stages for each trained cascade. The number of trained samples was picked by random and has no specific implication. The input image to the detector is initially converted to grayscale for processing. The output image is shown in color in addition to the colored bounding circles that are drawn to show the detected object of interest for clarity.

2.5.4.1 Leg Detector Trained with 890 Positive Samples

A set of fifty test images was created to test each classifier; the set was taken from the INRIA online dataset for humans [20]. Figure 2.6 shows a leg detector trained using 890 positive samples and tested on a number of INRIA dataset images. Subjectively, it is shown that in some cases such as (a) and (b), the detection is achieved with no false positives. In other cases, such as in (c), the detector fails to find the legs and creates two false decisions. Image (d) of the figure shows both a correct detection and a false positive. The last image, (e), shows a no detection case where the detector fails to identify the pair of legs present in the image.

(a) (b) (c)

(d) (e)

Figure 2.6 – Haar-like legs detector trained with 890 samples, (a) & (b) complete detection, (c) no detection and false positive, (d) detection with false positive, and (e) no detection.

2.5.4.2 Leg Detector Trained with 1890 Positive Samples

Figure 2.7 shows a leg detector trained using 1890 positive samples. As shown in the figure, image (a) shows double leg detection with two false positives, (b) also shows two legs detected with one false positive. Image (c) shows no detection which is probably because of the modest training used for this detector. In image (d) & (e), perfect detection is achieved with zero false positives. This detector showed a 36% increase in the detection rate compared to the one trained with 890 positive samples. A 20% increase in the false positive rate was also recorded and is probably due to the different structure of the cascade.

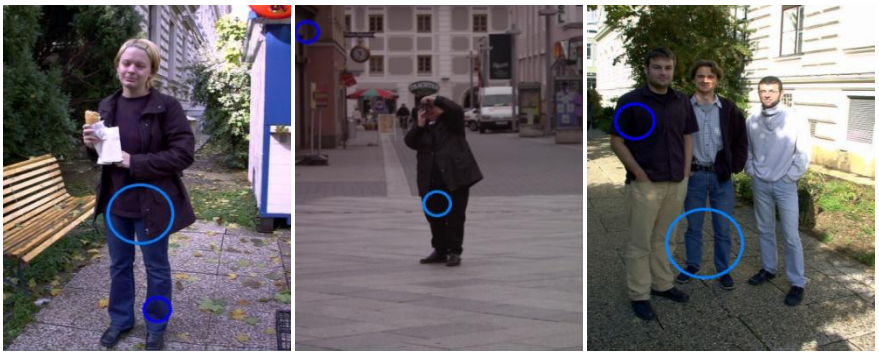
(a) (b) (c)

(d) (e)

Figure 2.7 – Haar-like legs detector trained with 1890 samples, (a) & (b) detection with some false positives, (c) no detection, and (d) & (e) complete detection.

2.5.4.3 Leg Detector Trained with 2890 Positive Samples

Figure 2.8 shows a leg detector trained using 2890 positive samples. Image (a) shows a perfect leg detection, image (b) shows complete detection of an image that the 1890 sample detector failed to capture. Image (c) and image (d) show detection with one false positive. In image (e), the legs of the man appearing in the image are detected while the children's were not.



(a) (b) (c)

(d) (e)

Figure 2.8 – Haar-like legs detector trained with 2890 samples, (a) complete detection (b), (c) & (d) detection with one false positive, and (e) detection with two missed.

This detector showed a 40% increase in the detection rate compared to the one trained with 890 positive samples. A 36% increase in the false positive rate was recorded and is probably also due to the structure of the overall cascade. The leg detectors trained using 890, 1890 and 2890 positive samples were used again in the testing process of the overall detection process.

2.6 Pedestrian Detection Results

Two experimental scenarios were used to test the Harr-like feature pedestrian detectors. The first scenario was an outdoor environment where pedestrians were crossing the street. In this experiment, a full body Haar-like feature detector is used. The second scenario was a lab environment and was aimed at detecting legs in an image.

2.6.1 Traffic Light Intersection

The main scenario that was used to show the accuracy of the Viola and Jones pedestrian detector was on a crosswalk located at an intersection in the city of Tulsa, Oklahoma. The Viola and Jones algorithm was tested using a video capture of pedestrians crossing the street. Network cameras installed by the city of Tulsa on the traffic lights provided a live video feed to a lab environment test bed. These cameras are normally used to detect vehicles in the turn lanes of the intersection to analyze the traffic flow and make light timing decisions. Future work to enable pedestrian detection at traffic lights can help regulate crosswalk light timers. As pedestrians are approaching the crosswalk or are actively crossing the street, the video can be used to regulate these pedestrians and the traffic more efficiently.

As pedestrians crossed the street, the full body pedestrian detector was used to find any humans in the captured frames. Red bounding boxes shown in Figures 2.9, 2.10 and 2.11 indicate positive detections. The eastbound traffic camera was directed toward the intersection where the pedestrians were crossing. The three figures are snapshots of the pedestrians at time instances 1, 10 and 20. As shown, not all the humans were detected due to the noise added by outdoor factors.

In Figure 2.9, six pedestrians can be fully seen whereas the detector only draws three bounding boxes. The first box to the left shows that the detector found one human but actually there are two present in that area of the image. Just to the right are three pedestrians with two of them clear and one almost fully occluded. The Haar-like detector is not immune to any type of occlusion and that is shown in this example. Thus, only two bounding boxes are drawn around that area.

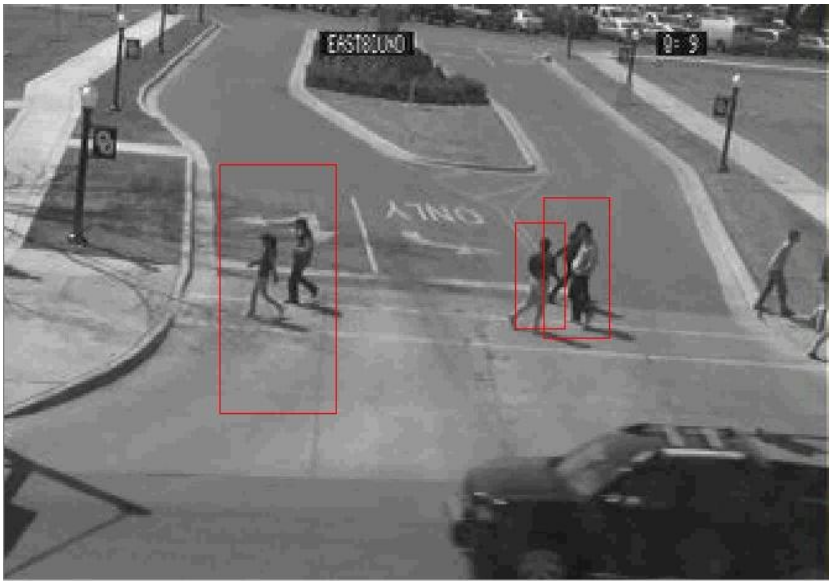


Figure 2.9 – Pedestrian traffic crossing the street at time instance 1.

The last pedestrian on the far right of the image is not detected at all which could be because of the pose and image noise. In this example, the detector accurately specifies three bounding boxes in which humans are assumed to be. There are no false positives recorded whereas the other objects present in the scene do not disturb the overall pedestrian detection process. Figure 2.10 shows where the pedestrians have moved to in the image at time instance 10. The detector again shows three bounding boxes but six pedestrians are present. At the right of the image there are two pedestrians crossing with one partially occluding the other. Also, their shadows add to the image noise in that region.

As a result, neither of the two were detected. The pedestrian crossing in the middle of the intersection is fully discovered by the detector and a bounding box is drawn around him. On the left side of the image there are three pedestrians, two of which are detected and the far left most one is not. This can also be attributed to the occlusion around the far most pedestrian's right leg. The bounding boxes can sometimes be misleading when it comes to scaling and can be improved to provide a better fit for the detected object. Again, no false positives were recorded for the captured frame.

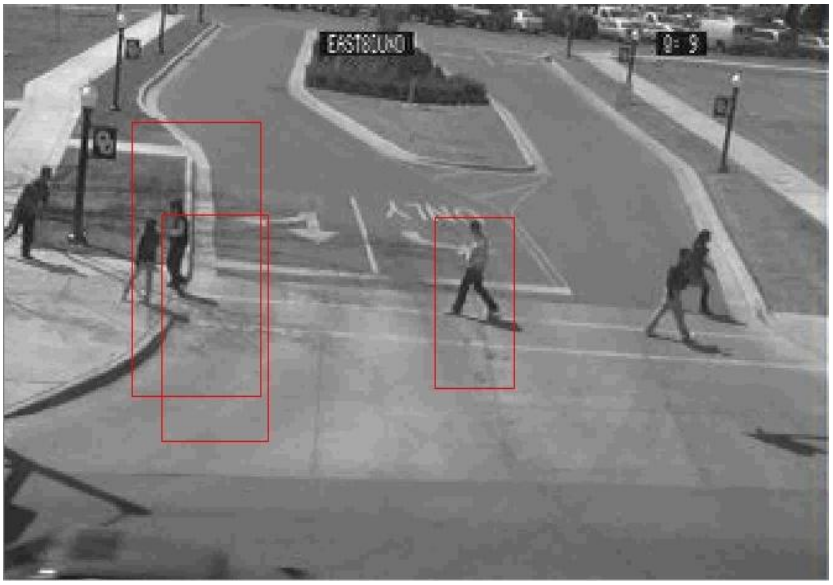


Figure 2.10 – Pedestrian traffic crossing the street at time instance 10.

Figure 2.11 shows a frame that was collected at time instance 20. Most of the pedestrians have crossed the intersection and are already present on the left side. Only one pedestrian is still walking toward the left side. The full body Haar-like detector found that person and drew a bounding box around him. To the left most of the image there are three other pedestrians that were not detected. This is because they are occluded by one another in addition to the noise and shadowing factors that appear in the image region.

Although it failed to detect some of the pedestrians, this detector performs moderately well and can be utilized in different ways as will be shown in the later chapters of this dissertation. In some cases, this type of detector is implemented for real-time applications where a video stream is fed as the input. The results obtained from this experiment are not too promising on their own but can be used to prove the validity of the overall smart surveillance system proposed. Note that the captured frames were already in grayscale and no conversion was required. Also note that in this scenario bounding boxes are used instead of circles to show the full body detection results.

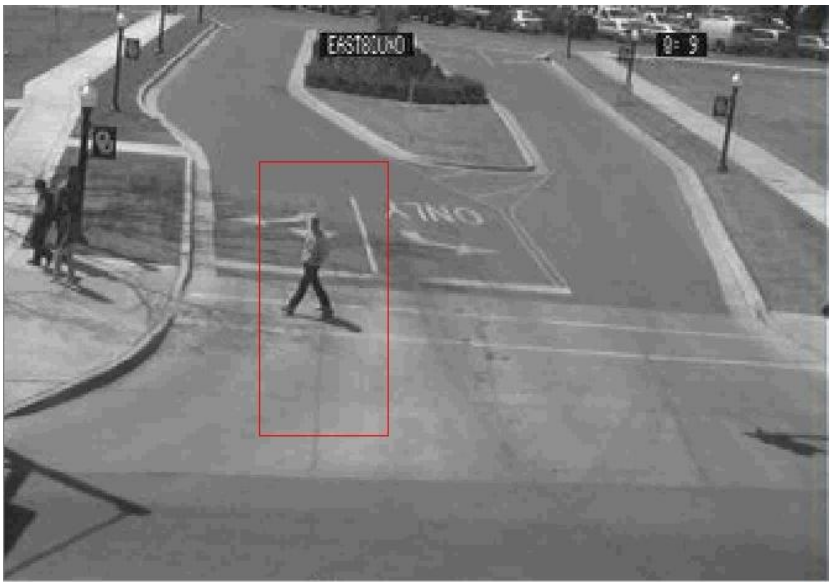


Figure 2.11 – Pedestrian traffic crossing the street at time instance 20.

2.6.2 Lab Environment

The second test bed for the Haar-like feature detector is within an indoor laboratory environment, Figure 2.12. The leg detector was applied to a video sequence and the results were very promising. The indoor scenario did not have as much noise as the outdoor scenario presented in the previous section, and thus, the detection rate was much higher.

The false positive rate was 9.5% whereas the detection rate was 93.8%. Two hundred and ten test images picked at random were collected from the video capture. The illumination factor was not an issue and had no effect on the detection process. Note that the scene background was not part of the captured positive images used in the training process. Varying the background in the training positive images helps improve the overall accuracy of the cascade of classifiers and decreases the false positive rate. The captured frames were converted to grayscale for the detector to process them and then the output image is shown as a colored image in addition to the resulting bounding circles.

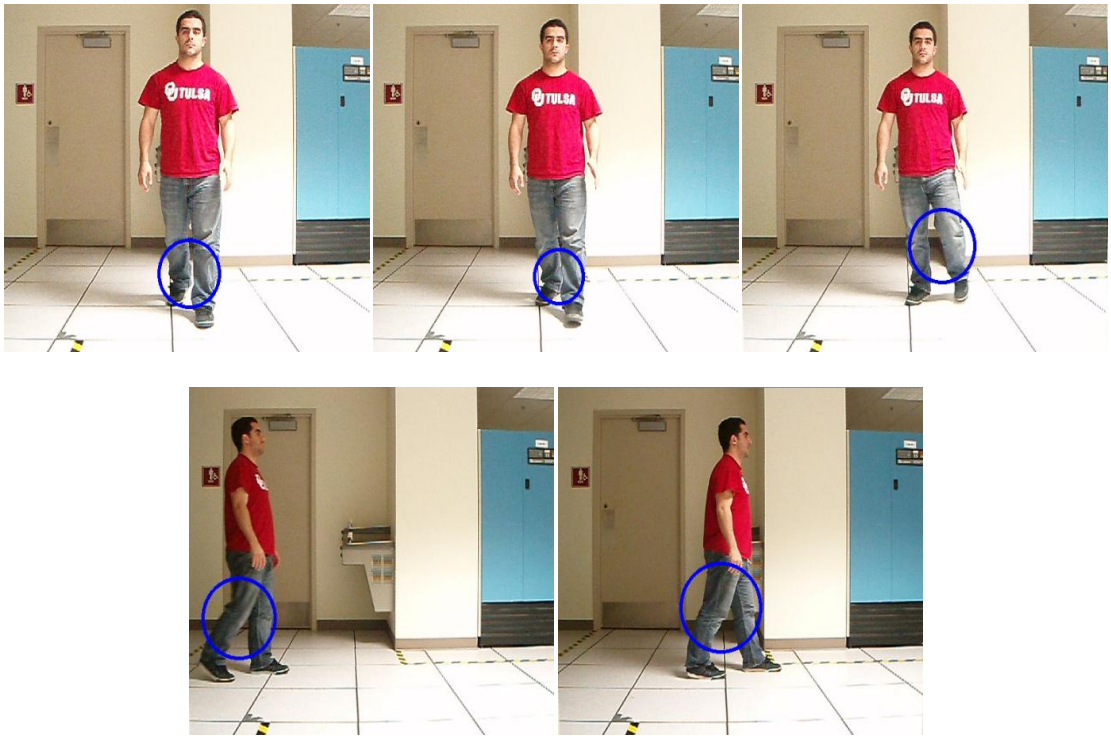


Figure 2.12 – Leg detector used in an Indoor Environment.

2.7 Summary

There are many different methods applied to perform pedestrian detection, a group of which leans to using Haar-like features. The HFPD is a fast detection method that can be used in real-time video surveillance applications. While it is efficient in that sense, HFPD struggles with accuracy and rotation. The training process of a HFPD can improve its accuracy and make it more adaptable with rotating objects of interest. Compared to indoor scenarios, outdoor scenarios introduce more noise and illumination variations that are setbacks for Haar-like feature detection. Increasing the number of training samples and varying the scene backgrounds resulted in a better object detector. The trained leg detector was improved to be more accurate in a lab scenario with a detection rate of 93.8%.

Haar-like features are used for detecting legs, upper body and full body in the overall detection system. The ability to detect objects of interest in real-time makes these detectors suitable for the proposed detection system. While the moving object tracker starts pointing toward the moving object, the HFPD starts the detection process to determine whether a pedestrian is present in the video capture or not. The use of the different filters and classifiers helps detect unauthorized personnel quicker and with a modest detection rate.

Chapter 3

Histogram of Oriented Gradients Adaptive Threshold Pedestrian Detector

3.1 Introduction

As discussed in the previous chapter, many image processing approaches have been proposed to achieve human detection. Other than using HAAR-like features to establish this type of detection, several researchers adopted the Histogram of Oriented Gradients (HOG) approach to more accurately find humans in a given image. The HOG detection approach was first introduced in 2005 and aimed at detecting objects based on their edge orientations [21]. The HOG approach can be compared to the Scale-Invariant Feature Transform (SIFT) approach proposed by David Lowe in 1999 [22]. These two approaches, the HOG and the Haar, share the same concept of extracting unique features to help in the decision-making process in determining whether an object of interest is present in an image. However, the HOG method segments the image in a different way and makes use of local contrast normalization to improve the overall performance of the system. As the name indicates, the HOG methodology is highly dependent on edge segmentation and detection. A number of histograms of oriented gradients are calculated to determine the location of the object of interest.

The remainder of the chapter is organized as follows: Section 3.2 presents some of the currently proposed research efforts for using HOG, Section 3.3 is an overview of the HOG algorithm, and Section 3.4 explains the proposed HOG detector. Section 3.5 shows the experimental results and Section 3.6 provides a summary.

3.2 Background

Currently, HOG is being used in multiple object detection applications resulting in fast and accurate detection [23-25]. The most common detection application for HOG is human detection. In [26], Bertozzi et al proposed using the HOG approach for achieving pedestrian detection and a Support Vector Machine (SVM) for classification. The authors used two types of camera sensors: visible and infrared. The overall system is decomposed into three main stages: detection, symmetry, and filters. The HOG and SVM are used in the filtering stage to finalize the detection process. Different cell and block sizes are tested to evaluate an optimal solution for the detection system. Experimental results show pedestrian detection rates of 91% in nocturnal scenes and 81% in daylight scenes.

Another application using HOG for object detection was proposed by Hu et al to perform motion detection under various illumination changes [27]. The detection process is defined by two stages: coarse detection and refinement. The first stage builds a background model of the image using groups of adaptive HOGs. Comparing the features extracted in the current frame and the next frame results in detecting the foreground objects. The refinement stage helps cancel all errors created by shadowing, redundancy, noise and other factors. Experimental results show robustness against illumination variations during moving object detection. The authors compared their results to both the Gaussian Mixture Model method and the codebook method.

In [28], Toya et al used HOG with stereo vision to recognize pedestrians in real-time video. This approach makes use of a stereo camera and laser radar. The radar can detect object presence by emitting a laser beam in a given 2D plane. After sensing an object, the detection system is initiated to determine whether a pedestrian is present or not. HOG and principal component analysis are applied to the foreground to detect pedestrians.

Using the HOG descriptor with other image processing techniques has led to better overall object detection systems. An example of this is the work of Lee et al in [29], which utilized HOG feature detection and human body ratio estimation. The system was tested on three different outdoor scenarios with a frame resolution of 420x360 pixels. The authors tested different threshold values and numbers of samples. Increasing the threshold value increased the detection rate and decreased the false detection rate. The highest achieved detection rate was 95% performed on 240 testing samples from one of the four datasets.

Other than being used in pedestrian detection systems, HOG features can also be utilized in different object detection systems. In [30], Rybski et al proposed using HOG features for vehicular orientation classification. The authors studied two sets of classifiers: the orientation-specific and the orientation-independent. The tested algorithm yielded a vehicular orientation classification rate of 88%. Several classifiers were used to determine orientation of each vehicle present in the picture. Such a process requires a lot of computation and could be made faster by using pyramid scales.

3.3 Overview of the HOG Algorithm

Initially, HOG was designed to detect pedestrians in static images and then was upgraded to perform human detection in video. The detector used converts still input frames to grayscale prior to the detection stage. The average detection time for the adopted approach is longer than needed for real-time assessment. Currently, HOG can be used in real-time applications primarily due to the fast parallel computation provided by General Purpose Graphical Processing Units (GPGPUs) [31].

3.3.1 Gradient Computation

The first step in the HOG algorithm is gradient computation. The simplest and most efficient way to accomplish that, as tested by Dalal and Triggs, is to apply a 1D, centered point, discrete derivative mask. Applying other types of masks such as the 3x3 Sobel mask does not lead to better overall system performance [21]. A derivative mask system is defined as follows:

$$\begin{cases} Y_v(i,j) = X(i,j+1) - X(i,j-1) \\ Y_h(i,j) = X(i+1,j) - X(i-1,j) \end{cases} \qquad \text{Eq. 3.1}$$

The equation system contains vertical and horizontal 1D derivative masks that can be applied pixel wise to an input image X. Y_v and Y_h are the resulting output image with the calculated pixel derivatives for rows i and columns j. The whole image is scanned to calculate each pixel orientation to be used in computing the later histograms. The derivative masks used can be expressed as:

$$[-1, 0, 1] \qquad \text{or} \qquad \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

3.3.2 Cell Histograms

After calculating the gradients, the algorithm defines a detection window of fixed size (64x128 pixels) to scan the image. The detection window is then divided into a number of 8x8 pixel groups called cells, Figure 3.1. A cell can be rectangular or circular in shape and can vary in size, although a 6x6 pixel group is considered an optimal solution for human detection. For the purpose of this study, the selected cells are rectangular. The next step in this system finds a 9-bin histogram of pixel orientations for each cell. The number of orientation bins selected suggests looking at 20 degree bins, 180/9, for each pixel orientation. The range from 0-180 degrees, for unsigned gradients, is divided by the 9-bin orientation in which linear gradient voting is represented. A weighted vote for each pixel is calculated based on the direction of the gradient element at its center. According to [32], a fast way to calculate histograms of regions of interest can be achieved using integral histograms.

3.3.3 Normalized Blocks

In order to pass the computed histograms of gradients into a classifier, cells are organized into a 3x3 arrangement called a block. Creating these blocks helps the algorithm become more immune to changes in illumination and contrast. The blocks overlap in the image producing more correlated spatial information to be used in the descriptor. It also improves the overall detection performance. Figure 3.1 shows an example of blocks containing 9 cells inside the detection window.

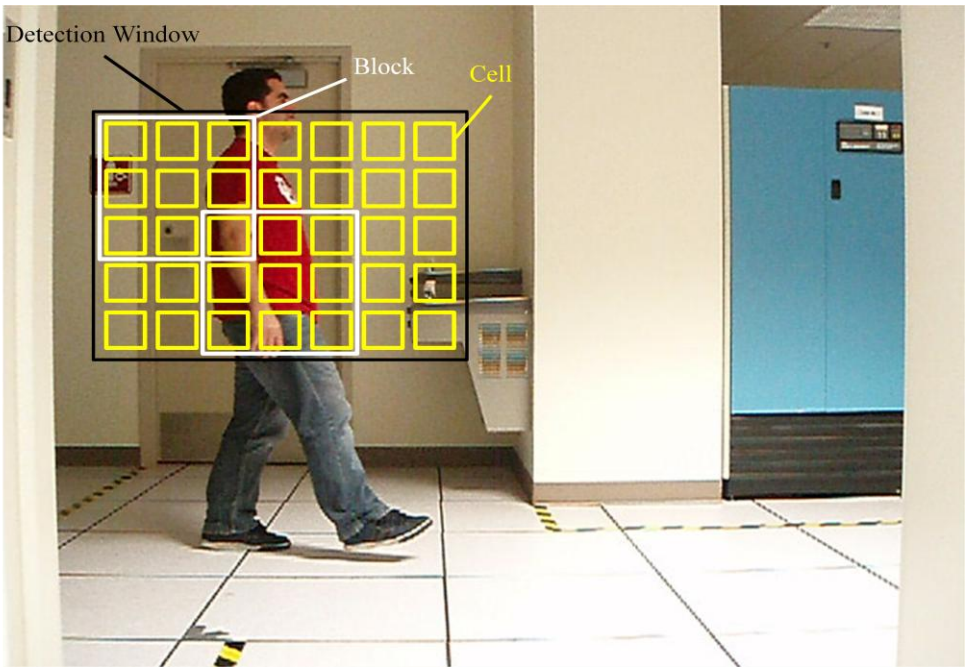


Figure 3.1 – HOG detection window with cells and blocks.

The 3x3 and 6x6 blocks worked best for Dalal and Triggs in their experimental results and they believe that varying the block size does not have as much effect on the detection as overlapping the blocks [21]. In some cases, increasing the number of cells present in the block decreases the overall performance of the detection system. Two arrangements of blocks were first proposed by Dalal and Triggs: rectangular and circular. The rectangular HOG, also known as R-HOG, can be set with different block dimensions but are best used in square arrangements. The R-HOG is adopted in the tested HOG human detector presented in this chapter. A block is represented by a multi-dimensional feature vector that is used in the classification step. Block normalization is needed to decrease the required computation, thus L-2 normalization on the block is performed, followed by a renormalization step. Each block is normalized and used in the collected feature vector. In [24], 2x2 cells were used to form a block and thus resulted in a 36 dimensional normalized feature vector, since four 9-bin histograms were used for the HOG detector.

3.3.4 Support Vector Machine Classification

The final step for the HOG algorithm is to use the feature vector as input to a Support Vector Machine (SVM) classifier to perform the decision making. The linear SVM is one of the most common, simple methods used for forming different classes of a dataset. Figure 3.2 shows an example of a linear SVM classifier that splits the circular green dataset from the square blue dataset. The hyperplane represents the border in the middle of the graph that separates the two classes. The margin is the maximum separation between the hyperplane and the classes' data points.

A misclassified data point is one that appears in a class different than its original class or the class it belongs to. The HOG algorithm feeds the descriptor vector to a trained linear SVM to determine human presence in a given test image. SVM has been used by many researchers in object detection and segmentation to deliver a classification method for various objects in input images [33-35].

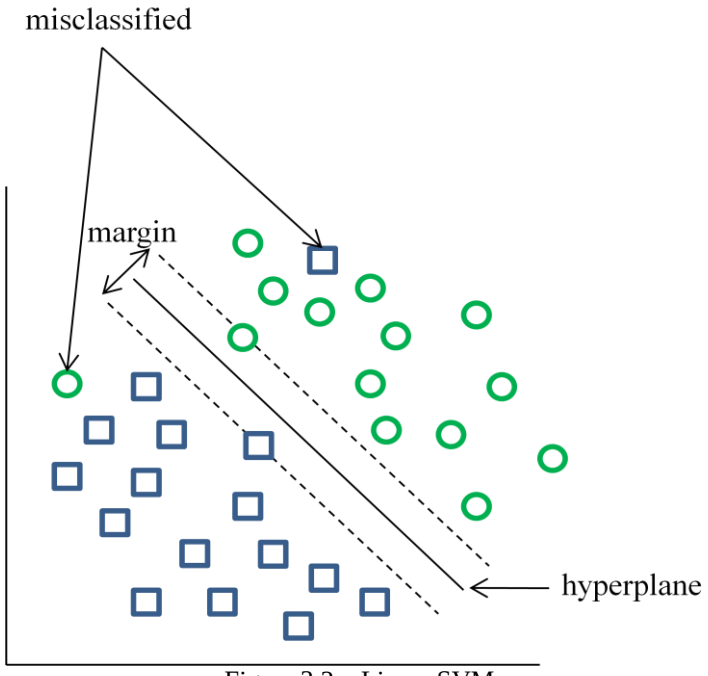


Figure 3.2 – Linear SVM.

3.4 Proposed HOG Implementation

A flow diagram of the HOG method is shown in Figure 3.3. The input image undergoes four major steps before reaching a verdict on whether a human is present in it or not. While this approach is claimed computationally heavy and requires a long time to converge on a decision, it is still suitable for our detection system , which makes use of its high accuracy and low false positive rate.

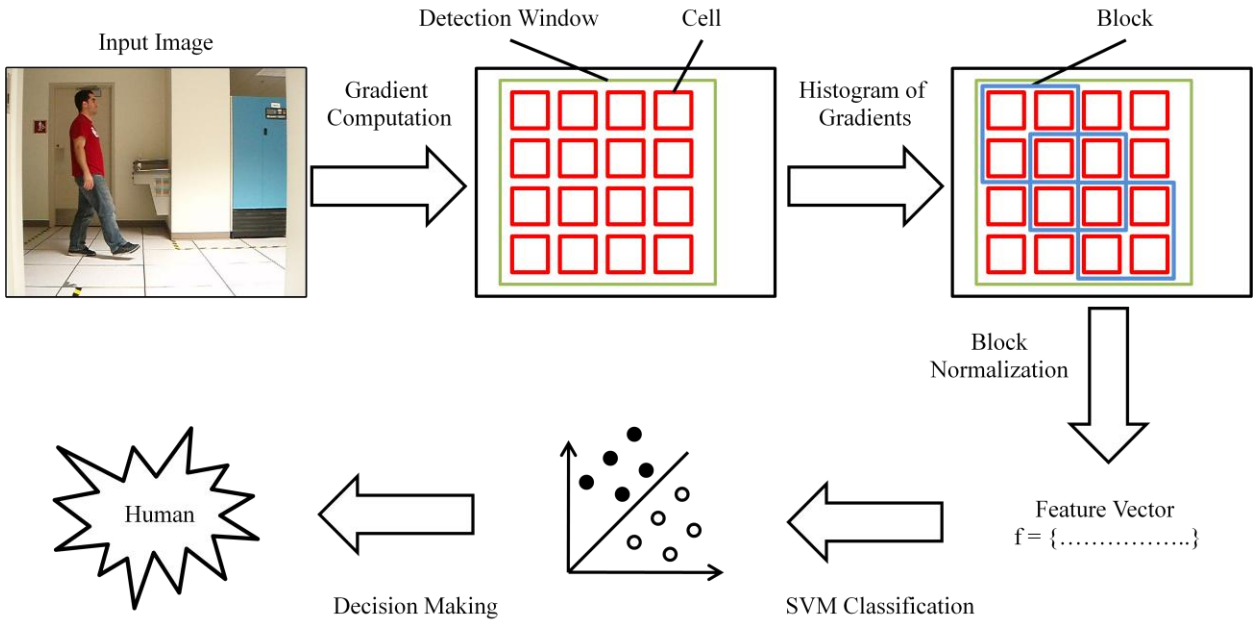
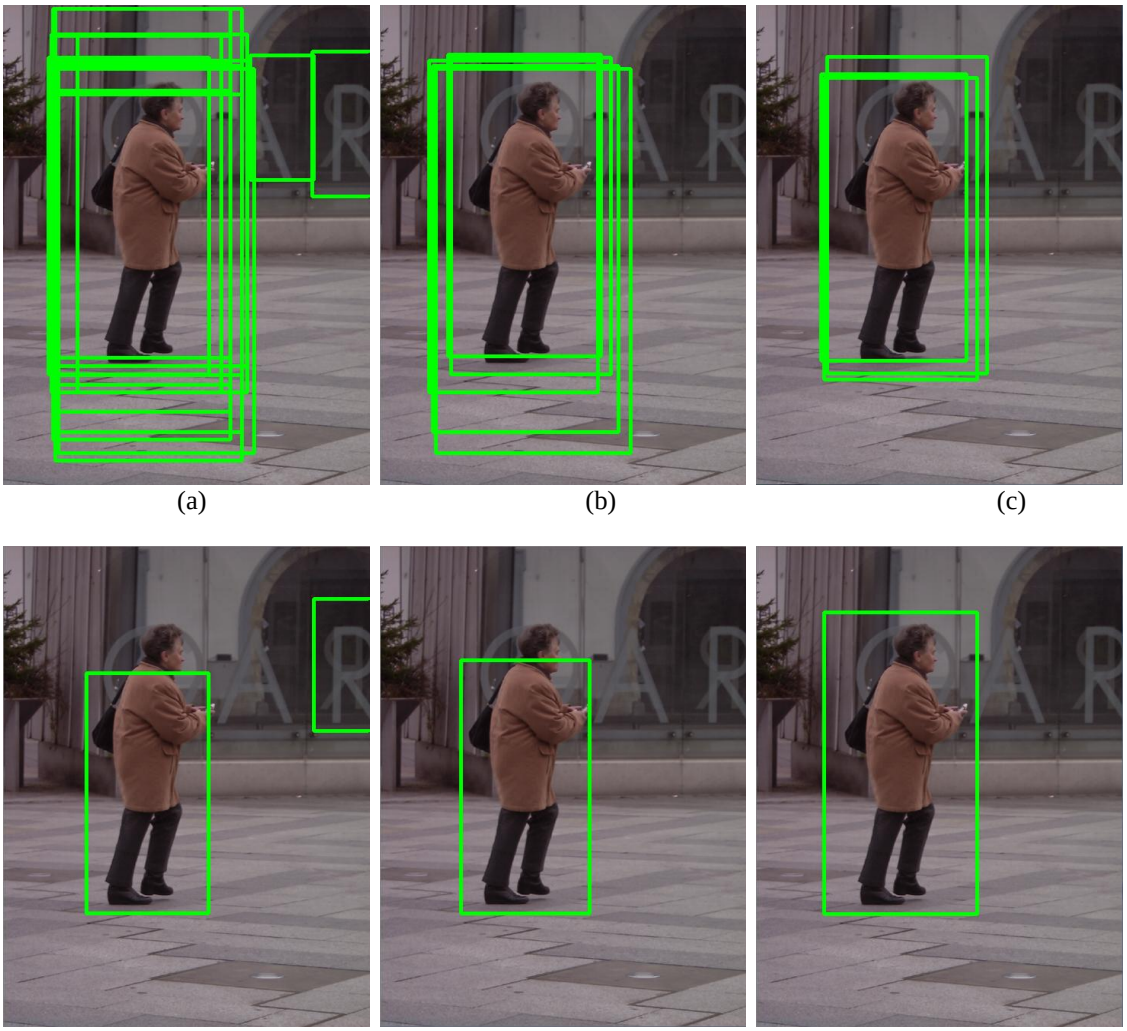


Figure 3.3 – Static Histogram of Oriented Gradients Approach.

According to [21], the proposed HOG scheme was tested and performed extremely well on two datasets: first on the MIT pedestrian database [36] and then on a new dataset created by Dalal and Triggs called the INRIA dataset. In this study, some of the HOG parameters are adjusted to adapt hit and group thresholds to fit well with the overall pedestrian detection system.

3.4.1 Adaptive Threshold

The hit and group thresholds are two parameters used in the HOG algorithm with a multi-scale detection function. Varying these two thresholds, result in better performance for the proposed overall human detection system. Figure 3.4 shows two input images used in a HOG detector with a hit threshold (ht) varying between 0, 1 and 2.



(a) (b) (c)

(d) (e) (f)

Figure 3.4 – Varying the hit threshold value in HOG human detection: (a) ht = 0 and gt = 0, (b) ht = 1 and gt = 0, (c) ht = 2 and gt = 0, (d) ht = 0 and gt = 1, (e) ht = 1 and gt = 1, (f) ht = 2 and gt = 1.

Notice that increasing the hit threshold value results in more deterministic results while decreasing it increases the false positives. Although a low threshold value produces more false positives, it increases the chances of finding humans in some situations, as shown in Figure 3.5. The test images used in Figures 3.4 and 3.5 are from the INRIA dataset [37].

(a) (b) (c)

(d) (e) (f)

Figure 3.5 – Varying the hit threshold value in HOG human detection: (a) ht = 0 and gt = 0, (b) ht = 1 and gt = 0, (c) ht = 2 and gt = 0, (d) ht = 0 and gt = 1, (e) ht = 1 and gt = 1, (f) ht = 2 and gt = 1.

3.4.2 Detection Optimization

Choosing different hit and group thresholds add more variability to the overall detection system and creates a need for optimizing these parameters. While in some cases, decreasing these thresholds can increase the detection rate, it can also increase the false positive rate. Increasing the thresholds is not an optimal solution either as some humans might be missed, increasing the non detection rate. As discussed later in the results, trying different thresholds adds more reliability to the detector.

The hit and group threshold values were picked after several detection tests using various images from online datasets and personal video recordings. When the group threshold value is set to 0, grouping is turned off and more bounding boxes of objects assumed to be humans are returned. The higher the group threshold, the lower the hit rate, resulting in fewer bounding boxes being returned, as shown in Figure 3.4 and 3.5.

Also from the figures, it is notable that increasing the hit threshold value has an effect similar to increasing the group threshold. In both images, when the two thresholds were set to 0, the person was detected more than once as well as other objects present in the picture that are not human. As both thresholds are increased, fewer bounding boxes are present and the false positive bounding box disappears.

3.5 Experimental Results

In the HOG experiments, three rates used in human detection techniques were observed: detection rate, false positive rate, and false negative rate. A detection rate is the rate of the detector finding a human present in the image. A false positive rate is the rate of the detector finding something that it thinks is a human but it is not. A false negative rate is the rate of the detector missing a human present in the image. These rates are determined subjectively and through a predetermined number of test images. The experimental results shown in this section are for three main scenarios: Indoor Scenario with a single pedestrian, Outdoor Scenario with two pedestrians, and Outdoor Scenario with three or more pedestrians. The videos were captured at the University of Oklahoma – Tulsa campus. The video background for the different scenarios was held stable to overcome any background noise that might affect the detection rate. Outdoor scenarios are considered to have a static camera sensor capturing the surveillance video. With static cameras, the background is assumed to not change during the sensing process. This facilitates using a simple object tracker which will find foreground objects in an image and track them through their motion paths. In the first scenario, a 640x480 video of a single pedestrian was captured. As shown in Figure 3.6, nine frames are tested under different illumination changes. The HOG detector, with a hit threshold of 0 and group threshold of 1, performed extremely well for the video captured in the lab environment. 475 frames were extracted from the video and 9 were picked at random to show the detection result.

Figure 3.6 – Pedestrian detector applied on frames: 4, 23, 32, 58, 160, 240, 320, 410 and 460 of a lab video capture.

Note that on the middle right image (frame 52), the detector identifies two humans where only one is present. Although the output is misleading on the number of humans present in the image, it is still pointing at the same human. The detector was able to find the human in all the test images. In some images, the detector drew two or more bounding boxes on the same human.

In the second scenario, a video of two pedestrians walking in an outdoor scene is captured and all the frames are tested. The same camera sensor was used for two different video captures in this scenario with two different resolutions: 640x480 and 320x240 pixels.

This was done to compare the detector's speed and accuracy when applied to different image resolutions. Intuitively, the larger the image size, the higher the computation load, due to the fact that the detection window slides over the whole image to compute the cell histograms. This results in a larger feature vector and makes the detector slower in detection time.

The pedestrian detector performed extremely well, as shown in Table 3.1. On the 320x240 video, the detection rate was 91.1% which includes detecting both pedestrians in each frame. As the two pedestrians moved farther away from the sensor, the miss rate increased leading to much lower detection rates. The false positive rate was kept considerably low, 0.90%, this was due to the detection region being smaller in scale. The average detection time at this resolution is 160 ms which is desirable for real-time detection.

Resolution	Total Number of Frames	Detection Rate	False Positive Rate	Average Detection Time
640x480	300	97%	5.3%	790 ms
320x240	300	91.10%	0.90%	160 ms

Table 3.1 – Detector performance on 640x480 and 320x240 resolutions.

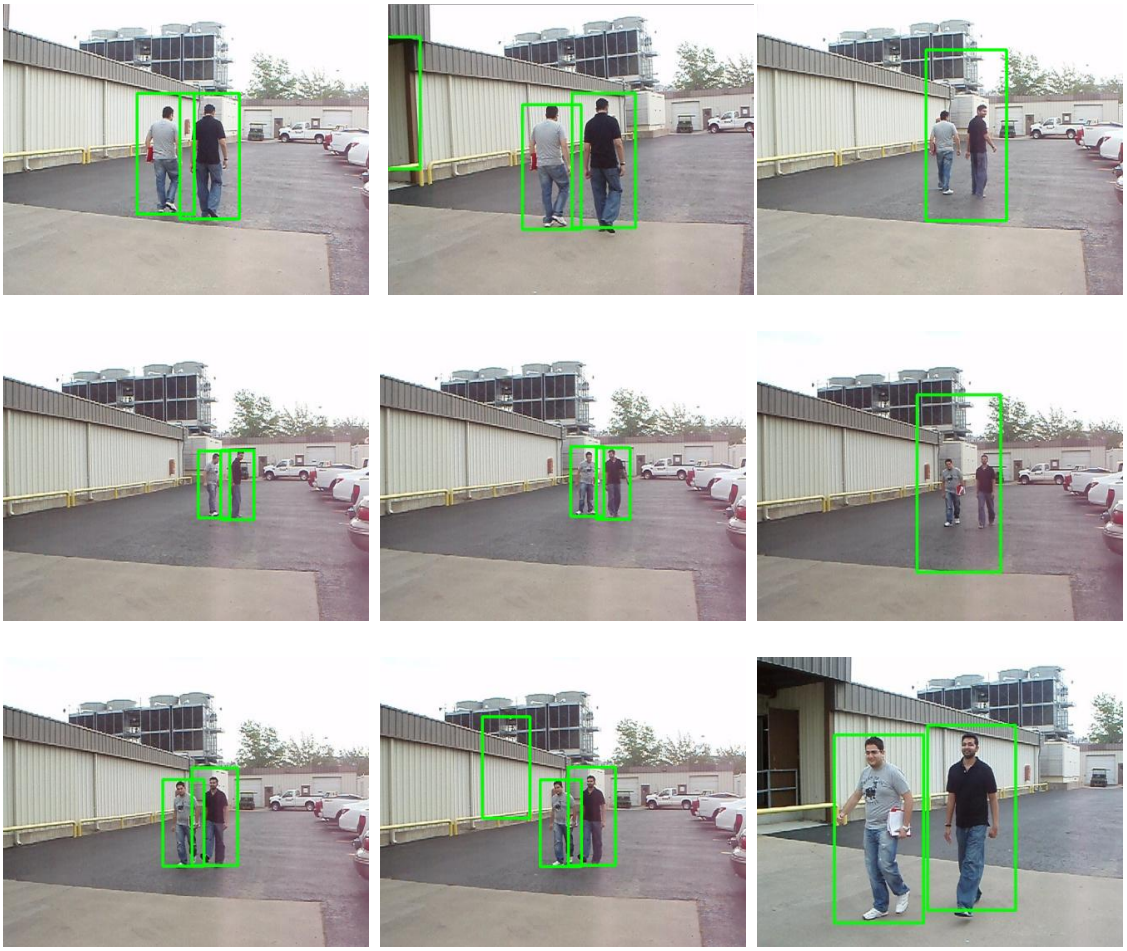


Figure 3.7 – Pedestrian detection of two humans in an outdoor scenario.

For the 640x480 resolution, the detection rate increased to 97% although the background scene is the same as the one found in the smaller resolution video. The false positive rate was approximately five times higher than the first case due to the increased number of detection windows used. The second and eighth images in Figure 3.7 show the two false positives. Also the third and sixth images show one bounding box drawn around both pedestrians rather than two bounding boxes. With this higher resolution, the average detection time increased to 790 ms, which is not desirable when trying to achieve real-time detection.

The last scenario studied for the HOG pedestrian detector was outdoors and has four pedestrians walking around the sensing area. As shown in Figure 3.8, the four pedestrian are walking randomly and are being detected in most of the pictures. In some cases, such as image 3, 4 and 6, a pedestrian is occluded by another object or pedestrian in the captured frame. The detector seemed to have various results according to how much the pedestrian is occluded. Another interesting factor in the detection process is the ability to detect humans as they start appearing in the surveillance region. The first image in Figure 3.8 is a perfect example of this and shows a human entering the scene from the right side and being surrounded by a bounding box.

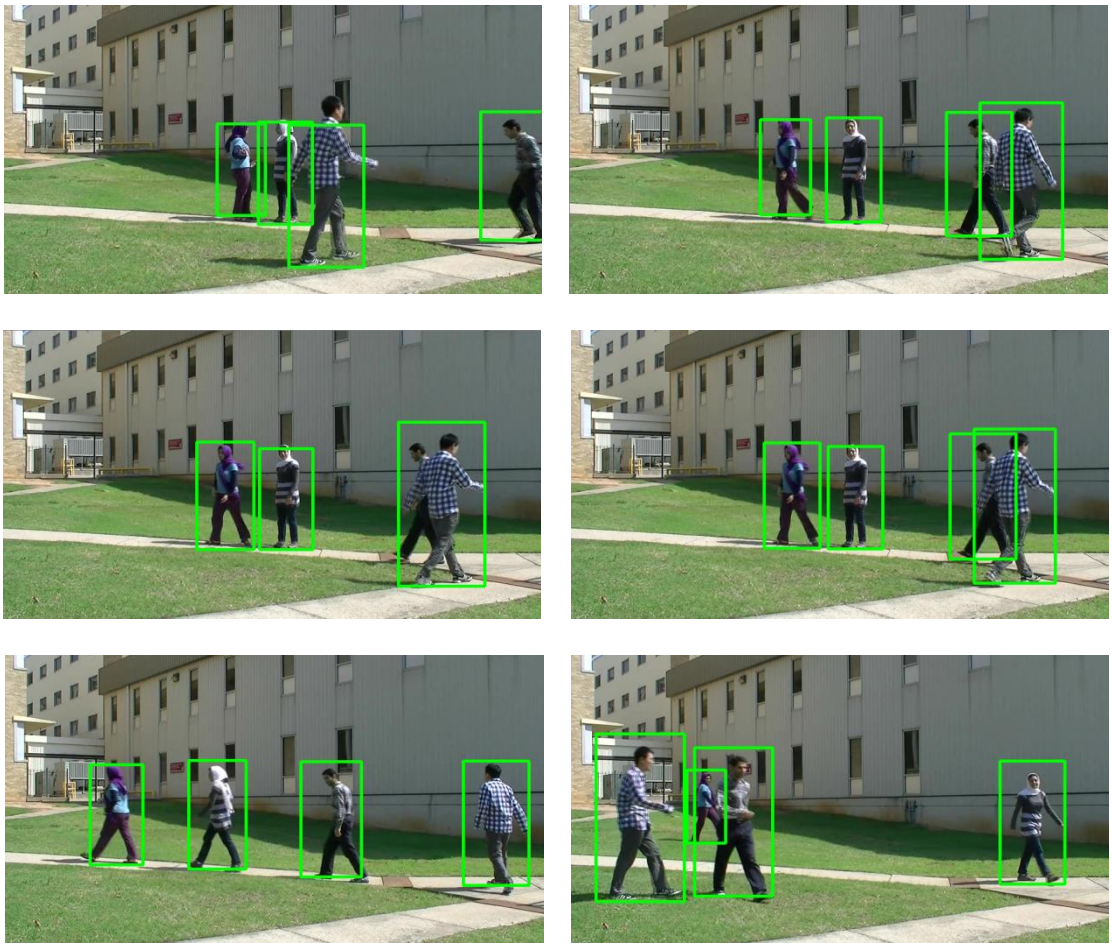


Figure 3.8 - Pedestrian detection of three or more humans in an outdoor scenario.

The HOG pedestrian detector performed extremely well in the last outdoor scenario where more than two pedestrians were present throughout the video. As shown in Table 3.2, 300 frames were tested and a detection rate of 93.4% was achieved. This rate is much higher than that of the two previous scenarios which is partially due to the higher resolution used in the sensing stage. Increasing the number of pedestrians located in the image does not seem to affect the detector's accuracy.

Resolution	Total Number of Frames	Detection Rate	False Positive Rate	Average Detection Time
848x480	300	93.4%	1.1%	1010 ms

Table 3.2 – Detector performance on a 848x480 resolution.

The false positive rate is only 1.1% which is an acceptable percentage that can be overlooked for the purpose of this study. One last thing to note in Table 3.2 is the average detection time which is approximately 200 ms higher than the detector in the second scenario of the results. The detection time is dependent on the size of the input image this is due to the increased number of visited detection windows required by the algorithm. Down sampling the images can help decrease the detection time to fit in a model for real-time or near real-time pedestrian detection. Investigation into GPGPUs may also reduce the time required to process an image.

3.6 Summary

In this chapter, the Histogram of Oriented Gradients was covered including methods suggested by other researchers using HOG. The proposed HOG was evaluated and resulted in high accuracy when used in indoor and outdoor pedestrian detection scenarios. Three main scenarios were tested for the detection method and low false positive rates were obtained. The detector was able to find humans in images of various resolutions and backgrounds.

Chapter 4

Object Tracking in Static Backgrounds

4.1 Introduction

Object tracking is the process of constantly locating moving objects in a given video stream. Object tracking is used in many applications in the field of image processing and is considered an important task for computer vision. Examples of these applications are: video surveillance, augmented reality, medical imaging, and scene analysis. In [38], Yilmaz et al. completed a survey on object tracking, in which they defined the major methods used for tracking. The three main classes of object tracking presented were: point tracking, Kernel tracking, and silhouette tracking. The tracker used in this research is considered a kernel tracker. Several challenges face object tracking that could alter the tracking process and result in faulty decisions. Some of these challenges are: noise, partial and full occlusion, illumination changes, and complex object motion.

The proposed object tracker for this research is robust to all these challenges except for full occlusion. For the purpose of this study, the tracker is not required to be susceptible to full body occlusion. The objective of the tracker is to locate and tag moving objects in each frame of the captured video. The two detectors, discussed in the previous chapters, are used to find the human in the area provided by the tracker where a moving object is located. The background of the captured video is assumed to be static and thus background motion can be disregarded. This simplifies the tracking process and decreases computational requirements.

The organization of the chapter is as follows: a look at the current and previous object tracking techniques are discussed in Section 4.2. Section 4.3 shows the three main categories of object tracking: point tracking, Kernel tracking, and silhouette tracking and classifies the type of tracking used in the research. The proposed tracking method is presented in Section 4.4 in addition to a discussion about Kalman filters and the mean-shift algorithm. Experimental results are given in Section 4.5 and a brief summary in Section 4.6.

4.2 Background

Counting moving objects, for example pedestrians, in an outdoor scenario can become a complex problem, especially with phenomena such as occlusion. In [39], Rabaud and Belongie suggested using a highly parallelized version of the Kanade-Lucas-Tomasi (KLT) tracker to define a set of feature trajectories to extract motion information. Doing so, the authors were faced with other challenges such as the unequal lengths and fragmented nature of the detected objects. The KLT tracker defines windows of operation to perform detection of the objects of interest. To establish a more efficient tracker for crowded scenes, the authors did some training for the classifier to have a better estimate of the detected pedestrian. The detected objects are presented in clusters at a given time t. The results collected were done on several datasets and achieved an error percentage of less than 23%.

In [40], Oren et al. proposed an architecture that is based on wavelet templates to perform pedestrian tracking. Sets of positive and negative images are used to perform training on the desired pedestrian detector. Positive images are sample images that contain a pedestrian, whereas negative images are those with no pedestrian. After creating the wavelet template, a matching process occurs resulting in detection of pedestrians with a calculated false positive rate. The authors also tried using a support vector approach to compare to simple template matching. The support vector produced a more sophisticated classifier that had better detection results. Both classifiers were tested for the set of low and high quality sample images. Wavelet templates resulted in approximately 53% detection rates for low quality images and 62% for high quality images. The support vector method achieved approximately 70% for low quality images and 82% for high quality images.

Infrared camera sensors can be used as sensing devices for object tracking. In [41], Nguyen and Havlicek computed low-level target and background features in the modulation domain. This was introduced as a new approach that is different from the traditional computation techniques which depends on the pixel domain and Fourier domain features. The proposed technique was used to separate the target from the background in the given images. The authors presented a proof of concept approach which qualitatively supported their claims of achieving object tracking in infrared images in the modulation domain. In [42], Nguyen et al. proposed an object tracking algorithm based on template tracking in the modulation domain. The authors compared template tracking in both the pixel and modulation domain. They also proposed a joint solution that uses both domains to perform object tracking.

The object of interest used in their experiments was the human face. The tracker operates in the pixel domain where it detects the face and then updates its results in the next video frame. The tracker then approximates the modulation domain correlation function. The experimental results showed better performance of the modulation domain tracker with an updated template compared to the pixel domain tracker with an updated template.

In [43], Mould et al. considered using a particle filter based tracker and improved its performance by incorporating modulation domain checks. The dual domain (pixel and modulation domains) tracker was tested on two long wave infrared Aviation and Missile Command (AMCOM) closure sequences. The dual-domain tracker showed better tracking capabilities than the pixel domain tracker when used in a video sequence for tracking a moving vehicle.

In the video sequence, a set of vehicles were moving along a certain path. The two trackers were tracking the vehicle in the lead; as the vehicles turn, the dual-domain tracker continues tracking the desired vehicle, whereas the pixel domain tracker starts tracking a different vehicle. The paper focuses on proving that with Amplitude Modulation-Frequency Modulation (AM-FM) consistency checks the performance of elementary track processors can be improved.

4.3 Object Tracking Categories

According to [38], object tracking can be categorized into three approaches: point, Kernel, and silhouette tracking. Objects are represented differently in each of the three tracking category. For example, point tracking represents object targets using points corresponding to the object in every frame. Moreover, Kernel tracking use geometric shapes, such as rectangles and ellipsoids, to represent the object of interest. Finally, as the name implies, silhouette tracking draws matching silhouettes of the tracked object in captured frames.

4.3.1 Point Tracking

Point tracking is based on representing objects in consecutive frames using points and associating these points with the previous ones, thus collecting position and motion information. This point correspondence for tracking objects is not an easy task to accomplish and can be done by either deterministic or statistical methods. Object detection for point tacking is performed as a separate task. This is similar to the approach taken in this research, although the tracking method is categorized as Kernel tracking. Figure 4.1 shows an example of point object tracking.

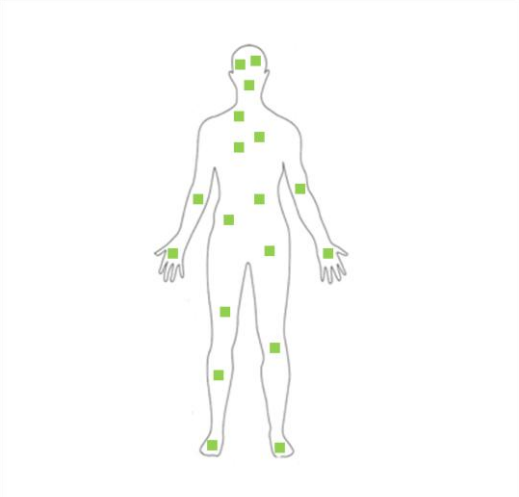


Figure 4.1 – Interest points marked in green using point tracking.

4.3.1.1 Deterministic Methods for Point Correspondence

The deterministic method uses motion constraints to assign costs of associating objects in the current and previous frame. The method introduces an optimization problem to find the minimum cost required. The solution of the problem presents a set of points in which each point is associated with one point in the previous frame.

One way to solve this problem is using greedy search methods as suggested by Shafique and Shah [44]. The authors proposed a non-iterative greedy algorithm to solve the optimization problem for real-time deterministic point tracking systems.

4.3.1.2 Statistical Methods for Point Correspondence

One of the challenges that object tracking faces is sensing noise which is a varying parameter that occurs during the video capture. To overcome this drawback, statistical methods are used to filter some of this noise out and enhance the obtained measurements. These methods also help predict the objects' states while the objects are maneuvering randomly and inconsistently around a given area.

Object properties, such as position and velocity, are measured to determine the state variation over time. Some examples of statistical correspondence filters are: Kalman filters, particle filters, and Joint Probability Data Association Filters. For this research, Kalman filters are used to determine the object position and size during the object trajectory post processing phase.

4.3.2 Kernel Tracking

Kernel tracking is the process of locating moving objects in every frame using primitive bounding regions, such as rectangles and ellipsoids to show the object in motion. Unlike point tracking, kernel tracking methods use template/histogram matching and multi-view appearance models. These models can track multiple objects independently (i.e., each object is tracked with no relation to the background or other objects) or jointly (i.e., objects are tracked even if occluded by other objects or by the background). Template tracking methods rely on online learning of the tracked object as it moves from one frame to another. Multi-view tracking methods, such as the Support Vector Machine (SVM), undergoes training with an offline dataset of positive and negative image samples of multiple views of the tracked object. Offline training negates the need to gather object information from recent observations. Figure 4.2 shows a moving object being tracked and assigned a tracking ID during a video capture.

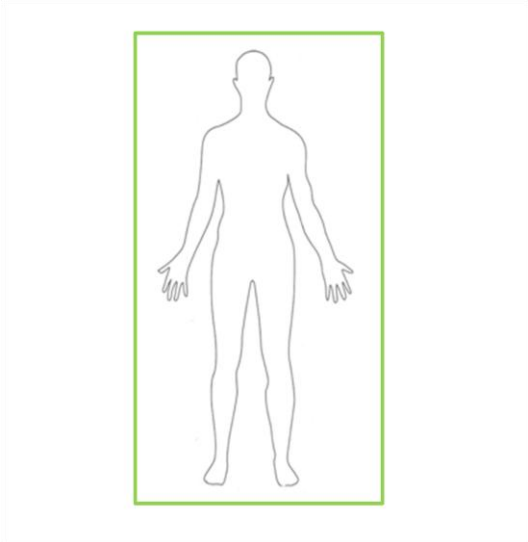


Figure 4.2 – Kernel-based tracking representation for a human.

4.3.3 Silhouette Tracking

Silhouette tracking is used when the object of interest is more complex in structure and requires a more detailed representation. Examples of such objects are humans. Shape matching and contour tracking are the two major methods used in silhouette tracking. In shape matching, a hypothesized model from previous frames is matched with the object of interest present in the given frame. Also, edge maps are created for the matched model to maintain tracking through appearance changes. These maps strengthen the model against changes in appearance and illumination. On the other hand, contour tracking initiates contours in previous frames and moves it to its new position in the current frame. Parts of the object in the frame are required to have appeared in the previous frames to maintain tracking. The state of the object is updated at each time instant where the posterior probability is maximized. State space and direct minimization of contour energy functional models can be used to perform contour object tracking. An example of silhouette tracking is shown in Figure 4.3.

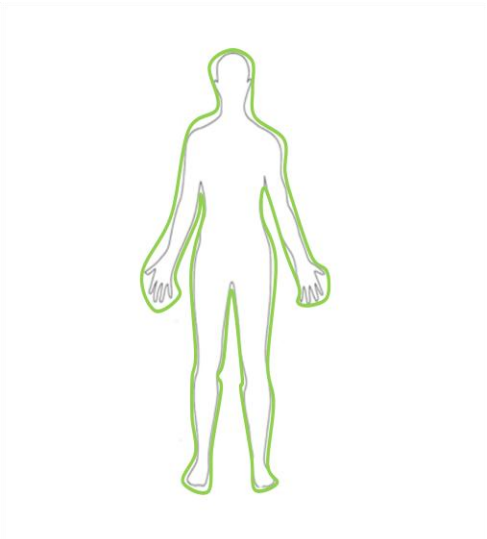


Figure 4.3 – A representation example of silhouette tracking.

4.4 Object Tracking Approach

In the literature [45-48], there are a great number of methods in which object tracking can be implemented. A simple object tracker can be sufficient since the human detection process is performed separately. The tracker is assigned to find any moving object in the video stream and report it. The detector then determines if a human is present or absent. In the following subsections, the two main functions used in the object tracker are explained to understand the tracker's functionality.

4.4.1 Kalman Filters

Kalman filters are considered one of the most commonly used techniques in object tracking. This section discusses this kind of filter and shows its use in similar applications to the one being proposed. A Kalman filter is a type of non-linear filter that makes use of a series of noisy measurements to estimate efficiently the state of a linear dynamic system. Kalman filters are utilized in several applications, most importantly in computer vision.

The filter has two main phases: "predict" and "update". The predict phase uses the previous time state estimate to output an estimate of the state at the current time. In the update phase, the collected measurement information at the current state is used to update the current prediction. The filter provides accurate, continuously updated information about the position and velocity of objects. Thus, from only a sequence of observations about the target's position from the collected frames, essential information related to the target can be acquired.

Although Kalman filters are more commonly used in the computer vision world, the unscented Kalman filter could be of more use where the state transition and observation models are non-linear. The unscented Kalman filter uses what is called the unscented transform to find the sigma points around the mean. These points are then spread through the non-linear functions to result in the covariance of the estimate. Figure 4.4 shows how a Kalman Filter can be implemented. The process is recursive in nature, taking the mean and covariance of a random variable *x* as initial data and using it to start updating the state. In a regular tracking algorithm, *x* represents the state of the detected target.

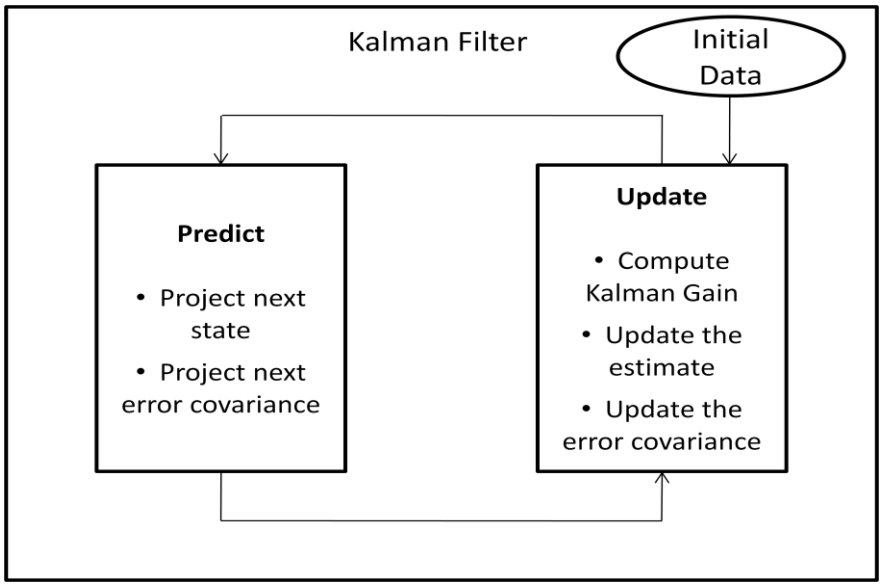


Figure 4.4 – Kalman Filter.

First, the next state mean and covariance are predicted. Then, the Kalman gain is calculated for every state while also updating the mean and covariance values. Each state is represented by a vector showing information about the location and dimensions of the tracked object. These vectors can be used in correlation with other vectors representing other tracked or untracked objects in the case of partial or total occlusion. The following is a mathematical understanding of the unscented Kalman filter.

First, let the mean and the covariance of a random variable x be $\bar{x}$ and P_x, respectively. If the dimension of x is n_x, then the sigma points (σ^i) used are $2n_x + 1$. These sigma points can be calculated using the following equations [49]:

$$\sigma^i = \bar{x} \qquad \text{Eq. 4.1}$$

$$\sigma^i = \bar{x} - \left(\sqrt{(n_x + \lambda)P_x}\right)(i) \qquad i = 1, \dots, n_x \qquad \text{Eq. 4.2}$$

$$\sigma^i = \bar{x} + \left(\sqrt{(n_x + \lambda)P_x}\right)(i) \qquad i = n_x + 1, \dots, 2n_x \qquad \text{Eq. 4.3}$$

where λ is a scaling factor.

Each sigma point is assigned a weight and both are used to approximate the updated values of the expectation and covariance. In [50], Meuter et al proposed a time-efficient approach to establish camera-based tracking from a moving vehicle. While an image processing application found the location of the pedestrians, the target movement was estimated according to the vehicle and target's relative motion. An unscented Kalman filter was then used to join the motion and measurement models. Results showed the feasibility of real time pedestrian tracking from a moving vehicle.

4.4.2 Mean-Shift Algorithm

The mean-shift algorithm was originally proposed by Fukunaga and Hostetler in 1975 [51]. This procedure was then used in the applications of computer vision and image processing [52-54]. The main function of the mean-shift algorithm is finding local maxima for a density function in a given dataset. Usually this is realized by finding the center of the tracked object in every frame. The mean-shift algorithm is a brute force technique used in object tracking. The technique assumes that the object of interest only moves slightly in consecutive frames. A kernel defined by this technique is to be used in estimating the direction in which the object is moving.

Mean-shift can use different feature spaces defined by the given kernels to represent the target object. An investigation window is created and a histogram of the object is computed in the first frame. The object histogram is computed from the pixel values defined by the kernel. Pixel weight computation can be based on color feature space [55]. After the initialization process, the object's new location is determined by finding a histogram of the object in the next frame that is most similar to the current one.

The mean-shift algorithm is then given by the following three steps:

1. Create an investigation window
2. Compute the mean vector of the object histogram within the window
3. Derive the new location to the mean and iterate until convergence

One of the major challenges that mean-shift tracking faces is the ability to maintain efficient tracking under complex backgrounds. In [56], Chen et al. proposed an improved mean-shift tracking algorithm that can handle object tracking in complex scenarios. The algorithm initiated a segmentation step and then looked at the object shape to create a level set asymmetric kernel to be used in tracking. In the traditional mean-shift algorithm, symmetric kernels are used which can be easily computed.

The level set kernels are commonly used in contour tracking and result in representing an object by its shape. In classic mean-shift tracking, the target object is represented by primitive geometrical shapes. The improved mean-shift tracking proposed by Chen showed flexible object tracking for images with various backgrounds.

4.5 Experimental Results

The adopted object tracker tries to separate moving foreground object from a relatively static background. The tracking process undergoes three major steps:

1- Foreground/background pixel labeling

2- Adjacent foreground pixel grouping

3- Object tracking and identification

The first step is the most complex and requires the highest computation among all three. In this step, each pixel is categorized as either part of the foreground or the background. This helps segment the frame and prepare the foreground pixels to be grouped and labeled as one object. The module used for the foreground/background detection is the same proposed by Li et al. in [57]. After the segmentation step, when a new object enters the scene, the adjacent pixels of the moving object are classified as foreground and are tracked. Grouping these pixels together is done using an object entrance detector based on connected components of the foreground mask.

Tracking the object can be performed using several methods, such as: simple connected component tracking, mean-shift algorithm, and particle filtering. Mean-shift tracking was selected due to its simplicity and efficiency for tracking objects in the tested surveillance schemes. The Kalman filter was used as a post processing procedure to update the position and size of the tracked object. The next step in the tracking process is generating the object trajectory and saving it in a track record. These records can be saved in a text file at a predefined location.

The final step handles the object tracking analysis by providing histograms of position and velocity of the objects in the captured video. Figure 4.5 shows the results of applying the object tracking algorithm to a video sequence in an indoor scenario. The moving object is the single human where the background is static. As shown in the figure, the tracked object is not fully occluded as it moves around in the scene. Note that when the object is entering the scene, as shown in the 7th image in the figure, it is considered as a new object and is given a different object ID.

Figure 4.5 – Object Tracking for a single moving object in a scene.

The object tracker maintained a precise location of the moving object throughout the entire video. The object tracker was also tested in an outdoor scenario with multiple moving objects. In the outdoor scenario, as shown in Figure 4.6, a considerable amount of background noise was introduced due to the wind moving the trees and needed to be taken into consideration. This challenge was solved by increasing the tracking latency which gave more time for the tracker to classify the entire tree as background and not just consider its movements.

This noisy measurement can also be negated by the detectors during the human detection process. In Figure 4.6, three single moving objects are tracked in an outdoor scenario. The images show the capability of the object tracker to locate and track moving objects with different color and texture information. Each object is given a different tracking ID as they enter the scene.

Figure 4.6 – Single Object Tracking.

The tracking ID helps maintain a record for each moving object in the scene before it goes out. This ID is ~~not~~ unique to each instance of an object and so if the same object exits and reenters the scene, it is given a new ID number. In Figure 4.7, two moving objects are being tracked with a unique tracking ID as they move closer to one another. In the third image shown, one object partially occludes the other object while tracking is still maintained for both.

Figure 4.7 – Object tracking in an outdoor scenario.

As the nearer object exits the scene, the tracker keeps its elliptical representation for some time before removing it. This is due to the post processing and tracking analysis steps that the tracker goes through prior to stopping the tracking record. Figure 4.8 shows additional examples of multi-object tracking in an outdoor scenario. The first image contains three moving objects while the tracker identifies only two of them. This is due to the tracker connecting both objects as they start to partially occlude one another during their motion path. The tracker uses a connected components approach to identify moving objects. Thus, adjacent pixels are connected and considered belonging to the same object. The objects are then separated and each is given a tracking ID. The other frames show fully tracked objects throughout the video stream.

The last image shown in the figure displays two moving objects: one entering the scene and one exiting it. The high sensitivity of the object tracker is vital to the proposed human detection system. Each moving object, especially those at the image boundaries, need to be checked for human presence/absence in the surveyed scene. Note that the shadows of the moving objects can sometimes be identified as a separate object. In this case, the tracker will report the identified object and the detectors can be used to discard these results.

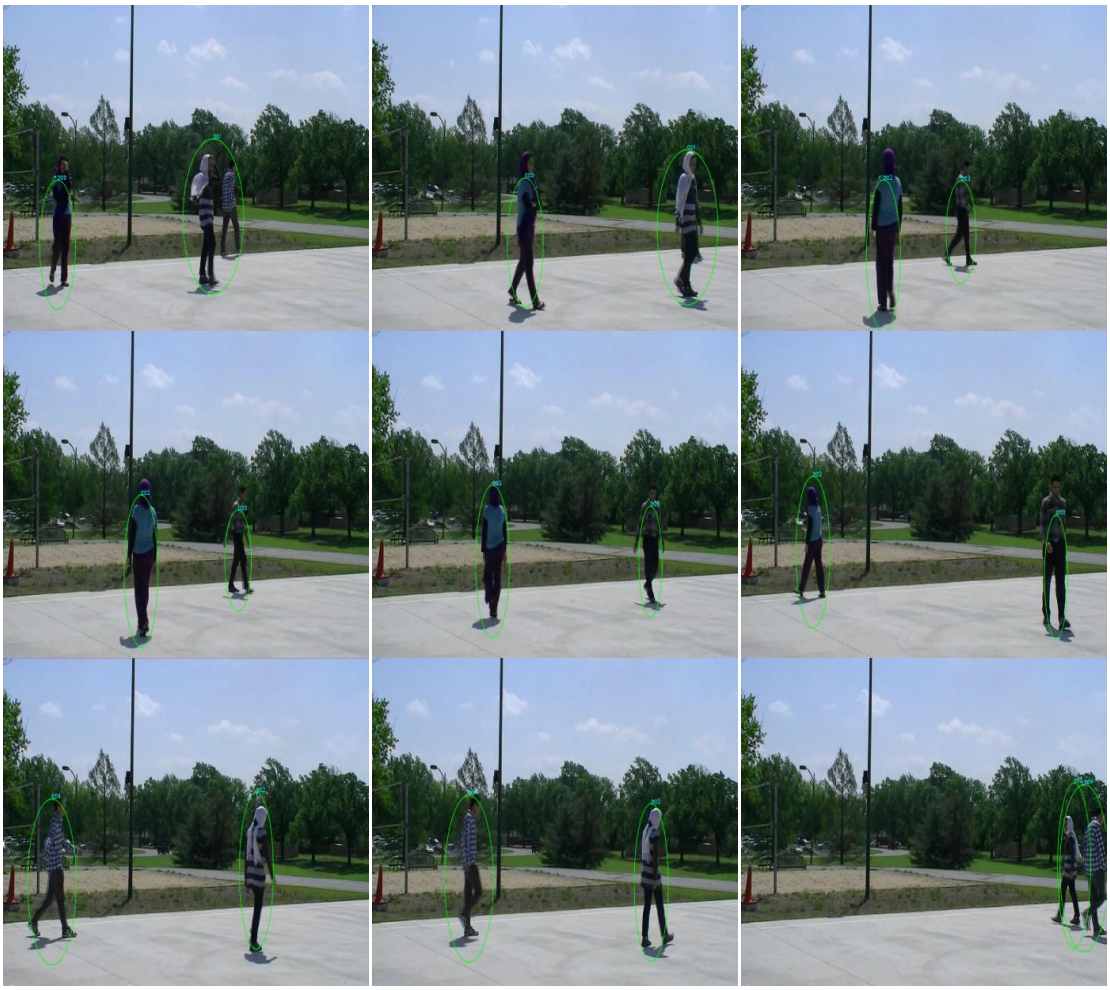


Figure 4.8 – Multiple object tracking..

4.6 Summary

In this chapter, a simple and efficient object tracker is presented. The tracker is capable of locating and assigning IDs to moving objects in the scene. Experimental results show how adaptive and reliable this technique can be when used in indoor and outdoor scenarios. This simple tracker enables a system to initiate the human detection process upon any triggered event caused by a moving object. The human detectors are responsible for detecting human presence within the area of the tracked object. The object tracker is an approach that makes use of a mean-shift algorithm and Kalman filters for kernel tracking.

Chapter 5

Human Detection and Tracking in a Feedback System

5.1 Introduction

Human detection and tracking are two major fields in computer vision and have numerous image and video processing applications. The two functions can be executed jointly or separately according to the application requirements. Combining both functions adds to the complexity of the training and testing phases of the overall system [58, 59]. Other approaches, such as [60, 61], separate human detection and tracking into two distinctive steps. Separating these functions, results in having two less complex processing systems. This proposed human detection and tracking approach combines two human detectors in conjunction with an object tracker.

As discussed in the previous chapters, the HOG and Haar detectors provide different detection rates with relatively low false positive and negative rates. Integrating both methods through a feedback messaging system increases presents a way to considerably reduce the number of false positive detections. Tracking on the other hand, is only done for moving objects. Some moving objects can be non-human at times, this is where the detectors are used to identify whether the tracked object is human or not. While the object tracker is running in real-time, the two detectors are not. The detectors process the next available frame in a video stream after the previous detection process has completed.

Figure 5.1 shows the setup of the detection and tracking system. As shown, the first step involves feeding the input frames to the object tracker for objects to be identified and tracked throughout their motion paths. Then, the HOG human detector validates human presence in the surveyed area. In the third step, the Haar human detector produces its results to confirm human presence/absence. The two detectors then compare their results to limit false positives created by either method during the detection process. This confirmation step is done using the feedback messaging system discussed later in the chapter.

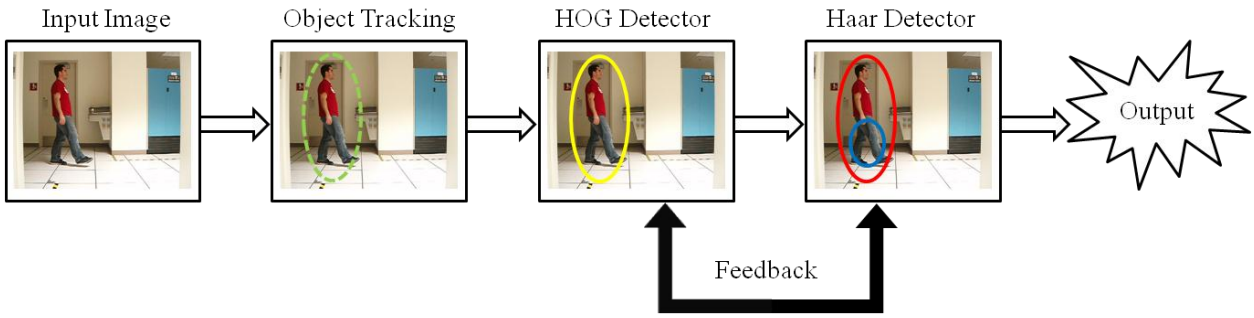


Figure 5.1 – Overall Setup of the Detection and Tracking System.

In this chapter, a literature overview is presented in Section 5.2. The feedback messaging system architecture is proposed in Section 5.3. Section 5.4 explains the detection and tracking integration methodology. Section 5.5 shows the experimental results of the overall systems. The results show that using two different human detectors with a simple object tracker can result in good human detection rates. Using these image and video processing techniques improves port security by implementing a smart surveillance system that can monitor, detect, and identify humans in a given area of investigation. A summary of the chapter is presented in Section 5.6.

5.2 Background

Using different schemes for detection and tracking has been introduced by many researchers. One of the most notable approaches is that of Li et al in [62]. The authors use the HOG detection technique to extract human features. Integral histograms are used to speed up the training and testing process for the HOG detector. To enable real-time detection and tracking, the authors used a Kalman filtering approach which also increased the detection accuracy. The proposed method was only tested for one person moving in the video capture.

Most of the current human detection methods are based on color or gray level segmentation. In [63], Bhuvaneswari and Abdul Rauf proposed a human detection and tracking system used in real-time applications. The authors extract edgelet features, silhouette oriented features, which are used in the human detection process. The detection output is considered a soft decision and depends on several confidence levels from different detection responses. These responses, in addition to the mean-shift algorithm, are also used in the tracking module. Experimental results show that increasing the number of input frames decreases the false alarm rate.

Achieving real-time human detection and tracking is essential in many video processing applications. In [64], Zhou and Hoang proposed using a codebook to classify humans present in the surveyed environment. The adopted appearance model for tracking individuals is based on color histograms computed per frame. The authors also propose a false positive detection method for two levels: object tracking and human tracking levels. Background subtraction was also used to segment the foreground and label moving objects and humans. The results showed robustness to partial occlusion.

Another approach that used Kalman filtering and image segmentation for human detection and tracking is the one proposed by Thombre et al in [65]. Each pedestrian's motion is classified using vision algorithms. The Kalman filter is used to evaluate the position and velocity path characteristic for each pedestrian within the field of vision of the camera. The system was tested in an outdoor scenario and showed the actual human detection and tracking techniques and the predicted results by the Kalman filter.

Human detection and tracking based on stereo vision systems can be a very efficient method to be used in real-time applications. In [66], Abd-Almageed et al proposed a human detection and tracking system that operates on mobile vehicles. The authors also presented a faster Adaboost human detector that is based on Fisher Linear Discriminants. That allowed for a less complex and faster classifier to be used for training and testing. Experimental results showed that false detection rate was reduced using the proposed method when compared to the other non-stereo approaches.

Particle filtering techniques have been used in object tracking by many researchers, such as in [67, 68]. In [69], Xu and Gao proposed a particle filtering human tracker with a HOG human detector. The HOG descriptor was used with a support vector machine for human classification. Experimental results showed that the particle filter is superior to the Kalman filter for human tracking. The particle filter was also better in tracking humans under partial occlusion. Three videos in outdoor scenarios were captured and the human detector and tracker were tested with an approximate tracking speed of 50 frames per second. The algorithm also showed adaptability to illumination changes, rotational human movements, and partial occlusion by other humans or background objects.

5.3 Feedback Messaging System Architecture

Feedback systems are used in several fields of study including control theory, climate science, mechanical engineering, software development and social sciences. An example of a commonly used feedback system in control theory is the proportional–integral–derivative (PID) controller [70]. Other examples, such as [71, 72], use feedback systems in image processing and computer vision applications.

5.3.1 Feedback Messaging System Overview

The proposed feedback system manages messages between the HOG and Haar detectors. These messages include several parameters such as: group and hit thresholds, detection scheme, bounding box size and location, number of bounding boxes, and detection decision. An overview of the system is shown in Figure 5.2. Parameters are sent back and forth between the two detectors to provide a better estimate for the detection process. Each detector can provide one of three outcomes: detection, false positive or false negative, although the individual detector is unaware of the actual outcome. These cases are determined subjectively for each detector by looking at each test image separately. In the proposed approach, the feedback system iterates once for every inspected frame.

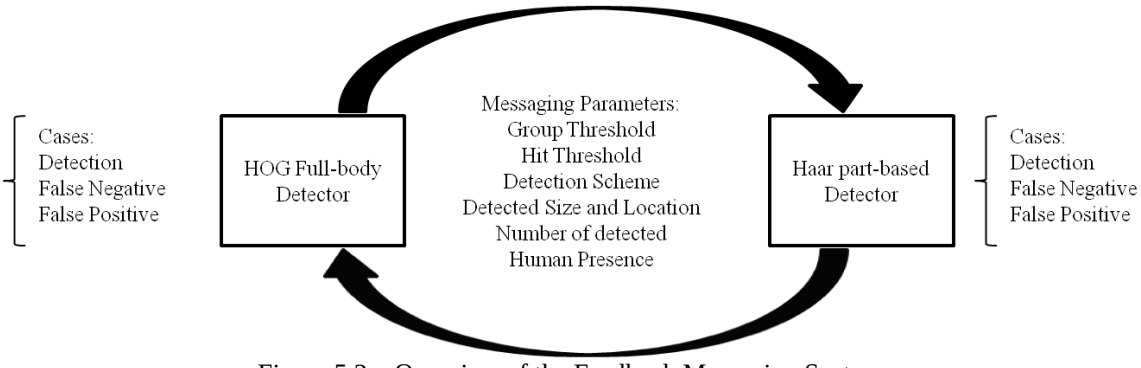


Figure 5.2 – Overview of the Feedback Messaging System.

5.3.2 Feedback Messaging

Several requirements need to be met to merge the two human detectors (HOG and Haar) to be able to accurately detect humans. The main objective in this system is to get both detectors to process one frame from the video and perform detection. Either detector can start first regardless of the detection time since the results from both should be present to be used. The HOG detector, throughout the testing process, showed better detection rate than the Haar-like feature detector. This can be used to make the HOG detector result a higher priority if needed in the system.

The messaging between the two detectors includes parameters used by each method as well as results produced by them. When a moving object enters the scene, the detectors try to decide whether that object is a human or not. The two detectors produce two separate decisions with the number of detected humans and their locations in the frame. If either or both find a human an alert will be sent. Thus, the detection rate can be increased. The HOG detector is a full body detector while the Haar is a leg detector. Thus, for both to agree, the bounding box for the Haar should be approximately within the bounding box of the HOG; more precisely, in the lower part of the box.

If each detector finds the human in a different spot, then the location relevance is taken into consideration to determine whether one should be canceled. For example, if the HOG finds a human in the tracking window and the Haar detector locates a human up in the air, then the Haar result is discarded. A scene analysis of the surveyed area is done subjectively prior to the system initiation. This is a simple step to help decrease the false positives produced by both detectors.

5.3.3 Alert System

The alert system is designed for object analysis of any movement occurring in the monitored area. The decisions made by the human detection and tracking schemes set system alerts. Before a description of the alert system is presented, a number of assumptions need to be stated.

5.3.3.1 Human Identification Assumption

The proposed human detection and tracking system is not capable of identifying which human is present in any given frame. This recognition step can be performed using other techniques, [73, 74], that are not applied in this research. Human identification requires knowing who exactly is in the image and to give that human a unique identifier for future reference. In this system, each tracked object is given an ID number whether that object is human or not. Any human activity in the given surveyed area confirmed by the two detectors and tracker is capable of triggering an alert.

5.3.3.2 Static Background Assumption

The object tracker used in the system is a simple implementation of Kalman filters and the mean-shift algorithm. It does not take into consideration camera movements. The camera sensors used in the system are assumed static at all time. Several approaches have been proposed in the literature, such as [75 - 77], that can be used to present a solution to this challenge. The proposed system can be improved using these approaches to handle camera motion during tracking.

5.3.3.3 Alert Scheme

The alert scheme is straightforward and is shown in Table 5.1. When the tracker locates a moving object in the surveyed area, an alert is sent out and the human detection process is initiated. If there is no motion in the secure area, then no alert is sent out. When the tracker locates a moving object and the human detection classifies it as human, an alert with a high priority is sent. Table 5.1 shows all investigated cases in the detection and tracking system. The three cases are:

1. No motion/human detection: In this case, the object tracker did not locate any moving objects. Thus no alerts are sent out to the security personnel.
2. Motion detection & no human detection: A moving object is being tracked but the detectors did not output a bounding box around that object to indicate that it is a human. An alert with low priority is sent specifying motion detection.
3. Motion detection & human detection: This case includes having the tracker identifying a moving object and the detectors finding a human in the same area. An alert with a high priority is sent.

Table 5.1 – Alert Cases for Motion and Human Detection.

Case	Object Tracker	Merged Detector	Alert Type
1	No Motion Detected	No Humans Detected	None
2	Motion Detected	No Humans Detected	Low
3	Motion Detected	Humans Detected	High

5.4 Detection and Tracking Integration

The proposed feedback messaging system combines the two detectors to work together in detecting human presence in given frames. The other image processing technique used in the overall system is object tracking. This step is initiated first in the system and used throughout the complete smart surveillance process. The tracker assigns an ID for every tracked object and saves the location of that object as it moves around the surveyed scene. An alert is sent out for each tracked object. The system then gets the first available frame and starts the detection process using the HOG full body and the Haar part-based detectors. Figure 5.5 shows the modules that make up the overall human detection and tracking system. The sensing module consists of fixed IP cameras to capture the video of the surveyed scene. The tracking module initiates the object tracker and identifies each object with an ID number. The detection module makes use of a full body and part-based human detectors incorporating a feedback messaging system. The last module, the output module, provides the conclusion of the system by sending out alerts to the user monitoring the system. These four modules are the building blocks for the proposed smart surveillance system.

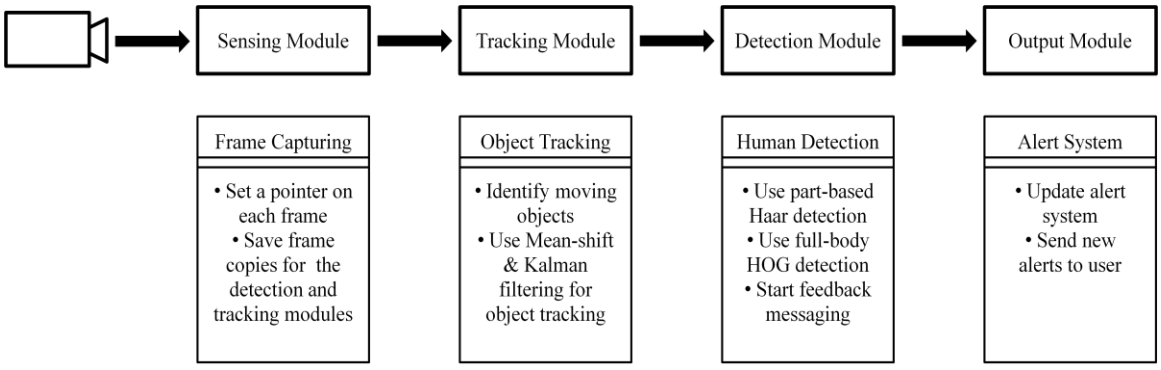


Figure 5.5 – Proposed modules for the Detection and Tracking System.

A flow chart of the integrated tracking and detection methods processing the input frames from the sensing module is given in Figure 5.6. The object tracker is running in real-time while the merged human detectors are not. First, the tracker starts inputting frames for processing; whenever a moving object is located a copy of the current frame (the first available frame) is sent to the merged detector. Inside the merged detector, the two detection approaches run separately on two copies of the frame being processed. The results from each detector are then fed to the feedback messaging system in addition to other parameters. The final decision is made by the merged detector after a period of time and given to the output module, which in turn decides on whether an alert of a determined priority needs to be triggered or not. By the time this detection process is finalized, more frames already have been captured. These frames are not buffered, thus the human detector looks at the next available frame.

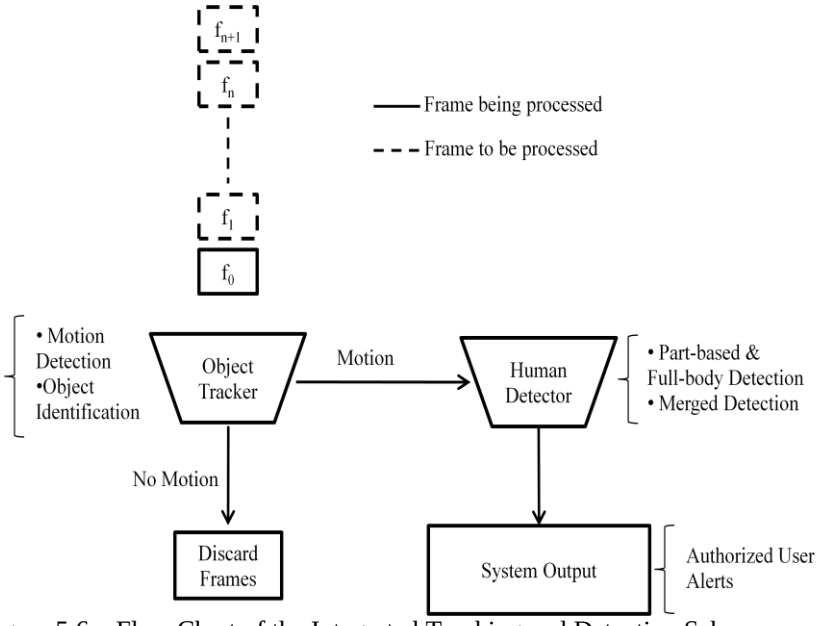


Figure 5.6 – Flow Chart of the Integrated Tracking and Detection Schemes.

5.4.1 Human Detection Timeline

For the tested scenarios shown in the next section, two cameras were used for sensing. The first camera captured video at 15 frames per second (fps) under two resolutions: 640x480 and 320x240 pixels. The second camera was used to capture at 848x480 pixels resolution at 30 fps. Varying the video resolution and frame rates shows the system's ability to integrate different sensing devices and their effects on the overall system performance.

The first case includes using a 15 fps camera with a 320x240 video resolution. The object tracker inputs all 15 fps for processing. The average detection time for the merged detector is 190 ms. Thus, the detector can input a frame for processing and finish before the third consecutive frame. Figure 5.7 shows the timeline where each frame is marked to be processed by either both the detector and tracker, or only the tracker. In this case, it take 190 ms for the merged detector to process one frame (e.g., frame 0). In this time two frames already passed, thus the detector grabs the next available frame which is the frame 3. The other missed frames are discarded for the detection process but used for the object tracking process.

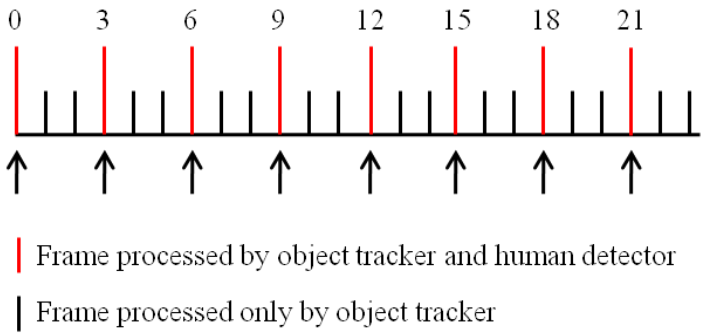


Figure 5.7 – Timeline for processing 15 fps video with a 320x240 resolution.

The second setup included having a 15 fps camera capturing a 640x480 pixels resolution video. The frames processed by both the detector and tracker are marked in red whereas the frames processed by the tracker are marked in black, as shown in Figure 5.8. In this case, the detection system requires 880 ms to process one frame. Thus, by the time the detector has processed one frame, 13 frames have passed. The detector takes the next available frame which in the example below is frame 14.

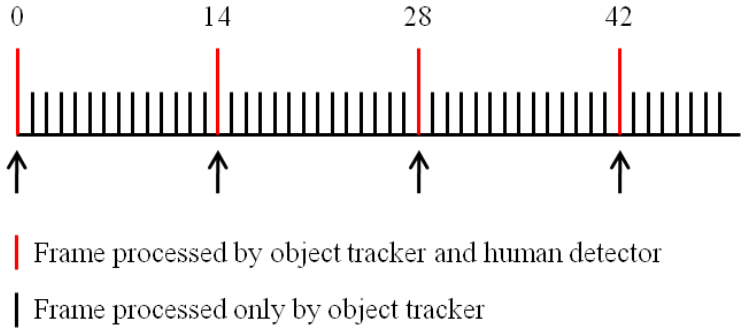


Figure 5.8 – Timeline for processing 15 fps video with a 640x480 resolution.

In the last case, a 30 fps camera is used to capture an 848x480 pixels resolution video. For this resolution, the human detector takes 1.15 s to process and output the results. Therefore, 34 frames pass while the detector is processing frame 0. The next available frame for processing will be frame 35 where all the other 34 frames are discarded. Figure 5.9 shows the timeline for this case.

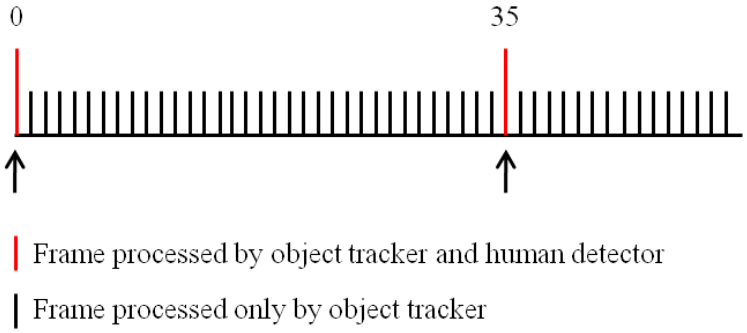


Figure 5.9 – Timeline for processing 30 fps video with an 848x480 resolution.

5.5 Experimental Results

The collected experimental results show the performance of the combined human detector compared with the two separated detectors. The feedback system maintained a high detection rate and decreased the false positive rate which results in a more robust detector. Indoor and outdoor scenarios with different image resolutions are tested.

5.5.1 Merged HOG and Haar Detectors Results in an Indoor Scenario

The first scenario tested for the two detectors was indoors, as shown in Figure 5.10. This scenario was used previously to test the HOG full body and the Haar leg detector separately. The collected results showed high detection rates in both cases and very low false positive and negative rates. The detection rate for the Haar leg detector was 93.8% for 210 test images and the false positives rate was 9.5%. The HOG detector was able to locate the human in every frame with an insignificant false positive rate.

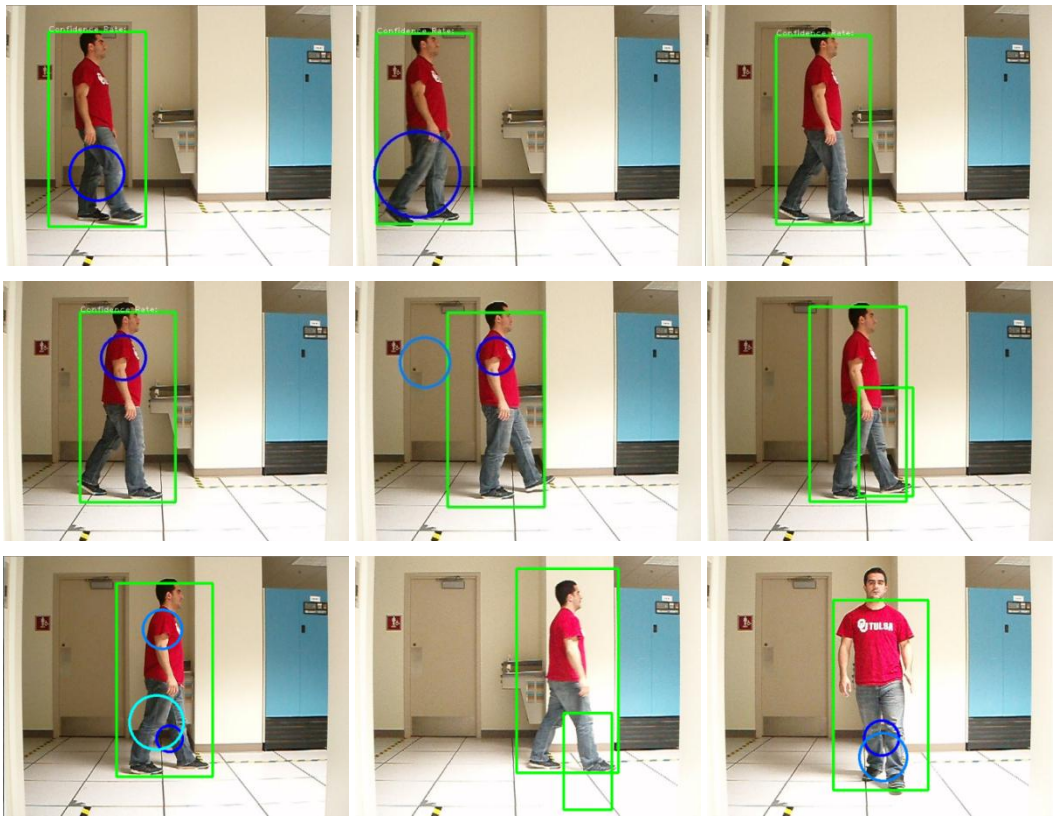


Figure 5.10 –HOG and Haar Used in an Indoor Scenario.

The test results for the indoor scenario were taken to show both detectors activities and how the algorithm works in different cases. For example, the first, second and ninth frames in the above figure show complete detection. The third, sixth and eighth frames show a detected human by the HOG detector and missed detection by the Haar detector as explained in Subsection 5.3.2. The fourth and fifth frames show two cases of HOG detection and Haar false detection. Note that in the fourth frame the false detected leg is the upper body and within the region of the human. In the fifth frame, a second false positive is shown by the Haar detector behind the human. This false positive is discarded during the feedback messaging algorithm while the other one, which is in the human detection region, is not. The seventh frame shows one HOG detection box and three Haar detection circles, two of which are true detection and one false positive that falls within the HOG box.

5.5.2 Merged Detectors Tested on Two Humans in an Outdoor Scenario

The second scenario used to test the two merged detectors was of two humans in an outdoor scenario. Figure 5.11 shows the detected false positive and negative results for both detectors. The first frame shows two HOG boxes for the two humans and that the Haar detector has missed both. The second and ninth frames are the only ones where both detectors agree on spotting both pedestrians. In the third frame, the HOG detector finds both humans whereas the Haar finds none and adds a false positive.

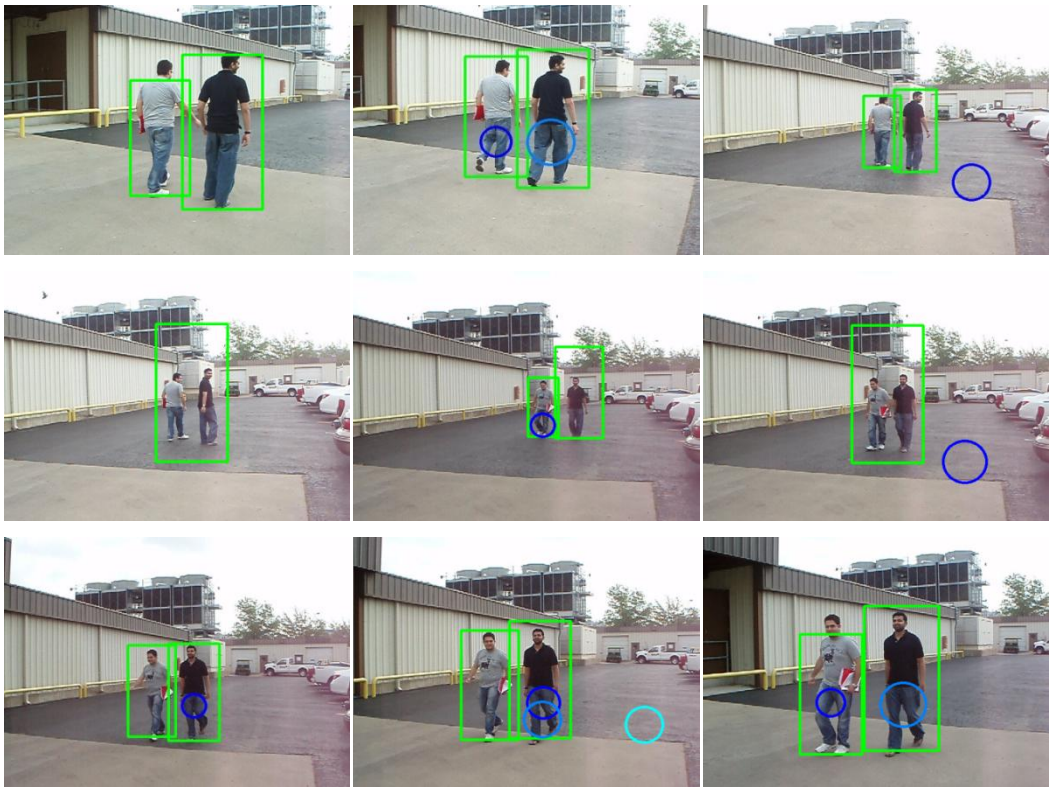


Figure 5.11 – Results of Applying both Detectors in an Outdoor Scenario.

The fourth and sixth frames show both humans detected as one using the HOG full body detector. In these two cases, only one alert is sent. In the fifth, seventh, and eighth frames, the HOG finds the two humans whereas the Haar detector only finds one. Two alerts are sent out to the authorized personnel. When tested separately using 300 test images, the detectors showed different detection, false positive and negative rates. The HOG outperformed the Haar detector in the detection and negative rates by almost 20% for each. Both detectors had approximately the same false positive rate of 6%.

Using the feedback messaging system, a more accurate human detector can be established by merging the two full and part-based detectors. The feedback system helps decrease the false positive rate for the combined detector. Table 5.3 shows the statistics for all three cases.

Table 5.2 – Detection Statistics for Separated and Merged Detectors.

Detector Types	Resolution (in pixels)	Total Number of Test Images	Detection Rate	False Positive Rate	False Negative Rate	Average Detection Time (in ms)
HOG Full Body Detector	640x480	300	97.0%	5.3%	3%	790
Haar Leg Detector	640x480	300	77.33%	6%	22.67%	50
Merged Detector	640x480	300	97.0%	0.67%	3%	880

Each of the 300 test images must ideally produce two alerts, one for each human in the captured frame. Thus, the expected total number of true alerts sent is 600. The false positive rate can be decreased using information from both detectors where the human is expected to be. Therefore, a huge reduction in the false positive rate can be observed.

On the other hand, the negative rate stays the same as the one for the more accurate detector, which in this case is the HOG full body detector. The final detection rate for the merged detector is 97%. The detection time for the final detector is approximately the sum of the detection time of both detectors in addition to a small margin taken for the feedback messaging system.

5.5.3 Merged Detectors Tested on Multiple Humans in an Outdoor Scenario

The last scenario investigated has multiple humans walking in an outdoor scene. Again, the two detectors are applied on several test frames to determine subjectively the false positive, false negative and detection rates. Figure 5.12 shows the results of merging the two detectors. As expected, the HOG detector produced a detection rate higher than that of the Haar leg detector. The HOG detection rate was 93.5% while the Haar had a detection rate of 62.8% for 300 test images. The false positive rate in both cases was less than 3%. Note that the Haar leg detector was not able to find all four pedestrians in the test images. This is due to the training dataset that only included one instance of the target object for each image.

Figure 5.12 – Results of Applying both Detectors for Multiple Human Detection.

In this scenario, four humans are walking around and at times partially or fully occluding one another. Table 5.3 shows the detection, false positive and negative rates in addition to the average detection time for each detector. The detection time is higher than the previous scenario due to an increase in the video resolution from 640x480 to 848x480 pixels. The system requires just over a second to determine whether one or more humans are present in frames of size 848x480 pixels.

Table 5.3 – Detection Statistics for Multiple Human Separate and Merged Detectors.

Detector	Resolution (in pixels)	Total Number of Test Frames	Detection Rate	False Positive Rate	False Negative Rate	Average Detection Time (in ms)
HOG Full Body Detector	848x480	300	93.5%	1.1%	6.5%	1010
Haar Leg Detector	848x480	300	62.8%	2.7%	37.2%	70
Merged Detector	848x480	300	93.5%	0%	6.5%	1150

Ideally, the number of produced alerts should be 1200, but in this case, the 300 test images contained 1, 2, 3 or 4 humans per frame. The total number of expected alerts is 838 alerts. Note that the detection rate for the merged detector is not much higher than that of the full body detector due to the high negative rate that was not decreased. On the other hand, the false positive rate was taken out by the feedback messaging system. The false positives from both detectors were not in the same location and also did not correspond with the location of the moving object given by the tracker.

5.6 Summary

In this chapter, the overall smart surveillance system was presented in addition to a detailed description of the building blocks of the system. This system can be used to enhance security at shipping ports. The system consisted of four modules: sensing, tracking, detection, and output. A simple object tracker was used to identify and track moving objects as they enter and leave the surveyed scene. A merged human detector based on a full body and part-based detector was used. Combining the two detectors decreased the false positive rate. Experimental results showed the detector's high accuracy and ability to find humans in a given frame. The output module contained an alert system given by the object tracker and the merged detector. The proposed human detection and tracking system proved to be an efficient and low cost method to improve security measures for many security applications.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this dissertation, a human detection and tracking system was presented to enhance security at ports of entry. This work is the first to demonstrate a detection system based on combining two detection methods using a feedback messaging system. The merged detector provided higher detection rate and lower false positive rate. Indoor and outdoor scenarios were used to test the human detector and tracker to show the system reliability and robustness. Thus, such a smart surveillance system can be utilized for applications to improve security measures. This smart surveillance system is essential for improving security measures in various monitoring applications. The system consisted of four main modules:

- Sensing Module: involved sensing devices, static cameras, which input a number of frames to be processed by the other modules.
- Tracking Module: introduced a simple object tracker that identified moving objects and tracked them in the given surveyed area. The locations of these objects were then sent to the detection module.
- Detection Module: consisted of two detection methods combined using a feedback messaging system. The merged detector achieved high detection and limited the errors of the two detectors.
- Output Module: produced a set of alerts with a confidence level to alert the security personnel of human activity in the monitored area.

The four modules provide a complete smart surveillance system that integrates several image and video processing techniques to achieve reliable human detection and tracking. The alert system based on this integration gives the security personnel different levels of alerts to be dealt with.

The proposed human detector and tracker can be used in various applications including: ports of entry, smart surveillance systems, intelligent transportation systems, and many others. The low cost approach that is utilized in this system makes it a desirable security solution for such applications. Also, the easy integration of software and hardware that is required is another important feature. A graphical user interface can be designed to give security personnel a subjective and controllable view of the monitored area. Appendix A contains a sample code for using both the HOG full body detector and the Haar-like part-based detector together. The code iterates once where the HOG detector and the Haar-like feature detector exchange bounding box locations and produce a joint human detection decision.

The feedback messaging system is a novel approach that combines two image processing techniques based on the detection parameters. The parameters, such as detected human location, are sent between the detectors to help set a certain confidence level for the sent alert. This helps create a more robust and reliable human detector that can be used in different applications. The proposed system can be used in a wide area of applications including: intelligent transportation systems, smart surveillance systems, roadside collision avoidance systems, home monitoring, and others.

6.2 Future Work

The dissertation objective was to find a way to detect and track humans in a given area of interest with high detection rate and to decrease errors produced by detectors. The research presented can still be improved by increasing the computation power using parallel programming and strengthening the multi-human detection rate.

6.2.1 Real-time Detection

The trained Haar-like feature detector can perform part-based human detection in real-time, while the full body HOG human detector cannot. The later requires an average detection time of 700 ms which is not suitable for real-time applications. The detection speed of the HOG can be improved using graphical processing units (GPUs) as proposed in [78 - 80]. Using parallel computing can speed up the HOG algorithm and GPUs are used mainly for parallel processing. GPUs can be used for a full real-time human detection and tracking system that incorporates both the HOG and Haar-like feature detection methods.

6.2.2 Detection Enhancements

The detection rates presented in this work was very high in indoor and outdoor scenarios and with one or multiple humans present in the frame. Though, the Haar-like feature detector was only trained with positive samples that contained one human. This showed to be a drawback in the experimental results where the detection rate dropped down from 84.3%, for single human detection, to 60.1%, for multiple human detection. A more robust Haar-like feature detector can be trained with a dataset that contains multiple humans that are marked and saved during the training process.

This variation in the training process will improve the part-based detection and thus the overall detection of the system. Also, this could help decrease the false positive rates. It is recommended to vary the backgrounds in the negative and positive images as discussed in Chapter 2 as well as having positive samples with multiple humans.

Another enhancement can be to speed up the detection process by limiting the area of investigation to only that of the tracker. This would decrease the number of detection windows processed by the human detector. In this case, only moving objects are looked at and not the entire image. Also, enhancing this system to be able to handle moving backgrounds can result in greater utilization in various applications.

References

[1] R. C. Huck, M. K. Al Akkoumi, R. W. Herath, J. J. Sluss Jr., S. Radhakrishnan, and T. L. Landers, “A Demonstration of a Low Cost Approach to Security at Shipping Facilities and Ports”, presented in SPIE Defense, Security and Sensing, Orlando, FL, April 2010

[2] R. C. Huck, M. K. Al Akkoumi, S. Shammaa, J. J. Sluss Jr., S. Radhakrishnan, and T. L. Landers, "A building block approach to security at shipping ports" presented in Unmanned/Unattended Sensors and Sensor Networks VI, Proceedings of SPIE Vol. 7480, Berlin, Germany, September 2009

[3] T. B. Moeslund, A. Hilton and V. Kruger, “A survey of advances in vision-based human motion capture and analysis”, Computer Vision and Image Understanding, Vol. 104, Issue 2, November 2006

[4] C. Cortes and V. Vapnik, "Support-Vector Networks", Machine Learning, Vol. 20, pp.273 - 297, September 1995

[5] J.A.K. Suykens and J. Vandewalle, “Least squares support vector machine classifiers”, Neural Processing Letters, vol. 9, no. 3, pp. 293 - 300, June 1999

[6] M.S. Ryoo and J. K. Aggarwal, “Observe-and-explain: A new approach for multiple hypotheses tracking of humans and objects”, Computer Vision and Pattern Recognition, Pages: 1 – 8, 2008

[7] Paul Viola, Michael J. Jones and Daniel Snow, “Detecting Pedestrians Using Patterns of Motion and Appearance”, IEEE International Conference on Computer Vision, Volume 2, pp. 734-741, October 2003

[8] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Volume 1, pp.I-511 - I-518 vol.1, 2001

[9] L. Yun and Z. Peng, “An Automatic Hand Gesture Recognition System Based on Viola-Jones Method and SVMs”, IEEE International Workshop on Computer Science and Engineering, Vol. 2, pp. 72-76, October 2009,

[10] T. Ephraim, T. Himmelman and K. Siddiqi, “Real-Time Viola-Jones Face Detection in a Web Browser”, IEEE Canadian Conference on Computer and Robot Vision, pp. 321-328, May 2009

[11] D. Hefenbrock, J. Oberg, N.T.N. Thanh and R. Kastner, “Accelerating Viola-Jones Face Detection to FPGA-Level Using GPUs”, IEEE International Symposium on Field-Programming Custom Computing Machines, pp. 11-18, May 2010

[12] M. Kolsch and M. Turk, “Analysis of rotational robustness of hand detection with a Viola-Jones detector”, proceedings of the International Conference on Pattern Recognition, Santa Barbara, CA, Volume 3, pp. 107 - 110, August 2004

[13] T. Mita, T. Kaneko and O. Hori, “Joint Haar-like features for face detection”, IEEE International Conference on Computer Vision, Beijing, pp. 1619 - 1626, Volume 2, October 2005

[14] S. Han, Y. Han and H. Hahn, “Vehicle Detection Method using Haar-like Feature on Real Time System”, World Academy of Science, Engineering and Technology, Issue 59, pp. 455 - 459, 2009

[15] X. Cui, Y. Liu, S. Shan, X. Chen and W. Gao, “3D Haar-like Features for Pedestrian Detection”, IEEE International Conference on Multimedia and Expo, pp. 1263 – 1266, July 2007

[16] W. C. Chang and C. W. Cho, “Multi-Class Boosting with Color-Based Haar-Like Features”, IEEE Conference on Signal-Image Technologies and Internet-Based System, Shanghai, pp. 719 - 725, December 2007

[17] Negative samples dataset, http://note.sonots.com/SciSoftware/haartraining.html

[18] Objectmarker for Haartraining, http://www.cs.utah.edu/~turcsans/DUC/

[19] OpenCV library, http://opencv.willowgarage.com/wiki/

[20] INRIA Person Dataset, http://pascal.inrialpes.fr/data/human/

[21] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Volume 1, pp. 886 - 893, June 2005

[22] D. Lowe, "Object recognition from local scale-invariant features", IEEE International Conference on Computer Vision, Volume 2, pp. 1150 - 1157, 1999

[23] H.X. Jia and Y.J. Zhang, “Fast Human Detection by Boosting Histograms of Oriented Gradients”, IEEE International Conference on Image and Graphics, August 2007, pp. 683-688

[24] Q. Zhu, M.C. Yeh, K.T. Cheng and S. Avidan, “Fast Human Detection Using a Cascade of Histograms of Oriented Gradients”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, Vol. 2, pp. 1491-1498

[25] C. Zhou, L. Tang, S. Wang and X. Ding, “Human Detection Based on Fusion of Histograms of Oriented Gradients and Main Partial Features”, IEEE International Congress on Image and Signal Processing, pp. 1-5, October 2009

[26] M. Bertozzi, A. Broggi, M. Del Rose, M. Felisa, A. Rakotomamonjy and F. Suard, "A Pedestrian Detector Using Histograms of Oriented Gradients and a Support Vector Machine Classifier", IEEE Intelligent Transportation Systems Conference, pp 143 - 148, September 2007

[27] L. Hu, W. Liu, B. Li and W. Xing, "Robust Motion Detection using Histogram of Oriented Gradients for Illumination Variations", IEEE International Conference on Industrial Mechatronics and Automation, Volume 2, pp. 443 - 447, May 2010

[28] A. Toya, Z. Hu, T. Yoshida, K. Uchimura, H. Kubota and M. Ono, "Pedestrian Recognition using Stereo Vision and Histogram of Oriented Gradients", IEEE International Conference on Vehicular Electronics and Safety, pp. 57 - 62, September 2008

[29] K. Lee, C. Y. Choo, H. Q. See, Z. J. Tan and Y. Lee, "Human Detection using Histogram of oriented gradients and Human body ratio estimation", IEEE International Conference on Computer Science and Information Technology, Volume 4, pp. 18 - 24, July 2010

[30] P. E. Rybski, D. Huber, D. D. Morris and R. Hoffman, "Visual classification of coarse vehicle orientation using Histogram of Oriented Gradients features", IEEE Intelligent Vehicles Symposium, pp. 921 - 928, June 2010

[31] K. Lillywhite, D. Lee and D. Zhang, "Real-time Human Detection Using Histograms of Oriented Gradients on a GPU", IEEE Applications of Computer Vision, pp. 1 - 5, December 2009

[32] F. Porikli, "Integral Histogram: A Fast Way to Extract Histograms in Cartesian Spaces", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Volume 1, pp. 829 - 836, June 2005

[33] H. Li and J. Cao, "Detection and Segmentation of Moving Objects Based on Support Vector Machine", IEEE International Symposium on Information Processing, pp. 193 - 197, October 2010

[34] E. Pasolli, F. Melgani, M. Donelli, R. Attoui and M. de Vos, "Automatic Detection and Classification of Buried Objects in GPR Images Using Genetic Algorithms and Support Vector Machines", IEEE Geoscience and Remote Sensing Symposium, Volume 2, pp. II-525 – II528, July 2008

[35] G. Zhu, C. Xu, Q. Huang and W. Gao, "Automatic Multi-Player Detection and Tracking in Broadcast Sports Video using Support Vector Machine and Particle Filter", IEEE International Conference on Multimedia and Expo, pp. 1629 - 1632, July 2006

[36] The MIT Pedestrian Dataset provided by the MIT Center for Biological and Computational learning, http://cbcl.mit.edu/software-datasets/PedestrianData.html

[37] The INRIA Person Dataset, http://pascal.inrialpes.fr/data/human/

[38] A. Yilmaz, O. Javed and M. Shah, “Object Tracking: A Survey”, ACM Journal of Computing Surveys, Volume 38, Number 4, 2006

[39] V. Rabaud and S. Belongie, “Counting Crowded Moving Objects”, IEEE Computer Vision and Pattern Recognition, volume 1, pp. 705 - 711, June 2006

[40] M. Oren, C. Papageorgiou, P Sinha, E. Osuna, and T. Poggio, “Pedestrian Detection Using Wavelet Templates” Computer Vision and Pattern Recognition, IEEE Computer Society Conference, pp. 193 - 199, June 1997

[41]Chuong T Nguyen and Joseph P. Havlicek, “Modulation Domain Features for Discriminating Infrared Targets and Backgrounds”, IEEE International Conference on Image Processing, pp. 3245 - 3248, October 2006

[42] C. T. Nguyen, J. P. Havlicek, and M. Yeary, “Modulation Domain Template Tracking”, IEEE International Conference on Computer Vision and Pattern Recognition, pp. 1 - 8, July 2007

[43] N. A. Mould, C. T. Nguyen, and J. P. Havlicek, “Infrared Target Tracking with AM-FM Consistency Checks”, IEEE Southwest Symposium on Image Analysis and Interpretation, pp. 5 - 8, March 2008

[44] K. Shafique and M. Shah, “A non-iterative greedy algorithm for multi-frame point correspondence”, IEEE International Conference on Computer Vision, Volume 1, pp. 110 - 115, October 2003

[45] H. Kim and K. Sim, “A particular object tracking in an environment of multiple moving objects”, IEEE International Conference on Control Automation and Systems, pp. 1053 - 1056, October 2010

[46] N. D. Binh, “A Robust Framework for Visual Object Tracking”, IEEE International Conference on Computing and Communication Technologies, pp. 1 - 8, July 2009

[47] Q. Xuena, L. Shirong, L. Fei, Z. Weitao and D. Fangfang, “A moving object tracking method based on sequential detection scheme”, IEEE Chinese Control Conference, pp. 2991 - 2996, July 2010

[48] J. U. Cho, S. H. Jin, X. D. Pham and J. W. Jeon, “Object Tracking Circuit using Particle Filter with Multiple Features”, IEEE International Joint Conference SICE-ICASE, pp. 1431 - 1436, October 2006

[49] E. A. Wan and R. Van Der Merwe, “The unscented Kalman filter for nonlinear estimation”, IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium, pp. 153 - 158, October 2000

[50] M. Meuter, U. Iurgel, S. B. Park and A. Kummert, “The Unscented Kalman Filter for Pedestrian Tracking from a Moving Host”, IEEE Intelligent Vehicles Symposium, pp. 37 - 42, June 2008

[51] K. Fukunaga and L. Hostetler, “The estimation of the gradient of a density function, with applications in pattern recognition”, IEEE Transactions on Information Theory, Volume 21, Issue 1, pp. 32 - 40, January 1975

[52] Y. Zhou, S. Jiang and M. Yin, “A Region-Based Image Segmentation Method with Mean-Shift Clustering Algorithm”, IEEE International Conference on Fuzzy Systems and Knowledge Discovery, pp. 366 - 370, October 2008

[53] W. Fei and S. Zhu, “Mean shift clustering-based moving object segmentation in the H.264 compressed domain”, IET Image Processing, Volume 4, Issue 1, pp. 11 - 18, February 2010

[54] M. Wang, C. Zhang and Y. Song, “Extraction of image semantic features with spatial-range mean shift clustering algorithm”, IEEE International Conference on Signal Processing, pp. 906 - 909, October 2010

[55] H. Yu, J. Wei and J. Li, “Object Tracking by Mean Shift Based on Color Distribution and Simulated Annealing”, IEEE International Conference on Future Information Technology and Management Engineering, pp.128 - 131, December 2009

[56] X. Chen; S. Yu and Z. Ma, “An improved mean shift algorithm for moving object tracking”, IEEE World Congress on Intelligent Control and Automation, pp. 5111 - 5114, June 2008

[57] H. Li, W. Huang, I. Y. H. Gu and Q. Tian, “Foreground Object Detection from Videos Containing Complex Background”, proceedings of the ACM international conference on Multimedia, New York, USA, November 2003

[58] J. Wang, G. Bebis and R. Miller, “Robust Video-Based Surveillance by Integrating Target Detection with Tracking”, IEEE Computer Vision and Pattern Recognition Workshop, pp. 137 - 144, June 2006

[59] L. Li, J. K. E. Hoe, S. Yan andX. Yu, “ML-fusion based multi-model human detection and tracking for robust human-robot interfaces”, IEEE Workshop on Applications of Computer Vision pp. 1 - 8, December 2009

[60] M. Gupta, L. Behera and V. K. Subramanian, “A Novel Approach of Human Motion Tracking with the Mobile Robotic Platform”, IEEE International Conference on Computer Modeling and Simulation, pp. 218 - 223, March 2011

[61] A. Utsumi and N. Tetsutani, “Texture adaptation for human tracking using statistical shape model”, IEEE International Conference on Pattern Recognition, Volume 2, pp. 973 - 976, 2002

[62] C. Li, L. Guo and Y. Hu “A New Method Combining HOG and Kalman Filter for Video-based Human Detection and Tracking”, IEEE International Congress on Image and Signal Processing, Volume 1, pp. 290 - 293, October 2010

[63] K. Bhuvaneswari and H. Abdul Rauf, “Edgelet based human detection and tracking by combined segmentation and soft decision”, IEEE International Conference on Control, Automation, Communication and Energy Conservation, pp. 1 - 6, June 2009

[64] J. Zhou and J. Hoang, “Real Time Robust Human Detection and Tracking System”, IEEE International Computer Society Conference on Computer Vision and Pattern Recognition, pp. 149 - 156, June 2005

[65] D. V. Thombre, J. H. Nirmal and D. Lekha, “Human detection and tracking using image segmentation and Kalman filter”, IEEE International Conference on Intelligent Agent & Multi-Agent Systems, pp. 1 - 5, July 2009

[66] W. Abd-Almageed, M. Hussein and M. Abdelkader, “Real-Time Human Detection and Tracking from Mobile Vehicles”, IEEE Intelligent Transportation Systems, pp.149 - 154, September 2007

[67] Y. Jin and F. Mokhtarian, “Variational Particle Filter for Multi-Object Tracking”, IEEE International Conference on Computer Vision, pp. 1 - 8, October 2007

[68] S. Fazli, H. M. pour and H. Bouzari, “Particle Filter Based Object Tracking with Sift and Color Feature”, IEEE International Conference on Machine Vision, pp. 89 - 93, December 2009

[69] F. Xu and M. Gao, “Human detection and tracking based on HOG and particle filter”, IEEE International Congress on Image and Signal Processing, pp. 1503 - 1507, October 2010

[70] Y. Li, K. H. Ang and G. C. Y. Chong, “PID control system analysis and design”, IEEE Control Systems Magazine, Volume 26, Issue 1, pp. 32-41, February 2006

[71] A. Raglin, M. Voronstov and M. Chouikha, “Winner take all in a large array of opto-electronic feedback circuits for image processing”, IEEE International Conference on Image Processing, Volume 2, pp. II-349 - II-352

[72] M. Al-Fandi, M. A. Jaradat, A. Abusaif and T. C. Yih, “A real time vision feedback system for automation of a nano-assembly manipulator inside scanning electron microscope”, IEEE International Multi-Conference on Systems Signals and Devices, pp. 1 - 5,June 2010

[73] D. A. Roark, A. J. O’Toole and H. Abdi, “Human Recognition of Familiar and Unfamiliar People in Naturalistic Video”, IEEE International Workshop on Analysis and Modeling of Faces and Gestures, pp. 36 - 41, October 2003

[74] Y. Mao and X. Huang, “Human Recognition Based On Head-Shoulder Moment Feature”, IEEE International Conference on Service Operations and Logistics, and Informatics, Volume 1, pp. 622 - 625, October 2008

[75] Y. R. Chen, C. M. Huang and L. C. Fu, “Upper body tracking for human-machine interaction with a moving camera”, IEEE International Conference on Intelligent Robots and Systems, pp. 1917 - 1922, October 2009

[76] L. Davis, V. Philomin and R. Duraiswami, “Tracking Humans from a Moving Platform”, IEEE International Conference on Pattern Recognition, Volume 4, pp. 171 - 178, 200

[77] C. S. Fahn and C. S. Lo, “A high-definition human face tracking system using the fusion of omni-directional and PTZ cameras mounted on a mobile robot”, pp. 6 - 11, June 2010

[78] H. Sugano, R. Miyamoto and Y. Nakamura, “Optimized parallel implementation of pedestrian tracking using HOG features on GPU”, IEEE Conference on Ph.D. Research in Microelectronics and Electronics, pp. 1 - 4, July 2010

[79] B. Bilgic, B. K. P. Horn and I. Masaki, “Fast human detection with cascaded ensembles on the GPU”, IEEE Intelligent Vehicles Symposium, pp. 325 - 332, June 2010

[80] S. Bauer, S. Kohler, K. Doll and U. Brunsmann, “FPGA-GPU architecture for kernel SVM pedestrian detection”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 61 - 68, June 2010

Appendix A

```
// This code project can be used to perform human detection using the Histogram of
//Oriented Gradients and the Haar-like feature detection algorithms.  Functions from the
//OpenCV library are used.
//

#define CV_NO_BACKWARD_COMPATIBILITY

#include "cv.h"
#include "cvaux.h"
#include "highgui.h"
#include "cxcore.h"

#include <iostream>
#include <cstdio>

#ifdef _EiC
#define WIN32
#endif

using namespace std;
using namespace cv;
```

```
//Define the Detect and Draw function for the HAAR detector
void detectAndDraw( Mat& img, CascadeClassifier& cascade, double scale, int x, int y,
int width);

//Set the destinaton path of the HAAR cascade
        String cascadeName = "C:\\Users\\Mouhammad\\Desktop\\2 Detectors –
        Copy\\TestingDetector\\cascade2890.xml";
//example:"C:\\Users\\Mouhammad\\Desktop\\2Detectors\\TestingDetector\\haarcascade_
fullbody.xml";


//Start the main function


int main( int argc, const char** argv )
{
        //Define the image file parameters for the HOG detector
        Mat img;
        FILE* imagef = 0;
        cv::Rect r;
        char _imagename[1024];
```

```
//Define the image file and cascade parameters for the HAAR detector
Mat frame, frameCopy, image;
const String scaleOpt = "--scale=";
size_t scaleOptLen = scaleOpt.length();
const String cascadeOpt = "--cascade=";
size_t cascadeOptLen = cascadeOpt.length();
String inputName;
CascadeClassifier cascade;
double scale = 1;

//Alert the user of any misuse of the command or if the image file is
not loaded
if( argc == 1 )
{
printf("Usage: 2Decs (<image_filename> | <image_list>.txt)\n");
return 0;
}

img = imread(argv[1]);
if( img.data )
{
strcpy(_imagename, argv[1]);
}
```

```
else
{
imagef = fopen(argv[1], "rt");
if(!imagef)
{
fprintf( stderr, "ERROR: the specified file could not be loaded\n");
return -1;
}
}

//Set the HOG decriptor with the default setting
cv::HOGDescriptor hog;
hog.setSVMDetector(HOGDescriptor::getDefaultPeopleDetector());

//Alert the user of any misuse of the command or if the image file is
not loaded
for( int i = 1; i < argc; i++ )
{

if( cascadeOpt.compare( 0, cascadeOptLen, argv[i], cascadeOptLen ) ==0)
cascadeName.assign( argv[i] + cascadeOptLen );
else if( scaleOpt.compare( 0, scaleOptLen, argv[i], scaleOptLen ) == 0)
{
```

```
if( !sscanf( argv[i] + scaleOpt.length(), "%lf", &scale ) || scale < 1)
scale = 1;
}
else if( argv[i][0] == '-' )
{
cerr << "WARNING: Unknown option %s" << argv[i] << endl;
}
else
inputName.assign( argv[i] );
}

//Alert the user if the cascade was not loaded
if( !cascade.load( cascadeName ) )
{
cerr << "ERROR: Could not load classifier cascade" << endl;
return -1;
}

//Get the input image and create an image window for the HAAR Detector
if( inputName.size() )
{
image = imread( inputName, 1 );
}
```

```
_cleanup_:
        for(;;)
        {
        char* imagename = _imagename;
        if(imagef)
        {
        if(!fgets(imagename, (int)sizeof(_imagename)-2, imagef))
        break;

        //Check the image file if it's valid or not for the HOG detector
        if(imagename[0] == '#')
        continue;
        int l = strlen(imagename);
        while(l > 0 && isspace(imagename[l-1]))
        --l;
        imagename[l] = '\0';
        }
        printf("%s:\n", imagename);
        if(!image.data)
        continue;
        fflush(stdout);
        vector<Rect> found, found_filtered;
```

```
//Set the time to measure the detection time
double t = (double)getTickCount();
//Start the HOG detector
int can = image.channels();
//
hog.detectMultiScale(image, found, 0, Size(8,8), Size(32,32), 1.05, 1);


//Calculate and display the detection time
t = (double)getTickCount() - t;
printf("tdetection time = %gms\n", t*1000./cv::getTickFrequency());
size_t i, j;


//Collect the detection rectangles
for( i = 0; i < found.size(); i++ )
{
Rect r = found[i];
for( j = 0; j < found.size(); j++ )
if( j != i && (r & found[j]) == r)
break;
if( j == found.size() )
found_filtered.push_back(r);
}
```

```
//Call the detectAndDraw function
if( !image.empty() )
{
detectAndDraw( image, cascade,  scale, r.x, r.y, r.width);
}

//Shrink the detection rectangles for the HOG detector since they
usually are a little larger in scale
for( i = 0; i < found_filtered.size(); i++ )
{
r = found_filtered[i];
cv::Point textpoint = r.tl();
r.x += cvRound(r.width*0.1);
r.width = cvRound(r.width*0.8);
r.y += cvRound(r.height*0.07);
r.height = cvRound(r.height*0.8);
rectangle(image, r.tl(), r.br(), cv::Scalar(0,255,0), 3);

CvFont font1;
cvInitFont( &font1, CV_FONT_VECTOR0, 0.4f, 0.4f, 0.0f,2 );
}
```

```
//Call the detectAndDraw function
if( !image.empty() )
{
detectAndDraw( image, cascade,  scale, r.x, r.y, r.width);
}

//Display the image with the HOG detection in a new window
imshow("HOG Detector", image);

//Wait until the user closes the HOG window
int c = waitKey(0) & 255;
if( c == 'q' || c == 'Q' || !imagef)
break;
}

//Close the file
if(imagef)
fclose(imagef);
//Destroy the HAAR window
cvDestroyWindow("HAAR Detector");
return 0;
}
```

```
//The detectAndDraw function takes the input image, the HAAR cascade
and the scale to detect the desired object and draw a bounding box
around it using the HAAR detection algorithm
void detectAndDraw( Mat& img,
CascadeClassifier& cascade,  double scale, int x, int y, int width)

{
int i = 0;
double t = 0;
vector<Rect> faces;
const static Scalar colors[] =  { CV_RGB(0,0,255),
CV_RGB(0,128,255),
CV_RGB(0,255,255),
CV_RGB(0,255,0),
CV_RGB(255,128,0),
CV_RGB(255,255,0),
CV_RGB(255,0,0),
CV_RGB(255,0,255)};
Mat gray, smallImg( cvRound (img.rows/scale),
cvRound(img.cols/scale), CV_8UC1 );
cvtColor( img, gray, CV_BGR2GRAY );
resize( gray, smallImg, smallImg.size(), 0, 0, INTER_LINEAR );
equalizeHist( smallImg, smallImg );
```

```
t = (double)cvGetTickCount();
cascade.detectMultiScale( smallImg, faces,
1.1, 2, 0
|CV_HAAR_SCALE_IMAGE
,
Size(30, 30) );
t = (double)cvGetTickCount() - t;
printf( "detection time = %g ms\n",
t/((double)cvGetTickFrequency()*1000.) );

for( vector<Rect>::const_iterator r = faces.begin();
r != faces.end(); r++, i++ )
{
Mat smallImgROI;
vector<Rect> nestedObjects;
Point center;
Scalar color = colors[i%8];
int radius;
center.x = cvRound((r->x + r->width*0.5)*scale);
center.y = cvRound((r->y + r->height*0.5)*scale);
radius = cvRound((r->width + r->height)*0.25*scale);
if (center.x>x && center.x < x+width)
circle( img, center, radius, color, 3, 8, 0 );
```

```
smallImgROI = smallImg(*r);
for( vector<Rect>::const_iterator nr = nestedObjects.begin();
nr!= nestedObjects.end(); nr++ )
{
center.x = cvRound((r->x + nr->x + nr->width*0.5)*scale);
center.y = cvRound((r->y + nr->y + nr->height*0.5)*scale);
radius = cvRound((nr->width + nr->height)*0.25*scale);
if (center.x>x && center.x < x+width)
circle( img, center, radius, color, 3, 8, 0 );
}
}
}
```