



HÖGSKOLAN
Dalarna

Motion Detection for Video Surveillance

Md. Junaedur Rahman

2008

Master Thesis
Computer Engineering
Reg No: E3651D



HÖGSKOLAN
Dalarna

MASTER THESIS
Computer Engineering

Programme Master in Computer Engineering specialization in Artificial Intelligence.	Reg. Number E3651D	Extent 30 ECTS
Name of student Md. Junaedur Rahman	Year-Month-Day 2008-11-05	
Supervisor Dr. Hasan Fleyeh	Examiner Professor Mark Dougherty	
Company/Department Deptement of Computer Science, Högskolan Dalarna		
Title Motion Detection for Video Surveillance		
Keywords Motion Detection, Video Surveillance, Background Subtraction, Shadow Detection		

Abstract

This thesis is related to the broad subject of automatic motion detection and analysis in video surveillance image sequence. Besides, proposing the new unique solution, some of the previous algorithms are evaluated, where some of the approaches are noticeably complementary sometimes. In real time surveillance, detecting and tracking multiple objects and monitoring their activities in both outdoor and indoor environment are challenging task for the video surveillance system. In presence of a good number of real time problems limits scope for this work since the beginning. The problems are namely, illumination changes, moving background and shadow detection.

An improved background subtraction method has been followed by foreground segmentation, data evaluation, shadow detection in the scene and finally the motion detection method. The algorithm is applied on to a number of practical problems to observe whether it leads us to the expected solution.

Several experiments are done under different challenging problem environment. Test result shows that under most of the problematic environment, the proposed algorithm shows the better quality result.

Acknowledgement

I would like to thank my creator, Almighty first who made me capable and responsible for those tasks I never had dreamed of. No word is enough to praise Him. I am happy to acknowledge the contributions of the following groups and individuals to the development of my thesis:

My Supervisor, Dr. Hasan Fleyeh, without his continuous motivation and inspiration throughout the program I may never had been pushed up to produce my best results. He is more than a supervisor to me who not only guided me through this difficult task with patience and perseverance but also shared a part of his valuable life time leading this project to a success.

My Professor, Mark Dougherty, whose recommendation I always keep on my bedside table for the continual motivation and recognition of my work. I still wonder how he managed me to have the understanding and solid grip on the complicated issues in accordance with a lot of fun.

My Teachers, whose important discussions and comments on certain issues had made the successful completion of the project, inevitable.

My Mother, whose love is my primary inspiration for this work.

My Father, to whom I respect most.

My sisters, whose support is worthwhile.

My all friends, well wishers and class mates, who had given me strong moral support and did not leave me in my bad times. I count on their comments on the final work and they helped me to refine my thesis and pointed out the weakness.

I want to thank you all from the bottom of my heart and expect you to continue supporting me like you did before.

Table of Contents

Chapter 1: Introduction.....	9
1.1 Motion detection:.....	10
1.2 Motion in real time environment: Problems.....	10
1.3 Video Surveillance:	11
1.3.1 Impact of video surveillance in commercial areas:	11
1.3.2 Video Surveillance Nowadays:.....	12
1.4 Overview of the real time surveillance system:.....	12
Chapter 2: Theoretical Background of Motion detection	13
2.1 Introduction	13
2.2 Gaussian Mixture Model:	15
2.3 Gaussian mixture background model	15
2.3.1 Model.....	15
2.4 Goals of object detection:	16
2.5 Background Subtraction:	16
2.6 Adaptive Mixture of Gaussians:	17
2.7 System Overview:.....	18
2.8 Method Illustration:	19
2.8.1 Preprocessing:.....	19
2.8.2 Recursive Techniques:.....	20
2.8.3 Foreground Detection	21
2.8.4 Data Validation:.....	22
2.9 Method recommendation for motion history update:	23
2.10 Live video analysis:	23
2.11 Suppression of False Detection:	25
2.12 Probabilistic Suppression of False Detection:	26
2.13 Silhouettes:	26
Chapter 3: Analysis of the previous works.....	28
3.1 Heikkila and Olli:	28
3.2 Pfinder.....	29
3.3 W4.....	29
3.4 LOTS	30
3.5 Halevy.....	31
3.6 Cutler	31
3.7 Wallower.....	32
3.8 Codebook-based Background Subtraction:	32
3.9 State-of-art:	33
3.10 Video Surveillance and Monitoring (VSAM):	34
3.11 Moving Target Classification and Tracking from Real-time Video:.....	34
3.12 KidRooms:.....	35
3.13 Grimson and Stauffer's work:	35
3.14 Discussion:.....	36
Chapter 4: Motion Detection and Shadow Elimination Methodology; Implementation.....	39
4.1 Working Environment:	39
4.2 Identify the image background:	39
4.3 Adaptive Gaussian Mixture Model.....	40
4.4 Background Subtraction Algorithm:	42
4.5 Foreground Segmentation:	43

4.6 Real time problems:	43
4.7 Proposed Solution:	44
4.7.1 Shadow:	45
4.7.2 Photometric color invariants	45
4.7.3 Geometric Properties of shadows	46
4.8 Division based brightness invariants:	47
4.8.1 Test Results:	48
4.8.2 Proposed Method of Shadow detection:	49
4.8.3 Proposed Method Analysis:	50
4.8.4 Improvement of the proposed algorithm	50
4.9 Final Algorithm of motion detection without shadows:	51
Chapter 5: Test results and discussion	53
5.1 Experiments in outdoor environment:	53
5.1.1 Busy Road:	53
5.1.2 Sunny day:	54
5.1.3 Traffic (Night):	55
5.1.4 Traffic (Day):	56
5.1.5 A Rainy Day:	56
5.1.6 A Rainy Day (Animated):	57
5.1.7 A Snowy Day:	58
5.1.8 Observations Outdoor:	59
5.2 Indoor Situation monitoring Experiments:	59
5.2.1 Indoor Room:	60
5.2.2 Large Hall (Indoor):	60
5.2.3 Observations Indoor:	61
5.3 Surveillance imagery from other sources:	61
5.3.1 Infrared:	62
5.3.2 Microscopic view:	62
5.3.3 Ultra Sonogram Image:	62
5.4 Observation of images from other sources:	62
5.5 Performance Evaluation:	63
5.5.1 Frame Rate per Second (FPS):	63
5.5.2 Mean and Standard Deviation:	66
5.5.3 FFT:	69
5.5.4 Comparison between different noise levels:	70
5.6 Discussion and recommendations:	70
5.6.1 Limitations:	70
Chapter 6: Conclusion and Future Work	72
6.1 Conclusion and recommendations	72
6.2 Future Works:	72
6.3 Some general open issues:	73
6.4 References:	74

List of Figures:

Name of the Figures	Pages
Figure 2.1: Three mixtures of Gaussians.	15
Figure 2.2: Flow diagram of generic background subtraction algorithm.	19
Figure 2.3: Silhouettes	27
Figure 4.1(a): Live image sequence	39
Figure 4.1(b): Subtracted background	39
Figure 4.2(a): The pixel value probability	41
Figure 4.2(b): The posteriori probabilities	41
Figure 4.3: The Background Subtraction algorithm designed in this work	42
Figure 4.4a,b,c,d: Live image sequence	43
Figure 4.5a,b: Live image sequence and effect of illumination change	44
Figure 4.6a,b: Live image sequence and Effect of moving shadows in foreground detection	44
Figure 4.7a,b,c: Live image Frame1, Frame2 and Live image Frame difference	45
Figure 4.8: Shadow lines definition	46
Figure 4.9: Input, c1, c2, c3, cc, L1, L2, L3, LL, R1', R2', R3', RR	48
Figure 4.9: Hue, Saturation, Value, HSV	49
Figure 4.10a,b,c: Live Image, Hue Image and extracted shadow	49
Figure 4.11a,b: Live Image Sequence and shadow detection in noise $b=70.5$	50
Figure 4.12a,b,c,d,e,f: Live image sequence and Histogram of Value Image;	51
Figure 4.13: Final work flow diagram of motion detection algorithm without shadows.	51
Figure 5.1a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p: Busy Road	53
Figure 5.2a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p: Sunny Day	54
Figure 5.3a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p: Traffic Night	55
Figure 5.4a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p: Traffic Day	56
Figure 5.5a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p: Rainy Day Live	56
Figure 5.6a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p: Rainy Day Animated	57
Figure 5.7a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p: Snowy Day	58
Figure 5.8a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p: Indoor Room	60
Figure 5.9a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p: Large Hall	61
Figure 5.10a,b,c,d: Infrared	62
Figure 5.11a,b,c,d: Microscope	62
Figure 5.12a,b,c,d: Ultra sonogram	62
Figure 5.13a,b,c,d,e,f,g: Performance evaluation Outdoor	63
Figure 5.14a,b: Indoor	65
Figure 5.15a,b,c: Other sources	65
Figure 5.16a,b: Performance comparison	66
Figure 5.17a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p: Mean and Standard Deviation	67
Figure 5.18a,b: Indoor	68
Figure 5.19a,b,c: Other sources	69
Figure 5.20a,b: FFT Day and FFT Night	69
Figure 5.21a,b: Noise level comparison	70

Chapter 1: Introduction

Human quest for an automatic detection system of everyday occurrence lead to the necessity of inventing an intelligent surveillance system which will make lives easier as well as enable us to compete with tomorrows technology and on the other hand it pushes us to analyze the challenge of the automated video surveillance scenarios harder in view of the advanced artificial intelligence.

Nowadays, it is seen that surveillance cameras are already prevalent in commercial establishments, with camera output being recorded to tapes that are either rewritten periodically or stored in video archives. To extract the maximum benefit from this recorded digital data, detect any moving object from the scene is needed without engaging any human eye to monitor things all the time. Real-time segmentation of moving regions in image sequences is a fundamental step in many vision systems. A typical method is background subtraction. Many background models have been introduced to deal with different problems. One of the successful solutions to these problems is to use a multi-color background model per pixel proposed by Grimson et al [1,2]. However, the method suffers from slow learning at the beginning, especially in busy environments. In addition, it can not distinguish between moving shadows and moving objects.

Image background and foreground are needed to be separated, processed and analyzed. The data found from it is then used further to detect motion. In this project work robust routines for accurately detecting and tracking moving objects have been developed and analyzed. The new method currently operates on video taken from a stationary camera. The traditional real time problems are taken under consideration including shadow interference while detecting motion. An improved shadow detection method is coordinated to handle the issue.

The method chosen to obtain the goal, the problems faced during the implementation and the primary idea of the solution is discussed in Part 1. A broad theoretical background has been presented in Part 2 with the relevant terminologies explained. In section 3 the previous works on this field have been discussed. The methods which have been analyzed and improved are explained which contains the primary motivation of this work. In section 4, the detail implementation technique of the work has been discussed. Section 5 briefly lists the significant outcomes achieved through the software demos that were performed both in indoor and outdoor at the busy run site and concludes in section 6 with plans for future research. The appendix contains the reference of published technical papers from certain research groups.

1.1 Motion detection:

Motion detection in consequent images is nothing but the detection of the moving object in the scene. In video surveillance, motion detection refers to the capability of the surveillance system to detect motion and capture the events. Motion detection is usually a software-based monitoring algorithm which will signal the surveillance camera to begin capturing the event when it detects motions. This is also called activity detection. An advanced motion detection surveillance system can analyze the type of motion to see if it warrants an alarm. In this project, a camera fixed to its base has been placed and is set as an observer at the outdoor for surveillance. Any small movement with a level of tolerance it picks is detected as motion.

Aside from the intrinsic usefulness of being able to segment video streams into moving and background components, detecting moving blobs provides a focus of attention for recognition, classification, and activity analysis, making these later processes more efficient since only “moving” pixels need be considered. There are three conventional approaches to moving object detection [3]: temporal differencing, background subtraction and optical flow. Temporal differencing is very adaptive to dynamic environments, but generally does a poor job of extracting all relevant feature pixels. Background subtraction provides the most complete feature data, but is extremely sensitive to dynamic scene changes due to lighting and extraneous events. Optical flow can be used to detect independently moving objects in the presence of camera motion; however, most optical flow computation methods are computationally complex, and cannot be applied to full-frame video streams in real-time without specialized hardware [3].

1.2 Motion in real time environment: Problems

Video motion detection is fundamental in many autonomous video surveillance strategies. However, in outdoor scenes where inconsistent lighting and unimportant, but distracting, background movement is present, it is a challenging problem. In real time environment where scene is not under control situation is much worse and noisy. Light may change anytime which cause system output less meaningful to deal with. Recent research has produced several background modeling techniques, based on image differencing, that exhibit real-time performance and high accuracy for certain classes of scene. The aim of this research work is to assess the performance of some of these background modeling techniques (namely, the Gaussian Mixture Model, temporal differencing, the Hybrid Detection, shadow detection and removal Algorithm) using video sequences of outdoor

scenes where the weather introduces unpredictable variations in both lighting and background movement. The results are analyzed and reported, with the aim of identifying suitable directions for enhancing the robustness of motion detection techniques for outdoor video surveillance systems. Motion in indoor and other situations are considered and analyzed as well.

1.3 Video Surveillance:

An appliance that enables embedded image capture capabilities that allows video images or extracted information to be compressed, stored or transmitted over communication networks or digital data link. Digital video surveillance systems are used for any type of monitoring. Broadly, video surveillance is the image sequences which are recorded to monitor the live activities of a particular scene. The importance of this digital evidence is given the first priority for any kind of occurrence. This digital information is recently become the field of interest to the researchers on the field of AI, Robotics, Forensic Science and other major fields of science.

1.3.1 Impact of video surveillance in commercial areas:

“What are you looking at?” — Graffiti by Banksy commenting on the neighboring surveillance camera in a concrete subway underpass near Hyde Park in London. The greatest impact of computer-enabled surveillance is the large number of organizations involved in surveillance operations.

The state and security services still have the most powerful surveillance systems, because they are enabled under the law. But today levels of state surveillance have increased, and using computers they are now able to draw together many different information sources to produce profiles of persons or groups in society.

Many large corporations now use various form of "passive" surveillance. This is primarily a means of monitoring the activities of staff and for controlling public relations. But some large corporations actively use various forms of surveillance to monitor the activities of activists and campaign groups who may impact their operations. Many companies trade in information lawfully, buying and selling it from other companies or local government agencies that collect it. This data is usually bought by companies who wish to use it for marketing or advertising purposes. Personal information is

obtained by many small groups and individuals. Some of this is for harmless purposes, but increasingly sensitive personal information is being obtained for criminal purposes, such as credit card and other types of fraud.

1.3.2 Video Surveillance Nowadays:

Modern surveillance cannot be totally avoided. However, non-state groups may employ surveillance techniques against an organization, and some precautions can reduce their success. Some states are also legally limited in how extensively they can conduct general surveillance of people they have no particular reason to suspect.

The constantly growing interest in the field of robotic vision is pushing the researchers hard to produce something which significantly fit in their requirement.

1.4 Overview of the real time surveillance system:

The real time video surveillance system presented in this project has been designed with robustness as the major design goal. Its main characteristic which supports the aim for robustness is the use of a two-stage multi-resolution approach along with the use of multiple cues. In order to perform the transition from motion detection to video content analysis a sequence of operations had to be maintained, namely:

- Adaptation of the ‘background update’ strategy
- Use of frame differencing features for object detection
- Use of event detection mechanisms
- Shadow detection
- Integration of modules to detect motion of the object.

Chapter 2: Theoretical Background of Motion detection

Background maintenance and subtraction is a common computer vision task. The usual pixel-level approach has been analyzed. First, some basic principles and requirements are extracted and then the contributions from the literature are summarized. Further, based on the presented principles, some standard theory and some recent results are analyzed. Firstly, algorithm which has used the parametric Gaussian mixture probability density is described. Recursive equations are used to constantly update the parameters and to select the appropriate number of components for each pixel. Then the algorithm which has used the old Grimson and Stauffer's work with an improvement is described. Finally, the method which was followed from their discussion has been showed and their results obtained have been analyzed.

2.1 Introduction

A static camera observing a scene is a common case of a surveillance system. Detecting intruding objects is an essential step in analyzing the scene. A usually applicable assumption is that the images of the scene without the intruding objects exhibit some regular behavior that can be well described by a statistical model [3]. If a statistical model of the scene has been revealed, an intruding object can be detected by spotting the parts of the image that don't fit the model. This process is usually known as "background subtraction".

Usually a simple bottom-up approach is applied and the scene model has a probability density function for each pixel separately [4]. A pixel from a new image is considered to be a background pixel if its new value is well described by its density function. For example, for a static scene the simplest model could be just an image of the scene without the intruding objects. The next step would be, for example, to estimate appropriate values for the variances of the pixel intensity levels from the image since the variances can vary from pixel to pixel. However, pixel values often have complex distributions and more elaborate models are needed [4].

The scene could change from time to time (sudden or slow illumination changes, static objects removed etc.) [4]. The model should be constantly updated to reflect the most current situation. The major problem for the background subtraction algorithms is how to automatically and efficiently update the model. Based on the extracted principles, analyzed and compared two efficient algorithms for the two models: Gaussian mixture and static background estimation. The Gaussian

mixture density function is a popular flexible probabilistic model. A Gaussian mixture was proposed for background subtraction in various studies. One of the most commonly used approaches for updating the Gaussian mixture model. A Gaussian mixture having a fixed number of components is constantly updated using a set of heuristic equations. Based on some additional approximations a set of theoretically supported but still very simple equations for updating the parameters of the Gaussian mixture is seen[4]. The important improvement compared to the previous approaches is that at almost no additional cost also the number of components of the mixture is constantly adapted for each pixel. By choosing the number of components for each pixel in an on-line procedure, the algorithm can automatically fully adapt to the scene.

Secondly, the simplest form of the reference image is a time-averaged background image. This method suffers from many problems and requires a training period absent of foreground objects. The motion of background objects after the training period and foreground objects motionless during the training period would be considered as permanent foreground objects. In addition, the approach cannot cope with gradual illumination changes in the scene. These problems lead to the requirement that any solution must constantly re-estimate the background model. Many adaptive background-modeling methods have been proposed to deal with these slowly-changing stationary signals. Friedman and Russell modeled each pixel in a camera scene by an adaptive parametric mixture model of three Gaussian distributions. The methods can cope well with the illumination changes; however, can not handle the problem of objects being introduced or removed from the scene. One solution is to use a multiple-color background model per pixel. Grimson et al. model can also lessen the effect of small repetitive motions; for example, moving vegetation like trees and bushes as well as small camera displacement.

In addition to Grimson et al. [1,2], many other authors have applied mixture models to model every pixel in camera scenes. Rowe and Blake applied the batch EM algorithm for off-line training in their virtual image plane. However, the model does not update with time and therefore leads to failure for external environments where the scene lighting changes with time.

Although a number of speed-up routines is presented, the approach was still of high computational complexity. The method which has been implemented and evaluated is based on Grimson et al.'s framework, the differences lie in the update equations, initialization method and the introduction of a shadow detection algorithm.

2.2 Gaussian Mixture Model:

Mixture of K Gaussians $\eta(\mu_i, \sigma_i, \omega_i)$ is a model in which independent variables are fractions of a total. Here η is the i-th Gaussian component with intensity mean μ_i and standard deviation σ_i . ω_i is the portion of the data accounted for by the i-th component [5]. It was expectation maximization (EM) algorithm that guarantees to converge to a local maximum in a search space. First it was convolved by Grimson and Stauffer's work (1999). In this way, the model copes also with multimodal background distributions; the number of modes (usually from 3 to 5). All weights ω_i are updated (updated and/or normalized) at every new frame [5]. At every new frame, some of the Gaussians "match" the current value (those at a distance $< 2.5 \sigma_i$) for them, μ_i and σ_i are updated by the running average [5]. The mixture of Gaussians actually models both the foreground and the background: now the question is how to pick only the distributions modeling the background. All distributions are ranked according to their ω_i / σ_i and the first ones chosen as "background" [5].

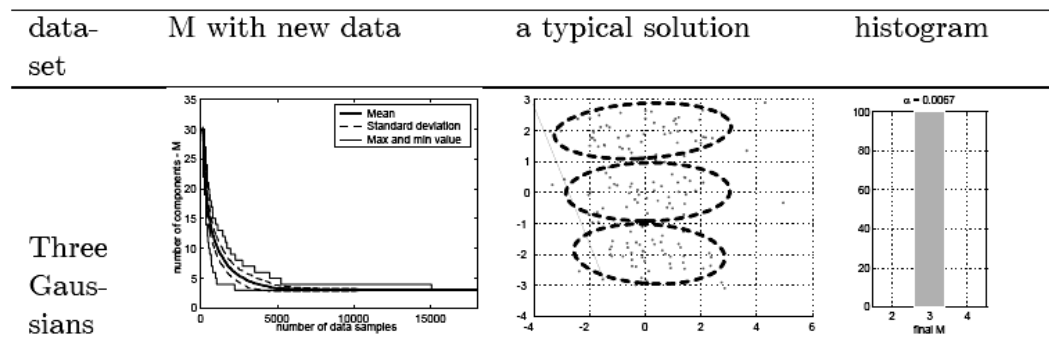


Figure 2.1: Three mixtures of Gaussians. Zivkovic 2003[6]

2.3 Gaussian mixture background model

Here, the Gaussian mixture model and its application for the background subtraction, is analyzed.

2.3.1 Model

A Gaussian mixture density with M components can be written as:

$$p(\vec{x}; \vec{\theta}) = \sum_{m=1}^M \pi_m N(\vec{x}; \vec{\mu}_m, C_m), \text{ with } \sum_{m=1}^M \pi_m = 1 \quad (2.1)$$

and $\vec{\theta} = \{\pi_1, \dots, \pi_m, \vec{\mu}_1, \dots, \vec{\mu}_m, C_1, \dots, C_M\}$. Where $\vec{\mu}_1, \dots, \vec{\mu}_m$ are the means and C_1, \dots, C_M are the covariance matrices describing the Gaussian distributions [6]. The mixing weights denoted by π_m are positive. Parameter estimates at time t will be denoted as $\hat{\vec{\theta}}^{(t)}$. The parameters are updated recursively according to the stochastic approximation procedure. Two rules, component generation and component deletion are added to adapt also the number of components and choose compact models for the data. The generation rule is inspired by the 'adaptive kernel' approach by Zivkovic [6]. The deletion rule is inspired by the recent results.

2.4 Goals of object detection:

The goals of the object tracking stage are to:

- ⌘ Determine when a new object enters the system's field of view, and initialize motion models for tracking that object.
- ⌘ Compute the correspondence between the foreground regions detected by the background subtraction and the objects currently being tracked [7].
- ⌘ Employ tracking algorithms to estimate the position of each object, and update the motion model used for tracking. The target is to model the overall motion of an object.

2.5 Background Subtraction:

As computer vision begins to address the visual interpretation of action applications such as surveillance and monitoring are becoming more relevant. Similarly, recent work in intelligent environments and perceptual user interfaces involve vision systems which interpret the pose or gesture of users in a known, indoor environment. In all of these situations the first fundamental problem encountered is the extraction of the image region corresponding to the object or persons in the room. Previous attempts at segmenting object from a known background have taken one of the three approaches mentioned previously. Most common is some form of background subtraction. For example, Grimson et al. uses statistical texture properties of the background observed over extended period of time to construct a model of the background, and use this model to decide which pixels in

an input image do not fall into the background class. The fundamental assumption of the algorithm is that the background is static in all respects: geometry, reflectance, and illumination.

The second class of approach is based upon image motion only presuming that the background is stationary or at most slowly varying, but that the object is moving.

In these methods no detailed model of the background is required. Of course, these methods are only appropriate for the direct interpretation of motion; if the object stops moving, no signal remains to be processed. This method also requires constant or slowly varying geometry, reflectance, and illumination.

The final approach, and the one most related to the technique presented is based upon geometry. Kanade, et al. employ special purpose multi-baseline stereo hardware to compute dense depth maps in real-time. Provided with a background disparity value, the algorithm can perform real-time depth segmentation or "z-keying". The only assumption of the algorithm is that the geometry of the background does not vary. However, the computational burden of computing dense, robust, real-time stereo maps requires great computational power.

2.6 Adaptive Mixture of Gaussians:

Each pixel is modeled separately by a mixture of K Gaussians

$$P(I_t) = \sum_{i=1}^k w_{i,t} \eta(I_t; \mu_{i,t}, \sigma_{i,t}^2) \quad (2.2)$$

where $K = 4$ and $K = 3 \dots 5$. It is assumed that $\sum_{i,t} = \sigma_{i,t}^2, I$ [5].

The background is updated, before the foreground is detected, as follows:

1. If I_t matches component i , i.e., I_t is within λ standard deviations of $\mu_{i,t}$, then the i th component is updated as follows:

$$w_{i,t} = w_{i,t-1} \quad (2.3)$$

$$\mu_{i,t} = (1 - \rho) \mu_{i,t-1} + \rho I_t \quad (2.4)$$

$$\sigma_{i,t}^2 = (1 - \rho) \sigma_{i,t-1}^2 + \rho (I_t - \mu_{i,t})^2 \quad (2.5)$$

where $\rho = \alpha \Pr(I_t | \mu_{i,t-1}, \sum_{i,t-1})$ [5].

2. Components which I_t don't match are updated by

$$w_{i,t} = (1 - \alpha)w_{i,t-1} \quad (2.6)$$

$$\mu_{i,t} = \mu_{i,t-1} - 1 \quad (2.7)$$

$$\sigma_{i,t}^2 = \sigma_{i,t-1}^2 \quad (2.8)$$

3. If I_t does not match any component, then the least likely component is replaced with a new one which has $\mu_{i,t} = I_t$, $\sum_{i,t}$ large, and $w_{i,t}$ low [5].

After the updates, the weights $w_{i,t}$ are renormalised.

The foreground is detected as follows. All components in the mixture are sorted into the order of decreasing $w_{i,t} / \left\| \sum_{i,t} \right\|$. So higher importance gets placed on components with the most evidence and lowest variance, which are assumed to be the background. Let

$$B = \arg \min_b \left(\frac{\sum_{i=1}^b w_{i,t}}{\sum_{i=1}^K w_{i,t}} \right) > T \quad (2.9)$$

for some threshold T . Then components 1..... B are assumed to be background. So if I_t does not match one of these components, the pixel is marked as foreground. Foreground pixels are then segmented into regions using connected component labelling. Detected regions are represented by their centroid.

2.7 System Overview:

Even though there exist a myriad of background subtraction algorithms in the literature [5], most of them follow a simple flow diagram shown in Figure 2.1.

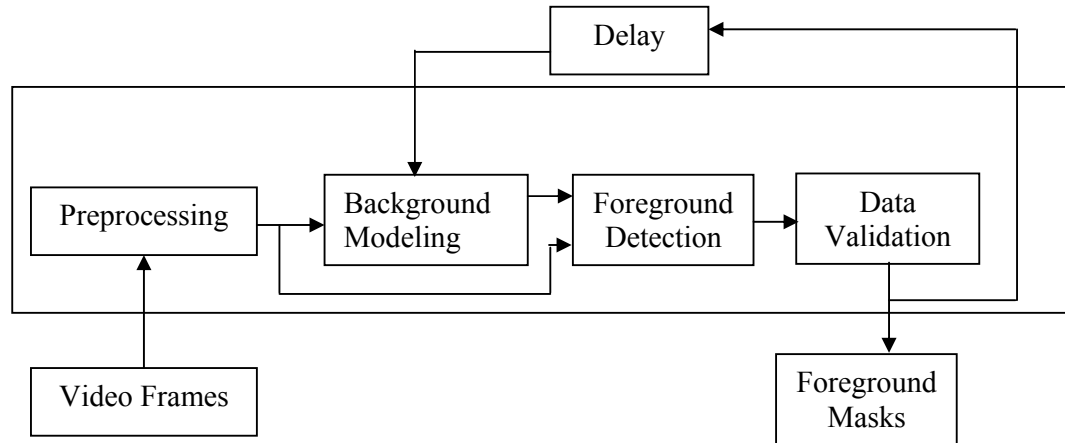


Figure 2.2: Flowdiagram of generic background subtraction algorithm.

2.8 Method Illustration:

The four major steps in a background subtraction algorithm are preprocessing, background modeling, foreground detection, and data validation. Preprocessing consists of a collection of simple image processing tasks that change the raw input video into a format that can be processed by subsequent steps. Background modeling uses the new video frame to calculate and update a background model. This background model provides a statistical description of the entire background scene. Foreground detection then identifies pixels in the video frame that cannot be adequately explained by the background model and outputs them as a binary candidate foreground mask. Finally, data validation examines the candidate mask, eliminates those pixels that do not correspond to actual moving objects, and outputs the final foreground mask. Domain knowledge and computationally-intensive vision algorithms are often used in data validation. Real-time processing is still feasible as these sophisticated algorithms are applied only on the small number of candidate foreground pixels. Many different approaches have been proposed for each of the four processing steps. Some of the representative ones in the following subsections has been reviewed.

2.8.1 Preprocessing:

In most computer vision systems, simple temporal and/or spatial smoothing are used in the early stage of processing to reduce camera noise. Smoothing can also be used to remove transient environmental noise such as rain and snow captured in outdoor camera. For real-time systems, frame-size and frame-rate reduction are commonly used to reduce the data processing rate. If the

camera is moving or multiple cameras are used at different locations, image registration between successive frames or among different cameras is needed before background modeling.

Another key issue in preprocessing is the data format used by the particular background subtraction algorithm. Most of the algorithms handle luminance intensity, which is one scalar value per each pixel. However, color image, in either RGB or HSV color space, is becoming more popular in the background subtraction literature. Some algorithms argue that color is better than luminance at identifying objects in low-contrast areas and suppressing shadow cast by moving objects. In addition to color, pixel-based image features such as spatial and temporal derivatives are sometimes used to incorporate edges and motion information. For example, intensity values and spatial derivatives can be combined to form a single state space for background tracking with the Kalman filter. Pless et al. combine both spatial and temporal derivatives to form a constant velocity background model for detecting speeding vehicles. The main drawback of adding color or derived features in background modeling is the extra complexity for model parameter estimation. The increase in complexity is often significant as most background modeling techniques maintain an independent model for each pixel.

2.8.2 Recursive Techniques:

To reduce computation burden and speed up the process recursive techniques are introduced in the process. Recursive techniques do not maintain a buffer for background estimation. Instead, they recursively update a single background model based on each input frame [5]. As a result, input frames from distant past could have an effect on the current background model. Compared with non-recursive techniques, recursive techniques require less storage, but any error in the background model can linger for a much longer period of time. Most schemes include exponential weighting to discount the past, and incorporate positive decision feedback to use only background pixels for updating.

Approximated median filter Due to the success of non-recursive median filtering, McFarlane and Schofield propose a simple recursive filter to estimate the median. This technique has also been used in background modeling for urban traffic monitoring. In this scheme, the running estimate of the median is incremented by one if the input pixel is larger than the estimate, and decreased by one if smaller. This estimate eventually converges to a value for which half of the input pixels are larger than and half are smaller than this value, that is, the median.

2.8.3 Foreground Detection

Foreground detection compares the input video frame with the background model, and identifies candidate foreground pixels from the input frame. Except for the non-parametric model and the MoG model, almost all the techniques use a single image as their background models [5]. The most commonly used approach for foreground detection is to check whether the input pixel is significantly different from the corresponding background estimate:

$$|I_t(x, y) - B_t(x, y)| > T \quad (2.10)$$

Where I is the image, B is the background and T is the threshold as usual. Another popular foreground detection scheme is to threshold based on the normalized statistics:

$$\frac{|I_t(x, y) - B_t(x, y) - \mu_d|}{\sigma_d} > T_s \quad (2.11)$$

where μ_d and σ_d are the mean and the standard deviation of $I_t(x, y) - B_t(x, y)$ for all spatial locations (x, y) . Most schemes determine the foreground threshold T or T_s experimentally.

Ideally, the threshold should be a function of the spatial location (x, y) . For example, the threshold should be smaller for regions with low contrast.

They use the relative difference rather than absolute difference to emphasize the contrast in dark areas such as shadow:

$$\frac{|I_t(x, y) - B_t(x, y)|}{B_t(x, y)} > T_c \quad (2.12)$$

Nevertheless, this technique cannot be used to enhance contrast in bright images such as an outdoor scene under heavy fog [5].

Another approach to introduce spatial variability is to use two thresholds with hysteresis. The basic idea is to first identify "strong" foreground pixels whose absolute differences with the background estimates exceeded a large threshold. Then, foreground regions are grown from strong foreground pixels by including neighboring pixels with absolute differences larger than a smaller threshold. The region growing can be performed by using a two-pass, connected-component grouping algorithm.

2.8.3.1 Updating the Background:

In the previous sections it was shown how to detect foreground regions given a recent history sample as a model of the background. This sample contains N intensity values taken over a window in time of size W . The kernel bandwidth estimation requires the entire sample to be consecutive in time, i.e., $N=W$ or sample $N/2$ pairs of consecutive intensity values over time W . This sample needs to be updated continuously to adapt to changes in the scene. The update is performed in a first-in first-out manner. That is, the oldest sample/pair is discarded and a new sample/pair is added to the model. The new sample is chosen randomly from each interval of length W/N frames. There are tradeoffs corresponding to the update decision regarding how fast to update and where to update in the image. The use of two different background models (short term and long term models) to overcome some of these tradeoffs is studied.

2.8.4 Data Validation:

Data validation is defined as the process of improving the candidate foreground mask based on information obtained from outside the background model. All the background models discussed so far have three main limitations: first, they ignore any correlation between neighboring pixels; second, the rate of adaption may not match the moving speed of the foreground objects; and third, non-stationary pixels from moving leaves or shadow cast by moving objects are easily mistaken as true foreground objects. The first problem typically results in small false-positive or false-negative regions distributed randomly across the candidate mask. The most common approach is to combine morphological filtering and connected component grouping to eliminate these regions. Applying morphological filtering on foreground masks eliminates isolated foreground pixels and merges nearby disconnected foreground regions [5]. Many applications assume that all moving objects of interest must be larger than a certain size which is partly wrong. Connected-component grouping can then be used to identify all connected foreground regions, and eliminates those that are too small to correspond to real moving objects.

When the background model adapts at a slower rate than the foreground scene, large areas of false foreground, commonly known as "ghosts", often occur [5]. If the background model adapts too fast, it will fail to identify the portion of a foreground object that has corrupted the background model. A simple approach to alleviate these problems is to use multiple background models running at different adaptation rates, and periodically cross-validate between different models to improve performance

[5]. Sophisticated vision techniques can also be used to validate foreground detection. If multiple cameras are available to capture the same scene at different angles, disparity information between cameras can be used to estimate depth. Depth information is useful as foreground objects are closer to the camera than background. But this is not the case for this work.

The moving-leaves problem can be addressed by using sophisticated background modeling techniques like MoG and applying morphological filtering for cleanup. On the other hand, suppressing moving shadow is much more problematic, especially for luminance-only video. A recent survey and comparison of many shadow suppression algorithms can be found in the later discussion.

2.9 Method recommendation for motion history update:

Many other algorithms, which have not been discussed here, assume that the background does not vary and hence can be captured apriori. This limits their usefulness in most practical applications. Very few of the papers describe their algorithms in sufficient detail to be able to easily reimplement them.

A significant number of the described algorithms use a simple Infinite Impulse Response (IIR) filter applied to each pixel independently to update the background and use thresholding to classify pixels into foreground/background. This is followed by some postprocessing to correct classification failures.

It was noted that the performance of the method was found to degrade if more than one secondary background was used. It was postulated that this is because it introduces a greater range of values that a pixel can take on without being marked as foreground. However, the adaptive mixture of Gaussians approach operates effectively with even more component models. From this it can be seen that using more models is beneficial only if by adding them the range (e.g., variance) of the individual components gets reduced such that the net range of background values actually decreases. It is the best suited of all for updating pixels without keeping any image buffer and offers less computational complexity.

2.10 Live video analysis:

Detection of moving objects in video streams is known to be a significant and difficult research problem. Aside from the intrinsic usefulness of being able to segment video streams into moving

and background components, detecting moving blobs provides a focus of attention for recognition, classification, and activity analysis, making these later processes more efficient since only “moving” pixels need be considered.

There are three conventional approaches to moving object detection:

- 1) **Temporal differencing:** Temporal differencing is very adaptive to dynamic environments, but generally does a poor job of extracting all relevant feature pixels.
- 2) **Background subtraction:** Background subtraction provides the most complete feature data, but is extremely sensitive to dynamic scene changes due to lighting and extraneous events.
- 3) **Optical flow:** Optical flow can be used to detect independently moving objects in the presence of camera motion; however, most optical flow computation methods are computationally complex, and cannot be applied to full-frame video streams in real-time without specialized hardware.

A robust detection system should be able to recognize when objects have stopped and even disambiguate overlapping objects — functions usually not possible with traditional motion detection algorithms. An important aspect of this work derives from the observation that legitimately moving objects in a scene tend to cause much faster transitions than changes due to lighting, meteorological, and diurnal effects.

A huge amount of video material is produced daily: television, movies, surveillance cameras etc. As the amount of the available video content grows, higher demands are placed on video analysis and video content management. A general review of the image based content indexing is given. The video indexing is reviewed for example in Brunelli et al.

Probably the most frequently solved problem when videos are analyzed is segmenting a foreground object from its background in an image. After some regions in an image are detected as the foreground objects, some features are extracted that describe the segmented regions. These features together with the domain knowledge are often enough to extract the needed high-level semantics from the video material. In this work, two automatic systems for video analysis and indexing is presented. In both systems the segmentation of the foreground objects is the basic processing step. The extracted features are then used to solve the problem.

The first system (described in the test result section) is a traffic video analysis system. The foreground objects that need to be detected are the vehicles on a highway. Usually, there is a huge

gap (“semantic gap”) between the low-level features extracted from the foreground objects and the high-level concepts [4]. However, for this domain it was possible to manually map the extracted features to the events that need to be detected (high-level concepts) in a simple way. The second system analyzes videos of tennis games (described in the test result section). It is difficult to manually generate the mapping from the features to the high-level concepts. Therefore the learning capability of Hidden Markov Models (HMMs) is exploited to extract high-level semantics from the raw video data automatically [4].

Although very specific, the two applications have many elements that are important for any surveillance/ monitoring system.

2.11 Suppression of False Detection:

There are still a lot of problems connected to this tricky implementation of motion detection. Clearly, a motion detection camera which is generally used nowadays cannot be used outdoors, or even the slightest change of sunlight may fire off the alarm [8]. Their most common use is during night time when there are little natural changes taking place that could set off false alarms. Still, today’s technology isn’t superior enough to make the difference between, say, your dog, running around the living room in the middle of the night, or a thief. Both will probably start the alarm. Therefore, it’s a good choice to keep the pets away from the monitored vicinity during the time motion detection is on.

Another constraint, or rather a fact anybody will probably want to avoid, is aiming the motion detection camera towards a window [8]. Even during night time, a window viewing towards the street can cause a lot of bogus alarms as lighting outside may change, the window may mirror light in the lens and so forth.

In outdoor environments with fluctuating backgrounds, there are two sources of false detections. First, there are false detections due to random noise which should be homogeneous over the entire image. Second, there is false detection due to small movements in the scene background that are not represented in the background model. This can occur, for example, if a tree branch moves further than it did during model generation. Also small camera displacements due to wind load are common in outdoor surveillance and cause many false detections. This kind of false detection is usually

spatially clustered in the image and it is not easy to eliminate using morphology or noise filtering because these operations might also affect small and/or occluded targets.

By thresholding background for detected pixels much false detection can be eliminated due to small motions. Unfortunately, some true detection can also be eliminated by this process, since some true detected pixels might be accidentally similar to the background of some nearby pixel. This happens more often on gray level images. To avoid losing such true detections, the constraint is added that the whole detected foreground object must have moved from a nearby location, and not only some of its pixels.

2.12 Probabilistic Suppression of False Detection:

The second stage of detection aims to suppress the false detections due to small and un-modeled movements in the scene background. If some part of the background (a tree branch for example) moves to occupy a new pixel, but it was not part of the model for that pixel, then it will be detected as a foreground object [8]. However, this object will have a high probability to be a part of the background distribution at its original pixel [8]. Assuming that only a small displacement can occur between consecutive frames, it is decided if a background object motion has caused a false detection by considering the background distributions in a small neighborhood of the detection.

Let x_t be the observed value of a pixel, x , detected as a foreground pixel by the first stage of the background subtraction at time t . By thresholding R_w for detected pixels much false detections can be eliminated due to small motions in the background. The constraint is added that the whole detected foreground object must have moved from a nearby location, and not only some of its pixels [9]. The component displacement probability is defined, PC , to be the probability that a detected connected component C has been displaced from a nearby location.

For a connected component corresponding to a real target, the probability that this component has displaced from the background will be very small.

2.13 Silhouettes:

Generally, silhouettes are dark images outlined against a lighter background. In this work, it is used for the blob analysis. Every detected moving object has a clear view in the light background of an

image. Extracting silhouettes technique is even quite useful for threshold analysis of consequent image sequences. It is especially useful to detect moving objects in a real time scene.

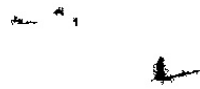


Figure 2.3: Silhouettes

Visual interpretation of people and their movements is an important issue in many applications, such as surveillance systems and virtual reality interfaces. The ability to find and track people is therefore an important visual problem. The problem is more difficult when there are small groups of people moving together or interacting with each other. In those cases individual people are not visually isolated, but are partially or totally occluded by other people. Silhouette based extraction of the moving objects thus come handy for the detection of moving group of people or a single person individually.

Chapter 3: Analysis of the previous works

Not using video surveillance to its full potential as a real-time threat detection system is unfortunate because video is an excellent tool in the fight to protect critical infrastructure. Most threatening activities begin with a prelude of hostile intelligence gathering - adversaries will often “case the joint” for a period of weeks or months before an attack. Appropriate video-based counter-measures can be used to detect these hostile patterns of activity. Furthermore, most hostile attacks begin with a perimeter breach, providing early opportunities for detection and interdiction. Again, video surveillance is an excellent tool to detect (in real-time) the nature and composition of a threat, its pattern of attack, whether it is a main force or merely a diversion, and monitor the progress of an attack and the effect of counter-measures. People are trying to make use of it over the last decade. Here few of the motion tracking strategies have been discussed which are being popular once and helped others ideologically to get things better. In this section, the glimpse of the total work done so far in this field have been put together and discussed the methods which have been followed primarily.

3.1 Heikkila and Olli:

In their work a pixel is marked as foreground if

$$|I_t - B_t| > \tau \quad (3.1)$$

where τ is a "predefined" threshold. The thresholding is followed by closing with a 3 X 3 kernel and the discarding of small regions.

The background update is,

$$|B_{t+1} = \alpha I_t + (1 - \alpha) B_t| \quad (3.2)$$

where B is the background image, I is the intensity and α , learning rate, is kept small to prevent artificial "tails" forming behind moving objects [23].

Two background corrections are applied:

1. If a pixel is marked as foreground for more than m of the last M frames, then the background is updated as $B_{t+1} = I_t$. This correction is designed to compensate for sudden illumination changes and the appearance of static new objects [23].
2. If a pixel changes state from foreground to background frequently, it is masked out from inclusion in the foreground. This is designed to compensate for fluctuating illumination, such as swinging branches [23].

3.2 Pfinder

Pfinder uses a simple scheme, where background pixels are modeled by a single value, updated by

$$B_t = (1 - \alpha)B_{t-1} + \alpha I_t \quad (3.3)$$

and foreground pixels are explicitly modeled by a mean and covariance, which are updated recursively [23]. It requires an empty scene at start-up.

It is a real-time system for tracking a person which uses a multi-class statistical model of color and shape to segment a person from a background scene. It finds and tracks people's head and hands under a wide range of viewing condition. There is a general purpose system for moving object detection and event recognition where moving objects are detected using change detection and tracked using first-order prediction and nearest neighbor matching [23]. Events are recognized by applying predicates to a graph formed by linking corresponding objects in successive frames.

3.3 W4

A pixel is marked as foreground if

$$|M - I_t| > D \text{ or } |N - I_t| > D \quad (3.4)$$

where the (per pixel) parameters M , N , and D represent the minimum, maximum, and largest interframe absolute difference observable in the background scene [23]. These parameters are initially estimated from the first few seconds of video and are periodically updated for those parts of the scene not containing foreground objects.

The resulting foreground "image" is eroded to eliminate 1-pixel thick noise, then connected component labelled and small regions rejected. Finally, the remaining regions are dilated and then eroded [23].

3.4 LOTS

Three background models are simultaneously kept, a primary, a secondary, and an old background [23]. They are updated as follows:

1. The primary background is updated as

$$B_{t+1} = \alpha I_t + (1 - \alpha) B_t \quad (3.5)$$

if the pixel is not marked as foreground, and is updated as

$$B_{t+1} = \beta I_t + (1 - \beta) B_t \quad (3.6)$$

if the pixel is marked as foreground. In the above, α was selected from within the range [0.0000610351 0.25], with the default value $\alpha = 0.0078125$, and $\beta = 0.25 \alpha$ [23].

2. The secondary background is updated as

$$B_{t+1} = \alpha I_t + (1 - \alpha) B_t \quad (3.7)$$

at pixels where the incoming image is not significantly different from the current value of the secondary background, where α is as for the primary background [23]. At pixels where there is a significant difference, the secondary background is updated by

$$B_{t+1} = I_t \quad (3.8)$$

3. The old background is a copy of the incoming image from 9000 to 18000 frames ago [23].

Foreground detection is based on adaptive thresholding with hysteresis, with spatially varying thresholds. Several corrections are applied:

1. Small foreground regions are rejected.
2. The number of pixels above threshold in the current frame is compared to the number in the previous frame. A significant change is interpreted as a rapid lighting change. In response the global threshold is temporarily increased.

3. The pixel values in each foreground region are compared to those in the corresponding parts of the primary and secondary backgrounds, after scaling to match the mean intensity. These eliminate artifacts due to local lighting changes and stationary foreground objects, respectively.

3.5 Halevy

The background is updated by

$$B_{t+1} = \alpha S(I_t) + (1 - \alpha)B_t \quad (3.9)$$

at all pixels, where $S(I_t)$ is a smoothed version of I_t . Foreground pixels are identified by tracking the maxima of $S(I_t - B_t)$, as opposed to thresholding. They use $\alpha = [0.3 \dots 0.5]$ and rely on the streaking effect to help in determining correspondence between frames [23].

They also note that $(1 - \alpha)^t < 0.1$ gives an indication of the number of frames t needed for the background to settle down after initialisation.

3.6 Cutler

Colour images are used because it is claimed to give better segmentation than monochrome, especially in low contrast areas, such as objects in dark shadows²¹.

The background estimate is defined to be the temporal median of the last N frames, with typical values of N ranging from 50 to 200.

Pixels are marked as foreground if

$$\sum_{C \in R, G, B} |I_t(C) - B_t(C)| > K\sigma \quad (3.10)$$

where σ is an offline generated estimate of the noise standard deviation, and K is an a priori selected constant (typically 10) [23].

This method also uses template matching to help in selecting candidate matches.

3.7 Wallower

Two auto-regressive background models are used:

$$B_t = -\sum_{k=1}^p a_k B_{t-k} \quad (3.11)$$

$$\hat{I}_t = -\sum_{k=1}^p a_k I_{t-k} \quad (3.12)$$

along with a background threshold

$$\varepsilon(e_t^2) = \varepsilon(B_t^2) + \sum_{k=1}^p a_k \varepsilon(B_t B_{t-k}) \quad (3.13)$$

$$\tau = 4\sqrt{\varepsilon(e_t^2)} \quad (3.14)$$

Pixels are marked as background if

$$|I_t - B_t| < \tau \text{ and } \left| I_t - \hat{I}_t \right| < \tau \quad (3.15)$$

The coefficients a_k are updated each frame time from the sample covariances of the observed background values. In the implementation, the last 50 values are used to estimate 30 parameters [23].

If more than 70% of the image is classified as foreground, the model is abandoned and replaced with a "back-up" model.

3.8 Codebook-based Background Subtraction:

The codebook BGS algorithm adopts a quantization/clustering technique, motivated by Kohonen, to construct a background model from long observation sequences [24]. For each pixel, it builds a codebook consisting of one or more code words. Samples at each pixel are clustered into the set of code words based on a color distortion metric together with a brightness ratio. Not all pixels have the same number of code words. The clusters represented by code words do not necessarily correspond to single Gaussian or other parametric distribution. Even if the distribution at a pixel were a single normal, there could be several code words for that pixel. The background is encoded

on a pixel by pixel basis. Thus a pixel is represented by a codebook which consists of one or multiple code words [24].

Detection involves testing the difference of the current image from the background model with respect to color and brightness differences. Unlike MOG or the kernel methods, the codebook method does not involve floating point calculation of probabilities which can be costly. Indeed, the probability estimate is dominated by the nearby training samples [24]. The CB method simply computes the distance of the sample from the nearest rescaled cluster mean. This is very fast and shows little difference in detection compared with the probability estimate. If an incoming pixel meets two conditions, it is classified as background - (1) The color distortion to some codeword is less than the detection threshold, and (2) its brightness lies within the brightness range of that codeword. Otherwise, it is classified as foreground [24]. To cope with the problem of illumination changes such as shading and highlights, the CB method does not use RGB values directly. Brightness is often the largest source of variation, not intrinsic color. Physically these are different as well. The CB method calculates a brightness difference (a ratio of RGB absolute values) and a color difference which rescales codeword RGB values to the brightness of the current, tested pixel.

3.9 State-of-art:

Organizations often spend millions of dollars on video surveillance infrastructure consisting of hundreds or thousands of cameras. These camera feeds are usually backhauled to a central monitoring location where some are recorded for a period of time on local view storage media, and some are displayed in real-time to one or more security personnel on a bank of video monitors [26].

No matter how highly trained or how dedicated a human observer, it is impossible to provide full attention to more than one or two things at a time; and even then, only for a few minutes at a time. A vast majority of surveillance video is permanently lost without any useful intelligence being gained. The situation is analogous to an animal with hundreds of eyes, but no brain to process the information [26]. This is where people stuck often with hundreds of camera for surveillance with standing right in front of several monitors. Within a moment the person getting lose of control of the whole scene. If someone stays behind his back he might have a funny feeling about this which is if the monitors are compared to the portraits and the people watching it are its admirer then the situation can be compared to an art museum. That's why it is called "The-State-of-Art" [26].

3.10 Video Surveillance and Monitoring (VSAM):

The Robotics Institute at Carnegie Mellon University (CMU) and the Sarnoff Corporation developed a system for autonomous Video Surveillance and Monitoring. The technical approach uses multiple, cooperative video sensors to provide continuous coverage of people and vehicles in a cluttered environment. This final report presents an overview of the system, and of the technical accomplishments that have been achieved.

Keeping track of people, vehicles, and their interactions in an urban or battlefield environment is a difficult task. The role of VSAM video understanding technology in achieving this goal is to automatically “parse” people and vehicles from raw video, determine their geo-locations, and insert them into dynamic scene visualization [9]. Robust routines for detecting and tracking moving objects have been developed. Detected objects are classified into semantic categories such as human, human group, car, and truck using shape and color analysis, and these labels are used to improve tracking using temporal consistency constraints. Further classification of human activity, such as walking and running, has also been achieved. Geo-locations of labeled entities are determined from their image coordinates using either wide-baseline stereo from two or more overlapping camera views, or intersection of viewing rays with a terrain model from monocular views. These computed locations feed into a higher level tracking module that tasks multiple sensors with variable pan, tilt and zoom to cooperatively and continuously track an object through the scene. All resulting object hypotheses from all sensors are transmitted as symbolic data packets back to a central operator control unit, where they are displayed on a graphical user interface to give a broad overview of scene activities. These technologies have been demonstrated through a series of yearly demos, using a test-bed system developed on the urban campus of CMU [9].

3.11 Moving Target Classification and Tracking from Real-time Video:

This method was introduced by Alan J. Lipton, Hironobu Fujiyoshi and Raju S. Patil. Historically, target classification has been performed on single images or static imagery. More recently, however, video streams have been exploited for target detection. Many methods like these, are computationally expensive and are inapplicable to real-time applications, or require specialized hardware to operate in the real-time domain. However, methods such as Pfister and Beymer et al. are designed to extract targets in real-time. The philosophy behind these techniques is the segmentation of an image, or video stream, into object vs. non object regions. This is based on

matching regions of interest to reasonably detailed target models. Another requirement of these systems is, in general, to have a reasonably large number of pixels on target. For both of these reasons, these methods would, by themselves, be inadequate in a general outdoor surveillance system, as there are many different types of targets which could be important, and it is often not possible to obtain a large number of pixels on target. A better approach is one in which classification is based on simple rules which are largely independent of appearance or 3D models. Consequently, the classification metric which is explored in this work is based purely on a target's shape, and not on its image content. Furthermore, the temporal component of video allows a temporal consistency constraint to be used in the classification approach. Multiple hypotheses of a target's classification can be maintained over time until the system is confident that it can accurately classify the target. This allows the system to disambiguate targets in the case of occlusions or background clutter.

A classification metric is applied the targets with a temporal consistency constraint to classify them into three categories: human, vehicle or background clutter. Once classified, targets are tracked by a combination of temporal differencing and template matching. The resulting system robustly identifies targets of interest, rejects background clutter, and continually tracks over large distances and periods of time despite occlusions, appearance changes and cessation of target motion. This study presents a much simpler method based on a combination of temporal differencing and image template matching which achieves highly satisfactory tracking performance in the presence of clutter and enables good classification. Hence the use of Kalman filtering or other probabilistic approaches is avoided.

3.12 KidRooms:

It is a tracking system based on closed-world regions". These are regions of space and time in which the specific context of what is in the regions is assumed to be known. These regions are tracked in real- time domains where object motions are not smooth or rigid, and where multiple objects are interacting. Bregler uses many levels of representation based on mixture models, EM, and recursive Kalman and Markov estimation to learn and recognize human dynamics [24].

3.13 Grimson and Stauffer's work:

The total out line of their work is summarized below:

- Model each background pixel by a mixture of K Gaussian distribution.

- The weight parameter of the mixture represents the time proportion.
- Static/Dynamic color object measure is termed as ‘fitness’.
- To adapt to changes in illumination an update scheme was applied.
- Every new pixel value is checked against the existing model.
- The first matched model will be updated.
- If no match found then a new Gaussian component will be added with the mean at that point, a large co-variance matrix and a small value of weighting parameter.

As it is evidence in their papers, Grimson et al.’s [1,2] tracker can not identify moving shadows from the objects casting them. The reason behind this is that no heuristic exists to label Gaussian components as moving shadows. One solution is to use a chromatic color space representation which reduces susceptibility [2]. As many color spaces can separate chromatic and illumination components, maintaining a chromatic model regardless of the brightness can lead to an unstable model especially for very bright or dark objects. This conversion also requires computational resources particularly in large images. The idea of preserving intensity components and saving computational costs lead us back to the RGB space. As the requirement to identify moving shadows, a color model is needed to be considered that can separate chromatic and brightness components. It should be compatible and make use of the mixture model. Moreover, the algorithm can not solve the three major problems discussed so far, namely sudden illumination change, moving background modeling and shadow detection. An improved motion detection strategy which can handle these issues successfully is proposed in this work.

3.14 Discussion:

Motion is a particularly important cue for computer vision. Indeed, for many applications, the simple fact that something is moving makes it of interest and anything else can be ignored. In such cases, it is common for moving objects to be referred to as the foreground and stationary objects as the background. A classic example of it is automatic traffic flow analysis in which motion is used to differentiate between vehicles (the foreground) and the roadway (the background). Higher-level processing could then be employed to categorize the vehicles as cars, motorcycles, buses, or trucks. Such a system might be used for determining patterns of traffic flow, or it could be adapted to automatically identify traffic law violations. Other applications where motion is important include gesture tracking, person tracking, model-based video coding and content-based video retrieval. In practice, the need to segment moving objects from a static background is so common; it has spawned a niche area of research where it is known as background subtraction, background

segmentation, or background modeling. As a priori knowledge of a scene's background does not often exist, the key for any background segmentation algorithm is how to learn and model it. The simplest approach involves calculating an average background frame whilst no moving objects are present. Subsequently, when objects enter the scene, they will cause the current frame to diverge from the background frame and their presence can be easily detected by thresholding the difference between them. However, any background or illumination change will severely and indefinitely degrade the accuracy of the algorithm. Therefore, practical implementations must continuously update the background frame to incorporate any permanent scene changes. Furthermore, assuming that the background is perfectly stationary is also flawed. For instance, a tree branch waving in the wind moves but is typically not important and so should be incorporated into the background model.

A single average background frame is clearly incapable of correctly modeling pseudo stationary backgrounds. In practice, the stationary assumption is often retained and subsequent processing is used to eliminate errors. Background segmentation is but one component of a potentially very complex computer vision system. Therefore, in addition to being accurate, a successful technique must consume as few processor cycles and as little memory as possible. An algorithm that segments perfectly but is very computationally complex is useless because insufficient processing power will remain to do anything useful with its results. Of the techniques, the two most promising were random updating and slope limited updating. Random updating replaces background pixels by the corresponding pixels in the current frame according to a pseudorandom sequence. As no reference is made to what data the pixels actually contain, errors in the background frame will occur. However, the errors are isolated and can be reduced using conventional morphological operations. In contrast, slope limited updating only adjusts the background frame when it differs substantially from the current frame and even then only by small amounts. Remarkably, given the level of computer technology at the time, they further showed that by using these techniques, it was possible to distinguish between vehicles and the roadway in real time. More recently, Chien et al. have revisited background subtraction. They surmised that the longer a pixel remained roughly constant, the more likely it is that it belongs to the background. Pixels are classified as stationary when the amount by which they change between consecutive frames falls below a threshold. Once a pixel has remained stationary for a sufficient number of frames, it is copied into the background frame. Although the above algorithms succeed in learning and refining the background frame, none of them is capable of handling pseudo stationary backgrounds. Stauffer and Grimson recognized that these kinds of backgrounds are inherently multimodal and hence they developed a technique which models each pixel by a mixture of Gaussians. However, if no match is found, then the minimum weighted Gaussian is replaced by a new one with the incoming pixel as its mean and a high initial variance.

Gaussians that are matched more frequently are nearby often occurring pixels and hence they are more likely to model the background. This algorithm has since been updated to suppress shadows and to improve its learning rate. A less obvious advantage of Stauffer and Grimson's technique is its ability to rapidly adapt to transient background changes. For example, if an object enters the scene and stops moving, it will eventually be incorporated into the background model. If it then moves again, the system should recognize the original background as corresponding Gaussians should still remain in the mixture. However, maintaining these mixtures for every pixel is an enormous computational burden and results in low frame rates when compared to the previous approaches. The algorithm of P. KaewTraKulPong and R. Bowden which is most similar to that of Stauffer and Grimson but with a substantially lower computational complexity is used. It will be shown that it has the capability of processing 320×240 video in real time on modest hardware.

Moreover, it involves calculating a reference image, subtracting each new frame from this image and thresholding the result. The result is a binary segmentation of the image which highlights region for non-stationary objects.

Chapter 4: Motion Detection and Shadow Elimination Methodology; Implementation

4.1 Working Environment:

The environment chosen for this thesis is visual C++. Free resource of opencv is used. The software developed “Cutting Horse” to test and analyze the images and the video sequences. Intel centrino M processor of 1.7 GHz is used on Windows XP environment. For the video surveillance a simple webcam is used. In practice a significant amount of noise is present in the un-sourced captured imagery which is good for the testing.

4.2 Identify the image background:

Background subtraction involves calculating a reference image, subtracting each new frame from this image and thresholding the result. What results is a binary segmentation of the image which highlights regions of non-stationary objects. The simplest form of the reference image is a time-averaged background image. This method suffers from many problems and requires a training period absent of foreground objects. The motion of background objects after the training period and foreground objects motionless during the training period would be considered as permanent foreground objects. In addition, the approach cannot cope with gradual illumination changes in the scene. These problems lead to the requirement that any solution must constantly re-estimate the background model. Many adaptive background-modeling methods have been proposed to deal with these slowly-changing stationary signals. Friedman and Russell modeled each pixel in a camera scene by an adaptive parametric mixture model of three Gaussian distributions.



Figure 4.1(a): Live image sequence



Figure 4.1(b): Subtracted background

In addition to Grimson et al, many other authors have applied mixture models to model every pixel in the camera scenes. The authors introduce a method to model each background pixel by a mixture of K Gaussian distributions (K is a small number from 3 to 5). Different Gaussians are assumed to represent different colors. The weight parameters of the mixture represent the time proportions that those colors stay in the scene. Unlike Friedman et al.'s work, the background components are determined by assuming that the background contains B highest probable colors. The probable background colors are the ones which stay longer and more static. Static single-color objects trend to form tight clusters in the color space while moving ones form widen clusters due to different reflecting surfaces during the movement. The measure of this was called the fitness value in their papers. To allow the model to adapt to changes in illumination and run in real-time, an update scheme was applied. It is based upon selective updating. Every new pixel value is checked against existing model components in order of fitness. The first matched model component will be updated. If it finds no match, a new Gaussian component will be added with the mean at that point and a large covariance matrix and a small value of weighting parameter.

4.3 Adaptive Gaussian Mixture Model

Each pixel in the scene is modeled by a mixture of K Gaussian distributions [5]. The probability that a certain pixel has a value of X_N at time N can be written as,

$$p(X_N) = \sum_{j=1}^K w_j \eta(X_N; \theta_j) \quad (4.1)$$

where w_j is the weight parameter of the k^{th} Gaussian component. $(x; \theta_k)$ is the Normal distribution of k^{th} component represented by

$$\eta(X; \theta_k) = \eta(X_N; \theta_j) = \frac{1}{(2\pi)^{\frac{D}{2}} \left| \sum_k \right|^{\frac{1}{2}}} e^{\frac{1}{2} (x - \mu_k)^T \sum_k^{-1} (x - \mu_k)} \quad (4.2)$$

Where μ_k is the mean and $\sum_k = \sigma^2 I$ is the covariance of the k^{th} component.

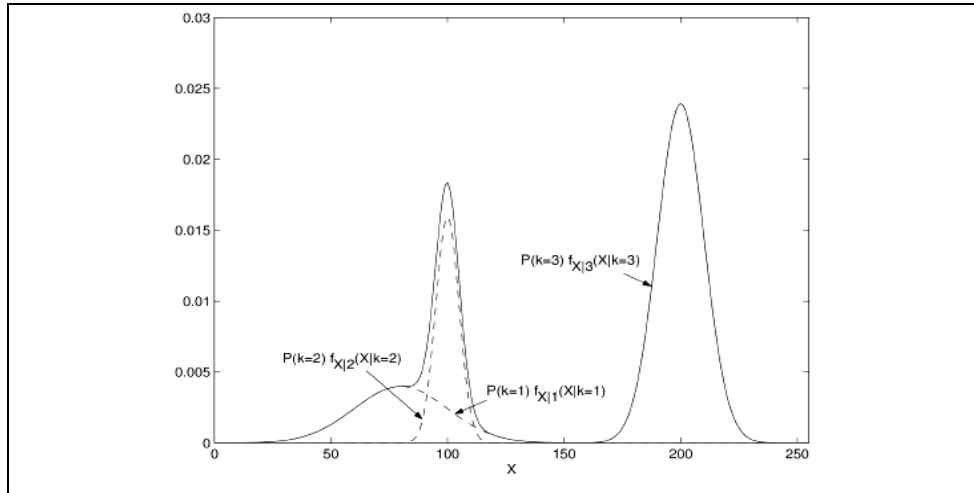
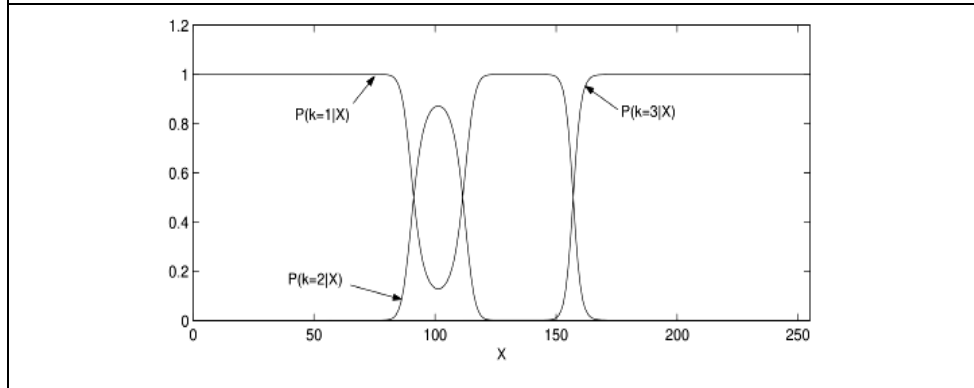


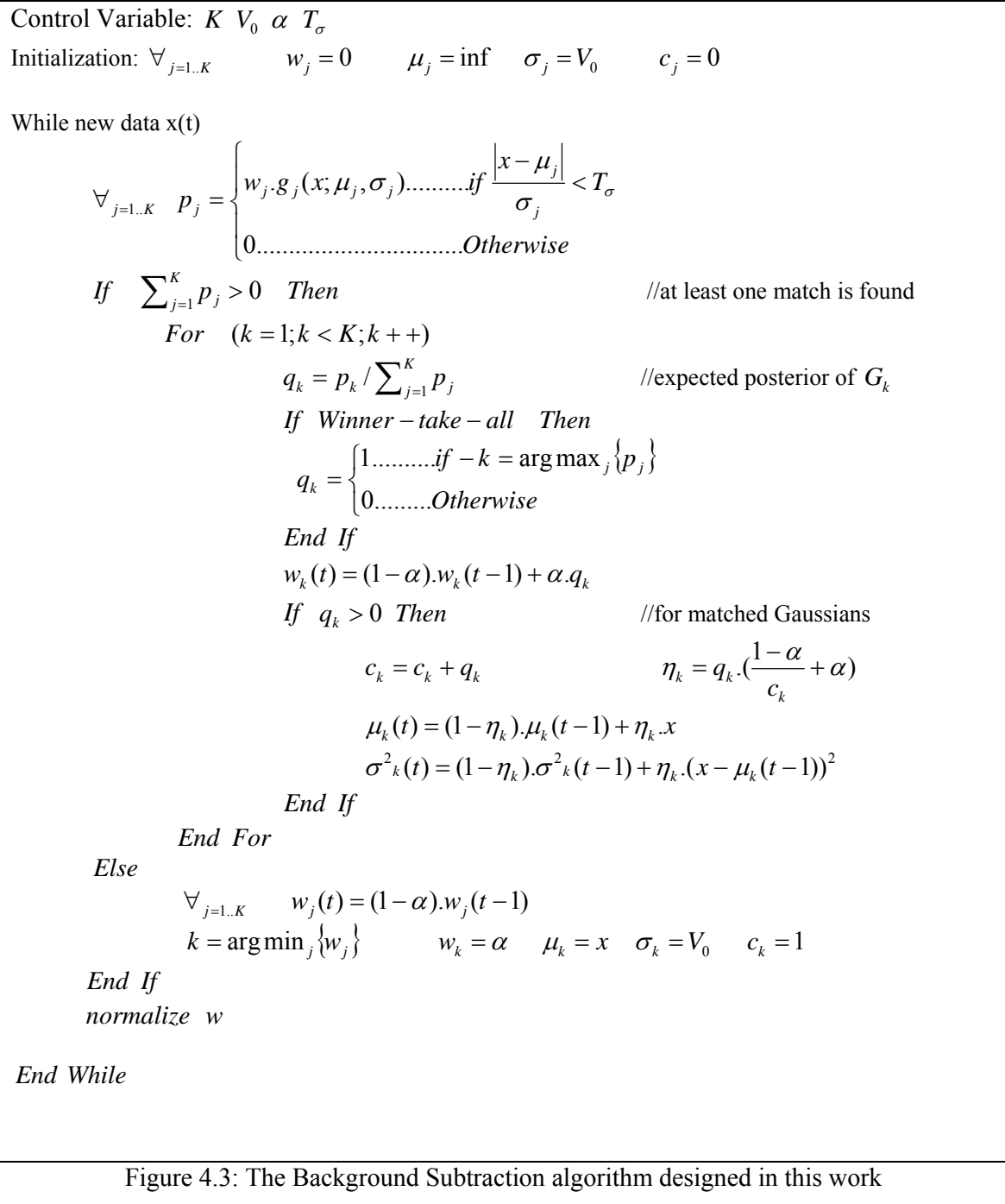
Figure 4.2(a): The pixel value probability illustrated for 1D pixel values[28]

$X \in \{0,1,\dots,255\}$, $K = 3$, $\omega_k = \{0.2,0.2,0.6\}$, $\mu_k = \{80,100,200\}$ and $\sigma_k = \{20,5,10\}$.

Figure 4.2(b): The a posteriori probabilities $P(k|X; \phi)$ plotted as functions of X for each $k = 1, 2, 3$ [28].

4.4 Background Subtraction Algorithm:

The traditional algorithm used by Grimson et al [1,2] has been implemented. The algorithm is shown below in figure 4.3:



4.5 Foreground Segmentation:



Figure 4.4a: Live image sequence



Figure 4.4b: Segmented Foreground (bg-th=0.7)



Figure 4.4c: Live image sequence

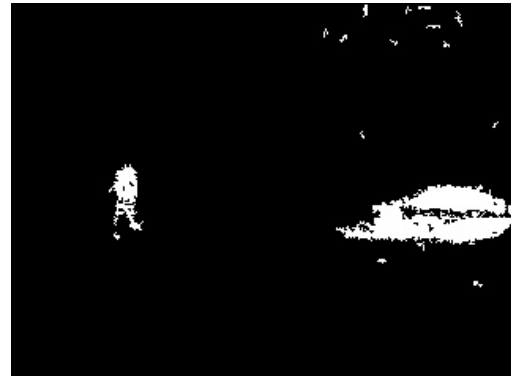


Figure 4.4d: Segmented Foreground (bg-th=0.7)

4.6 Real time problems:

In motion detection operation in real time faces its biggest challenge even after the background subtraction is performed. A moving background substance (i.e. moving vegetations, rains, sea weaves etc.) make the background subtraction fail. As we, in this project, used Mixture of Gaussians modeling each pixel, its produces a very very sensitive feedback. As a result, thousands of moving objects are detected as foreground objects.

In both indoor and outdoor scenes, the use of color cues for background segmentation is limited by illumination variations when lights are switched or weather changes slow or sudden. The background subtraction algorithm is too sensitive on illumination change. Every change in the illumination component of a pixel enables it to be detected as a foreground object. It has been noticed even in the light change for negligible distortion. For example, someone passes by any object then the reflection of it light causes an illumination change which affects the background.



Figure 4.5a: Live image sequence



Figure 4.5b: Effect of illumination change

The presence of shadows in an image causes distortion in the motion detection algorithm. Shadows provide relevant information about the scene represented in an image or a video sequence. They contain cues about the shape and the relative position of objects, as well as about the characteristics of surfaces and light sources. Despite this, in applications requiring the identification of objects, shadows modify the perceived shape and color, thus introducing a distortion in the object detection process. For this reason, the problem of shadow detection has been increasingly addressed over the past years.



Figure 4.6a: Live image sequence

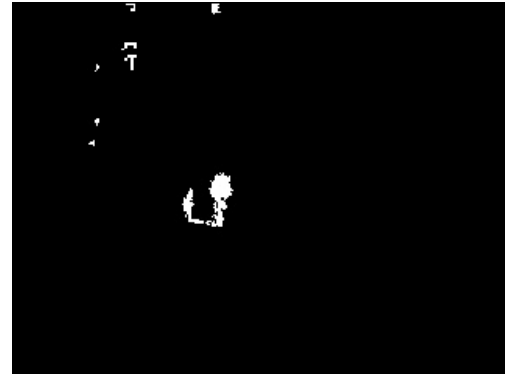


Figure 4.6b: Effect of moving shadows in foreground detection

4.7 Proposed Solution:

At this point the aim is to build an algorithm which not only detect motion in the scene but will also be able to solve the common problems which has been mentioned earlier. The image background has been taken and the running average is calculated as below:

$$B_{i+1} = \alpha * F_i + (1 - \alpha) * B_i \quad (4.3)$$

Where B_i is the background and F_i is the segmented foreground. It adds image to accumulator α with weights. Now the background model is computed as a chronological average from the pixel's history. The difference between the current frame and the previous frame is calculated and the area of motion is pointed out.



Figure 4.7a: Live image Frame1 Figure 4.7b: Live image Frame2 Figure 4.7c: Frame Difference

The change detection is always obtained using the obtained background using the algorithm of background subtraction. It is a new idea and a very good result has been achieved by using it. As this new method every pixel location in the model is based on its recent history so the problem of moving background and the illumination change are solved. Result is shown in the Result and discussion section.

The next problem need to deal with is shadow. This problem is found very critical so a detailed analysis of the shadow itself is done and a lot of experiments are done to identify and detect shadows in an image sequence.

4.7.1 Shadow:

A shadow occurs when an object partially or totally occludes direct light from a source of illumination. Shadows can be divided into two classes: self and cast shadows [27]. A self shadow occurs in the portion of an object which is not illuminated by direct light. A cast shadow is the area projected by the object in the direction of direct light. In the following, the relationship between shadows and lit regions is formalized in order to derive relevant shadow properties.

4.7.2 Photometric color invariants

A spectral property of shadows can be derived in the commented hypothesis by considering photometric color invariants. Photometric color invariants are functions which describe the color

configuration of each image point discounting shading, shadows, and highlights. These functions are demonstrated to be invariant to a change in the imaging conditions, such as viewing direction, object's surface orientation and illumination conditions [27]. Let us define F as one of the above mentioned photometric color invariants. F_l is the value assumed in a point in light, and F_s is the value in the same point in shadow. Then,

$$F_l = F_s \quad (4.4)$$

Examples of photometric color invariants are normalized rgb, hue (H), saturation (S), c1c2c3 and l1l2l3. In particular, among the different photometric invariant color features, the c1c2c3 model has been tested. The c1c2c3 invariant color features are defined as follows:

$$c_1(x, y) = \arctan \frac{R(x, y)}{\max(G(x, y), B(x, y))}, \quad (4.5)$$

$$c_2(x, y) = \arctan \frac{G(x, y)}{\max(R(x, y), B(x, y))}, \quad (4.6)$$

$$c_3(x, y) = \arctan \frac{B(x, y)}{\max(R(x, y), G(x, y))}, \quad (4.7)$$

for $R(x, y)$, $G(x, y)$ and $B(x, y)$ representing the red, green, and blue color components of a pixel in the image. It is known from Kender that normalized color rgb is unstable near the black vertex of the RGB space, where it is undefined, while hue is unstable near its singularities at the entire achromatic axis.

4.7.3 Geometric Properties of shadows

The geometric appearance of a shadow depends on objects and scene layout. However, it is possible to identify some geometrical characteristics of shadows, the

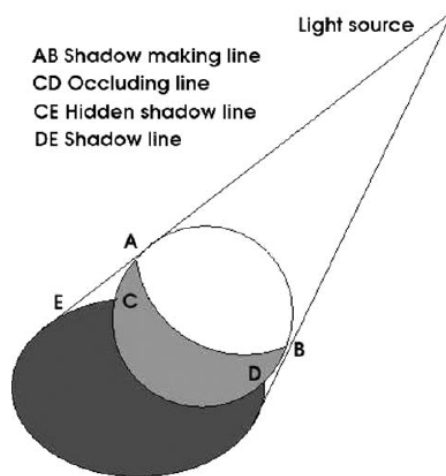


Figure 4.8: Shadow lines definition [27].

shadow boundaries, without any knowledge of the structure of the object or of the scene. Shadow boundaries can be classified into four classes: shadow-making lines, shadow lines, occluding lines, and hidden shadow lines [27]. These lines are depicted in Figure 4.8. Shadow-making lines, AB, separate the illuminated surface and the non-illuminated surface of an object. They appear to be the outlines of an object if the position of the observer is aligned with the direction of the light source. The projections of the shadow-making lines in the direction of light rays are called shadow lines, DE. Occluding lines, CD, separate an object from its cast shadow. A hidden shadow line, CE, is a shadow line corresponding to a non-visible shadow-making line [27].

4.8 Division based brightness invariants:

Another way to achieve invariant parameters is to normalize the RGB values to the intensity factor. This can be done by division of color signals. Examples of this type of parameters are normalized RGB (r g b), saturation calculated by:

$$S = \frac{\max\{R, G, B\} - \min\{R, G, B\}}{\max\{R, G, B\}} \quad (4.8)$$

or the new introduced colors like c1c2c3

$$c_1(x, y) = \arctan \frac{R(x, y)}{\max(G(x, y), B(x, y))}, \quad (4.9)$$

$$c_2(x, y) = \arctan \frac{G(x, y)}{\max(R(x, y), B(x, y))}, \quad (4.10)$$

$$c_3(x, y) = \arctan \frac{B(x, y)}{\max(R(x, y), G(x, y))}, \quad (4.11)$$

and l1 l2 l3:

$$l_1 = \frac{(R - G)^2}{(R - G)^2 + (R - B)^2 + (G - B)^2}, \quad (4.12)$$

$$l_2 = \frac{(R - B)^2}{(R - G)^2 + (R - B)^2 + (G - B)^2}, \quad (4.13)$$

$$l_3 = \frac{(G - B)^2}{(R - G)^2 + (R - B)^2 + (G - B)^2}, \quad (4.14)$$

which are suggested by Gever. Yet another brightness invariant model can be obtained by calculation:

$$r_1' = \frac{\max\{G, B\} - R}{\max(R, G, B)}, \quad (4.15)$$

$$r_2' = \frac{\max\{R, B\} - G}{\max(R, G, B)}, \quad (4.16)$$

$$r_3' = \frac{\max\{R, G\} - B}{\max(R, G, B)}, \quad (4.17)$$

It is shown that in some forest road scenes, segmentation by r_3' yields to better results in comparison to other brightness invariant parameters. However the results have been tested for detecting the shadow using the photometric color invariant parameters.

4.8.1 Test Results:

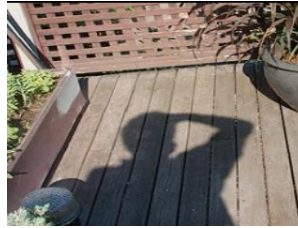


Figure 4.9: Input

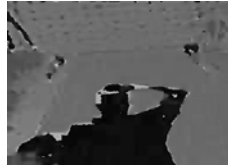


Figure 4.9a: C1



Figure 4.9b: C2



Figure 4.9c: C3

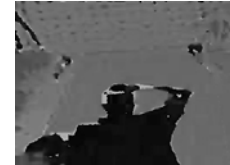


Figure 4.9d: CC

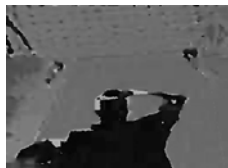


Figure 4.9e: L1



Figure 4.9f: L2

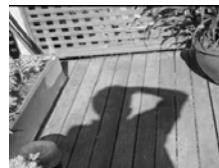


Figure 4.9g: L3



Figure 4.9h: LL

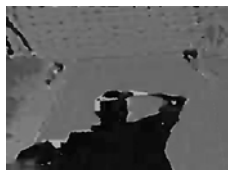


Figure 4.9i: R1'



Figure 4.9j: R2'

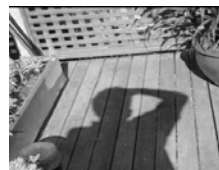


Figure 4.9k: R3'

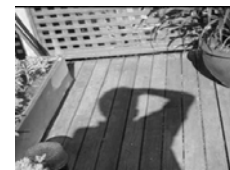


Figure 4.9l: RR

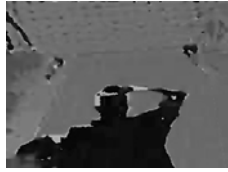


Figure 4.9m: H



Figure 4.9n: s



Figure 4.9o: v

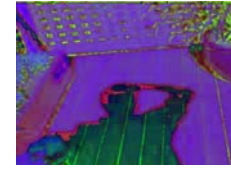


Figure 4.9p: HSV

These parameters are found not working very well in both still and moving sequences except the Value (V) image. The V image is used later on with the shadow detection algorithm. These were implied to the moving sequences as well as still pictures but could not show any satisfactory performance. The idea is left right after obtaining this result.



Figure 4.10a: Live image



Figure 4.10b: Hue Image

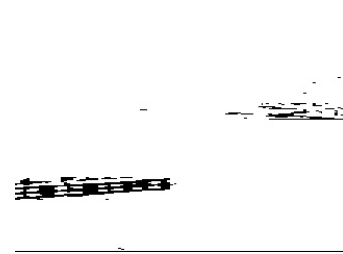


Figure 4.10c: Extracted Shadow

4.8.2 Proposed Method of Shadow detection:

The cast shadow segmentation algorithm of Ebrahimi et al [27] is first employed. The first level of the proposed strategy makes use of the property that shadows darken the surface upon which they are cast. This results in the identification of potential shadows. Let us refer to the image under analysis³ with $I(x,y) = (R(x,y), G(x,y), B(x,y))$, where R;G; B represent the three color channels and (x, y) indicates a generic pixel position. The intensity of each pixel $I(x, y)$ is compared to the intensity of a reference pixel, $I(x_r, y_r)$. The reference pixel (x_r, y_r) is defined differently for still images and video, and it is described in the following. Each camera sensor must have a lower response for a point in shade. The pixel (x, y) becomes a candidate shadow if its intensity is smaller than that of the reference pixel for all three channels. This results in the identification of a set of pixels, S_c , which are, candidate to be shadow pixels [27].

$$S_c = \{(x, y) : R(x_r, y_r) > R(x, y), G(x_r, y_r) > G(x, y), B(x_r, y_r) > B(x, y)\} \quad (4.18)$$

According to the proposed method the first scene of the sequence is considered as the reference image. The reference pixel (x_r, y_r) belongs to the reference image which represents the background of the scene. The reference image can be either a frame in the sequence or a reconstructed one [27]. The reference pixel (x_r, y_r) is at the same location as (x, y) in the image under analysis. The

analysis is performed only in the areas of the image which have been identified as changed by a motion detector. The identified areas correspond to both moving objects and their shadows. Candidate shadow points are detected by analyzing the image difference $D(x, y)$ computed as $D(x, y) = I(x_r, y_r) - I(x, y)$. In a noise-free case, the conditions,

$$R(x_r, y_r) - R(x, y) > 0, G(x_r, y_r) - G(x, y) > 0, B(x_r, y_r) - B(x, y) > 0 \quad (4.19)$$

would suffice to state that the pixel (x, y) belongs to S_c [27]. In real situations, the noise introduced by the acquisition process alters the above test, so that it becomes

$$R(x_r, y_r) - R(x, y) > b_1, G(x_r, y_r) - G(x, y) > b_2, B(x_r, y_r) - B(x, y) > b_3. \quad (4.20)$$

The vector $b = (b_1, b_2, b_3)$ takes care of the distortions introduced by the noise.

4.8.3 Proposed Method Analysis:

The method has been tested in moving shadows in the scene. The noise and distortion found in real time is enormous. The result obtained was not very satisfactory at all. It is known that the method is basically designed for both indoor and outdoor imagery where the interference of real time noise is controllable. But it seemed that the imaging process described is not suited with the real time images which were obtained from the surveillance camera. Instead the algorithm sometimes has produced a sheared image as below.



Figure 4.11a: Live image sequence

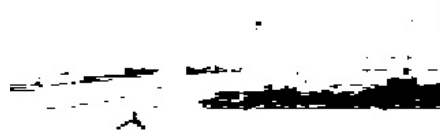


Figure 4.11b: Shadow detection in noise $b=70.5$

4.8.4 Improvement of the proposed algorithm

All the photometric color invariant parameters have been tried out and it is found that the r, g, b images are worthwhile. As proposed in the earlier works, the reference image considered to be the first image of the sequence. It is not explained what happened if there happens to be any sudden change in the background. It sounds logical if a solution is chosen which will not only keeps track of

the changes occurred but also update the background sequence as well. Instead of comparing the current frame with the previous frame, the current frame is compared with the background image which is updated periodically.

For the shadow detection method, the same background has been employed as the reference image and is compared with the value image of the current frame. The result obtained is much better and rather interesting. Only the moving object excluding shadows have been tried to detect.



Figure 4.12a: Live image sequence

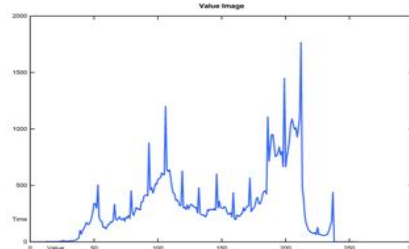


Figure 4.12b: Histogram of Value Image; $\mu = 148$ and $\sigma = 46$



Figure 4.12c: Shadow detection in noise $b=90.5$



Figure 4.12d: Live image sequence

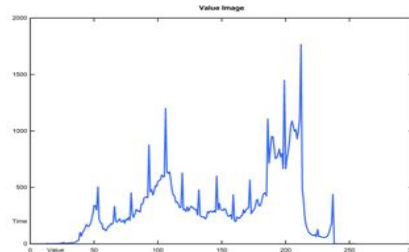


Figure 4.12e: Histogram of Value Image; $\mu = 148$ and $\sigma = 46$



Figure 4.12f: Shadow detection in noise $b=70.5$

4.9 Final Algorithm of motion detection without shadows:

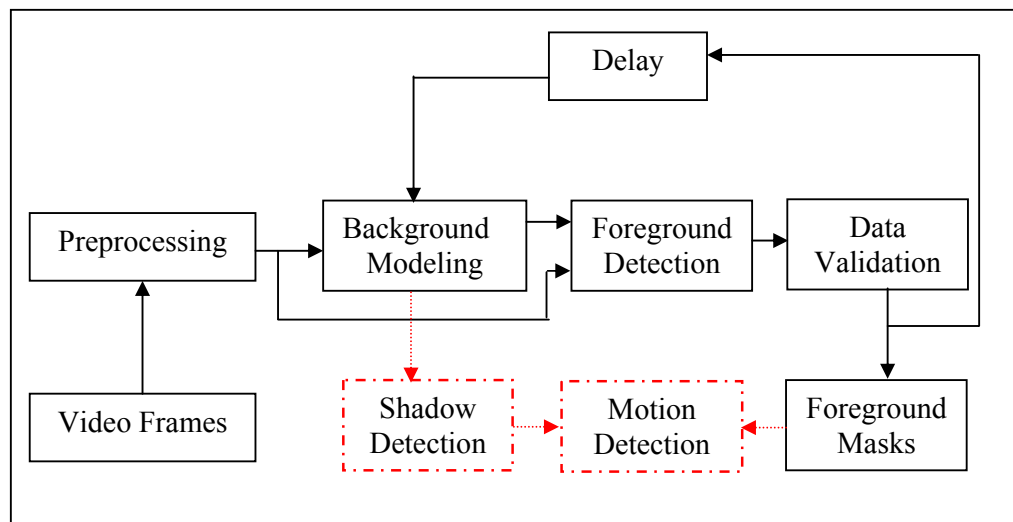


Figure 4.13: Final work flow diagram of motion detection algorithm without shadows.

The original work of Grimson and Stauffer's et al. is proven not enough to handle the problems itself as discussed. The final state of motion detection is achieved by adding two modules with the primary work. Shadow detection is done from the background image constructed through the new algorithm and the final motion detection module is done after performing the image differenceing technique is applied on the current image sequences. After the integration of these modules the improved motion detection algorithm is achieved.


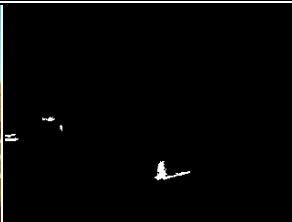



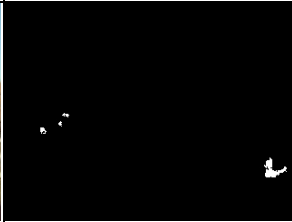



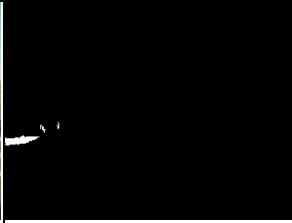


Chapter 5: Test results and discussion


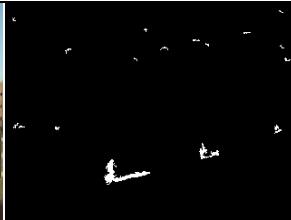

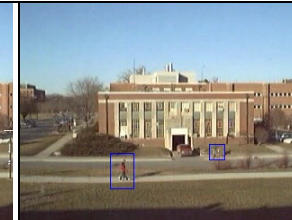
In this section, the algorithm's performance on a few standard problems is demonstrated. The algorithm is vastly tested on outdoor environment and indoor as well. The results of several trials for each data set are shown. For the real-world data-sets, the data is randomly sampled to generate longer sequences needed for the sequential algorithm.

5.1 Experiments in outdoor environment:




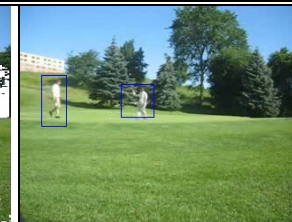

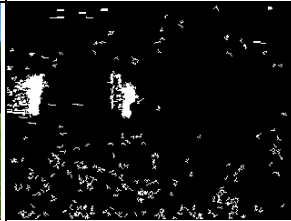

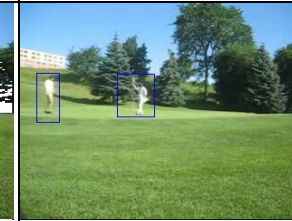



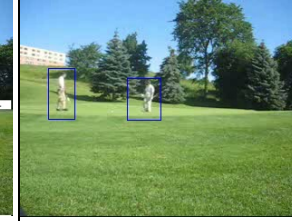

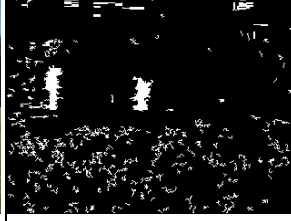

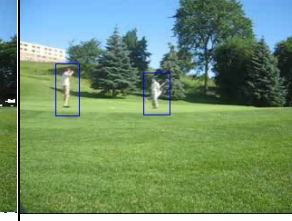
As it is said before, outdoor environment is very challenging as a significant amount of noise is present there. A lot of outdoor motion sequences have been taken under consideration in different scenarios where the real time problems were present. The newly developed algorithm has been tested in these situations under different noise levels. The results obtained are marked with the current noise level accordingly.

5.1.1 Busy Road:






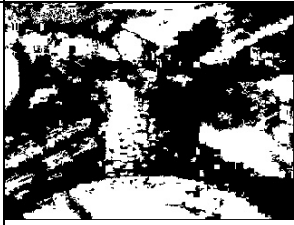



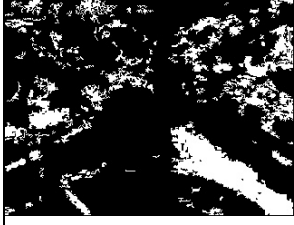



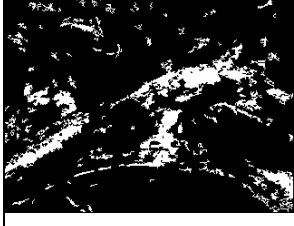


			
Figure 5.1a: Background	Figure 5.1b: Foreground	Figure 5.1c: Shadow Detection with noise $b=50.5$	Figure 5.1d: Detected Motion
			
Figure 5.1e: Background	Figure 5.1f: Foreground	Figure 5.1g: Shadow Detection with noise $b=70.5$	Figure 5.1h: Detected Motion
			
Figure 5.1i: Background	Figure 5.1j: Foreground	Figure 5.1k: Shadow Detection with noise $b=90.5$	Figure 5.1l: Detected Motion

			
Figure 5.1m: Background	Figure 5.1n: Foreground	Figure 5.1o: Shadow Detection with noise $b=100.5$	Figure 5.1p: Detected Motion



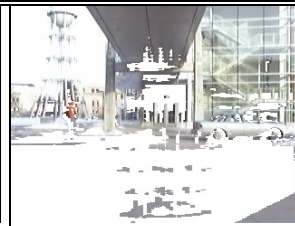
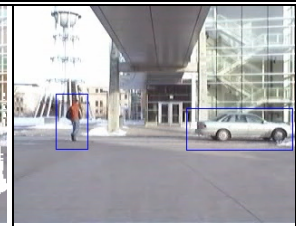


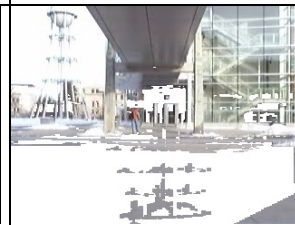

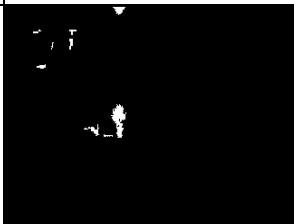
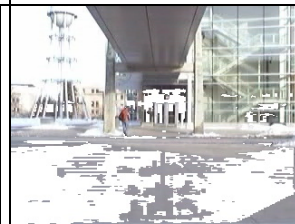


5.1.2 Sunny day:

			
Figure 5.2a: Background	Figure 5.2b: Foreground	Figure 5.2c: Shadow Detection with noise $b=50.5$	Figure 5.2d: Detected Motion
			
Figure 5.2e: Background	Figure 5.2f: Foreground	Figure 5.2g: Shadow Detection with noise $b=70.5$	Figure 5.2h: Detected Motion
			
Figure 5.2i: Background	Figure 5.2j: Foreground	Figure 5.2k: Shadow Detection with noise $b=90.5$	Figure 5.2k: Detected Motion
			
Figure 5.2l: Background	Figure 5.2m: Foreground	Figure 5.2n: Shadow Detection with noise $b=100.5$	Figure 5.2n: Detected Motion

5.1.3 Traffic (Night):

			
Figure 5.3a: Background	Figure 5.3b: Foreground	Figure 5.3c: Shadow Detection with noise $b=50.5$	Figure 5.3d: Detected Motion
			
Figure 5.3e: Background	Figure 5.3f: Foreground	Figure 5.3g: Shadow Detection with noise $b=70.5$	Figure 5.3h: Detected Motion
			
Figure 5.3i: Background	Figure 5.3j: Foreground	Figure 5.3k: Shadow Detection with noise $b=90.5$	Figure 5.3l: Detected Motion
			
Figure 5.3m: Background	Figure 5.3n: Foreground	Figure 5.3o: Shadow Detection with noise $b=100.5$	Figure 5.3p: Detected Motion

5.1.4 Traffic (Day):

			
Figure 5.4a: Background	Figure 5.4b: Foreground	Figure 5.4c: Shadow Detection with noise $b=50.5$	Figure 5.4d: Detected Motion
			
Figure 5.4e: Background	Figure 5.4f: Foreground	Figure 5.4g: Shadow Detection with noise $b=70.5$	Figure 5.4h: Detected Motion
			
Figure 5.4i: Background	Figure 5.4j: Foreground	Figure 5.4k: Shadow Detection with noise $b=90.5$	Figure 5.4l: Detected Motion
			
Figure 5.4m: Background	Figure 5.4n: Foreground	Figure 5.4o: Shadow Detection with noise $b=100.5$	Figure 5.4p: Detected Motion

5.1.5 A Rainy Day:

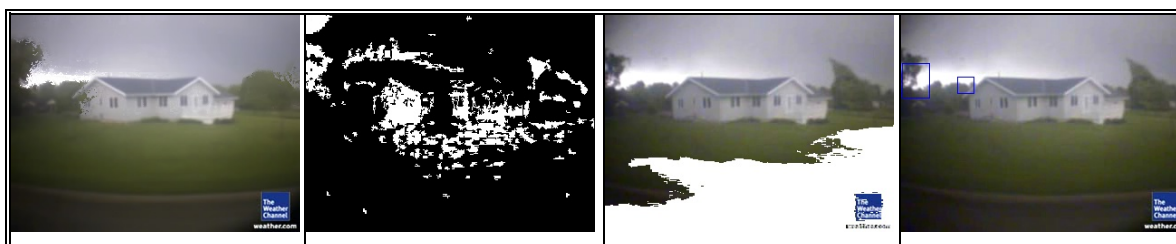

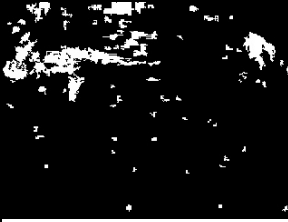



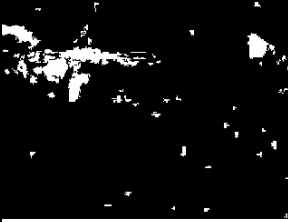



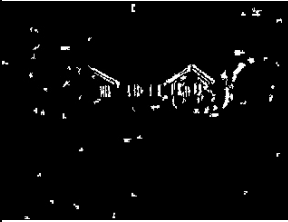


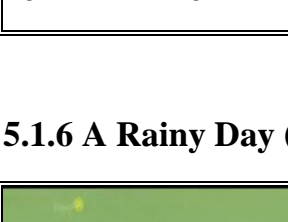
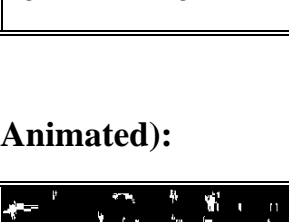
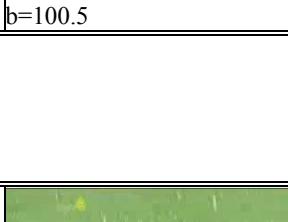
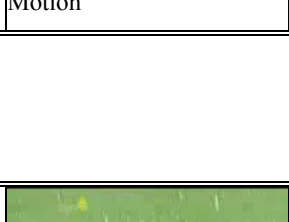


Figure 5.5a: Background	Figure 5.5b: Foreground	Figure 5.5c: Shadow Detection with noise $b=50.5$	Figure 5.5d: Detected Motion
			
Figure 5.5e: Background	Figure 5.5f: Foreground	Figure 5.5g: Shadow Detection with noise $b=70.5$	Figure 5.5h: Detected Motion
			
Figure 5.5i: Background	Figure 5.5j: Foreground	Figure 5.5k: Shadow Detection with noise $b=90.5$	Figure 5.5l: Detected Motion
			
Figure 5.5m: Background	Figure 5.5n: Foreground	Figure 5.5o: Shadow Detection with noise $b=100.5$	Figure 5.5p: Detected Motion
			

5.1.6 A Rainy Day (Animated):

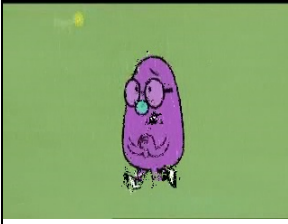
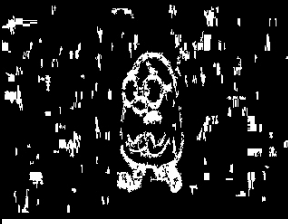

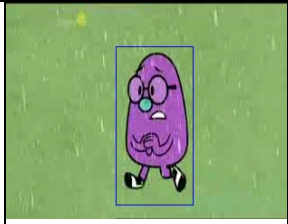
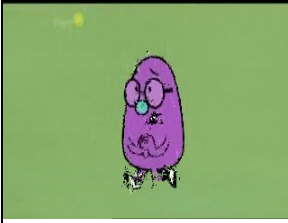
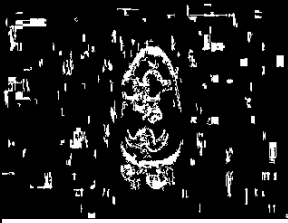

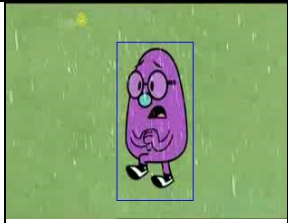
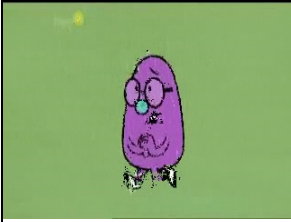
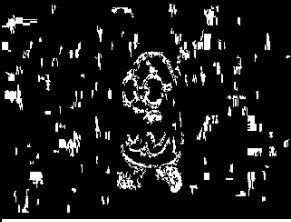

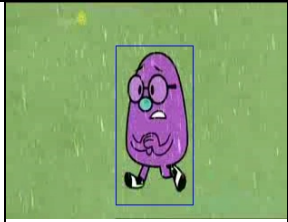
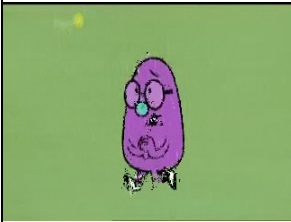
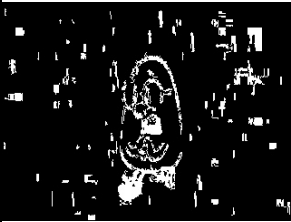
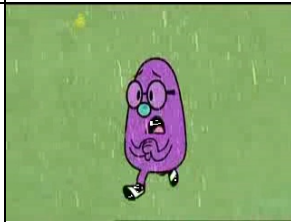
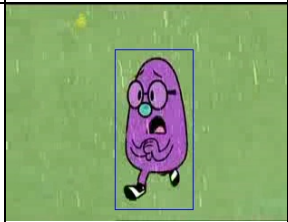
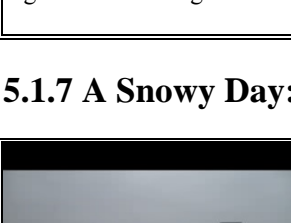
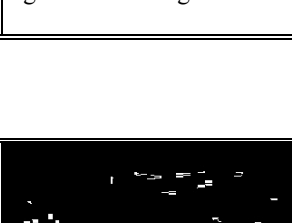

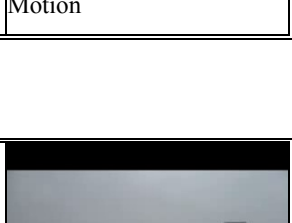

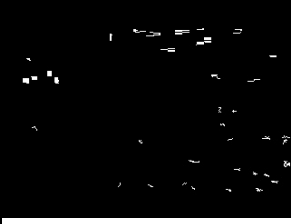



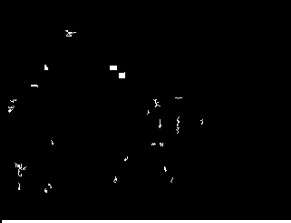


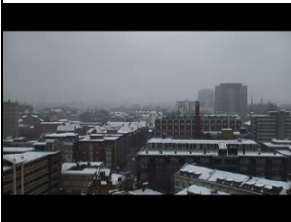
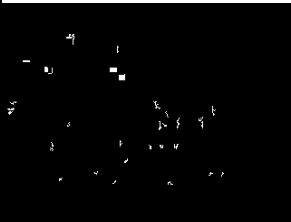
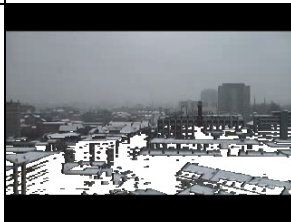
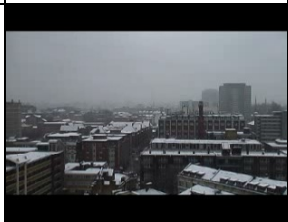








			
Figure 5.6a: Background	Figure 5.6b: Foreground	Figure 5.6c: Shadow Detection with noise $b=50.5$	Figure 5.6d: Detected Motion
			

Figure 5.6e: Background	Figure 5.6f: Foreground	Figure 5.6g: Shadow Detection with noise $b=70.5$	Figure 5.6h: Detected Motion
			
Figure 5.6i: Background	Figure 5.6j: Foreground	Figure 5.6k: Shadow Detection with noise $b=90.5$	Figure 5.6l: Detected Motion
			
Figure 5.6m: Background	Figure 5.6n: Foreground	Figure 5.6o: Shadow Detection with noise $b=100.5$	Figure 5.6p: Detected Motion
			

5.1.7 A Snowy Day:

			
Figure 5.7a: Background	Figure 5.7b: Foreground	Figure 5.7c: Shadow Detection with noise $b=50.5$	Figure 5.7d: Detected Motion
			
Figure 5.7e: Background	Figure 5.7f: Foreground	Figure 5.7g: Shadow Detection with noise $b=70.5$	Figure 5.7h: Detected Motion
			
Figure 5.7i: Background	Figure 5.7j: Foreground	Figure 5.7k: Shadow Detection with noise $b=90.5$	Figure 5.7l: Detected Motion
			

			
Figure 5.7m: Background	Figure 5.7n: Foreground	Figure 5.7o: Shadow Detection with noise $b=100.5$	Figure 5.7p: Detected Motion

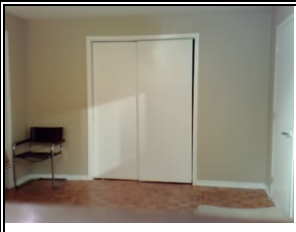
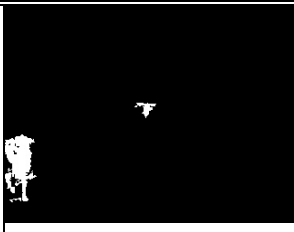

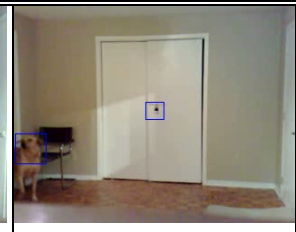
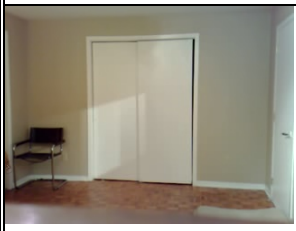
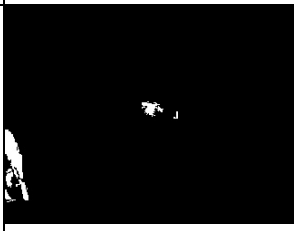


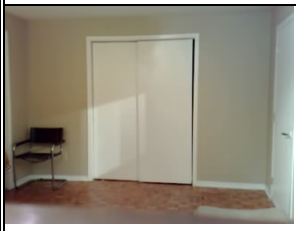

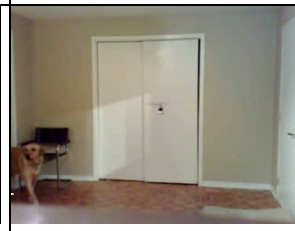


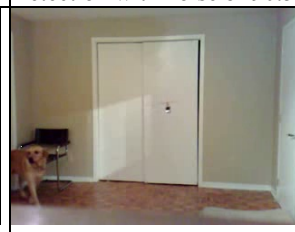
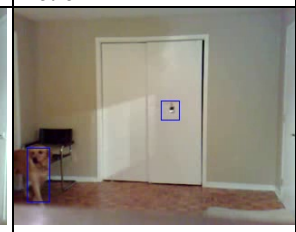
5.1.8 Observations Outdoor:

In busy road situation, it is seen that 90.5 noise level shows result much more close to the reality. The shadows are detected correctly (bus shadows and others) which makes the detection process more accurate. In sunny day situation both 90.5 and 100.5 matches the reality. Thing to be notices here is the moving trees and branches are not detected as foreground object. Even in the night time moving objects are detected successfully. The scene which is chosen is taken from a surveillance camera at a very high position in a busy traffic square. It is after shower and there is a reflection of light on the road was present which causes a wide illumination change as detected in the foreground. The new algorithm works very well in this situation too as shown in 5.1.3. Again the noise level 90.5 and 100.5 shows better result in both of the traffic situations. Videos of rainy situation (with the courtesy of the weather channel) are collected in both live and animated stream and the algorithm is tested on it. The result is quite optimistic. In the first real surveillance situation the rain drops and the moving trees are not detected as the foreground object. So no object was detected. It is clearly seen in the animated movie clip. The rain drops are seen in the live sequence but the background and foreground both are detected so well that it is only the moving object in the seen is detected. Noise level 100.5 shows the best result among all with no shadow detected. A surveillance camera was placed outside in a snowy day where random snow fall was occurred. With no trouble faced the algorithm did not pick any falling snow as a moving object.


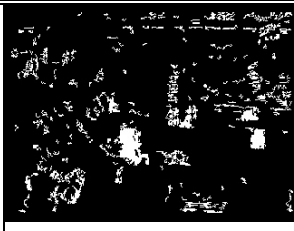

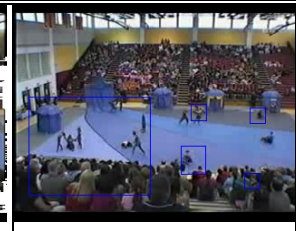
5.2 Indoor Situation monitoring Experiments:








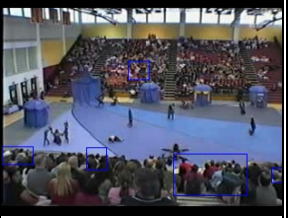




Here in this section the impact of the newly developed algorithm is closely monitored on the indoor surveillance.

5.2.1 Indoor Room:

			
Figure 5.8a: Background	Figure 5.8b: Foreground	Figure 5.8c: Shadow Detection with noise $b=50.5$	Figure 5.8d: Detected Motion
			
Figure 5.8e: Background	Figure 5.8f: Foreground	Figure 5.8g: Shadow Detection with noise $b=70.5$	Figure 5.8h: Detected Motion
			
Figure 5.8i: Background	Figure 5.8j: Foreground	Figure 5.8k: Shadow Detection with noise $b=90.5$	Figure 5.8l: Detected Motion
			
Figure 5.8m: Background	Figure 5.8n : Foreground	Figure 5.8o: Shadow Detection with noise $b=100.5$	Figure 5.8p: Detected Motion

5.2.2 Large Hall (Indoor):

			
Figure 5.9a: Background	Figure 5.9b: Foreground	Figure 5.9c: Shadow Detection with noise $b=50.5$	Figure 5.9d: Detected Motion

			
Figure 5.9e: Background	Figure 5.9f: Foreground	Figure 5.9g: Shadow Detection with noise $b=70.5$	Figure 5.9h: Detected Motion
			
Figure 5.9i: Background	Figure 5.9j: Foreground	Figure 5.9k: Shadow Detection with noise $b=90.5$	Figure 5.9l: Detected Motion
			
Figure 5.9m: Background	Figure 5.9n: Foreground	Figure 5.9o: Shadow Detection with noise $b=100.5$	Figure 5.9p: Detected Motion

5.2.3 Observations Indoor:

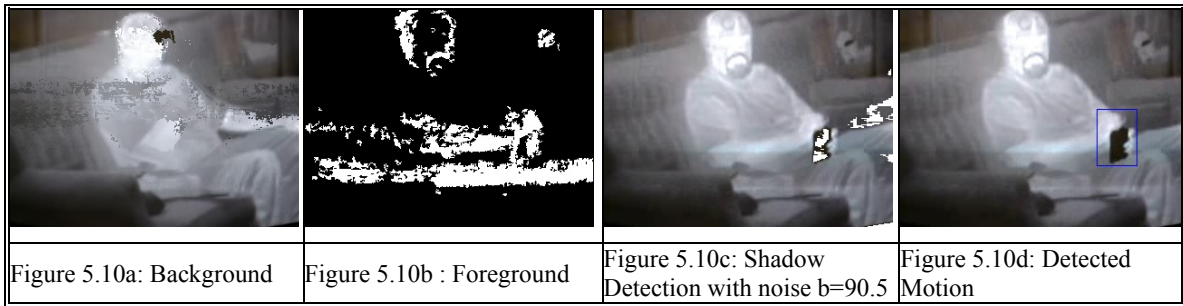
As the real time noise level is expectedly minimized in closed door situation so it is found that the algorithm works better in the range of 70-100. The rate of getting false alarm is significantly lower than outdoor. Change of ambient light in the large hall causes illumination change very often specially in the big hall situation. But this algorithm tackles this problem as well and shows correct result in most of the time.

5.3 Surveillance imagery from other sources:

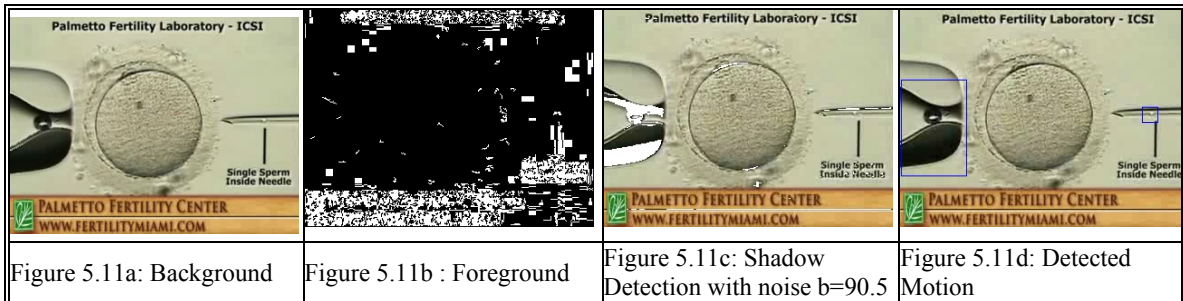
Images are taken for surveillance in not only indoor or outdoor situation but from also other sources and in other situations as well. There are new video cameras in the market which converts the live images directly to the infrared image. It gives the user the opportunity to keep an eye on expected objects under any circumstance where the live imagery may not give the expected result like in the dark areas. Microscopic cameras are also popular nowadays to monitor tiny organic particles. Ultra

sonogram images are now being watched and analyzed manually by human. The algorithm is tried on these untried image sources.

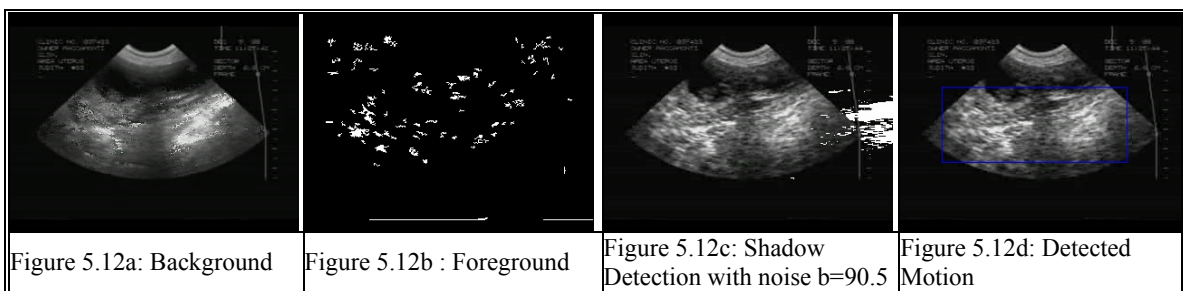
5.3.1 Infrared:



5.3.2 Microscopic view:



5.3.3 Ultra Sonogram Image:



5.4 Observation of images from other sources:

Infrared images are quite different from the normal images. It is clearly heat sensitive. Hot and cold objects and their impacts are easily distinguishable and different metals have different visibility in the infrared image. The impact of cold objects in the source image sequence has been analyzed and

the algorithm is applied there. The motion there is successfully detected even in infrared images as seen in 5.3.1. Microscopic videos are chosen to analyze with the courtesy of PALMETTO Fertility Center. Even the microscopic objects are successfully detected with their appropriate shadows. The quest is quite successful in testing the ultra sonogram videos as well. The motion of the object is accurately detected from the given source.

5.5 Performance Evaluation:

5.5.1 Frame Rate per Second (FPS):

Frame rate per second is measured for the first 200 frames of every sample video sequences used for testing. It is done on purpose to see time needed by the algorithm to process frames. The result obtained is shown below:

Outdoor:

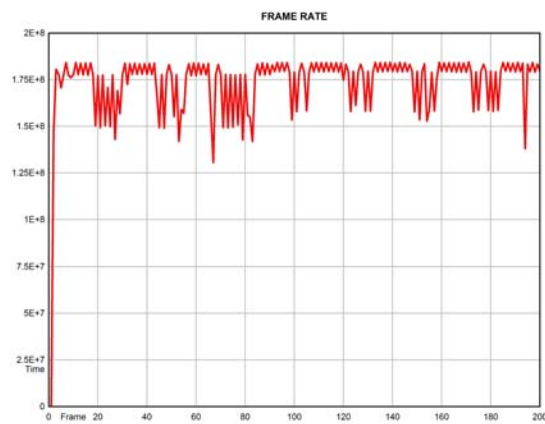


Figure 5.13a: Busy Road

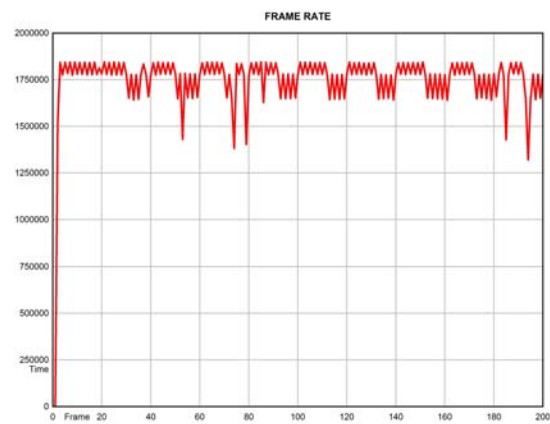


Figure 5.13b: Sunny Day



Figure 5.13c: Traffic (Night)



Figure 5.13d: Traffic (Day)

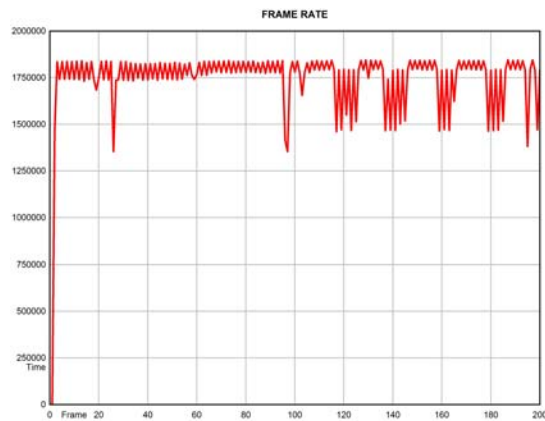


Figure 5.13e: Rainy Day (Live)

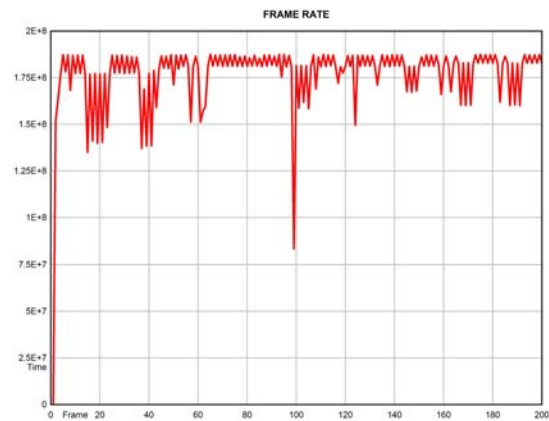


Figure 5.13f: Rainy Day (Animated)

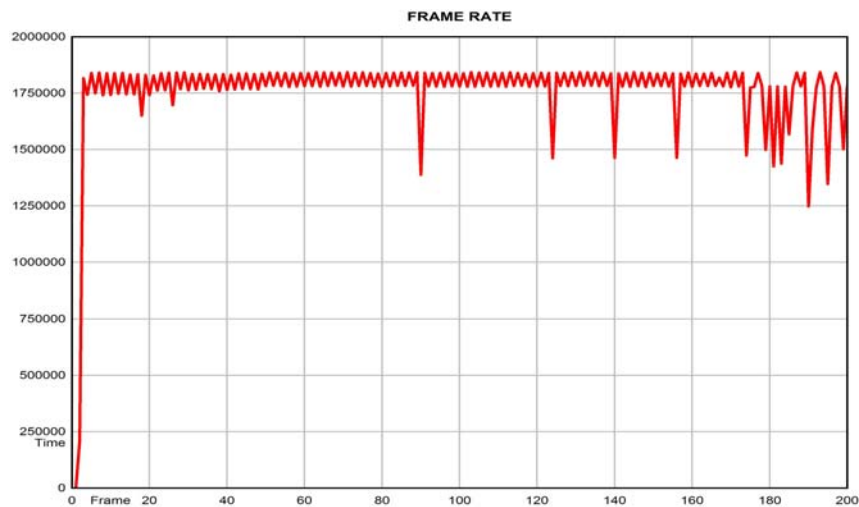


Figure 5.13g: Snow

It is clearly seen from the fps rate of the images that the algorithm converges at the rate of 17.5 fps. The busy road and traffic environment are noisy enough and frequent motion is detected in consecutive frames. This algorithm is time sensitive at the presence of the motion in the scene. Spikes are showing the sign of detected objects. Good thing is the behavior of the algorithm does not change for the presence of more than one moving object. It takes almost the same processing time for one and for several moving objects. The video of snowy environment is free of motion. A constant line through 17.5 fps was expected and it shows the same if the false alarms are ignored.

Indoor:

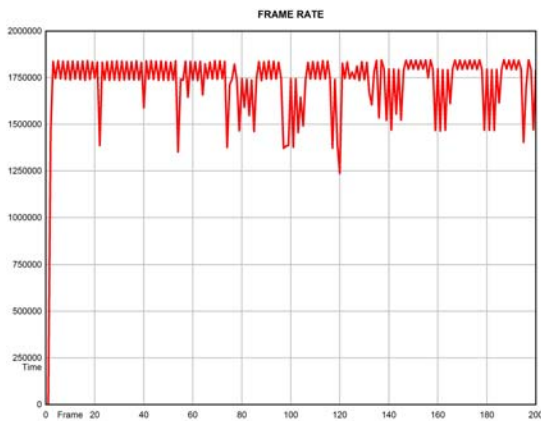


Figure 5.14a: Closed Room

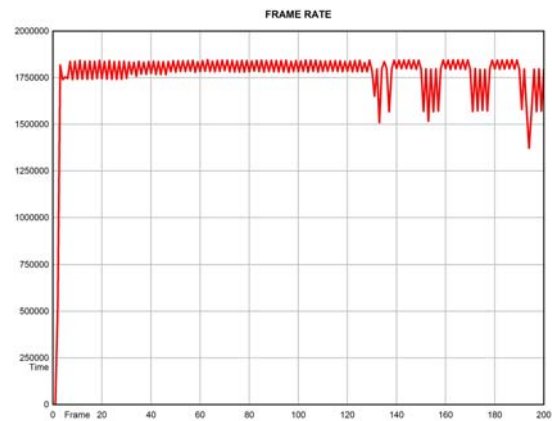


Figure 5.14b: Big Hall

Indoor situation is much easier to handle. Noise level is much lower than outdoor environment. The graph clearly depicts the situation. It usually responds to the motion otherwise remain stable. False alarms are not taken under consideration.

Other Imagery:

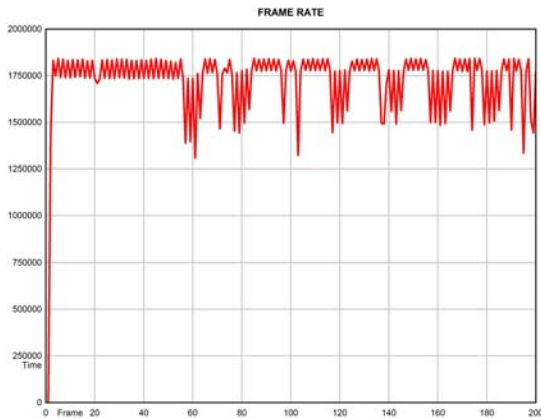


Figure 5.15a: Microscopic Images

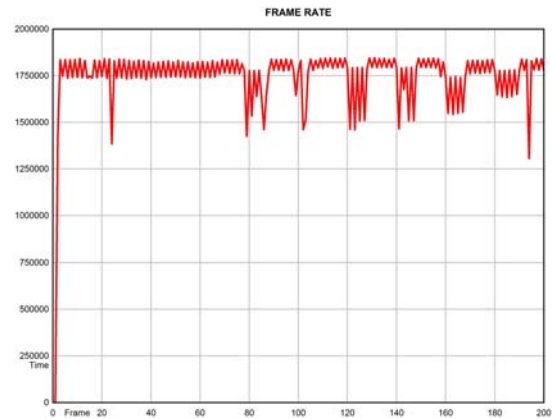


Figure 5.15b: Infrared Images

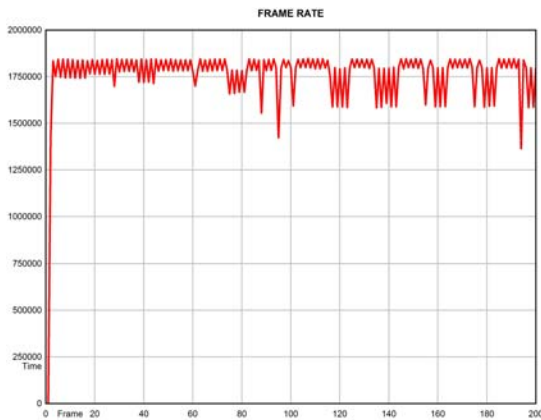


Figure 5.15c: Ultra sonogram Images

Among all the other source imagery, microscopic environment is proved much noisier than the others. Presence of moving objects is very frequent like the traffic conditions. Infrared images show a constant result so as ultra sonogram images.

Performance Comparison between outdoor and indoor videos:

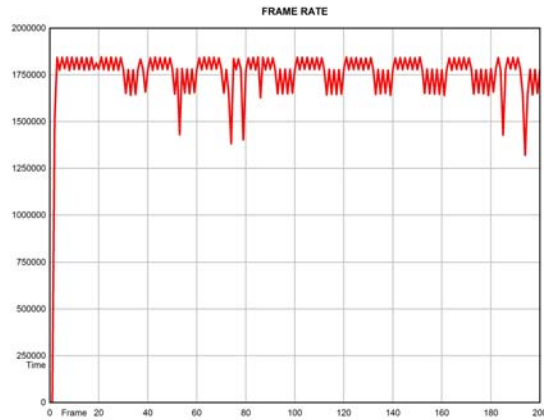


Figure 5.16a: Sunny Day

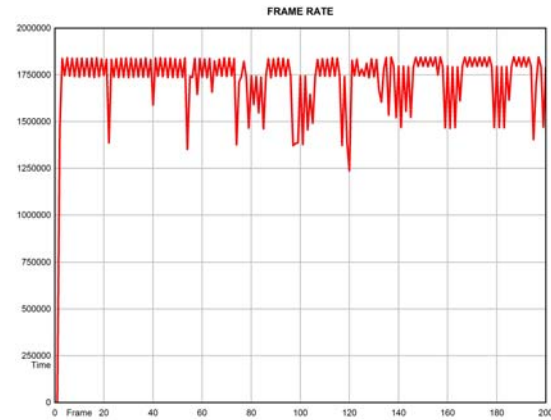


Figure 5.16b: Closed Room

It is obvious that outdoor environment is much noisier than a controlled environment like indoor videos. There much fluctuation in the processing time in the outdoor. On the contrary, ignoring the false alarms, a constant processing speed is observed in the indoor videos.

5.5.2 Mean and Standard Deviation:

To have the grasp of the data distribution of the processed image, the mean and standard deviation are analyzed in the test environments.

Outdoor:

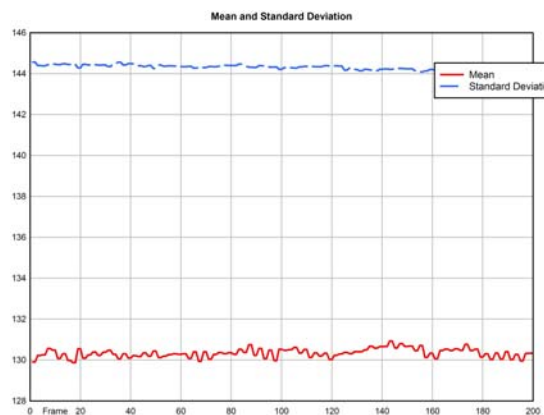


Figure 5.17a: Busy Road

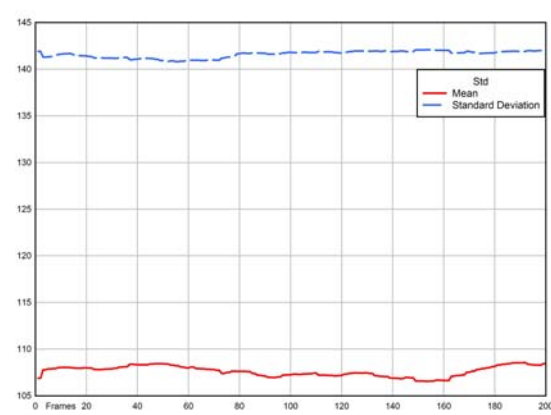


Figure 5.17b: Sunny Day

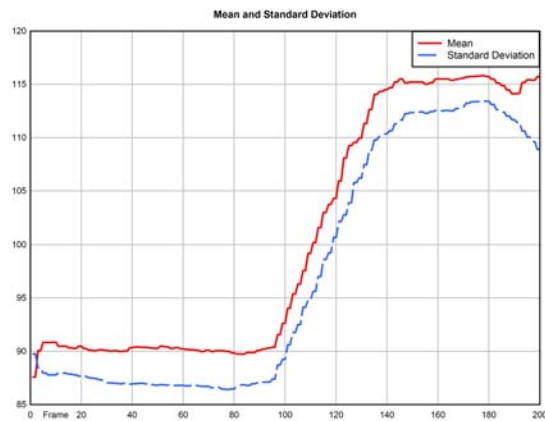


Figure 5.17c: Traffic (Night)

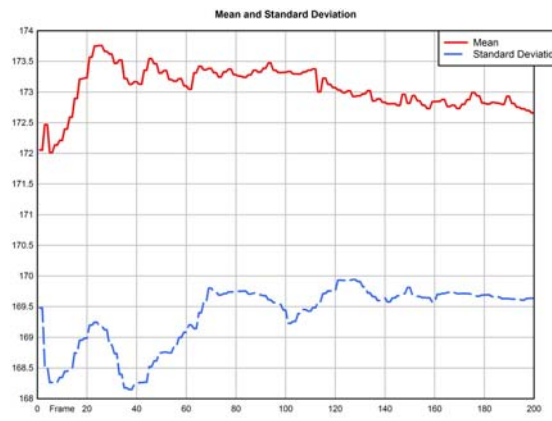


Figure 5.17d: Traffic (Day)

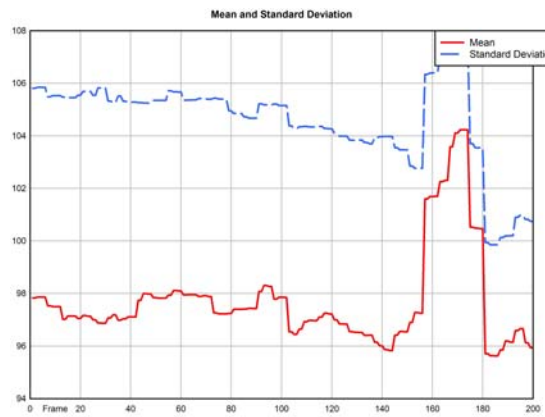


Figure 5.17e: Rainy Day (Live)

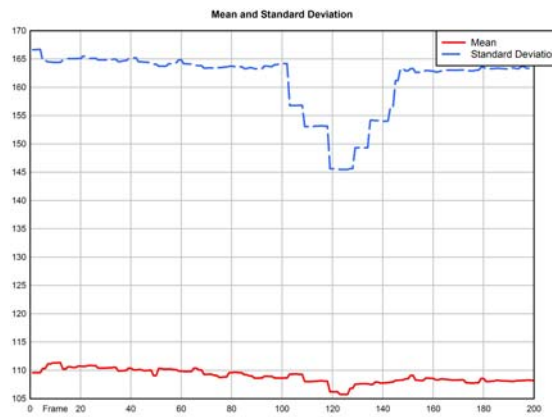


Figure 5.17f: Rainy Day (Animated)

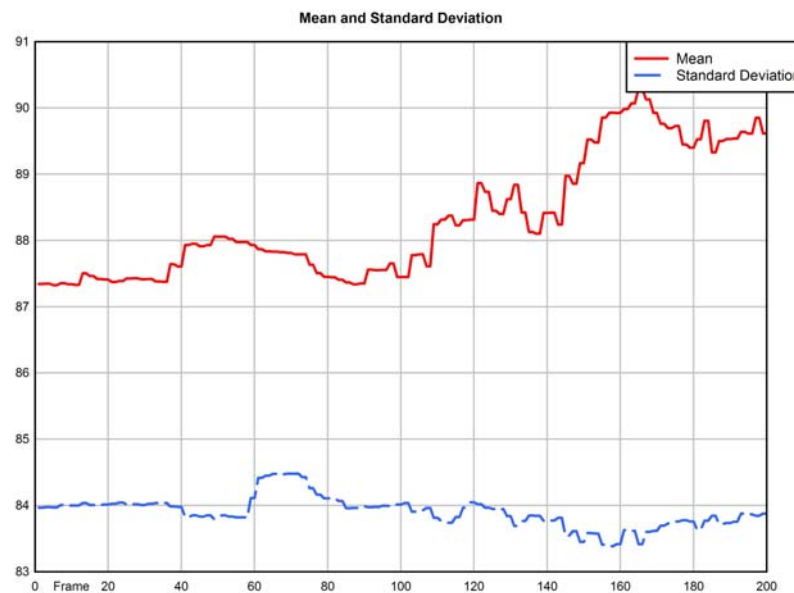


Figure 5.17g: Snow

The algorithm is found very sensitive in rapidly changing environment like traffic at night. Though background is still remains same, the rush of cars or deflection of other objects had a great impact

on the mean and the standard deviation of the image. The traffic night and the rainy day video suffer from the same problem. On the other hand, busy road, sunny day and snow show no distortion in the performance of the algorithm.

Indoor:

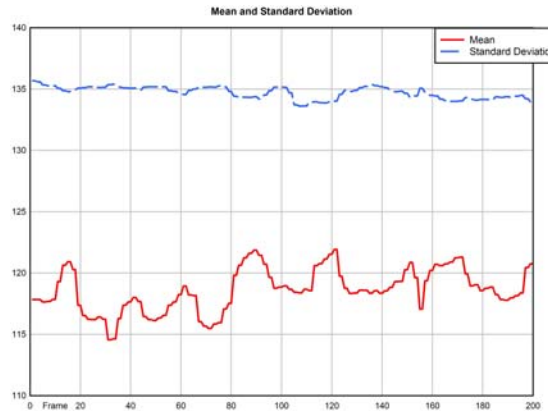


Figure 5.18a: Closed Room

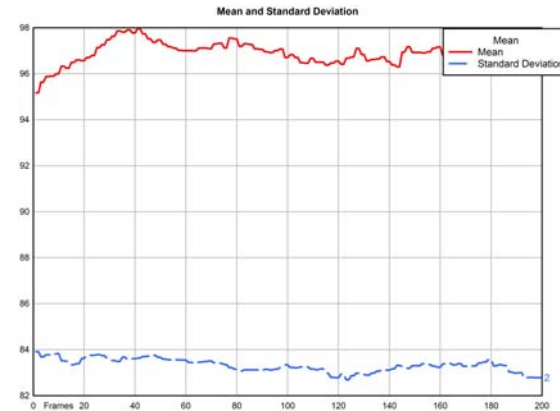


Figure 5.18b: Big Hall

The standard deviation of closed room shows a bit distortion than a bigger room environment. It found rather interesting that both mean and standard deviation of a closed room environment are much higher than that of a big hall environment.

Other Imagery:

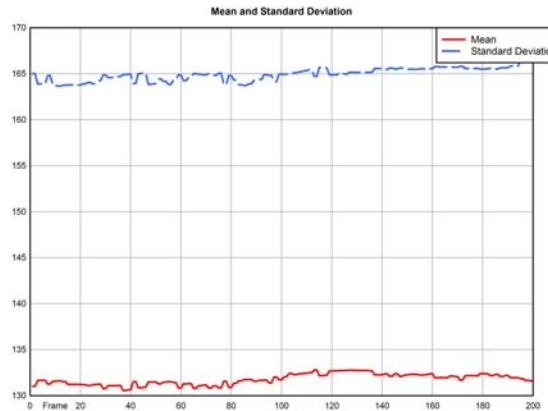


Figure 5.19a: Microscopic Images

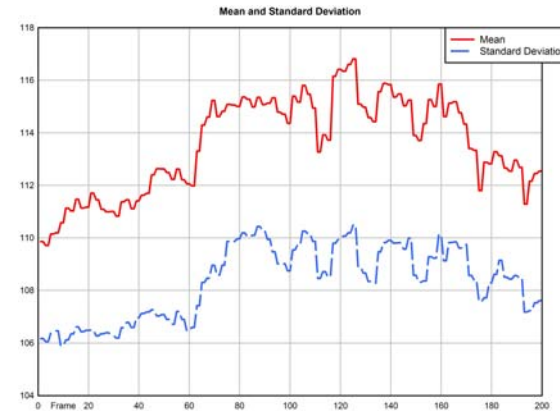


Figure 5.19b: Infrared Images



Figure 5.19c: Ultra sonogram Images

The mean and standard deviation of infrared and ultra sonogram images look rather messy than that of microscopic images, this is just because of the video sample. The samples taken are not good enough to produce a stable background. The algorithm works fine on each of them.

5.5.3 FFT:

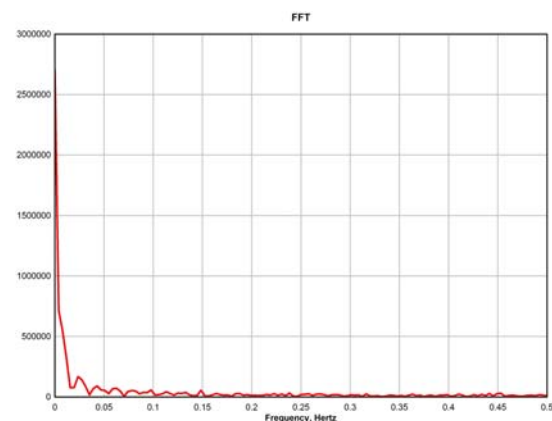
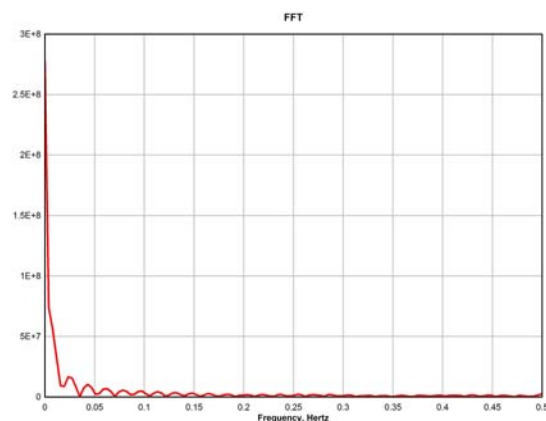


Figure 5.20a: FFT Image of Day video sequence Figure 5.20b: FFT Image of Night video sequence

Fast Fourier Transform just shows a decay at 0.2 Hz under any condition.

5.5.4 Comparison between different noise levels:

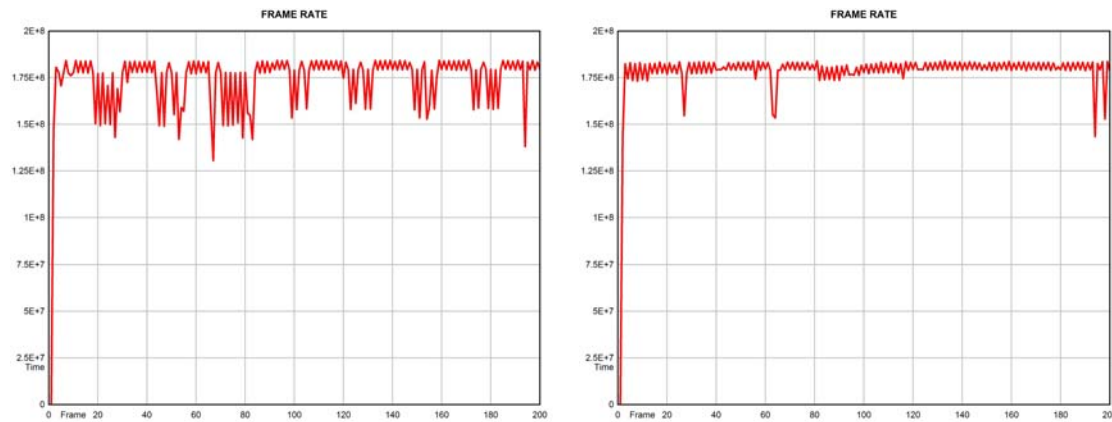


Figure 5.21a: FPS of Day video sequence; b=90.5

Figure 5.21b: FPS of Day video sequence; b=70.5

This is the fps of a busy road video sequence under different noise level. Result is pretty interesting. Not only the processing time is seen inconsistent in higher noise level but also the quality and detection result is much more acceptable in the lower noisy environment.

5.6 Discussion and recommendations:

A real-time motion detection algorithm is presented with necessary improvement. A simple shadow detection and background subtraction model is used to describe the appearance changes caused by movement in realistic light conditions. The algorithm was able to operate in various realistic conditions using cheap low-end equipment. Together with an automatic initialization procedure and re initialization when the target is lost, the algorithm seems to be a promising solution for a number of applications. The algorithm heavily reduced the dependency on relying on the initial image. Therefore, small movements in different weather condition were handled the best. However, there are situations which are yet to be dealt with.

5.6.1 Limitations:

Images of the initial phase have presented busy scene and a long run. Because of no clean images at the beginning, an artifact of the initial image left in Grimson et al.'s tracker lasted for over a hundred frames. Better segmentation can be seen from this method. The performance enhances dramatically with the shadow detection module. However, the algorithm has developed targeting specific objective to fulfill. That is why the method has got certain limitations as discussed below:

1. A stationary camera was considered while developing the algorithm, not the moving camera. The base of the camera certainly is needed to be fixed. Algorithm will not work if a moving background scene is introduced. Frequent zooming produces the same result.
2. Foreground detection method uses the connected component labeling analysis. As a result, several moving objects in the same area can be detected as one object sometimes.
3. The noise level is not generalized. It is needed to be tuned up for every different scenario. The noise level thus needed to be tuned up manually every time.
4. The whole algorithm is designed for the surveillance using Grimson et al background subtraction algorithm which uses GMM, is very sensitive to the light changes. So, if the object is too close to the camera it takes almost the whole image as a foreground therefore, the whole process is then getting pointless.
5. The algorithm can not solve the mirror object problem.
6. Panning, tilting and zooming causes a great deflection of the background. This out of scope for this project to solve.
7. Finding moving direction is not the target objective of this project.
8. Object classification is out of scope for this project as well.
9. False alarm detection is not done and left for future work.

Chapter 6: Conclusion and Future Work

6.1 Conclusion and recommendations

The field of motion detection in computer vision is quite new and emerging. The prospect is huge and there is so much to do. This thesis successfully opens the doorway for the people to explore the potentials hidden in it. It is been a topic of interest for forensic science over the last few years.

A new improved algorithm has presented for adaptive mixture models of background scene for the real-time tracking of moving objects. The algorithm runs under the framework of the real-time robust tracker proposed by Grimson et al. The results show the accuracy of the model using the update algorithm over the Grimson et al.'s tracker. A method is proposed to detect moving shadows using the existing mixture model. This significantly reduces additional computational burdens. Shadow detection need only to be performed upon pixels labeled as foreground and therefore with negligible computational overheads the moving shadows can be detected successfully. The shadow detection also reduces the effect of small repetitive motions in the background scene.

6.2 Future Works:

There are surveillance cameras which move sideways to cover up a larger area. An update in the proposed background subtraction algorithm will make it work. Some time the recognition of the object is needed. Specially, vehicles, human, suspicious objects, human face, body parts etc. are needed to be recognized. Bio technology has made life easier. Device input is taken from human as an identity. The collected data can be made more meaningful with good combination of this algorithm, retina scanning, thumb, hand and face scanning for example. Dynamic control system can be integrated with this machine vision algorithm to automate vehicles, surveillance system, life-saving apparatus, factory machineries and other automation equipment. This work foresees a huge prospect emerges even in the field of surgery, medicine and forensic department as well. This advanced field of AI has been the field of interest for quite a long period in Robotics. The new method will very easily fit in the existing robotic vision system.

Future work would involve a careful assessment of how onboard non stationary video processing cameras affects results. Cameras ordinarily supply images and videos that are compressed, as well as greatly processed away from being stationary images. Although the method does indeed work

under such processing it would be well to understand how non polarity impacts the method. A variation in-filling algorithm would likely work better than the present method for crossing shadow-edges, but would be slower.

The ultimate goal of this work is automated processing of motion of objects in un-sourced imagery such that shadows are removed. Results to date have indicated that, at the least, such processing can remove shadows and as well tends to "clean up" portraiture such that faces, for example, look more appealing after processing.

6.3 Some general open issues:

One of the monolithic goals of computer vision is to automatically interpret general digital images of arbitrary scenes. This goal has produced a vast array of research over the last 35 years, yet a solution to this general problem still remains out of reach. A reason for this is that the problem of visual perception is typically under-constrained. Information like absolute scale and depth is lost when the scene is projected onto an image plane. In fact, there are an infinite number of scenes that can produce the exact same image, which makes direct computation of scene geometry from a single image impossible. The difficulty of this "traditional goal" of computer vision has caused the field to focus on smaller, more constrained pieces of the problem. The hope is that when the pieces are put back together, a successful scene interpreter will have been created.

The importance of computer vision to the field of AI is fairly obvious: intelligent agents need to acquire knowledge of the world through a set of sensors. What is not so obvious is the importance that AI has to the field of computer vision. Indeed, it is believed that the study of vision and intelligence are necessarily intertwined. This article will look at the role that knowledge plays in computer vision and how the use of reasoning, context, and knowledge in visual tasks reduces the complexity of the general problem.

6.4 References:

- [1] E. Grimson, C. Stauffer, R. Romano, and L. Lee, "Using Adaptive Tracking to Classify and Monitoring Activities in a Site, Proc. Computer Vision and Pattern Recognition Conf., pp. 22-29, 1998.
- [2] E. Grimson and C. Stauffer, Adaptive Background Mixture Models for Real Time Tracking, Proc. Computer Vision and Pattern Recognition Conf., 1999.
- [3] Robert T. Collins, Alan J. Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burtl and Lambert Wixson, A System for Video Surveillance and Monitoring, The Robotics Institute, Carnegie Mellon University, Pittsburgh PA 1 The Sarnoff Corporation, Princeton, NJ
- [4] Zoran Zivkovic, Improved Adaptive Gaussian Mixture Model for Background Subtraction, Intelligent and Autonomous Systems Group University of Amsterdam, The Netherlands.
- [5] Sen-Ching S. Cheung and Chandrika Kamath, Robust techniques for background subtraction in urban traffic video, Center for Applied Scientific Computing Lawrence Livermore National Laboratory 7000 East Avenue, Livermore, CA 94550.
- [6] Zoran Zivkovic', Motion Detection and Object Tracking in Image Sequences, Ph. D Thesis
- [7] I. Haritaoglu, D. Harwood, and L. Davis, W4: Who, When, Where, What: A Real Time System for Detecting and Tracking People, Proc. Third Face and Gesture Recognition Conf., pp. 222-227, 1998.
- [8] Ahmed Elgammal, David Harwood, Larry Davis, Non-parametric Model for Background Subtraction, Computer Vision Laboratory University of Maryland, College Park, MD 20742, USA
- [9] Robert T. Collins, Alan J. Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt and Lambert Wixson. A system for Video Surveillance and Monitoring. The Robotics Institute, Carnegie Mellon University, Pittsburgh PA. The Sarnoff Corporation, Princeton, NJ.
- [10] Alan J. Lipton Hironobu Fujiyoshi Raju S. Patil. Moving Target Classification and Tracking from Real-time Video The Robotics Institute Carnegie Mellon University 5000 Forbes Ave Pittsburgh, PA, 5213.
- [11] Ismail Haritaoglu, Member, IEEE, David Harwood, Member, IEEE, and Larry S. Davis, Fellow, IEEE W4: Real-Time Surveillance of People and Their Activities. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 22, NO. 8, AUGUST 2000.
- [12] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In Proceedings of the European Conference on Computer Vision, 1992.
- [13] R. Cutler and L. Davis, View-Based Detection and Analysis of Periodic Motion, Proc. Int'l Conf. Pattern Recognition, 1998.
- [14] A. Elgammal, D. Harwood, and L. Davis, Non-Parametric Model for Background Subtraction, Proc. IEEE Frame Rate Workshop, 1999.

- [15] N. Friedman and S. Russell, Image Segmentation in Video Sequences: A Probabilistic Approach, *Uncertainty in Artificial Intelligence*, 1997.
- [16] I. Haritaoglu, D. Harwood, and L. Davis, W4S: A Real Time System for Detecting and Tracking People in 2.5D, *European Conf. Computer Vision*, 1998.
- [17] T. Horprasert, I. Haritaoglu, D. Harwood, L. Davis, C. Wren, and A. Pentland, Real-Time 3D Motion Capture, *Proc. Second Workshop Perceptual Interfaces*, Nov. 1998.
- [18] T. Horprasert, D. Harwood, and L.S. Davis, A Robust Background Subtraction and Shadow Detection, *Proc. Asian Conf. Computer Vision*, Jan. 2000.
- [19] S. Intille, J. Davis, and A. Bobick, Real-Time Closed-World Tracking, *Proc. Computer Vision and Pattern Recognition Conf.*, pp. 697-703, 1997.
- [20] A. Lipton, H. Fujiyoshi, and R. Patil, Moving Target Detection and Classification from Real-Time Video, *Proc. IEEE Workshop Application of Computer Vision*, 1998.
- [21] P. KaewTraKulPong and R. Bowden, An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection, *Vision and Virtual Reality group, Department of systems Engineering, Brunel University, Middlesex, UB8 3PH, UK.*
- [22] Kedar A. Patwardhan, Guillermo Sapiro and Vassilios Morellas, A Pixel Layering Framework For Robust Foreground Detection In Video, *Electrical and Computer Engineering and IMA, University of Minnesota, Minneapolis, MN 55455.*
- [23] Alan M. McIvor, Background Subtraction Techniques, *Reveal Ltd, PO Box 128-221, Remuera, Auckland, New Zealand.*
- [24] Thanarat Horprasert Chalidabhongse, Kyungham Kim, David Harwood, Larry Davis, A Perturbation Method for Evaluating Background Subtraction Algorithms, *Faculty of Information Technology, King Mongkut's Institute of Technology, Ladkrabang Bangkok 10520 Thailand, Computer Vision Lab, UMIACS. University of Maryland, College Park, MD 20742, USA.*
- [25] Philippe Noriega and Olivier Bernier, Real Time Illumination Invariant Background Subtraction Using Local Kernel Histograms, *France Telecom Research & Development 2, av. Pierre Marzin, 22300 Lannion, France.*
- [26] Alan J. Lipton, Craig H. Heartwell, Niels Haering & Donald Madden, *Automated Video Protection, Monitoring & Detection.*
- [27] Elena Salvador^a, Andrea Cavallaro^b, and Touradj Ebrahimi^a, Cast shadow segmentation using invariant, color features, ^a Signal Processing Institute, Swiss Federal Institute of Technology, Lausanne, Switzerland ^b Multimedia and Vision Laboratory, Queen Mary, University of London, London, UK.
- [28] P. Wayne Power, Johann A. Schoonees, Understanding Background Mixture Models for Foreground Segmentation, *Industrial Research Limited, PO Box 2225, Auckland, New Zealand.*
- [29] Dar-Shyang Lee, Member, IEEE, Effective Gaussian mixture learning for video background subtraction