Department of Computing

Virtual Image Sensors to Track Human Activity in a Smart House

Min Han Tun

This thesis is presented for the Degree of
Master of Science (Computer Science)
of
Curtin University of Technology

October 2007

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made. This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature Date

....

Dedicated to my beloved Mother...

and the loving memory of my Father...

Abstract

With the advancement of computer technology, demand for more accurate and intelligent monitoring systems has also risen. The use of computer vision and video analysis range from industrial inspection to surveillance. Object detection and segmentation are the first and fundamental task in the analysis of dynamic scenes.

Traditionally, this detection and segmentation are typically done through temporal differencing or statistical modelling methods. One of the most widely used background modelling and segmentation algorithms is the Mixture of Gaussians method developed by Stauffer and Grimson (1999). During the past decade many such algorithms have been developed ranging from parametric to non-parametric algorithms. Many of them utilise pixel intensities to model the background, but some use texture properties such as Local Binary Patterns.

These algorithms function quite well under normal environmental conditions and each has its own set of advantages and short comings. However, there are two drawbacks in common. The first is that of the stationary object problem; when moving objects become stationary, they get merged into the background. The second problem is that of light changes; when rapid illumination changes occur in the environment, these background modelling algorithms produce large areas of false positives. These algorithms are capable of adapting to the change, however, the quality of the segmentation is very poor during the adaptation phase.

In this thesis, a framework to suppress these false positives is introduced. Image properties such as edges and textures are utilised to reduce the amount of false positives during adaptation phase. The framework is built on the idea of sequential pattern recognition. In any background modelling algorithm, the importance of multiple image features as well as different spatial scales cannot be overlooked. Failure to focus attention on these two factors will result in difficulty to detect and reduce false alarms caused by rapid light change and other conditions.

The use of edge features in false alarm suppression is also explored. Edges are somewhat more resistant to environmental changes in video scenes. The assumption here is that regardless of

environmental changes, such as that of illumination change, the edges of the objects should remain the same. The edge based approach is tested on several videos containing rapid light changes and shows promising results.

Texture is then used to analyse video images and remove false alarm regions. Texture gradient approach and Laws Texture Energy Measures are used to find and remove false positives. It is found that Laws Texture Energy Measure performs better than the gradient approach. The results of using edges, texture and different combination of the two in false positive suppression are also presented in this work. This false positive suppression framework is applied to a smart house senario that uses cameras to model "virtual sensors" to detect interactions of occupants with devices. Results show the accuracy of virtual sensors compared with the ground truth is improved.

Published Work

The original work presented in Chapters 3 and 4 of the thesis are based on previously published work at the following refereed conference during the author's candidacy in the MSc. course.

Tun. M.H, G.A.W. West, T. Tan (2007), A Framework for False Positive Suppression in Video Segmentation, *IEEE* 14^{th} *International Conference on Image Analysis and Processing*, pp 501-505, Modena, Italy, 2007

Acknowledgements

As always, many people deserve a heart felt thanks.

First of all I would like to thank my supervisors Prof. Geoff West and Dr. Tele Tan. Through out my research Prof. West has given me much encouragement and guidance. He has invested a lot of time in me through out this journey. His advice and comments has been invaluable during my time at Curtin. Dr. Tele Tan has helped me at every turn of this research patiently filling in any gaps in my knowledge. He was always willing to meet, discuss, help, listen and encourage. Many people advised me to look for these qualities in supervisors and I am very fortunate that I found them all in both Tele and Geoff.

I would like to thank my parents. They have given me everything I could wish for in life. A special thanks goes to my Mother for all the sacrifices that she has made to give me a second chance at life. I truly appreciate you both.

My brother Chris and his dear wife Vanessa has been kind enough to share their home with me these past two years, putting up with all of my nonsense and eccentricities. Without Vanessa's constant supply of brain food, I would not have made it this far. They have made my life away from home much easier and all the more richer. Thank you both and I am forever indebted to you.

During the past two years, Curtin University has been my home and as such my thanks goes to all the denizens of the place. Wilson Leoputra, Kevin and Nan for all the helpful discussions relating to this work and sharing their little corner with me. Thorston, Wilson Waters, Alex, Graeme and the gang for all the Friday afternoon beers. I am going to miss those sessions, guys. And of course the technical staff deserve a special mention for putting up with my constant bombardment of emails to get my computer running perfectly.

Last but not least, a sincere thanks goes to Uncle Aung Gyi for taking the time out of his busy schedule to read, correct and comment on my thesis.

Contents

List of Algorithms

List of Figures

List of Tables

Chapter 1

Introduction

With the advancement in technology, computers have begun to play a more vital role in our daily lives. Nowadays, every thing from microwaves to space shuttles are controlled by computer software. Our homes are no different. Due to the advances in communications and sensor technology, the turn of the century has witnessed the concept of a smart home elevated to a whole new level.

A smart home is a dwelling where the appliances can communicate their status to a controller or with each other. A smart home really differs from a conventional home in its communications infrastructure. The major systems in the home such as heating, lighting, fire control, etc., can communicate and pass commands to each other and be controlled remotely. This type of smart home forms the first generation.

A second generation smart home has seen the birth of a more sophisticated system. Targeted towards the elderly and the infirm, these homes are capable of monitoring the well being of the occupants. Simple sensors placed around the house help to monitor the location and activity of the occupants. Homes such as these are termed Health Smart Homes. There are

many Health Smart Home systems available currently ranging from the simple use of burglar alarm type sensors to sensors based on highly sophisticated Radio Frequency Identification (RFID) systems. Their capabilities also range from those that simply detect movement to those that can actually aid us in our daily chores.

Although sensor technology has advanced to a stage where sensors have now become very small and easy to deploy, they still present significant challenges in a smart home setting. In a Health Smart Home, occupants are becoming more demanding and their needs may vary over time. As more sensors are added to meet these needs, the task of maintaining them have become more of a challenge. The cost of implementation will also increase with the rise in the number of sensors.

An alternate and cheaper solution to using a multitude of sensors is to use video surveillance and image processing techniques to achieve the same outcome. The computer vision community has faced this challenge and has been developing vision based systems to detect and analyse human movements and behaviours. West et al. (2005) explored the use of virtual sensors to detect human/appliance interactions in a smart home. Instead of a multitude of sensors placed around the home, they use a single ceiling mounted camera to capture the daily activities of the occupant. Polygonal regions are placed in areas of interest on the video stream captured by the camera. Image processing techniques are then applied to these areas to detect activities. These polygonal regions are termed virtual sensors.

The idea behind this project was to detect the interactions between people and appliances and to model the normal daily activities of people in a kitchen. This model will ultimately allow them to use artificial intelligence techniques to analyse and detect abnormality in a person's behaviour. For example, the system would be able to differentiate between someone who has forgotten to switch off the stove or is in the process of making breakfast.

The first step towards this aim is to build a reliable vision system that can detect human movements and thus their interactions with appliances. West et al. (2005) used virtual sensors to achieve this. By applying background modelling techniques on these regions, they are able to detect and monitor the movements in these regions. This system is proven to work well in detecting interaction between humans and appliances under normal lighting conditions in the environment but begins to fail under conditions of rapid light change and other environmental changes.

As in many computer vision systems, this failure is caused by the inability of the background modelling techniques to quickly adapt to rapid light changes. Under this condition, most background modelling algorithms begin to produce large areas of false positives.

1.1 Aims

This thesis extends the virtual system of West et al. (2005) by exploring ways of reducing the number of false positives produced under adverse lighting conditions with the aims being:

- To develop a framework that will allow easy integration of existing algorithms to suppress false positives. For example, when the light change in the environment is spatially non-uniform or rapid.
- To explore the suitability of edges and texture in false positive suppression. Since most false positives are caused by light changes, illumination independent image features should be considered to reduce false positives. To a certain extent, edges and texture are more robust against light change than any other image feature.

- To integrate edge and texture feature information into virtual sensors using the developed framework allowing it to function under adverse environmental conditions.

While there are systems that work towards achieving illumination invariance, they still cannot cope with spatially non-uniform light change. This is a particularly common problem in indoor environments such as the smart home. Simply switching lights on can have adverse effects on these systems such as the detection of large areas of the background as foreground. The failure of many systems to operate under these conditions highlights the fact that multiple image features must be used to achieve illumination invariance and thus suppress false positives.

1.2 Approach

Given the aims of this thesis, the approach gives utmost importance to the reduction of false positives in every stage. The flow of work is then divided into the following stages.

(i) Framework - An application framework based on the idea of sequential pattern recognition is developed.

(ii) Edge Analysis - The use of edge based features in the reduction of false positives is explored.

(iii) Texture Analysis - Textural features are deemed to be more resistant to light change. The use of texture in the reduction of false positives is explored at this stage.

(iv) Evaluation - The methods developed are extensively tested on videos captured in a laboratory environment as well as in a typical kitchen of a suburban home.

1.3 Significance

The significance of this work is in the methods used to suppress false positives under various environmental changes such as spatially non-uniform or rapid light changes. The false positive reduction techniques presented in this thesis will allow many applications to function under adverse lighting conditions. Particularly in smart homes, it would create a better detection of events and objects. This improved detection method will enhance the ability of the system to provide better, more reliable assistance to the occupants.

Object detection and segmentation forms the fundamental task every computer vision system must perform. Improving the accuracy of detection in this step will have profound effects on any additional processing that needs to be done. Reduced false positives means that higher level processing tasks such as object recognition, behaviour analysis, etc., will yield better results which in turn will greatly enhance further research into the extraction of high level semantics from video scenes. For example, we may infer what a person is doing from a video scene as "Person puts pot on the stove".

The architectural framework presented in this work provides a means of easily integrating existing imaging algorithms as well as future developments. The framework is designed based on a plugin architecture. All image processing algorithms inheriting from the *ImageOp* class are automatically recognised by the system at start up. Simply by inheriting from this class, new algorithms can be easily integrated.

1.4 Thesis Outline

The organisation of this thesis is as follows.

In Chapter 2, a more extensive treatment is made concerning the background modelling techniques used today. The need to consider multiple image features in dealing with false positives is discussed. Following this, an overview of research in the area of image segmentation and illumination invariant background modelling is given.

In Chapter 3, sequential pattern recognition and its applicability to false positive suppression is explored. By analysing a number of frames in a video sequentially, we may find, classify and ultimately remove false positives. Through the sequential pattern recognition approach the optimal combinations of algorithms to achieve maximum false positive suppression with minimal loss of true foreground is found.

Chapter 4 forms the first step towards false positive suppression. The use of edge based features to correctly classify false positives is explored in this chapter. Chapter 5 takes an in-depth look at using texture to achieve similar results. Both of these chapters present inital results of these techniques.

In some instances of light change, background modelling techniques produce large areas or blobs of false positives. Chapter 6 presents a technique to segment true foreground from these large blobs. Further experimental results of applying edge and texture based techniques, on their own and in combination, to videos with rapid light change are presented in Chapter 7.

Finally, in Chapter 8, conclusions are given along with some directions for future work.

Chapter 2

Literature Review

2.1 Introduction

Computer Vision is a relatively new field of research. It is defined as "the science and technology of machines that see" (Wikipedia, 2007). Although there has been earlier investigations, computer vision as a field of study did not flourish until the late 1970s mainly because computer were only able to begin to process large datasets such as images around that time.

Research from a variety of fields slowly evolved and formed into a new branch known as Computer Vision and thus it is a highly cross-disciplinary effort with intimate ties to artificial intelligence, neurobiology, physics and so on. Figure 2-1, adapted from Wikipedia (2007), shows the multi-disciplinary nature of Computer Vision.

The classical problem in Computer Vision, is that of determining whether or not the image data contains some specific object, feature or activity. These are trivial tasks for a human to solve but this is not so for a computer. Computer Vision applications must be highly specific to a task to be performed and are very hard to generalise. Although Computer Vision systems

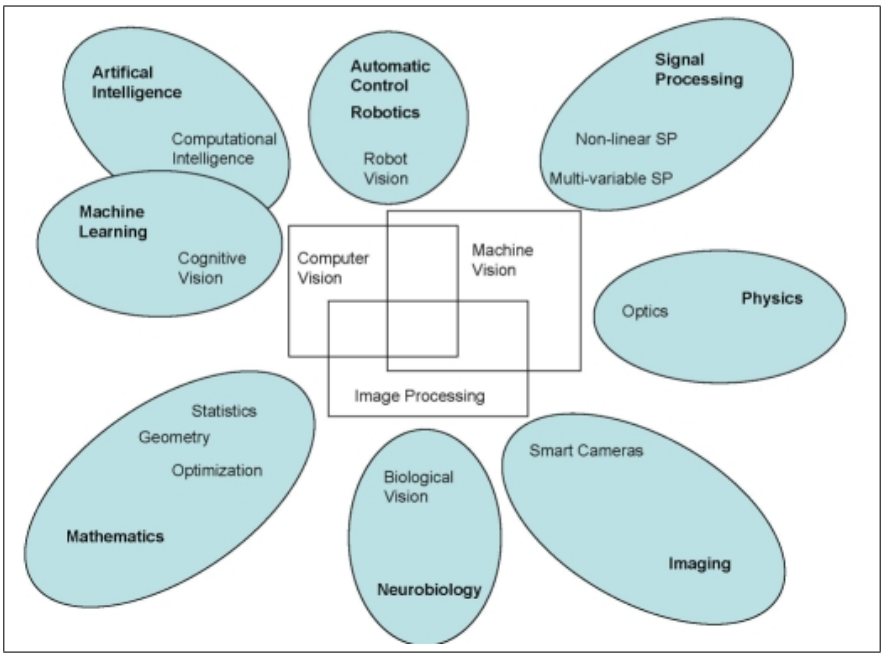


Figure 2-1: The cross-disciplinary nature of Computer Vision (adapted from Wikipedia, 2007)

are highly application dependent, there are typical functions that may be found in many vision systems:

i Image acquisition: A variety of image data can result, from multi-spectral images to image sequences, depending on the type of sensor (camera) used.

ii Pre-processing: Before any extraction of data can be done, the image typically needs to be processed to satisfy certain criteria such as:

- Re-sampling
- Noise reduction
- Contrast enhancement

iii Feature extraction: Image features at various levels of complexity can be extracted from the image such as edges, textures, corners, blobs and so on.

iv Detection and Segmentation: Regions of interest are labelled and segmented from the rest of the image.

v High level processing: This stage typically operates on a small set of blobs to extract high level semantics such as the recognition of objects, pose estimation and so on.

The degree of success achieved by any high level processing task is highly dependent on the quality of blobs it receives as input. Only if correctly segmented components are received can these higher level tasks produce correct results. This is why object detection and segmentation forms a critical step in many computer vision applications.

In recent literature, a large number of techniques and algorithms have been introduced in the area of background modelling and object detection as it has created widespread interest in the Computer Vision community due to its applicability in diverse disciplines. Background modelling finds its roots in simplistic methods like change detection. Change detection consists of a set of methods to detect pixels that have changed in a set of images that are taken over disparate time steps. However, in this thesis, we have focussed our attention on processing video streams where a sequence of images with much information can be obtained. Background modelling techniques are targeted more towards this type of scenario (Radke et al. 2005).

2.2 Background Modelling and Subtraction

The goal of a background modelling algorithm is to determine which pixels belong to the background while segmenting the foreground pixels into different objects. The majority of background modelling techniques calculate the background model using the Mixture of Gaus-

sian approach. In this approach the probability of observing an intensity value at $I_t(x, y)$ at a location (x, y) and at time t is the weighted sum of K Gaussian distributions (Radke et al. 2005).

At each time step, the pixel's class is determined by the Gaussian distribution that is most likely to explain the pixel's intensity value. Several authors (Kanade et al. 1998; Huwer and Niemann 2000; Cavallaro and Ebrahimi 2001) have proposed adaptive background modelling techniques using a single Gaussian distribution. Pixels that lie more than $+/-$ two standard deviations away from the mean are classified as foreground pixels. These techniques typically require only a few seconds of video of the background for them to initially learn the background. The mean and variance of background models are updated using simple adaptive filters capable of accommodating for slow changes in illumination.

Wren et al. (1997) proposed a method of tracking people and interpreting their behaviour known as Pfinder. Pfinder is built on a fundamental assumption that it will be processing a relatively static scene, for example an office, the majority of the time. This assumptions means the background scene will be less dynamic than the object to be detected and tracked. Pfinder includes a background estimation algorithm that made use of not just a single Gaussian distribution for the background model but also different Gaussian distributions for different foreground objects. Classification of pixels as foreground or background is accomplished by finding the model with the least Mahalanobis distance (Radke et al. 2005). People and moving objects can then be robustly detected by simply finding the distribution that most likely explains their intensity values.

Stauffer and Grimson (1999) improved on the Pfinder model to allow the background to be modelled by a mixture of Gaussian. By having multiple Gaussian distributions modelling the background. They were able to handle moving backgrounds such as waving trees or

waves. This model is perhaps the most widely used algorithm in today's computer vision applications particulary in visual surveillance systems. It is a representative algorithm of an adaptive background model utilising normal distributions to model multi-modal background scenes. Piccardi (2004) termed this background model as an *"Image Model"* as it provides a description of both foreground and background.

The probability of observing a certain pixel value, x, at time t by using a Mixture of Gaussians is:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(x_t, \mu_{i,t}, \Sigma_{i,t}) \tag{2.1}$$

where:

$$\eta(x, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{\frac{1}{2}(X_i - \mu_t)^\tau \Sigma^{-1} (X_i - \mu_t)} \tag{2.2}$$

and:

$$\omega_{i,t} = (1 - \alpha)\omega_{i,t-1} + \alpha(M_{k,t}) \tag{2.3}$$

with each K Gaussian describing one and only one foreground or background object. Typically K is set to 3 or 5. $M_{k,t}$ is 1 for the matching distribution and 0 otherwise, α denotes the learning rate and μ is the mean.

The Gaussians are multi-variate describing each of the Red, Green and Blue channels of a colour image. Since this model describes both foreground and background objects, a criterion must be given to distinguish between the foreground and background distributions. Stauffer and Grimson (1999) first ranks the distributions based on the ratio ω_i/σ_i. The first B distributions to meet the criteria of

$$\sum_{i=1}^{B} \omega_i > T \tag{2.4}$$

where T is a predefined threshold, is taken as the background distributions. The fundamental assumption here is that the higher and more compact the distributions, the more likely they are to describe background.

With each incoming frame two things must simultaneously occur. Firstly, the observed value, x_i , must be assigned to the best matching distribution. Secondly, model parameters must be estimated and updated. The distribution matching is approximated using:

$$(x_i - \mu_{i,t})/\sigma_{i,t} > 2.5 \tag{2.5}$$

The first ranking distribution satisfying this condition is taken as a match for x_t. The parameters $(\omega_{i,t}, \mu_{i,t}, \sigma_{i,t})$ are then updated only for this matching distribution. If no match is found, the lowest ranking distribution is replaced with a new one centred at x_i with a low weight and high variance. This assumes the pixels belong to a new object and hence need a new distribution to represent it.

Toyama et al. (1999) described an algorithm called Wallflower that makes use of a Weiner filter to predict a pixel's current value from a linear combination of its k previous values. Pixels whose predicted value is far worse than the expected error are deemed to be foreground pixels. The prediction coefficients are adaptively updated over time. Wallflower was designed to solve many of the canonical problems suffered by many vision systems ranging from moved objects and sleeping person problems, to shadows and light switch problems. This is the first system that attempted to analyse images at different spatial scales in the context of background modelling and subtraction.

At the region level, this system uses a series of frame differencing techniques to detect regions of movement. The first step in doing this is to take the difference between the current frame and the previous frame. Intersecting regions between the difference frame at time t and the difference frame at time $t-1$ are then taken as the foreground region. A histogram of the foreground regions found is computed and back projected locally so that false negatives are pushed into the foreground. Their most significant contribution by Toyama et al. (1999) is their five principles of background modelling:

i *No Semantics:* Semantic differentiation should not be handled at the background modelling level. Background modelling and subtraction is simply a component of larger systems aiming to achieve high level scene understanding.

ii *Proper Initial Segmentation:* Background subtraction should pick out all objects of interest allowing higher level algoithms to identify, track or ignore these objects.

iii *Stationarity Criteria:* An appropriate pixel level stationary criterion needs to be defined and pixels that satistify this criterion are declared background and ignored.

iv *Adaptation:* The background model must adapt to both sudden and slow changes in the background.

v *Multiple Spatil Scales:* Background models should take into account changes at different spatial scales.

These principles stemming from their development of Wallflower highlight several points that all background modelling techniques should take into account. Their fifth principle is one of the most significant because certain changes such as non spatially uniform light change cannot be detected at pixel level while these are common causes of false positives typically in indoor environments.

Haritaoglu et al. (2000) take a different approach from the above mentioned techniques of background modelling. Rather than having a mixture of Gaussian distributions model the background, they analyse a segment of video with only the background and calculate the minimum m and maximum M intensity values of the scene as well as the largest interframe difference δ. If any pixel is more than δ away from m or M it is considered foreground. They developed the algorithm W4 for the purpose of detecting and tracking multiple people to monitor their activities.

This system uses a two stage approach to background modelling. A pixel-wise median filter is applied to the first several seconds of the video, typically 20 to 40 seconds, to differentiate between moving and stationary pixels. The next stage processes the stationary pixels to build a background model.

This model also integrates two different techniques to update the background model to make it adapt to changes in the background. The two stages of background update are:

- **Pixel-based update:** This step updates the background model occasionally to deal with lighting changes. In the outdoor environments this system is targeted for, rapid changes in light rarely occur. A periodic update of the background model is sufficient to deal with slow light changes such as those caused by the time of day problem or clouds obscuring the sun.
- **Object-based update:** This step updates the background model for physical changes, such as abandoned objects.

In order to determine which method to use for a background update, the W4 constructs and maintains the following (Haritaoglu et al. 2000): a detection support map (gS), which maintains the number of times a pixel has been detected as background. A motion support map (mS) that represents the number of times a pixel has been foreground and a change history map (hS) that contains the amount of time, in frames, a pixels was last classified as foreground. The detection support map is used to determine the part of the background that should be updated with the pixel based approach. All three maps are used to work out the regions where object based update is to be used. During tracking W4 computes the background model separately for pixels that are classified as foreground and for those that are marked as background.

As with the majority of intensity based background modelling techniques, the W4 model also suffers from false positives when rapid illumination change occurs. To deal with this problem, W4 relearns the background model when more then 80% of the image is detected as foreground.

Several researchers (Cucchiara et al. 2003; Greenhill et al. 2004) propose the use of median value of the last n frames as the background model. This median value model maintains a set S of N samples for each pixel. The background model is the one that minimises the distance to all other pixels according to the equation (Greenhill et al. 2004):

$$D(a, b) = max(|a.c - b.c|) \tag{2.6}$$

where $c = R, G, B$, i.e c can take on the values of R, G and B. A threshold T is used to identify which image pixels are foreground pixels. The drawback of this method is that it has a relatively short term memory. Any evidence is replaced by new evidence in roughly $N/2$ samples. The complexity of updating the model is $O(N^2)$ but once this is done, classification is performed in constant time (Greenhill et al. 2004).

The SAKBOT system developed by Cucchiara et al. (2003) improves the Median Value model by adding adaptivity and object-level reasoning. A separate background model B_t is maintained from the statistical background model B_t^s. Foreground regions are labelled as moving objects, shadows, ghosts, etc. An adaptive update factor is included, adding B_t to S and weighting the distances used for the median function by a factor ω_b. Greenhill et al. (2004) further enhances this model to improve image quality during adaptation using a history of the mean illumination for $N/2$ images. A correction factor cf is added to the threshold T to adjust it to mean illumination shifts.

An approximation of the background probability distribution function can be obtained by the histogram of the most recent values of the pixels classified as background. As the number of

samples are limited, this approximation has some drawbacks: the histogram might provide poor modelling of the true unknown pdf. Elgammal et al. (2002) introduced the idea of using a nonparametric kernel density estimate to model the intensity of the background and each foreground object using the following equation:

$$P(x_t) = \frac{1}{n} \sum_{i=1}^{n} \eta(x_t - x_i, \Sigma_t) \tag{2.7}$$

The conventional nonparametric background modelling techniques have several limitations the first of which is its high computational complexity. Nonparametric background models rely on maintaining a foreground as well. This is unrealistic in real-world situations where there may not be enough pixel changes to train a foreground model. The third limitation is that when an object has similar colour as the background model, that foreground object cannot be detected. Luo et al. (2006) takes nonparametric methods a step further addressing the high computational complexity of these methods. They use an importance sampling method to generate a small subset of representative background samples from a large original dataset. This subset is then used to train the background modelling algorithm.

Cheng et al. (2006) proposes an online discriminative approach to background subtraction. Their work is based on one-class support vector machines (1-SVM) and introduces Implicit Online Learning with Kernels. The input feature dimensions are mapped to Hilbert feature space. A separating function $F(\cdot)$ is predicted as a weighted combination of examples where the past examples are associated with different weights derived formally from the large margin principle. To avoid having to maintain past samples when evaluating the separating hyperplane, a modification to obtain kernel classifiers based on limited support N, with $N << T$, and approximating f by heuristically truncating past examples is proposed.

As in previous techniques, this model seems to be dealing with a sum of Gaussians but there are substantial differences. Here, each Gaussian describes just one sample data with the same

Σ_t for all models whereas in previous methods the Gaussian describes the main mode of the pdf and is updated over time. Pixels with intensities that are not based on the density estimate are classified as foreground. An additional step that compares $I_t(x, y)$ with learned densities in a small neighbourhood provides a way of suppressing false positives.

Model updating is obtained by simply updating the buffer of the background values by selective update. The complete model estimation requires the estimation of Σ_t. This is a key challenge in Kernel Density Estimation (KDE). In Elgammal et al. (2002), the variance is calculated in the time domain by analysing a set of differences of two consecutive values. There are two drawbacks that this method faces. First, the background subtraction stage requires N Gaussian values with N typically being 50/100. This means that much storage space is needed and subtraction is very time consuming. Secondly, updating the KDE smoothing factor is also a very slow process.

Kim et al. (2005) uses a quantisation/clustering method to construct a codebook representing the background of a scene. The key features of this method are: (1) resistance to artifacts of acquisition, (2) ability to handle lighting changes, and (3) adaptivity and compression of the background model. For each pixel, a codebook consisting of one or more codewords is built from a long sequence of observations. Samples at each pixel are then clustered based on their colour distortions and a brightness bound. Detection is done through testing the difference of the current image from the background model with respect to colour and brightness differences.

Most background modelling techniques produce large areas of false positives when the environmental conditions such as illumination change. These algorithms will adapt to the changes. However, the image will contain many false positives during adaptation highlighting the need for further enhancement in background modelling algorithms.

2.3 Illumination Invariant Modelling

All of the methods mentioned above perform well under conditions of near constant lighting. However, rapid illumination changes produce large disparity between current pixel values and the background model resulting in large areas of the image being detected as foreground. The detection of these large areas interferes with subsequent processing tasks such as object segmentation and tracking. This is why it is extremely important to find ways of reducing false positives during the adaptation phase of background modelling techniques particularly in indoor environments.

The approach of Greenhill et al. (2004) using median values and adaptive thresholding performs better than Gaussian Mixture Models and Kernel Density Methods. Gordon et al. (1999) makes use of colour information as well as range to estimate the background and detect foreground. Colour (R,G,B) and range (Z) values of each pixel are recorded in a multidimensional histogram over a sequence of frames. A clustering method is then used to fit the data with an approximation of a mixture of Gaussians.

Ivanov et al. (1998) describe a method for illumination invariant background modelling using multiple fixed cameras. A disparity map of the empty background is built offline. This map is then used to find matches between the primary image (i.e. the image from the main camera) and the auxiliary image. Any pixel in the two images having the same colour and luminosity values are considered to be the background.

Illumination normalisation is another approach to background subtraction and change detection (Matsushita et al. 2004). To make the system robust to changes in illumination, an attempt is made to remove the illumination effects from the image sequence. The input image sequence is normalised in terms of the distribution of incident lighting. This approach

is composed of two parts:

(i) Estimation of intrinsic images. An image is assumed to be composed of two parts reflectance and illumination. Using the *Maximum Likelihood* estimation method, the illumination component of the image is extracted. These illumination images are then constructed into an illumination eigenspace using Principal Component Analysis. This process is carried out offline.

(ii) Once the illumination eigenspace is built, the illumination image is estimated directly from the input image using the illumination eigenspace. Shadow interpolation is also performed to obtain accurate illumination images.

Change detection is based on change in the reflectance image. This method requires a vast amount of training samples to handle large changes in illumination. The basic idea behind the Matsushita et al. (2004) approach is similar to Homomorphic filtering proposed by Toth et al. (2000). In their approach, Toth et al. (2000) apply a log function to the pixels to make the reflectance and illumination components additive. Assuming illumination changes slowly over space, a low pass filter is applied to filter out the illumination component and obtain the reflectance component. This reflectance component represents the background structure and is used to detect foreground in the image.

Order consistency is another good attribute to use for illumination invariant background modelling. Xie et al. (2004) uses order consistency coupled with statistical measures for camera noise to develop a foreground detection algorithm. The method is based on the observation that the ordering of pixel values is preserved under a range of physical conditions for which the mappings from scene radiance to image measurements are monotonic. It is unlikely that this consistency is observed for object pixels in the scene. Therefore, order consistency can

be used to discriminate between change caused by illumination and the appearance of an object. While this method works on most of the image, some neighbourhoods such as occlusion boundaries violate the fundamental assumptions and hence it does not work well in these areas.

Huang et al. (2007) improves on the Mixture of Gaussian (Stauffer and Grimson 1999) approach. In addition to colour information, motion information is also taken into account for modelling the dynamics of the background. Colour and motion are assumed to be independent and modelled by K Gaussian distributions respectively. The Bhattacharyya distance is used to measure the similarity between a region R_m segmented from the region and the corresponding region R_bm represented by the background model. The classification of foreground and background blobs is then taken as a graph labelling over a Region Adjacency Graph (RGA) based on a Markov Random Fields (MRF) statistical framework.

2.4 Edge and Texture Based Background Modelling

The lesson to be learnt from the aforementioned techniques and algorithms is that it is very difficult to model the background based on intensity values alone. Doing so will, more often than not, produce negative results under unstable environmental conditions. It is therefore important to consider other image features such as edges and texture to model the background.

Heikkila and Pietikainen (2006) presents a texture based background modelling algorithm based on a technique called Local Binary Patterns (LBP). LBP is a powerful means of describing texture. The operator labels an image block by thresholding the neighbourhood of each pixel with the centre pixel value and considering the result as a binary number. The

LBP operator is defined by the equation:

$$LBP(x_c, y_c) = \sum_{p=0}^{p-1} s(g_p - g_c)^{2^p} \tag{2.8}$$

where g_c corresponds to the grey value of the center pixel and g_p to the grey values of the P neighbourhood pixels. The function $s(x)$ is defined as:

$$S(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{2.9}$$

Figure 2-2 (adapted from (Heikkila and Pietikainen 2006)) illustrates the operator. The LBP codes calculated over an image block can be used as a texture descriptor for the block. LBP is invariant to monotonic changes in grey scale.

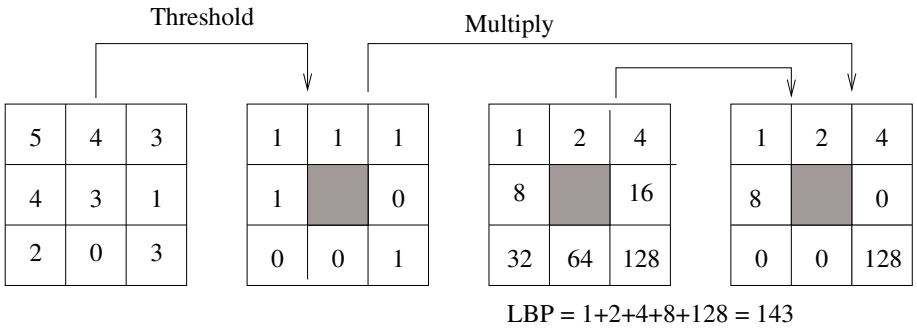


Figure 2-2: Example LBP calculation

The input image is discretised into equal blocks by using a partially overlapped grid structure. The LBP histogram computed over a circular region of radius R_{region} around the pixel is used as a feature vector. The block process is considered as a time series of LBP histograms. The background model for each pixel consists of an adaptive LBP histogram. The LBP histogram computed for a pixel in each new scene is compared against the K histograms using histogram intersection as a proximity measure. When a match is found, the best matching histogram is updated with new data.

An edge based region growing approach to video segmentation is presented by Fan et al. (2004). This paper outlines three significant achievements.

i The growing seeds are selected at the edge region automatically.

ii The region growing procedure is performed in the homogeneous regions and stops automatically when the seeds stop growing.

iii The detected object boundary is the true object boundary.

In this approach the input image is first smoothed using an averaging filter to remove the effects of high frequency noise. The Canny edge detector (Canny 1986) is then applied to the image to detect edges. The edge region is the place from where the region growing seeds will be selected. Therefore the edge region should surround the single pixel edges derived by the Canny operator. Four Sobel operators are applied to the image and the average of the filters is calculated. The region growing seeds are then selected automatically based on the edge image and edge region image. This algorithm is tested with images containing non-uniform illumination changes giving promising results.

Jabri et al. (2000) presents an edge based approach fused with colour information to locate and segment people in video scenes. The background is modelled into two parts: (1) an edge model and (2) a colour model. For each colour channel a mean and standard deviation of that colour component is maintained. The Sobel edge detector is applied to the videos to obtain the edge model. This yields the horizontal edge difference H and vertical edge difference image V. A weighted mean of H_i and V_i as well as the standard deviations are maintained. The colour channels and edge images are subtracted separately to obtain the background subtraction. A median filter is applied to the result to remove any salt and pepper noise.

Background information is very important in segmenting images. If sufficient background information is available, segmentation can be done more robustly. Alsaqre and Baozong (2003) suggests that spatial edge information is more important and effective than temporal information. Moving objects in an image have a coherence property thus a range of filters can be applied in the segmentation process. The author also suggests that edge information has two advantages. Firstly it is more resilient to light changes. Secondly it is independent of the number of objects in the scene.

Alsaqre and Baozong (2003) start their system with edge detection. The frame difference edge can be defined as:

$$DF_n = \psi(|F_n - F_{n-1}|) \tag{2.10}$$

where F_n and F_{n-1} represent current and previous frames respectively. The background subtraction edge map can be defined as follows:

$$BS_{n-1} = \psi(|F_{n-1} - B|) \tag{2.11}$$

where B is the background frame, and $\psi(.)$ is obtained by the Canny edge operator. From these images moving edges are extracted. Once the edge maps are defined, the object is ready for extraction. The first and last edge points are located for each row of the image and the pixels in between are assigned to the object.

A two stage approach to removing false alarm pixels caused by light change is presented by Zeng and Lai (2007). A mixture of gaussians approach is used as the initial background model. Once the initial pixels are removed, the image is analysed with a two stage classification approach to identify the false positive pixels. The first step is the pixel-wise classification. A number of manually labelled images is collected to train the classifier to recognise false positive pixels caused by light change. This is a labour intensive task. The pixel wise classification is able to remove pixels detections caused by slow lighting changes. To handle fast light changes, Zeng and Lai (2007) uses a region-wise classification.

The remaining pixels are grouped into regions based on their gain value using a region growing technique. The assumption is that the pixels belonging to the same light change should have the same gain value. The regions are then compared to the corresponding regions in the background and the average gain ($\overline{gain}$)is used for classification. Pixel regions with $\overline{gain} < T_{gain}$ is defined as the background region, where T_{gain} is a predefined threshold set at 0.5.

2.5 Summary

In this chapter, a discussion into the currently available background modelling techniques is given. While most of these methods function well under near constant environmental conditions, they tend to generate foreground images with large areas of false positives when the conditions change. Adaptive methods can recover from the effects of environmental change but the image quality during the adaptation phase is far from satisfactory. The following chapter discusses a framework to improve image segmentation accuracy during the adaptation phase by using additional image processing methods to remove false positives.

Chapter 3

Sequential Pattern Recognition

3.1 Introduction

Crucial to the approach to removing false alrms is the use of sequential pattern recognition. Pattern recognition typically involves two steps as depicted in Figure 3-1: (1) acquisition and pre-processing of raw data and (2) classification of captured data. During the pre-processing stage, raw data is captured and prepared for analysis. This stage may involve cleaning and formatting of the data and the extraction of features to be analysed. Once the features of interest are extracted, the classification stage begins. In this stage, the feature vectors are analysed and classified as belonging to one class or another.

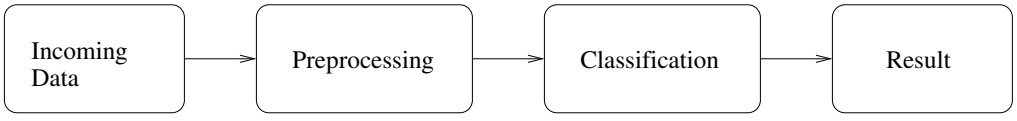
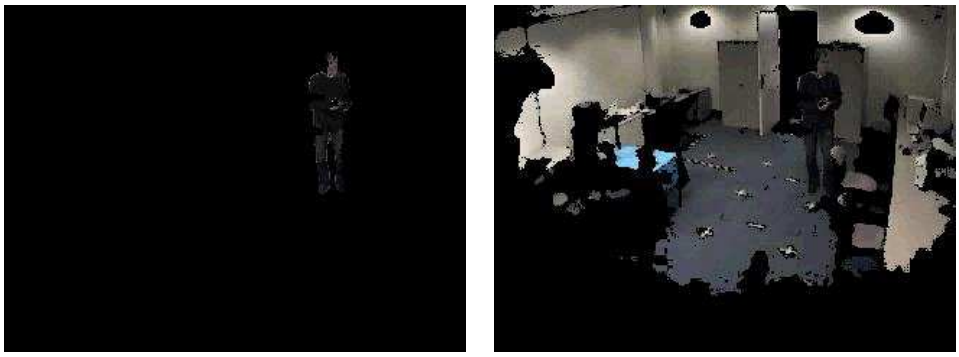


Figure 3-1: A Pattern Recognition Approach

In Computer Vision, this type of classification is used extensively to achieve many outcomes one of which is to identify moving objects. The typical feature vectors to be analysed constitute pixel intensities across video scenes. Based on the variations of these intensities, objects

are classified as foreground or background. Figure 3-2 presents the result of the Mixture of Gaussians method, a widely used background modelling algorithm (Stauffer and Grimson 1999) that eliminates pixels classified as background and leaves the foreground pixels.

The pattern recognition model used by this method is similar to that of Figure 3-1. This model works very well for computer vision applications working under stable environmental conditions but produces a lot of false positives when the conditions change. Figure 3-2 (a) presents the output of the the Gaussian Mixture Model of Stauffer and Grimson (1999) under normal lighting conditions and (b) the result at an instance of light change.



(a) (b)

Figure 3-2: Gaussian Mixture Model Detected Foreground (a) Foreground under stable conditions (b) Foreground at instance of light change

The reason for this change is that this model learns the feature vectors of previous frames of video and tries to classify the pixels in the current frame. When the illumination in the environment changes, for example, the features vectors change drastically and the classification model used in the previous time step no longer applies. Due to this, false positives are introduced. This highlights the fact that features may change at any instant and must be repeatedly analysed using different methods to get correct classification.

3.2 Sequential Pattern Recognition in False Positive Suppression

A sequential pattern recognition method is used in this thesis as a solution to the problem of misclassification in background modelling techniques. Sequential pattern recognition is typically used when patterns in the data may be costly or difficult to extract. In such cases, sequential pattern recognition analyses pattern/feature vectors sequentially, making decisions whether to continue the extraction of features or if it already has enough for classification. The data may also contain features that are best exploited in some linear fashion. This type of data is exemplified by time series data, on-line hand written data and video sequences (Smith and Yau 1972).

In Figure 3-3, the two distributions show the detection of feature values for foreground objects (Class C_1) and background (Class C_2). Where the overlap of the two distributions occur, misclassification will be produced. Foreground objects to the right of T_1 will be detected as background and background objects to the left of T_1 will be misclassified as foreground objects. T_1 is where the minimum error rate occurs. Moving the decision boundary to T_2 results in all foreground objects being detected correctly but results in a corresponding increase in the false alarm rate (background classed as foreground). The alternative is to use T_2 as the decision boundary and use increasingly comlex methods to eliminate false alarms.

In an attempt to minimise the false positives, i.e the shaded region in Figure 3-3, the pattern recognition model in Figure 3-1 is extended to include multiple classification techniques to correctly classify objects. The extracted feature vectors are iteratively analysed using these different algorithms removing as many of the false positives as possible while preserving the true foreground. The modified model is depicted in Figure 3-4.

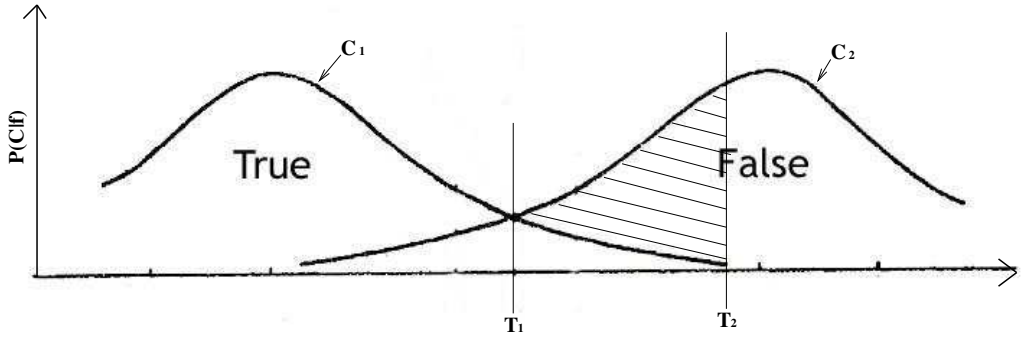


Figure 3-3: The Two Class Pattern Classification Problem.

(adapted from Smith and Yau (1972))

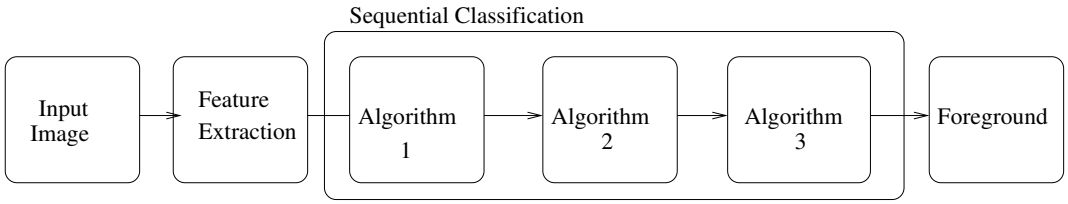


Figure 3-4: Sequential False Positive Suppression

The ability to apply these algorithms sequentially is facilitated by the fact that as we progress through the algorithms, the data to be processed will decrease. As each algorithm is applied some of the false positives are expected to be removed, reducing the amount of data to work on. Therefore algorithm 2 in Figure 3-4 will have less data to process than algorithm 1. This allows an increase in complexity of algorithms as the image progresses through the false positive (FP) removal steps. As shown in Figure 3-5 lesser data means more complex algorithms can be applied while maintaining a reasonable throughput.

Taking this into account, a framework for false positive suppression is developed and illustrated in Figure 3-6. False positives can be caused by a number of changes in the environment. This framework allows us to integrate any number of algorithms in any combination to deal with false positives. The main advantage of this framework is the ease with which different algorithms can be integrated.

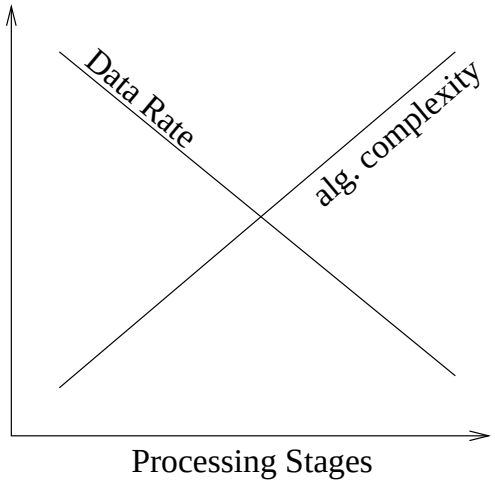


Figure 3-5: Data rate vs program complexity

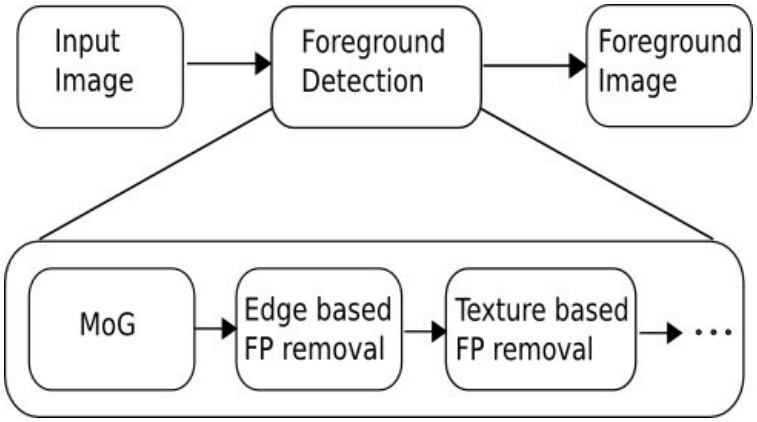


Figure 3-6: Sequential false positive removal

3.3 Implementing the Sequential Architecture

Java is chosen as the implementation language and a plugin design concept is chosen as the underlying architecture for this framework. The reasoning behind this choice is that plugins allow the algorithms to be loaded into memory when needed. This design also gives the user the ability of choose which algorithms to load. The primary advantage to this design is that additional algorithms can be integrated into the system with minimal effort.

The key to a successful plugin design is a minimal set of methods all plug ins must implement so that the incorporating application may search for and find plug ins. This is achieved by creating an Interface called ***ImageOps***. All image processing algorithms must inherit from this interface. When integrating new algorithms it is a simple matter of creating a wrapper class that inhertis from ***ImageOp*** and create a method that accepts and returns an image. The core implementation of the algorithm itself can remain untouched. When the system starts up, Algorithm 1 iterates through the plug ins folder and loads all the algorithms that inherit from ***ImageOps***.

Algorithm 1: Algorithm to load plugins

Input: Path to folder containing java archive files

Output: An array containing available plugin objects

```
begin
    AlgoList ⟵ null
    fileCount ⟵ getFileCount()
    for i ⟵ 0 to fileCount do
        file ⟵ readFile(i)
        if file instanceof ImageOp then
            AlgoList ⟵ file
end
```

Once everything is loaded, the user is presented with the list of available algorithms in the form of menus. The user selected algorithms are then placed in a queue and executed in the order of selection. This gives the user the ability to change the combination of algorithms to get desired results. Implementing this in the form of plugins also allows for easy integration of new algorithms in the future.

This system is implemented in java using the Java Media Framework (Sun 1999) to control video streams and the Java Advanced Imaging (Sun 2000) packages as a basis to implement the image processing algorithms. The framework is developed on a Linux platform and makes use of a Java Advanced Imaging package that is optimised for Linux.

3.4 Summary

This chapter introduced the idea of using sequential pattern recognition to suppress false positives. The main advantage of using this architecture is that multiple image processing algorithms can be integrated together to achieve a certain outcome, in this case, false positive suppression. The fact that computational time may be maintained through out the process is an added bonus.

In order to achieve false positive suppression, we advocate edge based and texture based methods. Based on this edge and texture information the differentiation between false positive and true foreground is made. The details of the edge based method are discussed in the next chapter.

Chapter 4

Edge Based False Positive Suppression

4.1 Introduction

Feature detection plays an important role in Computer Vision and image processing. Feature detection refers to the methods of computing abstractions of image information making local decisions at each pixel as to whether there is an image feature of a given type at that pixel. There is no universal or explicit definition of an image feature and the exact explanation may depend on the context and application in which feature detection is performed. Depending on the context of its use, a feature may be interpreted as (Davies 2005):

i **Edges:** are points where there is a boundary between two image regions that can be of arbitrary shape. Edges are usually defined as sets of points with strong gradient magnitudes. The wide variety of edge detection algorithms can be grouped into search based and zero-crossing based techniques. This image feature is extensively used in this

Chapter.

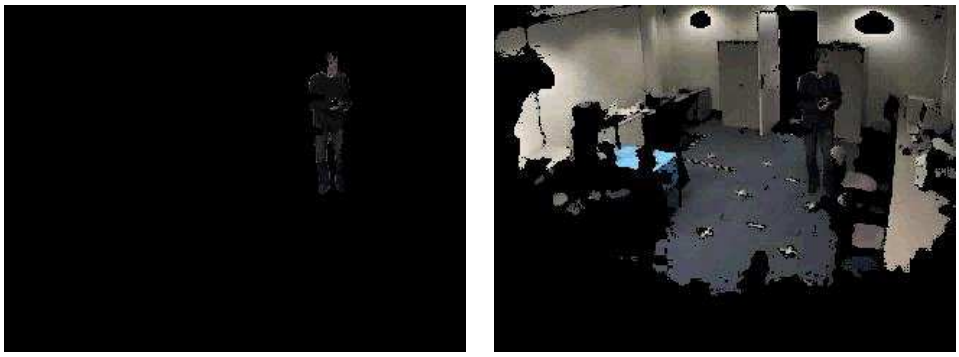
ii **Corners/points of interest:** During earlier years of development, these algorithms first detected edges and then analysed them to find rapid changes in direction (corners). Later developments no longer required explicit edge detection. Instead they look for high curvature in the image gradient to detect corners.

iii **Blobs/regions of interest:** Blobs are a natural way to describe image structures in terms of regions as opposed to corners that are more point-like. A variety of blob detection and tracking algorithms are also available.

iv **Ridges:** A ridge descriptor computed from a gray-level image can be seen as a generalisation of a medial axis. It is harder to extract ridges from an image than edges, corners or blobs.

Once the features have been detected, a local image patch around the feature may be extracted. The result is known as a feature vector. The feature vectors can then be used in a wide variety of tasks including but not limited to false positive suppression. In Computer Vision and image processing, false positives can be produced under a variety of conditions. For example, when the illumination in the room changes rapidly or the objects to be segmented occupy regions with non-uniform lighting, conventional background modelling and segmentation algorithms give poor results. As mentioned before, the Mixture of Gaussians method (Stauffer and Grimson 1999) in particular over segments in cases like these and thus gives large regions as foreground. Figure 4-1 shows an example of this over segmentation.

In this chapter an edge based approach is proposed to reduce the large areas produced by the Mixture of Gaussians algorithm. There are several advantages in using edges for this purpose. The most important of which is that edges are rather insensitive to illumination changes. Several edge detection methods such as Sobel edge detector, Canny edge detector (Canny



(a) (b)

Figure 4-1: (a) Result of MoG before light change (b) Result of MoG after light change

1986), Marr-Hildreth edge detector (Marr and Hildreth 1980) and Gabor filters (Turner 1986) were explored for use in this work. Gabor filters tend to be more resistant to environmental changes and can produce rich descriptions of textural and edge features (Ji et al. 2004; Kamarainen et al. 2006). Thus Gabor filters are used in this work.

4.2 Gabor Filters

Gabor filtering is a widely used approach to feature extraction, particularly for texture analysis in images. The two dimensional Gabor function can be thought of as a two dimensional sinusoidal signal of a particular frequency and orientation modulated by a Gaussian envelope. Gabor (1946) proposed a simple cell receptive field model consisting of harmonic oscillations within Gaussian envelopes (Turner 1986). A number of authors describe the use of a bank of Gabor filters to extract local image features (Ji et al. 2004; Kamarainen et al. 2006).

In signal analysis a function $f(x)$ defines its temporal (spatial in the case of an image) behaviour and its Fourier transform $F(\omega)$ denotes the frequency (spectral) behaviour. Ji et al. (2004) presents a Gabor expansion for time-frequency analysis based on the work by Gabor. This filter localises image information in both the time and space domains as well as the

frequency domain resulting in a simultaneous description of both temporal and spectral behaviour of a signal f. In this thesis a similar Gabor filter based on the work by Turner (1986) is used.

There are, however, a few important points to note in Turner's approach to Gabor filters. For all frequencies, the Gaussian envelope is kept constant. Turner (1986) suggested that this has the effect of reducing the bandwidth of the filters with increasing frequency. The are several advantages in having the same size filters. Firstly, the computational complexity is much less in applying the same size filters to an image than applying those of different sizes. More importantly, same size filters allow a more direct comparison of features across frequencies.

The Gabor filters also have a tendency to produce a DC offset response to uniform illumination. The sine component of a complex Gabor function has zero mean. However, the cosine component does not. This is the reason for the DC offset response in Gabor filters (Movellan 2006). This DC response is normalised by subtracting the mean value from each pixel.

The reasoning behind the use of Gabor filters is that they can produce a very strong edge map of an image and are more resistant to light changes than other edge detection algorithms. They can also be tuned to include or remove certain frequencies of an image. An input image is filtered with a bank of Gabor filters. There are several forms of 2-D Gabor filters. The filter used in the work is defined in Turner (1986) as:

$$G(x, y) = e^{\{-[(x-x_0)^2+(y-y_0)^2]/(2\sigma)^2\}} * \sin \omega[(x \cos \theta - y \sin \theta) + \rho] \tag{4.1}$$

where x_0 and y_0 specify the centre of the Gaussian. For the sinusoidal plane wave, ω is the frequency, and ρ is the phase of the plane wave. Once $G(x, y)$ is obtained the filtered image is created by $r(x, y) = G(x, y) * i(x, y)$ where $*$ denotes two dimensional convolution and $i(x, y)$ is the input image. This process can be applied at different frequencies and orientations resulting in a filter bank. This gives us a stack of filtered edge maps containing the object

edges which are defined at different frequencies and in different orientations.

Two Gabor implementations are tested. One in the spatial domain and the other in the frequency domain. In the spatial domain, the filter is applied as a 32×32 pixel sliding window. Every pixel in the cell is sampled using this sliding window. This places a restriction on the size of the cells. The cells need to be larger than the sliding window. In frequency space the filter and the cell needs to be of the same size. This is further restricted by the fact that the size of the filter must be a power of two if we are to achieve some performance gain from the Fourier transform. Otherwise, it does not give any noticeable improvement in speed over the spatial implementation.

Algorithm 2 outlines the creation of Gabor filters in the spatial domain. $SizeX$ and $SizeY$ denote the vertical and horizontal size of the filters. The values of a and f are the angle and frequency respectively. The size of the Gaussian envelope is defined by s. In Algorithm 2 line 9 generates the exponential portion of the Gabor filter: the Gaussian envelope. Lines 10 to 12 create the filter.

Algorithm 3 then uses algorithm 2 to generate a bank of Gabor filters for every angle and frequency pair. If for example, two frequencies and six angles are used, a bank of 12 filters is generated, and $minF$ and $maxF$ specify the range of frequencies to process. A two dimensional convolution between the input image and the filter bank is performed to obtain the edge map. Figure 4-2 shows a set of Gabor outputs in the frequency domain and a Gabor filtered image for all orientations of the filter set.

Algorithm 2: Gabor Filter Generator

Input: SizeX,SizeY,a,f,s

Output: $G(x, y)$ for an angle and frequency pair

begin

$mFrequency \longleftarrow (f * PI/2.0)$

$mYO \longleftarrow SizeY/2$

$mXO \longleftarrow SizeX/2$

for $i \longleftarrow 0$ *to* $sizeY$ **do**

for $j \longleftarrow 0$ *to* $sizeX$ **do**

$y \longleftarrow (i - mYO)$

$x \longleftarrow (j - mXO)$

$exponent \longleftarrow exp(-(x^2 + y^2)/s)$

$sincos \longleftarrow (mFrequency * (y * cos(a) - x * sin(a)))$

$G(x, y).real \longleftarrow (exponent * sin(sincos))$

$G(x, y).imag \longleftarrow (exponent * (cos(sincos) - exp((-1.0 * PI^2)/2.0))$

end

(a) (b)

Figure 4-2: (a) A Gabor Filtered Image (b) A Frequency Representation of Gabor Filters (D.C. centered at the middle of the image).

Algorithm 3: Gabor Filter Bank Generator

Input: sigma,NoOfAngles,NoOfFreqs,minFreq,maxFreq

Output: $G(x, y)$ for all angle and frequency combinations

begin

$SizeX \longleftarrow 32$

$SizeY \longleftarrow 32$

$s \longleftarrow (sigma * PI^2)$

for $i \longleftarrow 0$ *to* a **do**

for $j \longleftarrow 0$ *to* f **do**

$angle \longleftarrow ((i * PI)/NoOfAngles)$

$freq \longleftarrow (minF + (j * (maxF - minF))/NoOfFreqs)$

$G(x, y) \longleftarrow gaborfilter(SizeX, SizeY, angle, freq, sigma)$

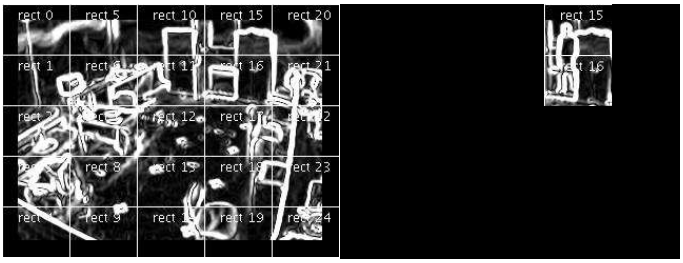
end

4.3 Removing False Positives

The edge maps produced by Gabor filters are used to determine the regions of the foreground image which are actually background. To accomplish this task, a grid of 5×5 cells is first applied to the image. The reason for having to apply this grid is that false positives are produced as large blobs that contain both true foreground and background. There needs to be a way of determining which portions of the blob are background. By dividing the blobs into small squares, the discrimination between true foreground and background can be made by analysing each square.

The Gabor filter is applied in each cell of the foreground image. Gabor produces edge maps in each frequency and orientation combination describing the features dominant in that particular orientation and frequency. In this work, these feature images are combined to get rotational invariance.The same is done for the corresponding cells in the reference image. The reference

image is the background image produced by Mixture of Gaussians. Corresponding cells of the reference image and the foreground image are then subtracted. The assumption here is that Gabor filters will produce the same responses if the two corresponding cells contain background only.



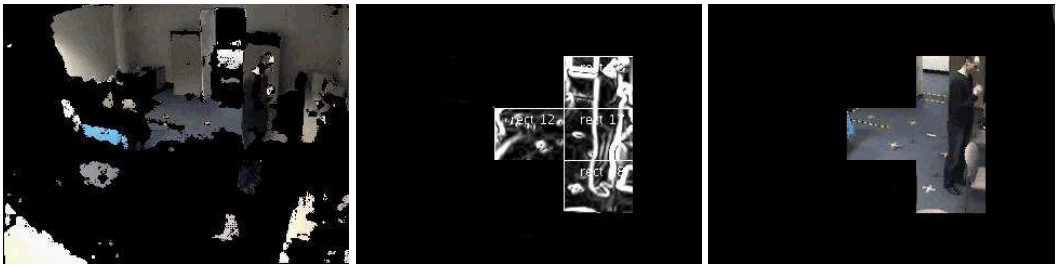
(a) (b)

Figure 4-3: (a) Gabor filtered and tiled background image (b) Foreground Image with background regions removed

A consequence of using an artificial grid is that in the cleaned image, the objects have a crude shape because of the large cells rather than the silhouette of the object. This is shown in Figure 4-4. Post processing of these outputs must be performed to regain the silhouette of the object. We will describe a post processing technique based on Graph Cuts in Chapter 6.

Figure 4-4 (a) shows the results of the Mixture of Gaussians method for instances of light change and (b) shows an edge map with false positives removed using algorithm 4. Although this method still leaves some false positives, they are significantly less than that of the Mixture of Gaussians on its own showing the usefulness of the sequential pattern recognition approach.

Once a clean edge map is obtained the objects of interest are extracted from the original image. Figure 4-4 (c) shows the final result of this image. As can be seen from this image, the resultant foreground contains the object of interest but retains the shape of the rectangular cells and not the silhouette of the object.



(a) (b) (c)

Figure 4-4: Image produced by Mixture of Gaussians at an instance of light change (a), a cleaned image using edge based method (b), and the cells showing the moving object of interest (c).

Algorithm 4: Edge based false positive removal

Input: CurrentImg,BackgroundImg

Output: A vector containing Gabor responses

begin

```
// create an artificial grid on image
BLOCKSIZE ⟵ 10
TILEWIDTH ⟵ imgWidth/BLOCKSIZE
TILEHEIGHT ⟵ imgHeight/BLOCKSIZE
TILES ⟵ createTiles(BLOCKSIZE,TILEWIDTH,TILEHEIGHT)
// get Gabor responses for each tile
for t ∈ TILES do
    fgPix ⟵ grabPixels(t, currImg)
    bgPix ⟵ grabPixels(t, bgImg)
    fgPix ⟵ applyGabor(fgPix)
    bgPix ⟵ applyGabor(bgPix)
    fgPix ⟵ subtract(fgPix − bgPix)
    // set the tile area in current image to the subtracted results
    CurrentImage ⟵ setImage(fgPix)
```

end

4.4 Summary

A discussion into using edge features for false positive suppression is made in this chapter. This technique can be used to reduce false positives in any number of situations and not just those caused by light changes. A set of Gabor filters is used to obtain an edge map of the current and previous frames. An edge strength comparison is then made to determine the regions of false positives and identified regions are removed leaving the objects of interest.

As a side effect of using an artificial grid, the remaining blobs retain a rectangular shape rather than the silhouette of the object. Techniques to overcome this problem are discussed in Chapter 6. The next chapter explores the uses of texture in false positive removal. Texture of an image is assumed to be the same, to a certain extent, regardless of the illumination in the environment (Tian et al. 2005). A false positive removal method is developed based on this fundamental assumption. In this work, both Sobel gradient (Tian et al. 2005) and Laws texture energy measures (Davies 2005) are used to achieve false positive suppression.

Chapter 5

Texture Based False Positive Suppression

5.1 Introduction

The texture of an image is an important feature and has been used in a variety of applications ranging from image segmentation to searching media databases. Image texture, in spite of its widespread use is a hard property to define. Websters's dictionary defines texture as "the visual or tactile surface characteristics and appearance of something". Only the visual dimension of texture is considered in computer vision. Sonka et al. (1999) describes texture as "properties that represent the surface or structure of an object". Davies (2005) presents texture as the surface of composition of the object that gives rise to visual properties. Essentially texture is the spatial variation in pixel intensities (gray values).

Texture is an important feature of images and has been the focus of research at least for the past four decades (Tan 1993). Texture plays an important role in many applications and

environments. In visual inspection of manufacturing processes, texture is used to inspect the products at every stage of production. Tan (1993) highlights several areas such as geology, petrography, metallography and lumber processing where texture plays an integral role.

One immediate use of texture is for recognising image regions using texture properties. For instance, we can identify cotton wool, straw matting, etc. in an image. This is known as texture classification. One could also find texture boundaries even if we cannot identify what that texture surface represents. This is the second type of problem texture aims to solve and is known as texture segmentation. The goal here is to obtain a boundary map of regions separated by different textures. Texture synthesis is used for image compression applications as well as in computer graphics where the realistic rendering of objects is important (Sonka et al. 1999).

Texture consists of texture primitives or texture elements called texels (Sonka et al. 1999). Texture is usually interpreted to measure qualities such as smoothness, coarseness, regularity and direction. Texture is heavily used in image segmentation and a wide variety of texture analysis methods are available:

i **Statistical Methods:** characterise smooth, coarse grainy, etc. using traditional statistical values such as mean and variance.

ii **Structural Methods:** detect patterns based on image primitives such as lines or other repeating structures in the image.

iii **Spectral Methods:** are based on properties of the Fourier transform such as symmetry, directionality.

Robust detection of moving objects in a video stream is a significant issue in video surveillance.

Environments with rapid light changes presents considerable challenges for object detection and analysis. Several authors (Heikkila et al. 2004; Tian et al. 2005; Heikkila and Pietikainen 2006) suggest that texture is more resistant to environmental changes such as illumination change than any other image property. This is a very useful fact in suppressing false positives in adverse environments. In this chapter, two texture based approaches to suppressing false positives due to light change are presented.

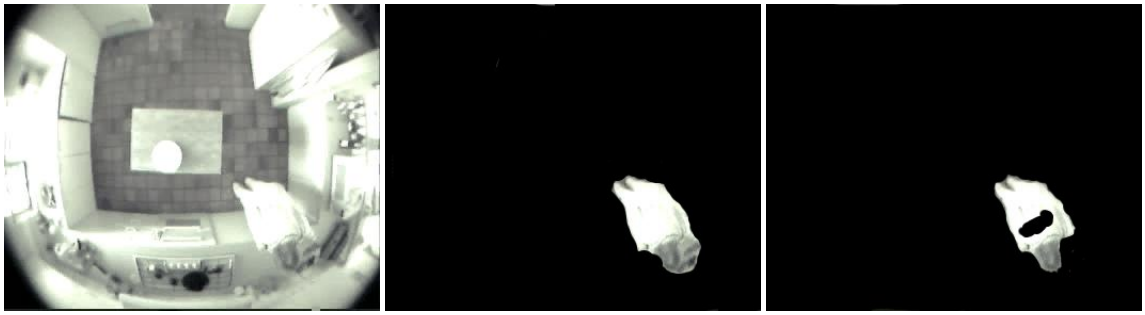
5.2 Texture Gradient Approach

Tian et al. (2005) presented a novel idea to extend the approach of Stauffer and Grimson (1999) to background modelling. The main extension presented in this paper is to suppress false alarms caused by light changes. Most methods that try to handle light changes do so by modifying the background modelling algorithm itself. The disadvantage of this approach is that only one image feature is considered in modelling the background. Tian et al. (2005) adds additional processing on top of a conventional background modelling technique. Pixel intensities are used to model the background and texture is used to suppress false positives produced at instances of light change.

The idea behind this approach is that texture of an image, regardless of illumination change, remains the same. Therefore, background pixels and false alarms should have similar textural features. Tian et al. (2005) suggest that the gradient value is less sensitive to light changes and can be used to derive local texture difference measures accurately. The texture similarity measure is defined by:

$$S(X) = \frac{\sum_{\mu \in W_x} 2||g(u)|| \cdot ||g_b(u)|| cos(\theta)}{\sum_{\mu \in W_x} (||g(u)||^2 + ||g_b(u)||^2)} \tag{5.1}$$

where W_x denotes the $M \times N$ neighbourhood centered at pixel X, g and g_b indicate the



(a) (b) (c)

Figure 5-1: (a) Original Image (b) Hand segmented ground truth (c) image segmented with texture approach

gradient vector of the current frame and reference frame respectively, and θ is the angle between the vectors. The gradient vectors $g(X) = (g^x(X), g^y(X))$ and the partial derivatives $g^x(X)$ and $g^y(X)$ are obtained from applying the Sobel operator to the images.

The foreground false positive areas, caused by rapid lighting changes, will have the same texture as the reference frames and therefore $S(X) \approx 1$. The foreground pixels with $S(X) \geq T_s$ are removed from the foreground image. In their work Tian et al. (2005) set the threshold, T_s, at 0.7.

This approach works well on images that have highly textured backgrounds. Their own experiments are carried out on scenes containing checkered floors as the background. On scenes that are less textured this method produces less promising results. The results are also dependent on the size of the window used to analyse the image and the foreground objects. For example, when applied on scenes from a ceiling mounted camera where the objects are larger than the evaluation window, this approach creates holes in the middle of the objects as can be seen in Figure 5-1.

5.3 Laws Texture Energy Measure Approach

In 1980 Laws presented a novel texture energy approach to texture analysis. He designed several filters designed to highlight points of high texture energy in images using simple filters such as Gaussian, edge detection and Laplacian type filters (Davies 2005). Laws's texture energy measure determines texture properties by assessing average grey-level, edge, spot, ripples and waves in texture (Sonka et al. 1999). Laws approach to texture analysis formed the basis for much work in later years. His approach was highly efficient.

Laws's introduced three sets of masks in his work. Amongst the three 3×3 masks are used in this work. Laws' created a simple set of 1 by 3 filters, namely:

$$\begin{aligned} L_3 &= [1\ 2\ 1] \\ E_3 &= [-1\ 0\ 1] \\ S_3 &= [-1\ 2\ -1] \end{aligned} \tag{5.2}$$

The initial letters of these masks represent *L*ocal averaging, *E*dge detection and *S*pot detection respectively. Additionally Laws created *R*ipple and *W*ave detection with the 1×5 masks. By multiplying these vectors with the first term as the column vector and the second as the row vector, a 3×3 filter mask is obtained. This creates a complete set of nine masks. These masks contain one whose components do not sum to zero. This is less useful in texture as it will give results dependent on pixel intensities (Davies 2005). Once these masks are obtained the following stages are used to get the texture description of the image.

i **Convolution:** Given an image of N by M dimensions that we want to perform texture analysis on, i.e. extract texture features from each pixel, each of the 25 convolution kernels (masks) are applied to the image. This results in a set of 25 $N \times M$ greyscale

images. These images form the basis for texture analysis.

ii **Windowing:** This step replaces every pixel in the 25 $N \times M$ images with a Texture Energy Measure (Laws 1980). This is done by looking in a local neighbourhood around each pixel and summing together the values using the formula:

$$TEM(x, y) = \sqrt{\sum_{i=-7}^{7} \sum_{i=-7}^{7} (grey(x + i, y + i)^2)} \tag{5.3}$$

This is essentially a smoothing function using a 15×15 window. An alternate method using absolute values is also suggested by Laws (1980). Although the absolute value approach is faster, using the squared magnitude corresponds to true energy and give a better response. Figure 5-2 shows the results of applying three of the nine Laws covolution masks. These convolution masks are obtained by multiplying the 1×3 masks together. For example, E3L3 mask is obtained when the E3 filter and a transpose of the L3 filter are multiplied together.

iii **Normalisation:** All convolution kernels are zero mean with the exception of one. Laws (1980) suggested the use of this kernel as a normalisation kernel. Normalising each of the TEM images with this kernel, pixel by pixel, results in contrast normalisation for that feature.

iv **Combine Similar Features:** Most texture applications do not need to use directionality of textures. In this case, similar features can be combined to remove directionality bias from the features.

Pietikainen et al. (1983) later proved that Laws Texture Energy Measures are more powerful than measures based on pairs of pixels such as co-occurrence matrices.

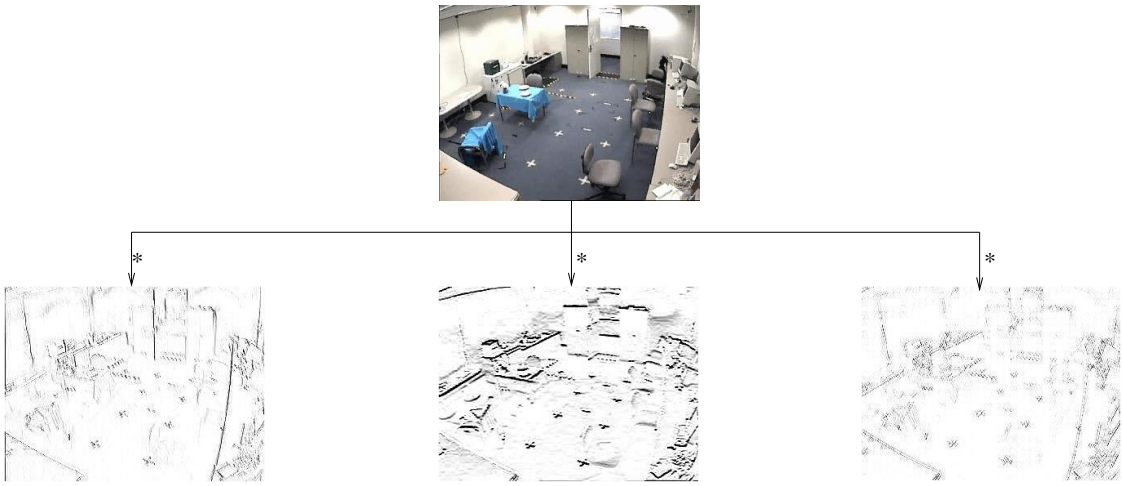


Figure 5-2: Output of Laws convolution and windowing operations, * indicates convolution. (a) result of applying the E3E3 mask (b) result of applying the E3L3 mask and (c) result of applying the E3S3 maks.

5.4 Removing Background Regions Based on Texture

As mentioned in the above stage, applying Laws filters to the image produces five images each describing one pixel in five different texture descriptions. The background image produced by the Mixture of Gaussian method (Stauffer and Grimson 1999) is filtered with the Laws texture masks. The current incoming frame containing false positives are also processed with the same filters. The resultant images are then compared for similarity. Assuming that false positive pixels caused by light changes have the same texture as the corresponding background pixels, false positives can be removed.

Each new image is processed with the Laws texture approach and the 14 texture energy measure images created. These images are then compared with those of the reference image. Background pixels detected as foreground are then removed from the current image.

5.5 Summary

In this chapter, a texture based image analysis approach is presented to suppress light changs in video scenes. The main reason for using texture in this work is that it is somewhat more resistant to illumination changes than any other image property. Textural features are collected using Laws's texture energy measures and these texture properties are compared to those of the reference frame. Pixel regions having the same textural properties of that of the reference frame are removed from the foreground mask.

Using texture for this purpose produces more stable results as shown in detail in Chapter 7. This is primarily due to the fact that texture features are more stable under illumination changes. However, it is harder to compare texture if the background is not highly textured. When the foreground object has similar texture as the reference frame, that foreground object tends to be mistaken as a false positive. Objects with high texture, for example patterns on a shirt, on a low textured background and vice versa will be picked up since all we are looking for are differences in texture.

Chapter 6

Refined Object Segmentation with Graph Cuts

6.1 Introduction

Chapters 4 and 5 discussed the use of edge and texture based image features to reduce the amount of false positives in object detection by enhancing the Mixture of Gaussians method. In using these techniques, a grid is placed on the image to reduce the size of blobs to be analysed. The cells in the grid deemed to be the background are eliminated and those that are foreground retained. As a result of using this grid, the remaining foreground objects retain a block shape rather than the silhouette of the object.

The challenge now is to automatically regain the shape of the object. A number of methods to localise object boundaries have been developed over the years such as snakes, active contours and geodesic active contours. (Boykov and Funka-Lea 2006). The methods all come with their own set of features. A method based on graph theory known as Graph Cuts (Kolmogorov

and Zabih 2004) is used in this thesis. Graph Cuts work by segmenting nodes in the graph based on the cost of edges between them.

The advantage Graph Cuts have over other segmentation methods is that when the edge cost between the pixels are defined, it can be based on any image property ranging from colour to texture. If an object has a particular texture for example, the edge costs between the pixels belonging to that object can be defined based on its texture. The same is true for colour. Therefore this segmentation method can be adapted to give optimal results depending on the available information as well as the environment being monitored.

6.2 Graph Cuts

Graph cuts are an efficient and powerful tool for image segmentation and its use in Computer Vision was first proposed by Boykov et al. (2001). The segmentation of objects may involve some form of labelling of pixels with the constraint that these labels vary smoothly and preserve object boundaries. These problems can be naturally defined as energy minimisation problems and implemented using two minimisation algorithms called *swap* move and *expansion* move.

Boykov et al. (2001) uses these minimisation techniques to develop a maximum flow algorithm for use in Graph Cuts. The reader is directed to Boykov et al. (2001) for a detailed discussion and implementation of the maximum flow algorithm. The following section discusses the use of Graph Cuts for object segmentation.

Let $G = (V, E)$ be a weighted graph with two vertexes known as terminals. A cut $C \subset E$ is a set of edges such that the two terminals are separated in the graph. The sum of the

edges in C forms the cost of the cut C. The minimum cut problem is to find the cheapest cut that will segment the two terminals (Boykov et al. 2001). Figure 6-1 (a) shows a directed graph created from the image. Each pixel is represented as a node in the graph. Graph Cut techniques then perform a cut between the *source* and *sink/terminal* nodes, i.e a s/t cut, to segment the image.

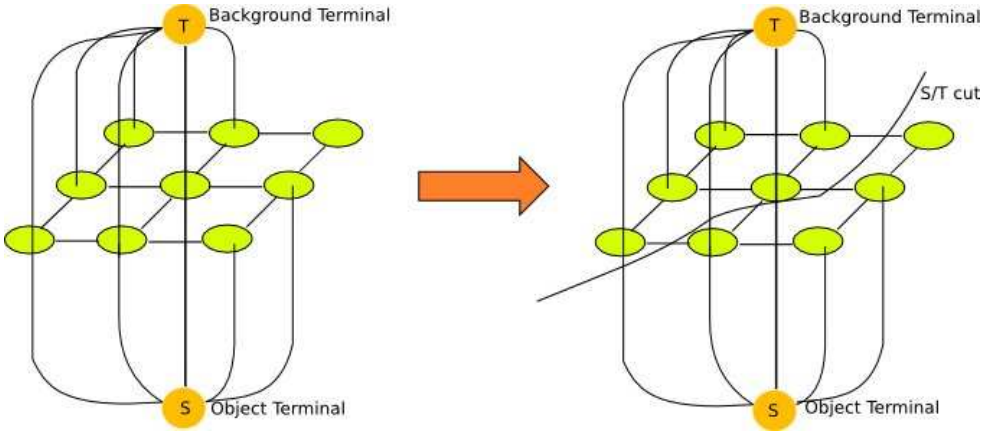


Figure 6-1: (a) A directed graph $G = (V, E)$ (b) An s/t cut on G

In this work, we make use of the max flow algorithm for graph cuts developed by Boykov and Kolmogorov (2004). The input image is constructed as a weighted graph taking each pixel as a node. In Algorithm 5 we first create a graph $G = (V, E)$ based on the input image. Each pixel P is represented as a node in the Graph G. Line 5 in Algorithm 5 accomplishes this task.

Once the nodes are created, the edges, E, between each node need to be created. The weights of each edge defines the correlation between the nodes it connects. Therefore we must take particular care in defining these weights. The connection between each node and the terminal nodes must also be defined. The correlation, i.e the strength of connection, between pixels or nodes are defined based on their colour. The assumption is that the pixels in a particular region belonging to the same object will have similar colour. In order to create

this relationship the colour of each pixel is first converted to its gray value using:

$$g = (.299 * P_r) + (.587 * P_g) + (.114 * P_b) \tag{6.1}$$

where P is the pixel under consideration and P_r, P_g and P_b represent the red, green and blue components of the pixel respectively. Line 12 of Algorithm 5 performs this task.

The gray values of the neighbouring pixels or nodes are also calculated. The eight neighbourhood of a pixel is represented in Figure 6-2. The centre cell represents the current pixel under consideration. Each cell beginning from the immediate right of the centre cell going clockwise represents the position of neighbourhood nodes relative to the current node.

(-1,-1)	(0,-1)	(1,-1)
(-1,0)	(x,y)	(1,0)
(-1,1)	(0,1)	(1,1)

Figure 6-2: Eight Neighbourhood Pixels

By adding the coordinates of each cell to (x, y), the position of the current pixel, we can obtain the real position of each of the neighbouring pixels. Once these coordinates are obtained, the colour value of each pixel can be extracted. Lines 15 and 16 in Algorithm 5 calculate the position of each neighbouring pixel. The difference between the gray values of the current pixel and its neighbour then defines the weight of the edge between them. If the difference is close to zero, the probability of both pixels belonging to the same object is very high. If the difference value is high, then they are deemed to be unrelated.

This suggests that there is a reciprocal relationship between the edge cost and gray difference.

The edge cost is calculated with:

$$cost = e^{(((1.0*(G_i - G_n)^2)*DIST)/2\sigma^2)*100} \tag{6.2}$$

where G_i and G_n are the gray values of current and neighbouring pixels respectively. $DIST$ is the distance between the current node and the neighbouring node. If the neighbour is diagonally from the current pixel, the $DIST$ is $\sqrt{2}$ and otherwise 1. The value of σ is twice the mean intensity of the image. Lines 20 to 23 of Algorithm 5 calculate the cost and creates the edges with those costs.

A connection between each node and the terminal nodes, namely source and sink nodes needs to be made and their edge weights defined. In order for graph cuts to work, foreground and background seeds need to be placed in the image. The weighting between *source* and background seed nodes needs to be high, and low with *sink*. For foreground seeds the reverse is true. If a node is neither background or foreground seed its connection with *source* and *sink* gets equal weighting. Ways of defining foreground and background seeds are discussed in Section 6.3.

6.3 Seed Definition and Placement

Graph cuts group pixels based on the weights of the edges that connect them. It then segments the image into two groups as belonging to the *source* and *sink* nodes. Pixels that are defined as foreground seeds have high connections to the *source* node and those pixels with high connections to these seed nodes get cut as belonging to the *source* as well. Background seeds work the same way with the *sink* node.

It is very important that seeds are placed in the correct positions. If a foreground seed gets

Algorithm 5: Directed Graph Creation from Image

Input: image,width,height

Output: A weighted Graph

begin

```
    for i ⟵ 0 to height do
        for j ⟵ 0 to width do
4           graph ⟵ AddNode
    for y ⟵ to height do
        for x ⟵ to width do
7           gray1 ⟵ GetGrayValue(currentNode)
            for j ⟵ 0 to 8 do
9               nx = x + neighbourhood[j].x
10              ny = y + neighbourhood[j].y
                if nx >= 0 AND nx <= width AND ny >= 0 AND ny <= height then
                    gray2 ⟵ GetGrayValue(neighbourNode)
                    grayDiff = abs(gray1 − gray2)
                if j%2 = 0 then
15                  cost = exp(−1.0 ∗ ((grayDiff²))/TWO_SIGMA_SQ) ∗ 100)
                else
17                  cost = exp(−1.0 ∗ (((grayDiff²)/TWO_SIGMA_SQ) ∗ 1.2) ∗ 100)
                n ⟵ n + cost
19              graph ⟵ AddEdge(currentNode,neighbourNode,cost,cost)
            if currentNode is foreground then
21              graph ⟵ SetEdgeWeight(currentNode, n, 0)
            else if currentNode is background then
23              graph ⟵ SetEdgeWeight(currentNode, 0, n)
            else
25              graph ⟵ SetEdgeWeight(currentNode, 10, 10)
```

end

placed in a background region, that region will get segmented as foreground. If background seeds get placed on foreground objects, those objects get suppressed as background. Therefore, much care needs to be taken in placing seeds. The majority of systems that make use of Graph Cuts are interactive (Boykov and Jolly 2001; Rother et al. 2003; Talbot and Xu 2006). The user defines where the seeds are to be placed.

However, in many computer vision applications and in the Smart Home situation in particular, the system needs to be able to operate autonomously if it is to successfully and unobtrusively monitor behaviours of the occupants. There is a need to place the seeds automatically. Gabor filters and the Laws Texture Energy Measure used in the previous stages of this work produce a very strong edge map of the foreground. With this edge map, we can estimate the relative regions of foreground seeds as well as the background seeds.

To accomplish this, we first scan the edge map for foreground edges. The fundamental assumption here is that the edge maps would have been sufficiently cleaned in the previous stages so that it contains only the foreground edges. This image is scanned line by line for edge pairs. Algorithm 6 shows how this is done.

Since the edge map is a gray image, the pixels with some intensity value are considered to be part of an edge. The image is scanned from left to right. The objects are hypothesised to have at least two edges forming its boundaries. The first edge found of an object is termed the leading edge and the last edge is the trailing edge. This is depicted in Figure 6-3.

From Figure 6-3, it can be seen that there is some noise left in the edge map. As in line 20 of Algorithm 6 a seed is placed only when both leading and trailing edges are found. To avoid placing seeds between noise data, the edge pairs are ranked based on the intensity. Seeds are placed between the top ten pairs.

Algorithm 6: Foreground and Background Seed Placement

Input: edgeImage,width,height

Output: foregroundSeed,backgroundSeed

```
begin
    foregroundSeed ⟵ new foreground seed image
    backgroundSeed ⟵ new background seed image
    lead ⟵ false
    trail ⟵ false
    leadposition ⟵ 0
    for y ⟵ 0 to height do
        for x ⟵ 0 to width do
            if current pixel intensity is not zero then
                if not lead then
                    lead ⟵ true
                    while pixel at x not zero do
13                      leadposition ⟵ x
14                      x ⟵ x + 1
                else if not trail then
                    tail ⟵ true
17                  tailposition ⟵ x
                    while pixel at x not zero do
19                      x ⟵ x + 1
            if lead AND trail then
21              plantseed()
                lead ⟵ false
                trail ⟵ false
end
```

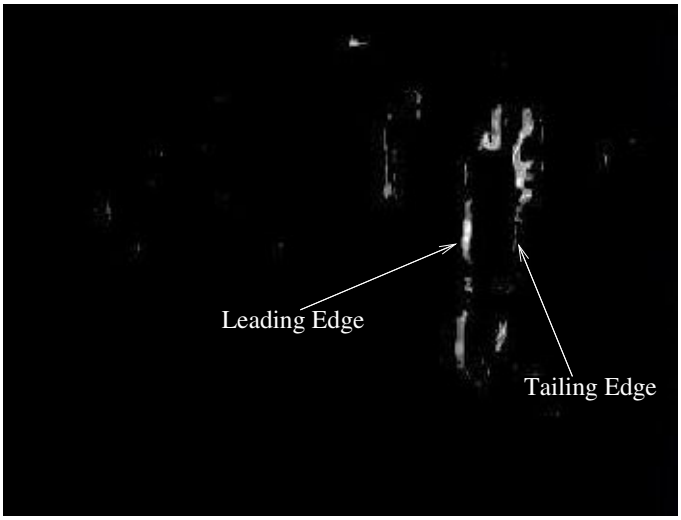


Figure 6-3: Leading and Tailing Edge Pairs

Once the edge pairs are found and ranked, Line 21 of Algorithm 6 places the foreground and background seeds. The pixels at the mid-point between leading and trailing edges are marked as foreground seeds. The pixels just before the leading edge and immediately after the trailing edge are marked as background pixels. After proper seed placement is done, the maxflow algorithm by Boykov et al. (2001) can be called to segment the image. Figure 6-4 (c) shows the final result of this process.

6.4 Summary

In this chapter a discussion into a method of obtaining a silhouette of an object is given. Graph Cuts are extensively used to achieve refined boundary segmentation. Graph cuts work well in segmenting the object as long as the seeds are in the correct positions. However if there are background regions in close proximity with objects that have similar colours, this algorithm takes those regions as foreground as well. This is the reason for the false alarms, the two lines near the object, seen in 6-4 (c).

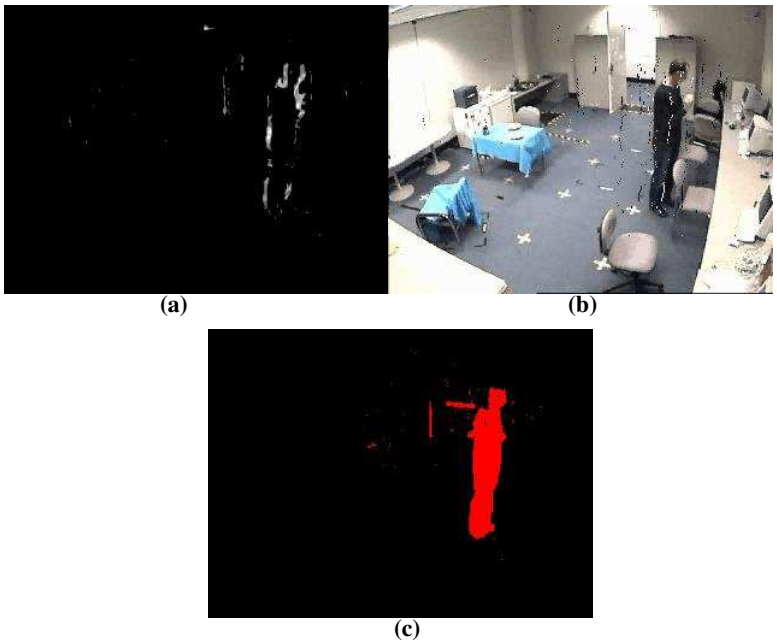


Figure 6-4: Graph Cut Results (a) An Edge Map (b) Seeds Placed on Image (c) Segmented Image

In the following chapter, the experimental results of this work is given. This system is tested on videos of a typical suburban home.

Chapter 7

Experimental Results

7.1 Introduction

The previous chapters discussed various techniques to detect foreground objects in videos and suppress false positives during the adaptation phase of the Mixture of Gaussians background modelling algorithms. In this chapter the experimental results of this work are presented. The first section presents the results of the edge based approach and the results of texture based approach are discussed in the following section. Finally the results of applying this approach in the smart home senario with virtual sensors are shown.

7.2 Results of the Edge Based Approach

Figure 7-1 shows the results of the edge based approach. This video was recorded in the mpeg4 format at a frame rate of 25 frames per second in the smart house lab. Column (a) shows the original image, (b) is the ground truth where the images are hand segmented to

establish a base line comparison, (c) and (d) represents the output of the Mixture of Gaussian approach and the results from applying the edge based false positive reduction technique on top of Mixture of Gaussians respectively. This figure is a sequence where there are spatially non-uniform light changes.

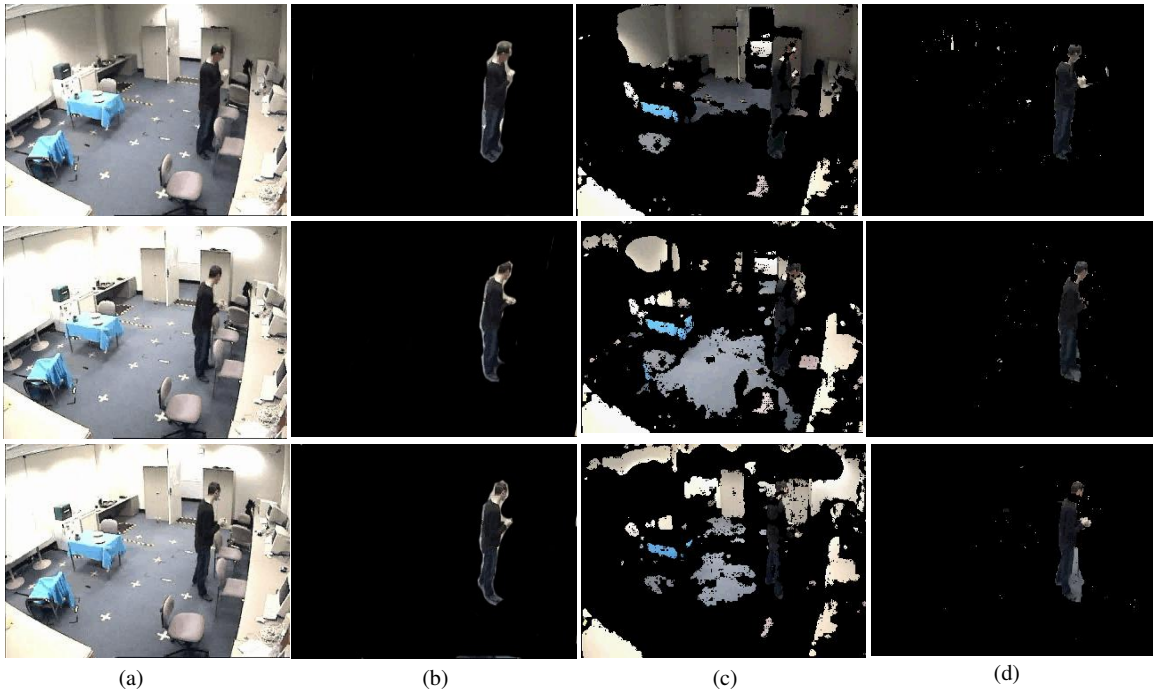


Figure 7-1: Edge based false positive reduction results of Video 1.

It can be seen from Figure 7-1 that while this approach still has some false positives it is significantly less than that of the Mixture of Gaussians alone. Moreover the resultant foreground image more closely matches the ground truth. Three Gaussian distributions are used for the Mixture of Gaussians and the learning rate set at 200 frames. Figure 7-2 is the result of using the edge based approach in the kitchen video. When the fridge is open it creates a sweeping light beam across the room.

Table 7.1 shows the results from different videos taken in the lab environment as well as a typical kitchen. Videos 1 and 2 are taken in the lab environment with sudden and spa-

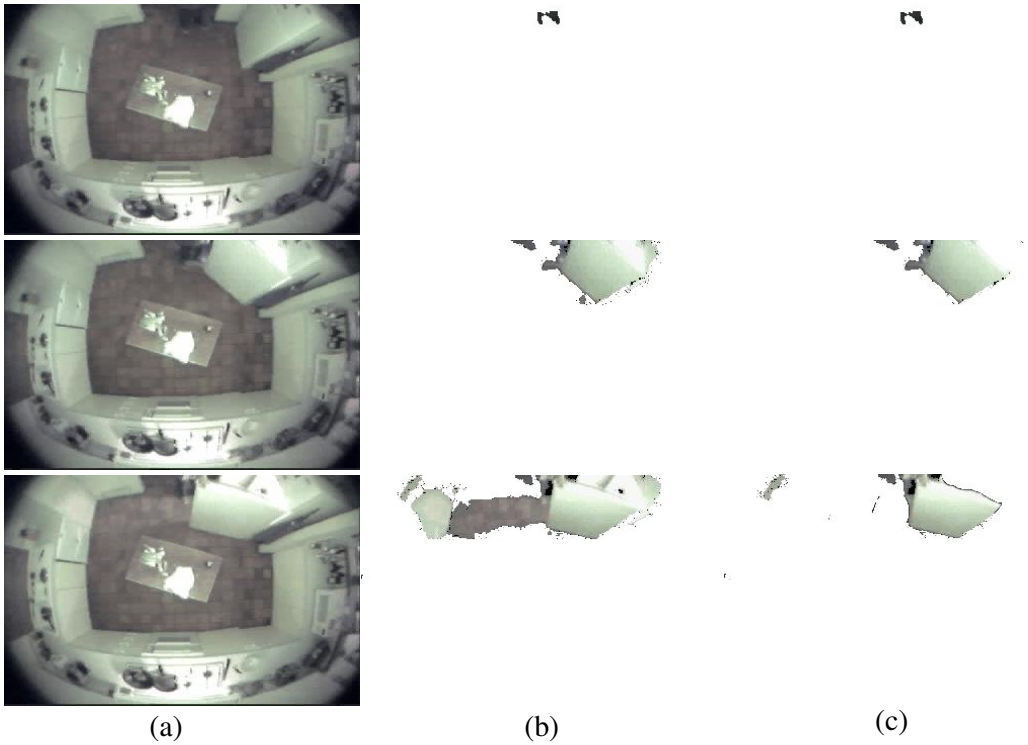


Figure 7-2: Edge based false positive reduction results of video 3. (a) original image (b) Mixture of Gaussian and (c) Mixture of Gaussian with FP reduction.

Table 7.1: False positive pixels detected using edge based approach

Videos	No of Frames	MoG (%)	MoG with edge base FP suppression (%)
1	3000	51.97	2.11
2	2700	39.7	3.5
3	3500	53.8	2.34
4	1500	60.32	3.31

tially non-uniform changes in light. The frames are first manually segmented to get the true foreground. This gives us the true positive or foreground pixels. The amount of pixels detected minus the true positive pixels are marked as false positives. These false positives are presented as a percentage of the total pixels in Table 7.1. Video 1 contains a single person moving around the room whereas video 2 contains multiple people. Videos 3 and 4 are from a ceiling mounted camera capturing the daily activities of a quintessential suburban home. Selected sequences from videos 1, 2 and 3 are shown in Figures 7-1, **??** and 7-2 respectively. Table 7.1 shows that there are significant reduction in false alarm pixels over the standard Mixture of Gaussians algorithm.

7.3 Results of Texture Based Approach

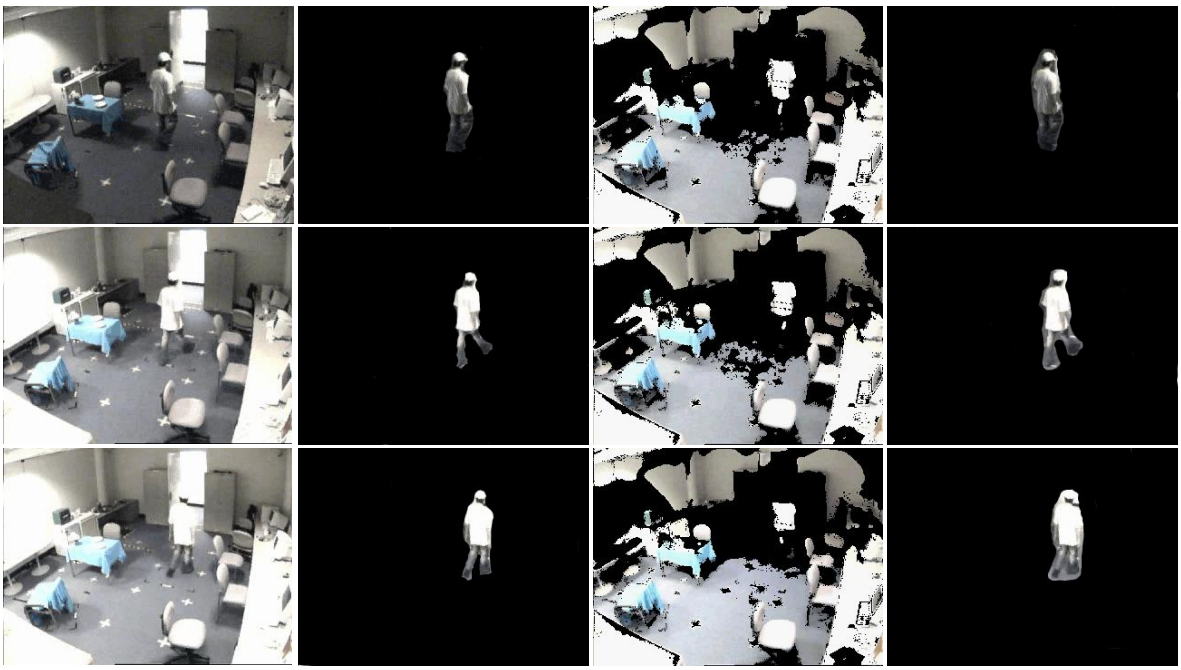
Figure 7-3 shows the results of the Laws texture approach. This approach is much more robust than using edges. The disadvantage of the edge based approach is that light fluctuations may cause misdetection of some edges making correct seed placement difficult. Laws approach on the otherhand, can robustly locate the position of the object. It does however, create a halo effect around the object as can be seen in Figure 7-3 (d).

The texture based method consistently performs better than the edge based approach as can be seen in Figures 7-3 and 7-4. However, when implementing the texture method, care must be taken to choose the appropriate analysis window size. If the window is too small, some texture characteristics can be missed. If the window is smaller than the foreground object, it can create holes in the foreground, if the foreground and background have similar textures.

Figure 7-5 shows the results of using texture in the kitchen videos. The same videos are processed with the texture approach and Table 7.2 shows the average percentage of pixels

(a) (b) (c) (d)

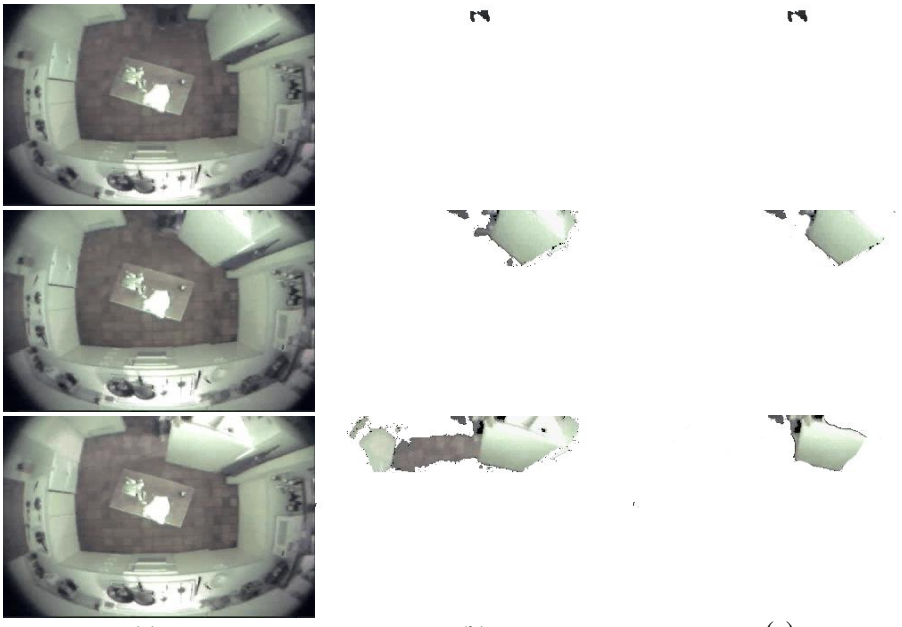
Figure 7-3: Texture based false positive reduction results of video 1



(a) (b) (c) (d)

Figure 7-4: Texture based false positive reduction results of video 2

detected as false positives in each video. Again there is a significant reduction in the number of false alarm pixels by using texture over the standard Mixture of Gaussians algorithm.



(a) (b) (c)

Figure 7-5: Texture based false positive reduction results of video 3. (a) Original Image (b) Mixture of Gaussians (c) Mixture of Gaussian with FP reduction.

Table 7.2: False positive pixels detected using texture based approach

Videos	No of Frames	MoG (%)	MoG with texture base FP suppression (%)
1	3000	51.97	1.5
2	2700	39.7	2.8
3	3500	53.8	2.34
4	1500	60.32	1.31

7.4 Results of Using Edge and Texture Approaches in Combination

The previous sections of this chapter showed the results of using Edge based approach and Texture based approach individially. This section presents the results of using these two approaches in different combinations.

Table 7.3: False positive pixels detected using edge and texture based approach

Videos	No of Frames	MoG (%)	MoG with combination FP suppression (%)
1	3000	51.97	0.5
2	2700	39.7	0.8
3	3500	53.8	1.0
4	1500	60.32	0.6

Table 7.4: False positive pixels detected using texture and edge based approach

Videos	No of Frames	MoG (%)	MoG with combination FP suppression (%)
1	3000	51.97	0.7
2	2700	39.7	1.0
3	3500	53.8	1.0
4	1500	60.32	0.8

As can be seen from Tables 7.3 and 7.4 applying the edge based approach first then the texture approach performs better than the other way around. It is observed that texture based approach is better at removing false alarms left by the edge approach.

7.5 Results of Using This System in Smart Homes for Activity Detection

West et al. (2005) used a virtual sensor system to detect interactions between humans and appliances around the house. Virtual sensors, which are polygonal regions, are placed in areas of interest such as the stove, fridge and so on in the video. The Mixture of Gaussian (Stauffer and Grimson 1999) is then applied within the virtual sensor regions to detect human movements. The videos processed are those from a ceiling mounted camera in a quintessential suburban kitchen. The vidoes are five hours long and a GUI-based method is developed to manually determine the ground truth efficiently. West et al. (2005) generated two types of ground truth:

- The *ANY* ground truth represent events where people are manually observed to be in the video and interaction with a device defined if they overlapped a Virtual Sensor region. This ground truth is generated from the knowledge of where the virtual sensors are and contains events such as people using, being near or passing a device. Human judgement is used to decide the activity. There should be a high correlation between this and the event log generated from the Virtual Sensors if the system is working correctly.
- The *USING* ground truth only contains events where a person is actually using an appliance. Hence there will be less USING events than ANY events as USING events are a subset of the ANY events.

The event logs from the Virtual Sensors are then compared with the ground truths. Figure 7-6 show the correlation of the state of each device for a one second time stamp. The correlation is between ground truth (lowercase device names) and Virtual Sensors (uppercase device names). Figure 7-6 (a) shows the comparison of ground truth and Mixture of Gaussians only

and (b) shows the ground truth and Mixture of Guassians wih FP suppression.

DISH	dish	FRIDGE	fridge	PANTRY	pantry	SINK	sink	STOVE	stove	
0.1	**0.6**	0.3	0.2	0.5	0.4	0.3	0.2	0.0	0.0	DISH
0.6	**0.1**	0.0	0.0	0.3	0.6	0.3	0.4	0.1	0.1	dish
0.3	0.0	**1.0**	**0.6**	0.3	0.0	0.2	0.0	0.0	0.0	FRIDGE
0.2	0.0	**0.6**	**1.0**	0.2	0.0	0.1	0.0	0.0	0.0	fridge
0.5	0.3	0.3	0.2	**1.0**	**0.4**	0.1	0.0	0.0	0.0	PANTRY
0.4	0.6	0.0	0.0	**0.4**	**0.1**	0.1	0.1	0.0	0.0	pantry
0.3	0.3	0.2	0.1	0.1	0.1	**1.0**	**0.7**	0.0	0.0	SINK
0.2	0.4	0.0	0.0	0.0	0.1	**0.7**	**1.0**	0.1	0.1	sink
0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	**1.0**	**0.7**	STOVE
0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	**0.7**	**1.0**	stove

(a)

DISH	dish	FRIDGE	fridge	PANTRY	pantry	SINK	sink	STOVE	stove	
0.1	**0.7**	0.3	0.2	0.5	0.4	0.3	0.2	0.0	0.0	DISH
0.7	**0.1**	0.0	0.0	0.3	0.6	0.3	0.4	0.1	0.1	dish
0.3	0.0	**1.0**	**0.6**	0.3	0.0	0.2	0.0	0.0	0.0	FRIDGE
0.2	0.0	**0.6**	**1.0**	0.2	0.0	0.1	0.0	0.0	0.0	fridge
0.5	0.3	0.3	0.2	**1.0**	**0.8**	0.1	0.0	0.0	0.0	PANTRY
0.4	0.6	0.0	0.0	**0.8**	**0.1**	0.1	0.1	0.0	0.0	pantry
0.3	0.3	0.2	0.1	0.1	0.1	**1.0**	**0.7**	0.0	0.0	SINK
0.2	0.4	0.0	0.0	0.0	0.1	**0.7**	**1.0**	0.1	0.1	sink
0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	**1.0**	**0.9**	STOVE
0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	**0.9**	**1.0**	stove

(b)

Figure 7-6: Correlations for the video compared with *ANY* ground truth (a) Mixture of Gassian only (b) Mixture of Gaussian with FP suppression

The values of the matrices range from 0.0 to 1.0 where 0.0 indicates no correlation and 1.0 indicate perfect correlation. The values in the off diagonal away from the leading diagonal are small indicating low correlation between unrelated events. The West et al. (2005) comparison matrix is somewhat optismistic as it considers the status of each device at each time step independently. For example, in 7-6 (a) if an event in the Virtual Sensor log is also contained in the ground truth regardless of the time it occurs, it is taken as correct.

This could mean that false positives in the Virtual Sensor log could also be taken as correct. In Figure 7-6 (b) false positive values are removed and the comparison made. It can be seen that the matrices indicate that applying the false positive suppression algorithms in this system does not negatively effect the system's ability to pick out relevant events. Figure 7-7 (a) shows a comparison of detected events of the ground truth and the virtual sensors using the Mixture of Gaussians only and Figure 7-7 (b) shows the comparison between ground truth and Mixture of Gaussian with FP suppression. As can be seen from the Figures, the accuracy of detected events is much improved by using False Positive suppression techniques as there are less false alarms.

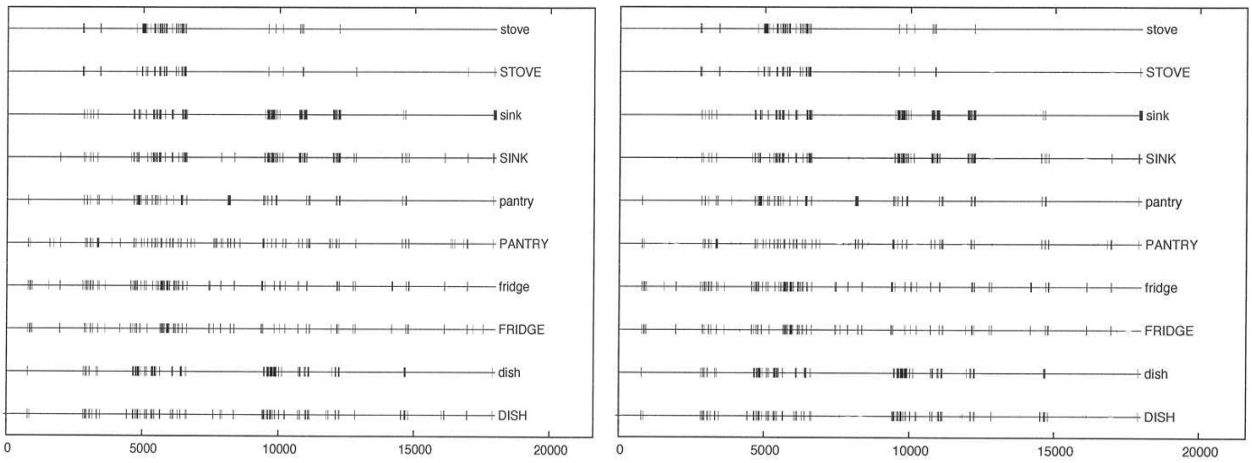


Figure 7-7: Events detected using the Virtual Sensor alongside ground truth for using, near and passing events (a) Ground truth vs Virtual Sensors (b) Ground truth vs Virtual Sensors with FP Suppression

It can be seen from Figure 7-7 (a) that in the West et al. (2005) model, detected events match well with those of the ground truth, however, there are also many false positives detected. Figure 7-7 (b) shows that modified model while leaving some false positives, is more accurate.

7.6 Summary

This chapter shows the results of this research. First the results of using the edge based approach to false positive suppression is given. Using this approach we are able to significantely reduce the amount of false positives in video streams. However, there are still a few false positives left and more over the object itself gets undersegmented in some cases. Texture is also used to eliminate false alarms. Texture appears to give more stable results than the edge based approach and there is little or no undersegmentation of the object. It does however, tend to create a halo around the object.

The results of using False Positive suppression in the Virtual Sensor system is also presented in this chapter. It can clearly be seen that using this method, a majority of the false alarms detected can be removed. The next chapter concludes this thesis and suggests future work to improve the performance.

Chapter 8

Conclusions and Future Directions

8.1 Summary

Current day background modelling and object detection algorithms work fairly well when the video scene that they model has stable environmental conditions. However, in reality the world around us is changing all the time and the background models must be able to adapt to these changes. As the technology advances, many algorithms that are capable to modelling slow changes in illumination and well as scenes with moving backgrounds have emerged (Stauffer and Grimson 1999; Kim et al. 2004; Tian et al. 2005). These systems are primarily aimed to be used in outdoor environments for tasks such as surveillance and traffic monitoring.

West et al. (2005) proposed the use of these technologies along with machine learning techniques to monitor the well being of elder and infirm people in their homes. This system is aimed at improving the quality of their lives and reduce their dependence on the care takers. Using currently available background modelling technologies in the indoor environment such as the smart home presents a new set of challenges particularly concerning the lighting in

the environment that is not stable or changes rapidly. For example, opening the fridge door creates a sweaping beam of light across the room.

In an attempt to reduce the false positives produced by these algorithms, we learned two things:

i Background modelling algorithms must consider integrating different image features such as edges, texture, etc. rather than building the model on just one feature.

ii Different spatial scales must be considered in modelling the background. Only then are we able to correctly model the changes that are spatially non-uniform. This idea was first introduced by Toyama et al. (1999) and is still valid to this day.

In this thesis, a framework based on sequential pattern recognition is presented as a solution to the problem of reducing false positives caused by changes environmental conditions in video processing. Complex image processing algorithms are sequentially applied to the incoming video stream to remove false positives.

An edge based approach to analyse images and removed false alarms is introduced. A Gabor filter with six angles and two frequencies is applied to the image to get an edge map of the current frame. The Mixture of Gaussian model also produces a updated background image. This image is processed with the Gabor filter and compared to the current frame. The two images are discretised and corresponding blocks are subtracted. The Gabor gives the same responses as the background image in false positive regions of the current frame. These blocks are then removed from the current frame. The Gabor responses for each image are combined for rotation invariance.

A texture based approach is also presented in this thesis. Laws texture energy maps are applied to both the current image and the background as before. Three by three Laws masks are used to process the image. Similar features are then combined to reduce the amount of data to be process and also to get rotation invariance. The current image and background images are compared as before and false positive blocks removed.

As a result of discretising the image, the remaining foregound objects retain the shape of the block instead of the silhouette of the object. To overcome this problem, Graph Cuts are used. Graph Cuts are an efficient image segmentation algorithm based on graph theory. Most algorithms that use graph cuts are interactive. The user labels some of the pixels in the image as object and background. Based on the manually placed seeds, the algorithm segments the image. In this research a method for automatic placement of seeds is presented.

8.2 Future Work

This thesis has presented two approaches to suppress false positives in video segmentation. A sequential framework is developed to allow for the integration of multiple algorithms. More complex and efficient algorithms can be added to this framework to improve the accuracy of suppressing false alarms.

When seed placement is done for graph cuts, sometimes false positive regions get marked as foreground. Even though the edges are ranked and the top ten edge pairs taken, false positive edges may be among the top ten. Objects move slowly over a number of frames where as the false positive edges tend to appear randomly in frames. We can take advantage of this temporal cohesion of objects so that the seeds are placed only in approximately the same regions over a number of frames. This will further reduce the number of false positives.

Texture can be extracted in many different ways. The Laws texture energy measures are used in this work. Other texture descriptors such as the Local Binary Patterns may offer better performance under conditions of changing illumination. These techniques could be integrated into this work.

There are many other image properties that can be investigated and added to this work such as colour information, co-variance metrices and so on. The virtual sensors used in this work can accurately determine when people are near appliances and interpret them as events. However, it cannot differentiate whether the person is actually using the device or just passing by. There is a need for the developmet of rules to determine the actual activity. This is an area for further investigation.

From the experiments we find that processing the videos take a fair amount of time. This is primarily due to the fact that the data rate does not reduce as much as anticipated before. One possible solution to this is to introduce a pre-processing step to detect scenes with light change and use false positive suppression on those frames only.

Bibliography

Alsaqre, F. E. and Y. Baozong (2003). Moving object segmentation for video surveillance and conferencing applications. In *International Conference on Communications Technology*, Volume 2, Beijing, China, pp. 1856–1859.

Boykov, Y. and G. Funka-Lea (2006). Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision 70*(2), pp 109–131.

Boykov, Y. and M.-P. Jolly (2001). Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *International Conference on Computer Vision (ICCV)*, Volume 1, pp. 105–112.

Boykov, Y. and V. Kolmogorov (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 26, 9, pp. 1124–1137.

Boykov, Y., O. Veksler, and R. Zabih (2001). Fast approximate energy minimization via graph cuts. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 23, pp. 1222–1239.

Canny, J. F. (1986). A computational approach to edge detection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 8, pp. 679–698.

Cavallaro, A. and T. Ebrahimi (2001). Video object extraction based on adaptive background and statistical change detection. In *SPIE Visial Communications and Image Processing*, pp. 465–475.

Cheng, L., S. Wang, D. Schuurmans, T. Caelli, and S. Vishwanathan (2006). An online discriminative approach to background subtraction. In *IEEE International Conference on Video and Signal Based Surveillance.*, pp. 2.

Cucchiara, R., C. Grana, M. Piccardi, and A. Prati (2003). Detecting moving objects, ghosts and shadows in video streams. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 25, pp. 1337–1342.

Davies, E. R. (2005). *Machine Vision: Theory Applications and Practicalities* (3rd ed.). Elsevier.

Elgammal, A., R. Duraiswami, D. Harwood, and L. S. Davis (2002). Background and foreground modelling using nonparametric kernel density estimation for visual surveillance. In *Proc. IEEE*, Volume 90, No 7, pp. 1151–1163.

Fan, S. L. X., Z. Man, and R. Samur (2004). Edged based region growing - a new image segmentation method. In *Proceedings of the 24th ACM SIGGRAPH International Conference on Virutal Reality and Its Applications in Industry*, pp. 302–305.

Gordon, G., T. Darrel, M. Harville, and J. Woodfill (1999). Background estimation and removal based on range and colour. In *Proceedings of the IEEE Comptuer Society Conference on Computer Vision and Pattern Recognition*, Volume 2, pp. 459–464.

Greenhill, S., S. Venkatesh, and G. West (2004). Adaptive model for foreground extraction in adverse lighting conditions. In *8th Pacific Rim International Conference on Artificial Intelligence (PRICAI)*, Volume 3157/2004, Auckland, pp. N2. SpringerLink.

Haritaoglu, I., D. Harwood, and L. S. Davis (2000). W4: Real-time surveillance of people and their activities. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 22, pp. 809–830.

Heikkila, M. and M. Pietikainen (2006). A texture-based method for modeling the background and detecting moving objects. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 28, pp. 657–662.

Heikkila, M., M. Pietikainen, and J. Heikkila (2004). A texture based method for detecting moving objects. In *British Machine Vision Conference*, Kingston University, London, pp. 187–196.

Huang, S., L. Fu, and P. Hsiao (2007). A region-level motion-based background modelling and subtraction using mrfs. In *IEEE Transactions on Image Processing*, Volume 16, issue 5, pp. 1446–1456.

Huwer, S. and H. Niemann (2000). Adaptive change detection for real time surveillance applications. In *Proceedings of Visual Surveillance*, Dublin, pp. 37–43.

Ivanov, Y., A. Bobick, and J. Liu (1998). Fast lighting independent background subtraction. In *IEEE workshop on visual surveillance*, Bombay, India, pp. 49–55.

Jabri, S., Z. Duric, H. Wechsler, and A. Rosenfeld (2000). Detection and location of people in video images using adaptive fusion of color and edge information. In *Proceedings of the International Conference on Pattern Recognition*, Volume 4, Barcelona, Spain, pp. 627–630.

Ji, Y., K. H. Chang, and C.-C. Hung (2004). Efficient edge detection and object segmentation using gabor filters. In *Proceedings of the 42nd Annual Southeast Regional Conference*, Huntsville, Alabama, USA, pp. 454–459.

Kamarainen, J. K., V. Kyrki, and H. Kalviainen (2006). Invariance properties of gabor filter-based features: overview and applications. In *IEEE Transactions on Image Processing*, Volume 15, pp. 1088–1099.

Kanade, T., R. Collins, A. Lipton, P. Burt, and L. Wixson (1998). Advances in cooperative multi-sensor video surveillance. In *Proceedings, DARPA Image Understanding workshop*, Volume 1, pp. 3–24.

Kim, K., T. H. Chalidabhongse, D. Harwood, and L. Davis (2004). Background modelling and subtraction by codebook construction. In *International Conference on Image Processing*, Volume 5, Genova, pp. 3061–3064.

Kim, K., T. H. Chalidabhongse, D. Harwood, and L. Davis (2005). Real-time foreground-background segmentation using codebook model. *Real time imaging 11, No 3*, 172–185.

Kolmogorov, V. and R. Zabih (2004). What energy features can be minimized by graph cuts. In *Transactions on Pattern Analysis and Machine Intelligence*, Volume 26, pp. 147–159.

Laws, K. (1980). *Textured Image Segmentation*. Ph. D. thesis, University of Southern Califonnia, Los Angeles.

Luo, X., S. M. Bhandarkar, W. Hua, and H. Gu (2006). Nonparametric background modelling using the condensation algorithm. In *Proceedings of the IEEE International Conference on Video and Signal Based Surveillance*, pp. 3–9.

Marr, D. and E. Hildreth (1980). Theory of edge detection. In *Proceedings of Royal Society London*, pp. 187–217.

Matsushita, Y., K. Nishino, and M. Sakauchi (2004). Illumination normalization with time-dependent intrinsic images for video surveillance. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 26, pp. 1336–1347.

Movellan, J. R. (2006). Tutorial on gabor filters. web page. mplab.ucsd.edu/wordpress/tutorials/gabor.pdf [last accessed 24/08/2007].

Piccardi, M. (2004). Background subtraction techniques: A review. In *IEEE International Conference on Systems, Man and Cybernetics*, pp. 3099–3104.

Pietikainen, M., A. Rosenfeld, and L. Davies (1983). Experiments with texture classifiction using averages of local pattern matches. In *IEEE Transctions on Systems, Man and Cybernetics*, Volume 13, No 3, pp. 421–426.

Radke, R. J., S. Andra, O. Al-Kofahi, and B. Roysam (2005). Image change detection algorithms: a systematic survey. In *IEEE Transactions on Image Processing*, Volume 14, pp. 294–307.

Rother, C., V. Kolmogorov, and A. Blake (2003). "grab cut" - interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics*, Volume 23, pp. 309–314.

Smith, S. E. and S. S. Yau (1972). Linear sequential pattern classification. In *IEEE Transactions on Information Theory*, Volume 8, pp. 673–678.

Sonka, M., V. Hlavac, and R. Boyle (1999). *Image Processing, Analysis and Machine Vision*. ITP.

Stauffer, C. and W. E. L. Grimson (1999). Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Volume 2, pp. 246–252.

Sun (1999). Java media framework. [Online] http://java.sun.com/products/java-media/jmf/ [last accessed 13/05/2007].

Sun (2000). Java advanced imaging. [Online] http://java.sun.com/ [last accessed 13/05/2007].

Talbot, J. F. and X. Xu (2006). Implementing grabcut. online [http://students.cs.byu.edu/ jtalbot/research/Grabcut.pdf] Date Accessed [13/06/2007].

Tan, T. (1993). *Colour Texture Analysis in Machine Vision*. Ph. D. thesis, University of Surrey, Guildford Surrey, UK.

Tian, Y., M. Lu, and A. Hampapur (2005). Robust and efficient foreground analysis for real-time video surveillance. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Volume 1, San Diego, pp. 1182–1187.

Toth, D., T. Aach, and V. Metzler (2000). Illumination-invariant change detection. In *Proceedings of the Fourth IEEE Southwest Symposium Image Analysis and Interpretation*, pp. 3–7.

Toyama, K., J. Krumm, B. Brumitt, and B. Meyers (1999). Wallflower: Principals and practice of background maintenance. In *Computer Vision, The Proceedings of the Seventh IEEE International Conference on*, Volume 1, Kerkyra, Greece, pp. 225–261.

Turner, M. R. (1986). Texture discrimination by gabor functions. *Biological Cybernetics 55*(2-3), pp 71–82.

West, G., C. Newman, and S. Greenhill (2005). Using a camera to implement virtual sensors in a smart home. In *3rd International Conference on Smart Homes and Health Telematics*, Magog, Quebec, Canada, pp. 83–90.

Wikipedia (2007). Computer vision. web. http://en.wikipedia.org/wiki/Computervision.

Wren, C., A. Azarbayejani, T. Darrell, and A. Pentland (1997). Pfinder: Real-time tracking of the human body. In *IEEE Trans. Pattern Analysis and Machine Intgelligence*, Volume 19, Issue 7, pp. 780–785.

Xie, B., V. Ramesh, and T. Boult (2004). Sudden illumination change detection using order consistency. *Image and Vision Computing 22*, 117–125.

Zeng, H.-C. and S.-H. Lai (2007, 15-20 April). Adaptive foreground object extraction for real-time video surveillance with lighting variations. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, Volume 1, pp. I–1201–I–1204.