Object Enumeration From Video Sequences

BY

PATEL CHIRAG I.

07MCE014

NIRMA
UNIVERSITY
INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

May 2009

Object Enumeration From Video Sequences

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

BY:

PATEL CHIRAG I.

07MCE014

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

May 2009

Certificate

This is to certify that the Major Project entitled "OBJECT ENUMERATION FROM VIDEO SEQUENCES" submitted by Mr. Patel Chirag I. (07MCE014), towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University of Science and Technology, Ahmadabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. S.N. Pradhan
Guide and Professor,
Department of Computer Engineering,
Institute of Technology,
Nirma University, Ahmedabad

Prof. D. J. Patel
Professor and Head,
Department of Computer Engineering,
Institute of Technology,
Nirma University, Ahmedabad

Dr K Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad

Abstract

The computer vision community has expended a great amount of effort in recent years towards the goal of counting people in videos. Much more recently, algorithms have been developed to identify objects in videos robustly. The goal of this project is to implement a system based on one of those algorithms, in order to count the objects in an offline video.

The system addresses the problem of counting the number of objects in an image frame. This system presents a human detection model, that is designed to work with people . The system proposed does learning through templates. The model makes use of Haar based features to form templates performs matching of Haar-transformed images. Objects can be detected irrespective of the texture and color of there clothing as well as orientation.

This system attempts to provide a Wavelet based human detection system. Human beings are non rigid objects and as such deteting them is a hard problem, due to the various possible combinations that arise out of clothes being worn, there texture, the orientation of the individual. To overcome this, we need a systems that is invariant to the colour differences, this is made possible by using Haar transforms. These have the property that they extract information from a given image, which is invariant to the absolute colour, and makes use of only color changes. The problem of handling multiple orientatios can be tackled by having a sufficiently large database of people in different orientations. Having a learning system simplifies the task of adding more templates as and when needed to handle new cases that may arise. Multi resolution Haar transform were found for human templates and Pyramidal search was caried out to match human beings. Human detection and counting has numerous advantages in real life problems.

Acknowledgements

It gives me immense pleasure in expressing my gratitude towards Dr S. N Pradhan, Professor, Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance, support and motivation throughout the Major project-2. I am thankful to him for the valuable time he spent with me for my thesis, for suggestion that shaped the project.

I like to give my special thanks to Mr. Tejas Gandhi, student IIT Kanpur, for his suggestions to improve quality of work and providing constant motivation and support. I am also thankful to Dr. K Kotecha, Director, Institute of Technology for his kind support in all respect during my study.

I am thankful to all faculty members of Department of Computer Science and Engineering, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- PATEL CHIRAG I
07MCE014

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The aim of this work is to automatically count the number of Object in an offline video in constrained enviroment. To do so, it will be necessary, to first detect the foreground objects in the video, and then finally classify the objects and count the number of Objects. There should be an automation system whose input is essentially the name of the video, and an ouput of the count of the number of objects in the video [1].

Video Processing is one of the most challenging areas in Computer Vision. It deals with identifying an object of interest. It has wide application in the fields like Traffic surveillance, Security, Criminology etc.

Automatic detecting and tracking vehicles in video data is a very challenging problem in computer vision with important practical applications, such as traffic analysis and security. Video cameras are a relatively inexpensive surveillance tool. Manually reviewing the large amount of data they generate is often impractical. Thus, algorithms for analysing video which require little or no human input is a good solution. Video processing systems are focussed on background modelling, moving object classification. The increasing availability of video sensors and high performance video processing hardware opens up exciting possibilities for tackling many video understanding problems, among which vehicle tracking and target classification are very important. Most occurrences of moving objects in our data are pedestrians, vehicles and some complex objects. In addition to vehicle counts, a much larger set of traffic parameters like vehicle classifications, lane changes, etc.,

This system attempts to provide a Wavelet based object/human detection system. object/human beings are non rigid objects and as such detecting them is a hard problem, due to the various possible combinations that arise out of clothes being worn, there texture, the orientation of the individual. To overcome this, we need a systems that is invariant to the colour differences, this is made possible by using Haar transforms. These have the property that they extract information from a given image, which is invariant to the absolute colour, and makes use of only color changes.

Previous work in this area has been based on the detection of motion or texture of the ojects.Probabilistic models like Gaussian mixture models can be used to detect motion and segment foreground and background objects An- other approach used is based on hand-crafted models.These methods are based on direct use of low-level process of interpreting image intensities and are not robust as they are sensitive to spurious details.Moreover they give ambiguous results.

Approach To capture high level knowledge about a object class such as shape of the person in the image scene we would like to make use of wavelet templates.The intensities of the interior features of a persons body significantly differs from that of that of the background.A wavelet basis function like Haar wavelet captures the relationship between neighbouring regions of the image.Wavelet function can compactly represent the structural commonalities amongst different classes.Also , wavelet coefficients can be computed efficiently.This approach does not rely on charecteristics such as motion , color , texture. We would like to combine both approaches . A motion-color approach (like GMM) to achieve background subtraction and segmentation would be used to identify foreground objects. These will be used to train the generation of wavelet templates.Thus the system would be self-learning i.e without the need of explicitly selecting objects (pedestrians) for training purposes.

1.1 SCOPE OF WORK

Object Enumeration System can be applied at retailer to plan staffing within each day or high seasons, for e.g. Diwali or sale. The Object Enumeration is also a measure of the effect of advertisement.

Railway stations, Shopping mall etc. can use this system to find out optimal time for cleaning and maintenance by analyzing the flow of Objects/People.

There are various reasons for counting people. In retail stores, counting is done as a form of intelligence-gathering. The use of people counting systems in the retail environment is necessary to calculate the Conversion Rate, i.e. the percentage of a store's visitors that makes purchases. This is the key performance indicator of a store's performance and is far superior to traditional methods, which only take into account sales data. Traffic counts and conversion rates together tell you how you got to your sales. i.e. if year-over-year sales are down: did fewer people visit my store, or did fewer people buy? Although traffic counting is widely accepted as essential for retailers, it is estimated that less than 25

Since staff requirements are often directly related to density of visitor traffic, accurate visitor counting is essential in the process of optimizing staff shifts. services such as cleaning and maintenance typically must be done when traffic is at its lowest or, occasionally, at a certain level. Planning of these activities necessitates accurate people counting.

For many locations such as bars or factories, it is essential to know how many people are inside the building at any given time, so that in the event of an evacuation due to fire they can all be accounted for. This can only be automated with the use of extremely accurate people counting systems. Although, no people counting system is 100

Many public organizations use visitor counts as evidence when making applications for finance. In cases where tickets are not sold, such as in museums and libraries, counting is either automated, or staff keep a log of how many clients use different services.

Many shopping center marketing professionals rely on visitor statistics to measure their marketing. Quite often shopping center owners measure marketing effectiveness with sales. A trend has been to include visitor statistics to scientifically measure marketing effectiveness. Marketing metrics such as CPM (Cost Per Thousand) and SSF (Shoppers per Square Foot) are becoming very useful key performance indicators.

Many real life scenarios require counting number of people/objects in an area. eg : number of people entering or present in a room or stadium . This may be useful in wide range of fields such as banking , to track the number of customers ; monitoring and tracking the people in highly secured zones . Counting the number of people can be achieved if we can identify people (or objects in general) in a sequence of images and obtain the count of these foreground objects.

Some of the major applications are Human Intrusion detection, Tracking usage of Resource / Preferences of people, (Unobtrusive monitoring) Optimizing working of road crossing signals, Getting a rough count of the number of people in an enclosed area (malls and bank).

1.2 ORGANIZATION OF WORK

The project work has been implemented into two major phases. The initial phases consists of Preprocesing of the input video typically we convert the RGB frame to intensity images, we can also blur image if required to remove some effect of non-static background .on this image we separate background and foreground by using background model for background subtraction.

The Second Phase, We initially process the result to remove noise from it. Then is segmentation module object cropped from the image. Next module performs labeling of objects which is used for training /creating database which would be used by matching.

The matching module will match the actual image with the image or its feature in database and based on its result will classify the foreground object as Objects and maintain the count.

The two most significant modules here are Background subtraction and Matching . The two modules are more demanding than others in terms of efforts.

Limitation

- People/Objects of small sizes can not be detected reliably If threshold background subtraction is lowered to form smaller templates , smaller images are detected.However, this results in generation of large number of false detections .
- Detection fails if a template of that shape is not present in the database As the detection is done by template matching , the detection will fail if appropriate template is unavailable in the database .

- Occlusion If a person/object gets occluded by a shape the detection fails as it

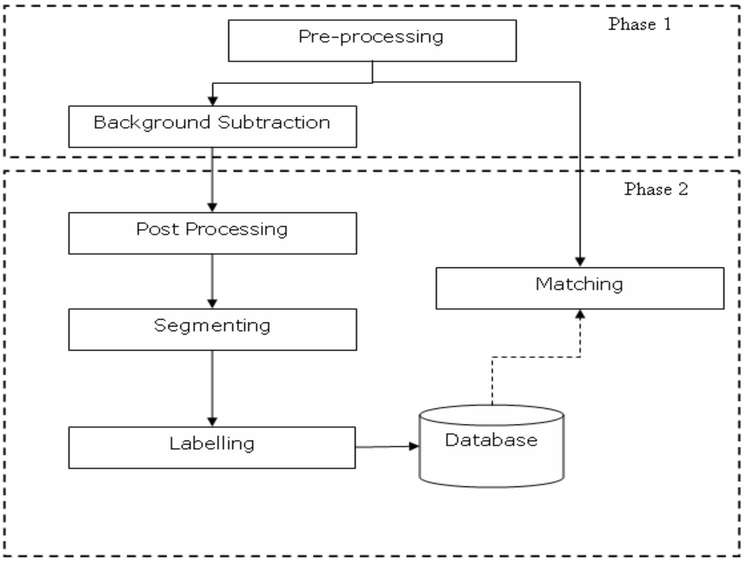

Figure 1.1: Organization of Work

1.3 OutLine Of Thesis Report

The rest of the thesis report is organized as follows.

Chapter 2, *Review of Literature*, This describes the problem definition and existing methodologies. This chapter explains various techniques used for background subtraction and matching. It also gives an insight details that which techniques is useful and why.

Chapter 3, *Background Subtraction using Gaussian Mixture Model*, This chapter describe about the gaussian mixture model and its implementation. Here it shows it details and result.

Chapter 4, *Haar Transform*, This chapter describe the template generation technique using haar transform.

Chapter 5, *Matching Using Normalized Cross Correlation*, This chapter describes template matching technique like normalized cross correlation and It shows the computational aspect of the matching technique. Improvements on the matching technique.

Chapter 6, *System Implementation*, This chapter describe system implementation and its details.

Chapter 7, *Test and Results Analisys*, This chapter describes results and its analisys.

Chapter 8, *Conclusion And Future Work*, This chapter describe what the outcome of the system and in this system what can happen in the future.

Chapter 2

Review of Literature

2.1 Object Detection

The problem of object detection from images or video has seen a high degree of interest over the years.the fundamental problem is how to characterize an object class.In contrast to the case of pattern classification, where we need to decide between a relatively small number of classes, the detection problem requires us to differentiate between the object class description for object detection must have large discriminative power to handle the cluttered scenes it will be presented with. furthermore , in modeling complicated classes of objects (e.g faces,pedestrians,Complex object) the intra-class variability itself is significant and difficult to model. Since it is not known how many instances of the class are presented in the scene, if any, the detection problem cannot easily be solved using methods such as maximum-a-posteriori probability or maximum likelihood models.Consequently, the classification of each pattern in the image must be done independently; this makes the decision problem susceptible to missed instances of the class and false positives [1].

There has been a body of work on people detection ([2],[3]and [4]); these approaches are heavily based on motion and hand crafted models.

One of the successful systems in the area of trainable object detection in cluttered scences is the face detection system of Sung and Poggio.They model face and non-face patterns in a high dimensional space and derive a statistical model for the class of frontal human faces. Similar face detection systems have been developed by others(Vailant,Rowley,Moghaddam and Osuna).

Frontal human faces, despite their variability, share very similar patterns(shape and the spatial layout of facial features) and their color space is very constrained. This is not the case with pedestrians in figure 2.1 shows several typical images of people inour databse. These images illustrate the difficulties of pedestrian detection; there is significant variability in the patterns and colors within the boundaries of the body. the detection problem is also complicated by absence of constraints on the image background.Given these problems, direct analysis of pixel charcteristics(e.g intensity,color and texture) is not adequate.

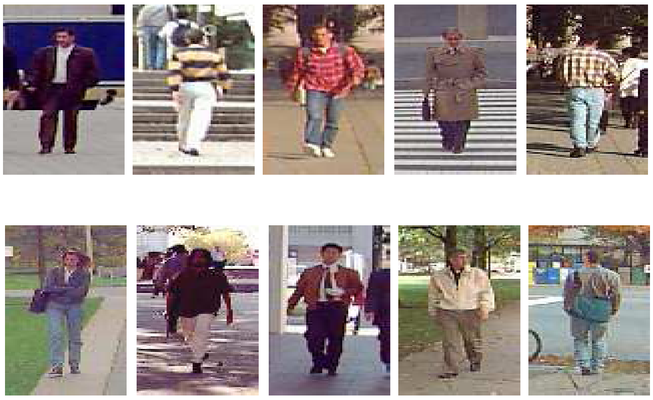

Figure 2.1: Examples of images of people in the database

2.2 Ratio Template

A ratio template encodes the ordinal structure of the brightness distribution on a face. It consists of a set of inequality relationships between the average intensities of a few different face-regions. This design was motivated by the observation that while the absolute intensity values os different regions change dramatically under varying illumination conditions, their mutual ordinal relationships(binarized ratios) remain largely unaffected. thus , for instance, the forehead is typically brighter than the eye-socket regions for all nut the most contrived lighting setups. A small set of such relationships, collectively called a ratio template,provides a powerful constraint for face detection.The emphasis on the use of qualitative relationships also renders the ratio template construct perceptually plausible(the human visual system is poor at judging absolute brightnesses comparisons).These include a formalization of the template structure in terms of simple primitives, a rigorous learning scheme capable of working with real images, and also the question of appicability to other, possibly more complex, object class such as pedestrians,vehicles.

2.3 Overview Wavelet Template

Now present an extension of the ratio template,known as the "wavelet template", and address some of these issues in the context of pedestrain detection[1]. The wavelet template consists of a set of regular regions of different scales that correspond to the support of a subset of significant wavelet functions. the relationships between different regions are expressed as constraints on the values of the wavelet coefficents. the wavelet template can compactly express the structural commonality of a class of objects and is computationally efficent.It is learnable from a set of examples and provides an effective tool for the challenging problem of detecting pedestrians in cluttered scences. It believe that the learnable wavelet template represents a frame work that is extensible to the detection of complex object classes other than pedestrians.

2.3.1 The Haar Dictionary

As motivated by the work on the template ratio,there is a need for an image representation which captures the relationship between average intensities of neighboring regions. This suggests the use of a family of basis functions, such as the HAAR wavelets, which encode such relationships along different orientations. The haar wavelet representation has also been used for image database retrieval, where the largest wavelet coefficients were used as a measure of similarity between two images.In other work , the wavelet representation is used to capture the structural similarities between various instance of the class in figure 2.2 , we depict the 3 types of 2 - dimensional Haar wavelets. These types include basis functions which capture change in intensity along the horizontal direction. the vertical direction and the diagonals(or corners). Since the wavelets that the standard transform generates have irregular support, we use the non-standard 2- dimensional DWT where, at a given scale, the transform is applied to each dimension sequentially before proceeding to the next scale. The results are Haar wavelets with square support of all scales.

The standard Haar basis is not dense enough for our application. For the 1-dimensional transform,the distance between two neighboring wavelets at n is 2^n. For better spatial resolution, we need a set of redundant basis functions , or an overcomplete dictionary, where the distance between the wavelets at scale is $1/4^{2n}$. We call this a quadruple density dictionary. As one can easily observe, the straightforward approach of shifting the signal and recomputing the DWT will not generate the desired dense sampling. However, one can observe that in the standard wavelet transform,

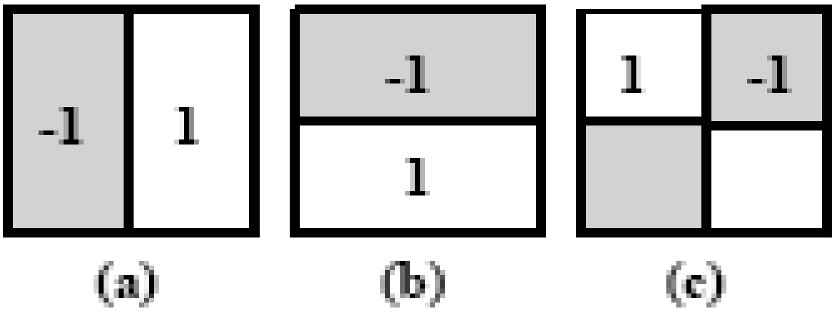

Figure 2.2: Haar-basis

after the scaling and wavelet coefficients are convolved with the corresponding filters there is a stepof downsampling . If we do not downsample the wavelet coefficents we generate wavelets with double density,where wavelets of level n are centered every $1/2^{2n}$. To generate the quadruple density dictionary,we compute the scaling coefficients with double density by not downsampling them. The next step is to calculate double density wavelet coefficent on the two sets of scaling coefficents-even and -odd separatly. By interleaving the results of the two transforms we get quadruple density wavelet coefficents. For the next scale we keep only the even scaling coefficents of the previous level and repeat the quadruple transfrom on this set only; the odd scaling coefficents are dropped off. Since only the even coefficent are carried along at all the scales, we avoid an"explosion" in tthe number of coefficient,yet provide a dense and uniform sampling of the wavelet coefficents at all the scles. As with the regular DWT, the time complexity is O(n) in the number of pixels n. The extension for the 2-dimensional transform is straightforward.

2.4 Wavelet Template

The ratio template defines a set of constraints on the appearance of an object by defining a set of regions and a set of relationships on their average intensities [5]. The relationships can require, for example, that the ratio of intensities between two specific regionsfalls within a certain range. We address the issues of learing these relationships, using the template for detection, and its efficent computation by establishing the ratio template in the natural framework of Haar wavelets. Each wavelet coefficentdescribes the relationship between the average intensities of two neighboring regions. If we compute the transform on the image intensities, the Haar coefficents specify the intensity differencs between the regions; computing the transform on the log of the ration of the intensities. Furthermore, the wavelet template can describe regions with different shapes by using combinations of neighboring wavelets with overlapping support and wavelets of different scales. The wavelet template is also computationally efficent since we compute the transform once for the whole image and look at different sets of coefficents for different spatial locations.

2.5 Learning the complex or pedestrian template

As shown in template database, it is easy to observe that there are no consistent patterns in the color and texture of pedestrians or their backgrounds in arbitary cluttered scenes in unconstrained enviroments. This lack of clearly discernible interior features is circumvented by relying in (1) differences in the in tensity between pedestrian bodies and their backgrounds and (2)consistencies within regions inside the body boundaries. We interpret the wavelet coefficents as either indicating an almost uniform area, i.e "no-change", if their absolute value is relatively small, or as indicating "strong change" if their absolute value is relatively large. The wavelet template we seek to identify will consist soley of wavelet coefficents(either vertical,horizontal or corner) whose types("change","nochange") are both clearly identified and consistent along the ensemble of pedestrian images; these comprise the important coefficents [1].

The basic analysis to identify the template consists of two steps first, we normalize the wavelet coefficents relative to the rest of the coefficents in the patterns;second, we analyze the averages of the normalized coefficents along the ensemble. We have collected a set of 564 color images of object for use in the template learning. All the images are scaled and clipped to the dimensions 128*64 such that the object are centered and approximately the same size In analysis, we restrict ourselves to the wavelets at scales of 32*32 pixels and 16*16 . For each color channel of every image, we compute the quadruple dense Haar transform and take the coefficent value to be the largest absolute value among the three channels. the normalization step computes the average of each coefficent's class(vertical,horizontal,corner*16,32) over all the object patterns and divides every coefficent by its corresponding class averages. We calculate the averages separately for each since the power distribution between the different classes may vary..

To begin specifying the template, we calculate the average of each normalized coefficent over the set of objects. A base set of 597 images of natural scenes that do not contain any object were gathered to compare with object patterns and are processed as above. Table 1(a)and 1(b) show the average coefficent values for the set vertical Haar coefficent of scale 32*32 for both the non-object and object classes. Table 1(a) & Table 2(b) shows that the process of averaging the coefficents within the pattern and then in the ensemble does not create spurious patterns; the average values of these non-object coefficents are near 1 since these are random images that do not share any common pattern. The object avergaes,on the other hand , show a clear pattern,with strong response in the coefficents corresponding to the sides of the body and weak response in the coefficents along the center of the body.

1.18	1.14	1.16	1.09	1.11
1.13	1.06	1.11	1.06	1.07
1.07	1.01	1.05	1.03	1.05
1.07	0.97	1.00	1.00	1.05
1.06	0.99	0.98	0.98	1.04
1.03	0.98	0.95	0.94	1.01
0.98	0.97	0.96	0.91	0.98
0.98	0.96	0.98	0.94	0.99
1.01	0.94	0.98	0.96	1.01
1.01	0.95	0.95	0.96	1.00
0.99	0.95	0.92	0.93	0.98
1.00	0.94	0.91	0.92	0.96
1.00	0.92	0,93	0.92	0.96

Table I: Coefficent

0.62	0.74	0.60	0.75	0.66
0.76	0.92	0.54	0.88	0.81
1.07	1.11	0.52	1.04	1.15
1.38	1.17	0.48	1.08	1.47
1.65	1.27	0.48	1.15	1.71
1.62	1.24	0.48	1.11	1.63
1.44	1.27	0.46	1.20	1.44
1.27	1.38	0.46	1.20	1.44
1.18	1.51	0.46	1.48	1.18
1.09	1.54	0.45	1.52	1.08
0.94	1.38	0.42	1.39	0.93
0.74	1.08	0.36	1.11	0.72
0.52	0.74	0.29	0.77	0.50

Table II: Coefficent

The template derived from learing uses a set of 29 coefficents that are consistent along the ensemble either as indicators of ”change” or ”no-change”, There are 6 vertical and 1 horizontal coefficents at the scale of 32 *32 and 14 vertical and 8 horizontal at the scale of 16*16. These coefficents serve as the feature vector for the ensuring classification problem.

2.6 The Detection system

Once we have identified the important basis functions, are identified they could be use various classification techniques to learn the relationships between the wavelet coefficents that define the object class. the architecture of the detection system;the classifier that could be used for(the support vector machine,ANN etc.)and traning process.

2.6.1 System Architecture

The system detects people in arbitary positions in the image and in different scales. To accomplish this task, the system is trained to detect a complex object centered in a fixed size pixel window . Once the traning stage is completed, the system is able to detect complex object at arbitary positions by shifthing according of there window size, thereby scanning all possible locations in the image . This is combined with iteratively resizing the image to achive multi-scale detection.

2.6.2 System Training

To train the system we use a database of complex object images of objects from outdoor and indoor scenes. The initial non-object in the training database are patterns from natural scenes not containing object. The combined set of positive and

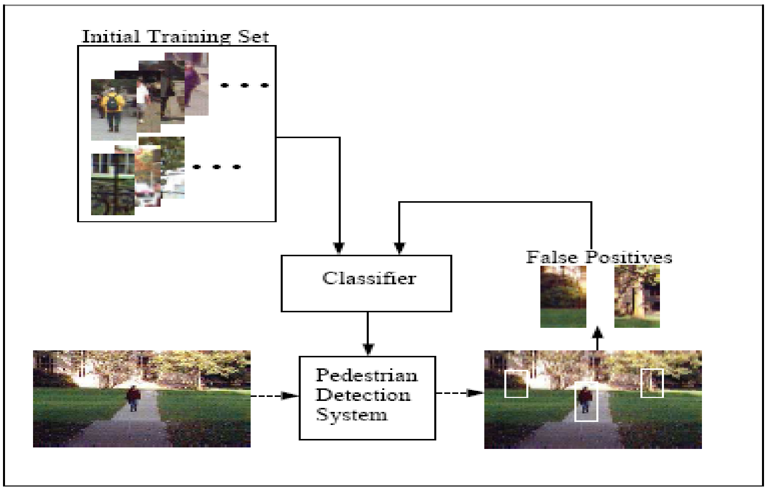

Figure 2.3: Incremental bootstrapping to improve the system performance

negative examples form the initial training database for the classifier. A key key issue with the training of detection systems is that , while the examples of the target class, in this case pedestrian are well defined, there are no typical examples of non-pedestrians.After the initial traning, we run the system over arbitrary images that do not contain any objects. Any detections are clearly identified as false positives and are added to the database of negative examples and the classifier is then retrained with this larger set of data. These iterations of the bootstrapping procedure allows the classifier to construct an incremental refinement of the non-object class until satisfactory performance is achieved. This bootstrapping technique is illustrated in figure 2.3.

2.6.3 Classification schemes

In this section we described the identification of the significant coefficents that characterize the complex-object class. These coefficents are used as the feature vector for various classification methods.

Basic template matching

The simplest classification scheme is to use a basic template matching measure [1],[6]. As the normalized template coefficents are divided into two categories; coefficients above 1 in the table (indicating strong change) and below 1(weak change). For every novel window, the wavelet coefficents are compared to the object template. The matching value is the ratio of the coefficents in agreement. A similar approach was used in for face detection with good results.

Support vector machines

Instead of the simple template matching paradigm we can use a more sophisticated classifier which will learn the relationship between the coefficients from given sets of positive and negative examples. The classifier can learn more refined relationships than the simple template matching scheme and therefore can provide more accurate detection [1],[7].
The classification technique we use is the support vector machine developed by Vapnik. This developed technique has several features that make it particularly attractive. Traditional training techniques for classifier, such as multilayer perceptrons, use empirical risk minimization and only guarantee minimum error over the traning set. In contrast , the SVM machinery uses structural risk minimization which minimizes a bound on the generalization error and therefore should perform better on moved data. Another interesting aspect of the SVM is that its decision surface depends only on the inner product os the feature vestors. This leads to an important extension since we can replace the Euclidean inner product by any symmertic positive definite

kernel K(x,y). This use of a kernel is equivalent to mapping the feature vectors to a high dimensional space, thereby significantly increasing the discriminative power of the classifier.

2.7 Various Techniques for Background Subtraction

In this section we describe the various technique to find out the foreground and background object from the images [8],[9].

2.7.1 General

Background subtraction is a widely used method for identifying moving objects in a video stream. It is the first significant step in many computer vision applications, including video surveillance, motion capture etc. The performance of these applications is dependent on the background subtraction algorithm being robust to illumination changes, small movements of background elements (e.g. swaying trees), the addition or removal of items in the background (e.g. parked car), and shadows cast by moving objects. Computational efficiency is also of high priority.

The most common paradigm for performing background subtraction is to build an explicit model of the background. Moving objects are then detected by taking the difference between the current frame and this background model. Typically, a binary segmentation mask is then constructed by classifying any pixel as being from a moving object when the absolute difference is above a threshold. Background subtraction algorithms differ in how they define and update the background model.

Despite the success enjoyed by background subtraction algorithms, it is becoming clear that post-processing is required in order to improve their performance. This post-processing can range from shadow detection algorithms operating at the pixel level to connected component labeling which identifies object-level elements. The results of post-processing can be used to directly improve the quality of the segmentation mask and fed back into the background subtraction algorithm in order to facilitate more intelligent updating of the background model.

Background subtraction, although being simply defined as a difference between the background image without objects of interest and an observed image, has many difficult issues to overcome, making it a problem that has inspired a wealth of research. For instance, the type of situation for which it is needed exposes many problems and a background subtraction algorithm that works well in one scenario may not necessarily work as well in another.

As this is an important phase of the project and has a significant impact on the outcome of the project, this report primarily focuses on background subtraction. This report looks at background subtraction with respect to videos, comparing an obtained background image to video frames, as opposed to background subtraction for still images

Although there are many algorithms for background subtraction, they all follow a general pattern of processing as shown in Figure .

Firstly, video frames captured from a camera are input to the background subtractor. Preprocessing stages are used for filtration and to change the raw input video to a process able format. Background modeling then uses the observed video frame to calculate and update the background model that is representative of the scene without any objects of interest. Foreground detection is where the pixels that show a significant difference to those in the background model are flagged as foreground. Data validation is used to examine the found objects of interest and to eliminate any false matches. A foreground mask can then be output in which pixels are assigned as foreground or background.

The foreground detection stage can be described as a binary classification problem whereby each pixel in an image is classified as foreground or background. Formally for every pixel p in image I, each pixel is either 0 (background) or 1(foreground). After this mask is obtained, background pixels are usually set to white or black to allow focus on the foreground object.

2.7.2 Problem and Solutions

Many algorithms are based on the basic principle of subtraction pixel values in the observed image from pixel values in the background image. However the nature of the realistic environments in which these systems are used introduces many problems which cause the incorrect classification of a pixel as foreground. This section will briefly describe some of these problems and some of the solution that literature study suggests to alleviate them [8],[9].

- **Changes in Illumination** Alter the colour compostion of the background. In color and intensity based algorithms this change causes a large difference in the subtraction and therefore increases the number of false detections. e.g Turning on a light or a cloudy day.

- **Relocation of a Background Object.** Cause changes in two regions, the new position of the object and the former. Both positions will be picked up as foreground due to change in color. e.g. A stationary object is moved to some nearby location.

- **Non-static Background.** Causes fluctuation in the pixel values causing change in color based detection algorithm that results in false matches in these areas. e.g Tree leaves moving due to wind.

- **Simliar background and foreground color.** These pixels will not be classified as foreground as they are not dissimilar enough. e.g. If someone is wearing clothes that are similar to background color.

- **Shadows** Objects can cast shadow areas which are darker than the background color in that area they will be wrongly classified as foreground pixels due to illumination change in the shadow region. e.g. A person moving in sunlight.

Many methods have been suggested to appease the problems described above that can be added to pre and post-processing stages. Shadows are one of the biggest issues and as such have inspired a wealth of research in the area of shadow removal alone. During preprocessing, Smoothing of the images can be used to reduce the transient environmental noise such as rain. Many algorithms use a Gaussian blur first to average out fluctuating pixel values to alleviate big differences [**?**]. Alternatively when temporal data can be exploited in a video, if a pixel's value is constantly changing over time then it can be assumed it is part of a non-static background object. The background model can deal with events such as objects changing positions by implementing an effective update rule to change the model over time. Background modeling is an area of research itself. One example of an update process is to track object locations. If an object moves and then remain constantly in the same position over a length of time it can be considered to be a part of background. Illumination changes can be handled by exploiting illumination invariance within the color space used. Post processing can be used for data validation to eliminate false positive matched. This can be in the form of the rejection of isolated foreground pixels as they can be assumed to be noise or thresholding on foreground region size. As the subtraction usually only looks at a single pixel, this stage can also examine the value of the neighbors [9].

The first step in developing a background subtractor is to build a model of the background. Since there are no preset background images to use, the subtractor will have to generate a model automatically.

Various methods for background subtraction had been studied and analyzed, The methods , their advantages and disadvantages are noted below.

2.8 Navie Frame difference Technique

Many background subtraction algorithms reduce down to simple subtraction of the pixel in the expected background image from the pixel in the observed image and any significant change indicates that an object of interest has been identified. This is the most nave approach.

First takes the Frame 1^{st} as a Base Frame, then compared the base with rest of the framewe will compare that frame one by one up to n^{th} Frame, then we will find where the base frame is varying from other Frame, At that value subtitute the value of that in the base Frame. But do not include the change that is going to happen in the other than frame then base frame.

But this technique can not provide decent result, if there is not significant different between successive frames.

2.9 Mean Method

In this method the mean of all pixels from frame-1 to frame-n. After that we will compare that mean model to each and every frame in that, we will specify one threshold value if the current pixel fall in that range then we can say that is belongs to the background otherwise we will specify it as foreground object.

This technique gives good result then the previous technique. These techniques also required less no of frame/sec.But these technique will not work well when we significant change in successive frame.

2.10 Graph-Cut Method

This is the one of the technique for background subtraction.This is simplest and cheaptest technique because in this technique it required the background of what ever video it have given to it to find the foreground and background object.

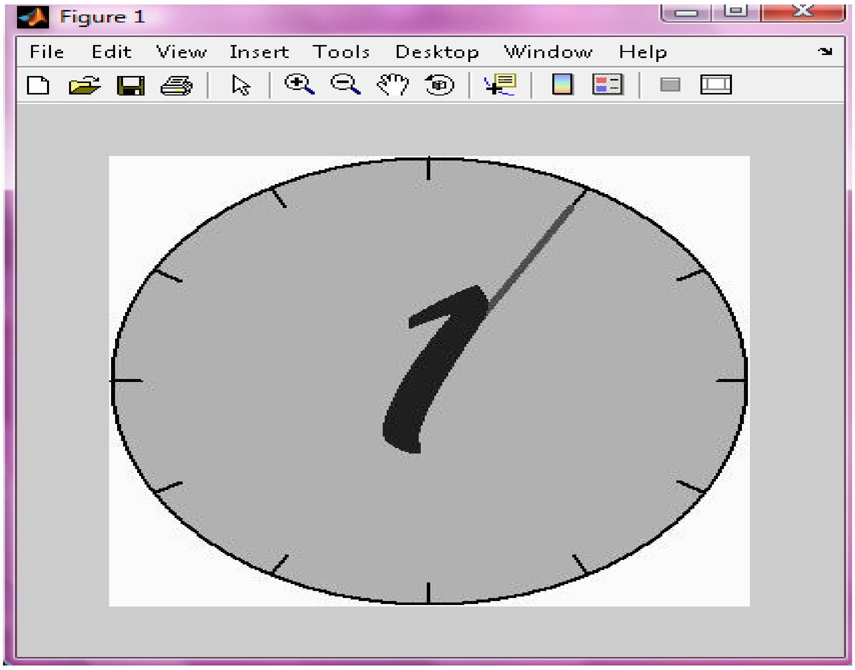

Figure 2.4: Base Frame

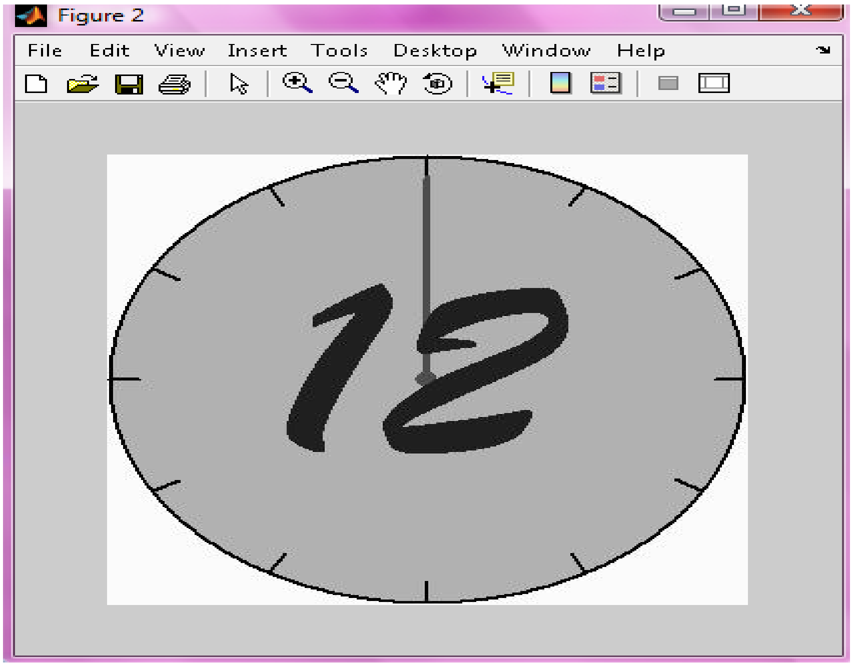

Figure 2.5: nth Frame

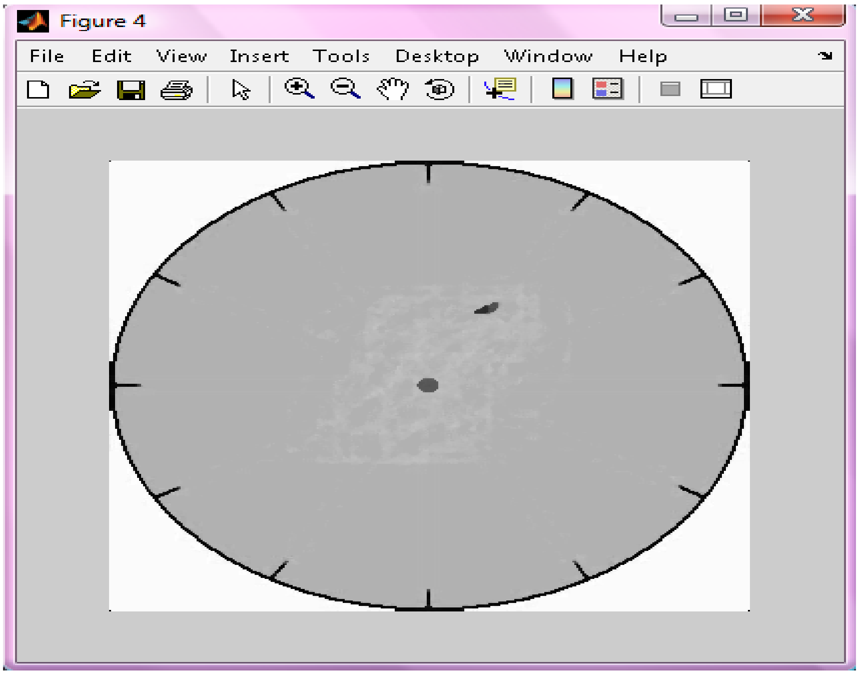

Figure 2.6: Approximate Background

This technique is not useful because in many video we can not get the background without the object or we have video that start from any where in medial of it. In this type of situation these technique is not useful. If we get the background of any video then this technique work that efficently and it save time also.

Chapter 3

Background Subtraction Using GMM

This chapter basically discribes the Background Subtraction using GMM. In this chapter topics included are basics of GMM, Its mathematical proof, Algorithm and Output

3.1 Basic Of Gaussian Mixture Models

Background subtraction is a commonly used class of techniques for segmenting out objects of interest in a scene for applications such as surveillance. It involves comparing an observed image with an estimate of the image if it contained no objects of interest. The areas of the image plane where there is a significant difference between the observed and estimated images indicate the location of the objects of interest. The name background subtraction" comes from the simple technique of subtracting the observed image from the estimated image and thresholding the result to generate the objects of interest.

A Gaussian Mixture Model copes up with multimodal background; hence it is widely used in background subtraction. It calculates each pixel-value from all the sample pixels' mean and variance [8].

GMM is created for each pixels and updated with each new frame. At every new frame some of the Gaussians matches the current value, for them, mean and variance are updated by the running average.

Usually the intensity plot of a pixel is a multimodal plot as shown in the Figure 3.1 . Hence a single Gaussian is unable to capture its multimodal behavior causing the requirement for Gaussian Mixture model.

Even the Literature studies shows that Gaussian Mixture Model is more suitable in such kind of system; hence we will use it in our system.
Basically there are many techniques for the background subtraction. That we have already discuss in the previous chapter. In this chapter we basically focus on the Gaussian Mixture Model.

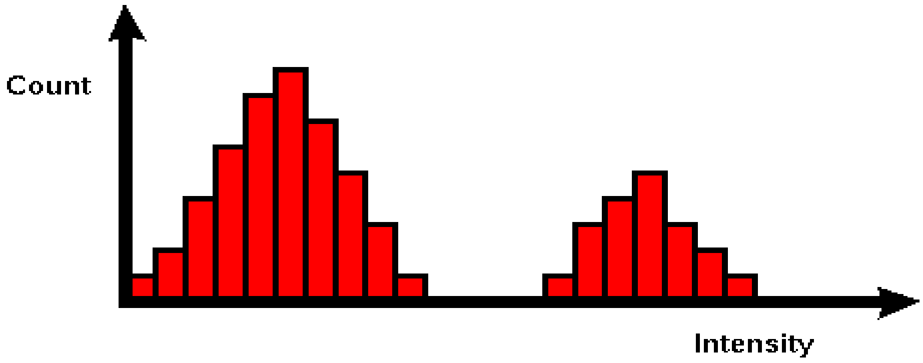

Figure 3.1: Intensity plot of a pixel is a multimodal plot

3.2 How GMM Updation

Basically GMM has three parameter that is going to update pixels by pixels and frame by frame. The Parameter are Mean,Co-Variance and Mixing parameter.

3.2.1 Mathematical Proof of the Gaussian Mixture Model

This section describe the mathematical proof of the GMM. So, Far we know that in GMM there are basically three parameters Co-variance,mean and mixing parameter. In these proof we show that how to update these three parameter accordingly [9].

1 . Learning Gaussian Mixture Models

The GMM G(t) = (C$_i$(t))$_{i=1}{}^m$, is a finite set of clusters of size m, where a cluster at the tth instant is given by,

$$C_i(t) = (\mu_i(t), \delta_i(t), \pi_i(t)) \tag{3.1}$$

Where, μ_i(t), σ_i(t) and π_i(t) are the respective mean vector, co-variance matrix and the mixing parameter of C$_i$(t) at the tth instant.

1.1 Inialization

The GMM is initialized with a single Cluster C$_1$(1) = (X$_1$;σ_{init}; 1:0), where X$_1$ is the data vector at t = 1 and σ_{init} is the initial co-variance matrix whose values are assigned from the domain knowledge.

1.2 Update

In this sub-section we deduce the equations for updating the GMM G(t-1) learned till the (t-1)th instant to G(t) with the current data vector X$_t$.We consider the data

vector to be belonging to the cluster C$_j$(t-1),if (X$_t$ - μ_j(t-1)^T σ_j(t-1)$^{-1}$(X$_t$-μ_j(t-1))¡nλ,, where λ is a user defined threshold and n is the dimension of the data vector (X ϵ R^n). Now, we consider the following cases.
In the fist case, We assume that $\exists_j$:X$_t$ ϵ C$_j$(t-1).Let, N$_i$(t) be the number of data vectors that has been assigned to C$_i$(t) till the tth instant. Thus, we have,

$$\pi_i(t) = \frac{N_i(t)}{t} \tag{3.2}$$

$$\pi_i(t) = \frac{(t-1)\pi_i(t-1) + \delta(i-j)}{t} \tag{3.3}$$

$$\pi_i(t) = (1-\alpha_t)\pi_i(t-1) + \alpha_t(\delta(i-j) \tag{3.4}$$

Where $\alpha_t = \frac{1}{t}$, δ(i-j) is Kronecker's delta.

Now, we update the mean and co-variance in C$_j$(t-1) only. To update the mean, we proceed as follows.

$$\mu_i(t) = \frac{1}{N_i(t)} \sum_{X \epsilon C_j(t)} X \tag{3.5}$$

$$\mu_i(t) = \frac{N_j(t-1)\mu_j(t-1) + X_t}{t\pi_j(t)} \tag{3.6}$$

$$\mu_i(t) = (1-\beta_j(t))\mu_i(t-1) + \beta_j(t) X_t where \beta_j(t) = \frac{\alpha_t}{\pi_i(t)} \tag{3.7}$$

Similarly, we can update the co-variance matrix. From definition, we can compute the co-variance matrix at the tth instant as,

$$\delta_j^2(t) = \frac{1}{N_j(t)} \sum_{X \epsilon C_j(t)} (X - \mu_j(t))(X - \mu_j(t))^T \tag{3.8}$$

$$\delta_j^2(t) = \frac{1}{N_j(t)} \sum_{X \epsilon C_j(t)} (XX^T - \mu_j(t)\mu_j(t)^T) \tag{3.9}$$

$$N_j(t)(\delta_j^2(t) - \mu_j(t)\mu_j(t)^T) = \sum_{X \epsilon C_j(t-1)} XX^T + X_t X_t^T \tag{3.10}$$

$$N_j(t)(\delta_j^2(t) - \mu_j(t)\mu_j(t)^T) = N_j(t-1)(\delta^2(t-1) - \mu_j(t-1)\mu_j(t-1)^T) + X_t X_t^T \tag{3.11}$$

Now further manipulating, by subtituting the update rule for μ_j(t) , it can be shown that the updated co-variance matrix is given by,

$$\delta_j^2(t) = (1 - \beta_j(t))(\delta_j^2(t-1) + \beta_j(t)((X_t - \mu_j(t-1)(X_t - \mu_j(t-1))^T) \tag{3.12}$$

In the second case, it may happen that ∃j : X$_j$ ϵ C$_j$(t-1).In such cases,we initialize a new cluster C$_k$(t)=(X$_t$,$\delta^2{}_{init}$,α_t). If G(t-1) contains less than m clusters, then we add C$_k(t)$ to it. Otherwise, C$_k$(t) replaces the cluster with the lowest weight. More so, in this particular case, the mixing parameters of all other clusters are penalized π_i(t)=(1-α_t)π_i(t-1),i $\neq$ k.

3.3 Implmentaion of Background Subtraction Algorithim

Thus we have now developed the recursive equation for updating of GMM. The algorithm for Background Subtraction is as follows.

Algorithm of Gaussian Mixture Model Thus we have now developed the recursive equation for updating of GMM. The algorithm for Background Subtraction

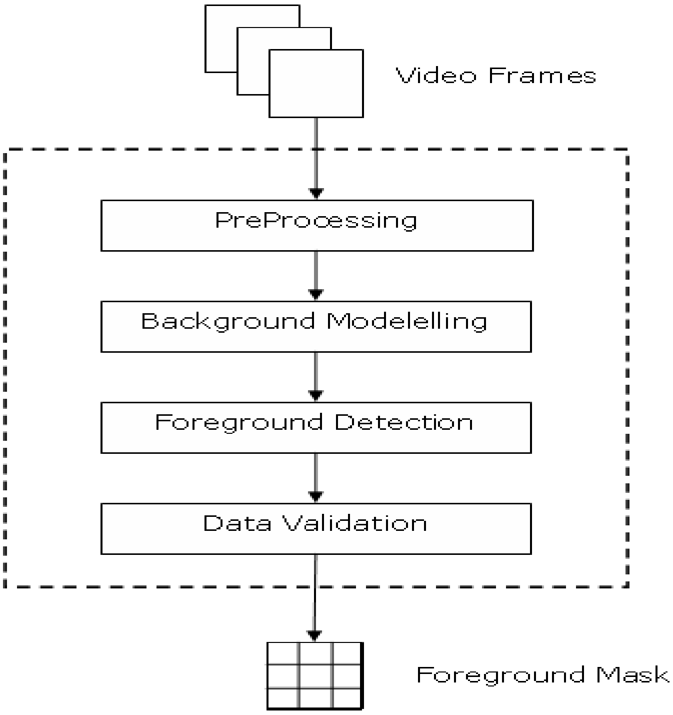

Figure 3.2: Basic outline of Background Subtraction Algorithm

is as follows.

```
ReadInitialFrame();
InitailizeGMM();

while(FramesLeft)
(
ReadNextFrame();
UpdateGMM();
)
Reopen()

while(FrameLeft)
(
ReadNextFrame();
ApplyGMM();

    WriteOutputFrame();
)
```

After this step erosion/dilation is performed, this will help in removing noise. dilation, in general, causes objects to dilate or grow in size; erosion causes objects to shrink. The amount and the way that they grow or shrink depend upon the choice of the structuring element.

3.4 Output after GMM

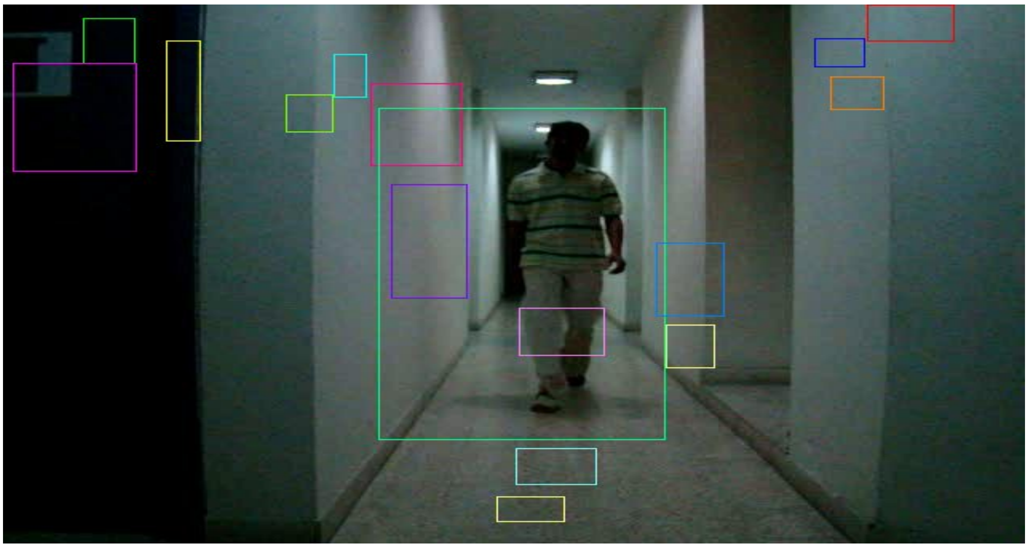

Figure 3.3: Output after GMM

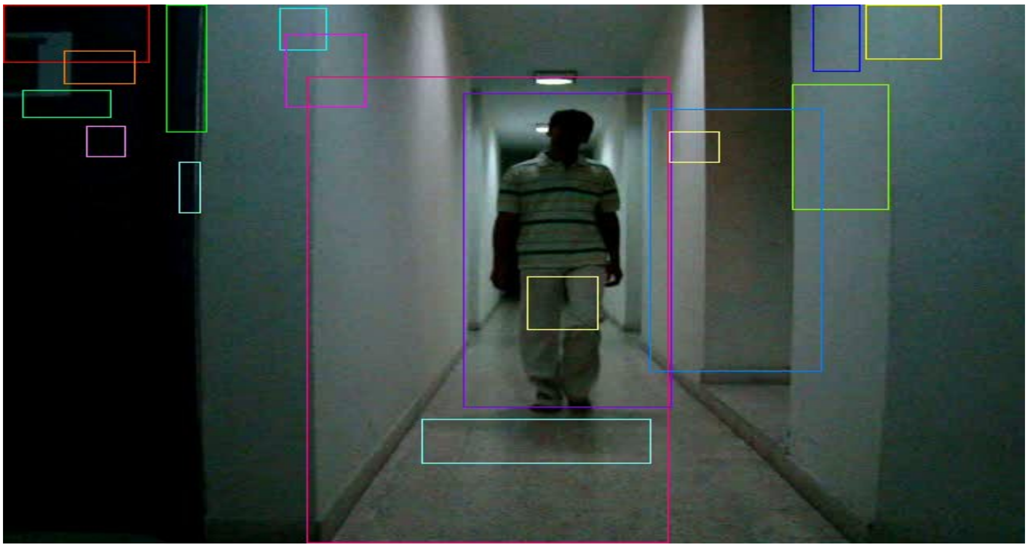

Figure 3.4: Output after GMM

Chapter 4

Haar Transform and Template Generation

We provide a brief, non-mathematical, summary of what wavelet analysis is all about. Examples are discussed from imaging.

4.1 What is wavelet analysis

Wavelets are a mathematical tool for hierarchically decomposing functions. they allow a function to be described in terms of a coarse overall shape, plus details that range from broad to narrow. regardless of whether the function of interest is an image, a curve , or a surface, wavelets offer an elegant technique for representing the levels of detail present. This primer is intended to provide people working in computer graphics with some intuition for what wavelets are , as well as to present the mathematical foundations necessary for studying and using them. In this part ,we discuss the simple case of Haar wavelets in one and two dimensions,and show how they can be used for image compression [5].

Although wavelets have their roots in approximation theory and signal processing , they have recently been applied to many problems in computer graphics . These graphics applications include image editing ,image compression,and image querying ; automatic level-of-detail control for editing and rendering curves and surfaces; surfaces reconstruction from contours; and fast methods for solving simulation problems in animation and global illumination. For a discussion of wavelets that goes beyond the scope of this primer.

We set the stage here by first presenting the simplest form of wavelets, the Haar basis. We cover one-dimensional wavelet transforms and basis functions, and show how these tools can be used to compress the representation of a piecewise-constant function.Then we discuss two-dimensional generalizations of haar basis,and demonstrate how to apply these wavelets to image compression.

4.2 Wavelets in one dimension

A Haar wavelet is the simplest type of wavelet. In discrete form, Haar wavelets are related to a mathematical operation called the Haar transform. The Haar transform serves as a prototype for all other wavelet transforms. Studying the Haar transform in detail will provide a good foundation for understanding the more sophisticated wavelet transforms. In this chapter we shall describe how the Haar transform can be used for compressing audio signals and for removing noise. Our discussion of these applications will set the stage for the more powerful wavelet transforms to come and their applications to these same problems. One distinctive feature that the Haar transform enjoys is that it lends itself easily to simple hand calculations. We shall illustrate many concepts by both simple hand calculations and more involved computer computations.

4.2.1 Haar transform

In this section we shall introduce the basic notions connected with the Haar transform, which we shall examine in more detail in later sections. First, we need to define the type of signals that we shall be analyzing.

We shall be working extensively with discrete signals. A discrete signal is a function of time with values occurring at discrete instants [1],[5]. Generally we shall express a discrete signal in the form

$$f = (f1, f2........., fn) \tag{4.1}$$

where N is a positive even integer which we shall refer to as the length of f. The values of f are the N real numbers f1,f2,.....,fn. These values are typically measured values of an analog signal g, measured at the time values t=t1,t2,...t_N. That is, the values of **f** are.

$$f_1 = g(t_1), f_2 = g(t_2), ..., f_N = g(t_N). \tag{4.2}$$

Like all wavelet transforms, the Haar transform decomposes a discrete signal into two subsignals of half its length. One subsignal is a running average or trend; the other subsignal is a running difference or fluctuation.

Let's begin with trend. The first trend, a^1=(a1,a2,.....,a$_{N/2}$, for the signal f is computed by taking a running average in the following way. Its first value, a1, is computed by taking the average of first pair of values of f:(f1+f1)/2; and then multiply it by root 2 .Thus , a$_1$=(f1+f2)/root(2). Similarly, its next value a$_2$ is computing by taking the average of the next pair of values f:(f3+f4)/2; and multiply by root(2) . Continuing in this way,all of the values of a^1 are produced by taking averages of successive pairs of values of **f**,and then multiplying these averages by root(2). A formula for the values of a^1 is

$$a_m = \frac{f_{2m-1} + f_{2m}}{\sqrt{2}} \tag{4.3}$$

For example, suppose f is defined by eight values, say
f = (4, 6,10, 12, 8, 6, 5, 5).

The average of its first two values is 5, the average of the next two values is 11, the average of the next two values is 7, and the average of the last two values is 5. Multiplying these averages by $\sqrt{2}$, we obtain the first trend subsignal

a^1=(5$\sqrt{2}$,11$\sqrt{2}$,7$\sqrt{2}$,5$\sqrt{2}$).

The other subsignal is called the first fluctuation. The first fluctuation of the signal f , which is denoted by d^1=*(d_1,d_2,..........,$d_{n/2}$), is computed by taking a running difference in the following way. Its first value,d_1 is found by taking half the difference of the first pair of values os f:(f_1-f_2)/2; and multiplying it by $\sqrt{2}$.Continuing in this way, all of the values os d^1 are produced according to the following formula:

$$d_m = \frac{f_{2m-1} - f_{2m}}{\sqrt{2}} \tag{4.4}$$

For example, suppose f is defined by eight values, say
f=(4,6,10,12,8,6,5,5),
its first fluctuation is

d^1=(-$\sqrt{2}$,-$\sqrt{2}$,$\sqrt{2}$,0).

4.3 Haar Transform, 1-level

The Haar Transform is performed in several stages, or levels. The first level is the mapping H_1 defined by

$$f = (a^1|d^1) \tag{4.5}$$

from a discrete signal **f** to its first trend a^1 and first fluctuation d^1. example we have shown in above section.

The mapping H_1 in above equation has an inverse. Its inverse maps the transform signal(a^1—d^1) back to the signal **f**, via the following formula:

$$f = (\frac{a_1 + d_1}{\sqrt{2}}, \frac{a_1 - d_1}{\sqrt{2}}, ..., \frac{a_n/2 + d_n/2}{\sqrt{2}}, \frac{a_n/2 - d_n/2}{\sqrt{2}}) \qquad (4.6)$$

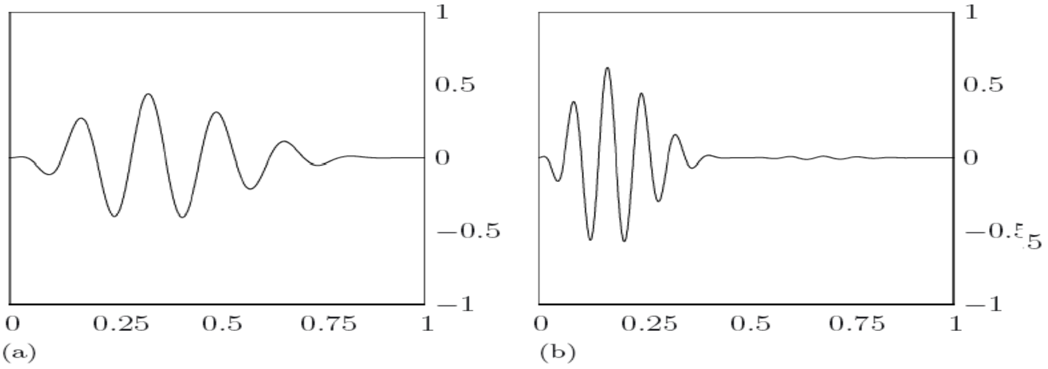

Figure 4.1: (a) Signal (b) 1- Level Haar

4.4 Haar Transform, multiple levels

Once we have performed a 1-level Haar transform, then it is easy to repeat the process and perform multiple level Haar transforms. After performing a 1-level Haar transform of a signal **f** we obtain a first trend a^1 and a first fluctuation d^1. The second level of a Haar transform is then performed by computing a second trend a^2 and a second fluctuation d^2 for the first trend a^1 only [6].

For example, if f = (4, 6, 10, 12, 8, 6, 5, 5) is the signal considered above, then we found that its first trend is $a^1 = (5\sqrt{2}, 11\sqrt{2}, 7\sqrt{2}, 5\sqrt{2})$. To get the second trend a^2 we apply Formula that shown in above to the values of a^1. That is, we add successive pairs of values of a^1 = $(5\sqrt{2}, 11\sqrt{2}, 7\sqrt{2}, 5\sqrt{2})$ and divide by $\sqrt{2}$ obtaining a^2 = (16, 12). To get the second fluctuation d^2, we subtract successive pairs of values of a^1 = $(5\sqrt{2}, 11\sqrt{2}, 7\sqrt{2}, 5\sqrt{2})$ and divide by $\sqrt{2}$ obtaining a^2=(-6,2). Thus the 2-level haar transform of **f** is the signal.

$$(a^2|d^2|d^1) = (16, 12| - 6, 2| - \sqrt{2}, -\sqrt{2}, \sqrt{2}, 0) \tag{4.7}$$

For this signal **f**, a 3-level Haar transform can also be done , and the result as

$$(a^3|d^3|d^2|d^1) = (14\sqrt{2}|2\sqrt{2}| - 6, 2| - \sqrt{2}, -\sqrt{2}, \sqrt{2}, 0). \tag{4.8}$$

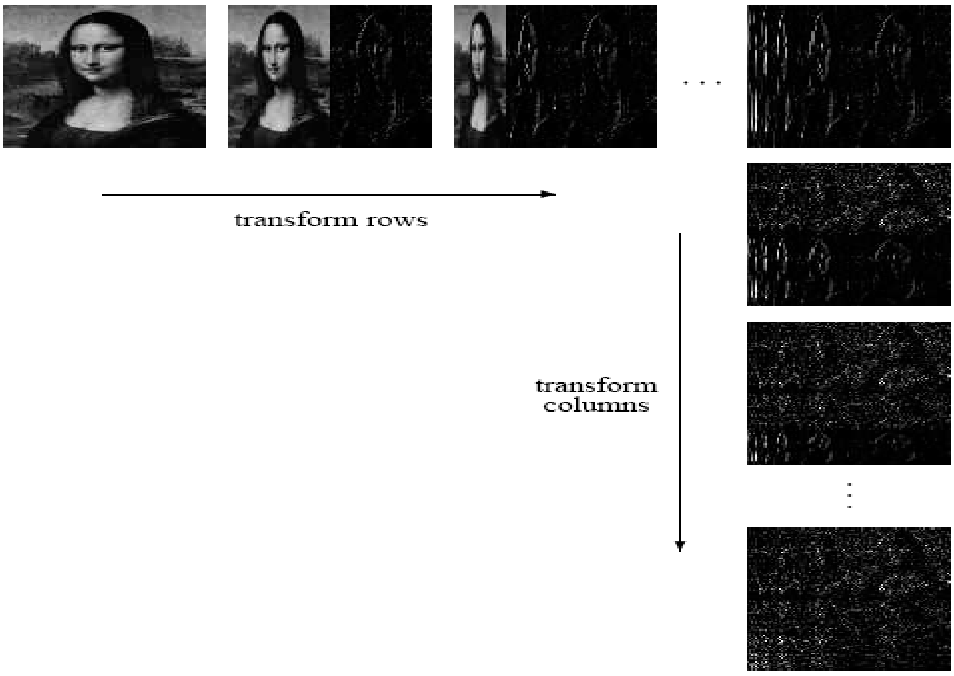

Figure 4.2: Standard decomposition of an image

4.5 Procedure For Standard Decomposition

```
procedure standarddecomposition(C:array[1...h,1..w]of reals [3]) for row 1 to h do
decomposition(C[row,1...w])
end for
for col 1 to w do
decomposition(C[1...h,col])
end for
end procedure
```

4.6 Template Generation Algorithm

Build Template Database (Learning)

- In order to build a template database, we used Background Subtraction to identify Foreground objects. The foreground objects were marked in the original input using coloured rectangles. and simultaneously foreground portion of the input image was cropped and saved.

- Once Background subtraction has run on all the input image set, we use the GUI generated for Template generation

- Main purpose of GUI was to assist the system in identifying False Positives, and help the system in learning from True Positives only, so that invalid data did not get into the templates database.

- The GUI application displays the image with foreground object marked with rectangles. The user is then shown a seperate window having ability to show the cropped images. In this window we can select the cropped images which are TRUE positives and label them.

- Labeling is the process of assigning a class to each cropped image, there can be multiple classes. We can have a class from frontal views of people, side views, people on cycle, etc. Note that this feature can be used to extend the system to detect and count non human objects as well.

- Once the cropped images are either rejected or labelled (put into appropriate directories) we compute the Haar of the cropped images. Steps followed in finding Haar Transform of an image are as follows: Convert the image to grayscale.

- Depending on the level of the Haar, we find the average of two adjoining regions of the gray scale image, We then find the difference in the average of the two regions and save it in place of the original image pixel.

- The haar transform can be computed using basis functions of a varing number of shapes, however we have restricted our use to only rectangular Haar features - Horizontal, Vertical and Diagonal Haar features. In case of Vertical Haar transform we find the difference between two adjoining blocks placed in different rows. Horizontal Haar is obtained by finding the difference between the two adjoining blocks placed in different columns. Diagonal Haar was computed in similar way for blocks in different row and column.

- we then found the average image of all the Haars, and saved the image so obtained, this was repeated for different sized Haars. We computed the Haar for levels 1 to 3 and saved them.

- To speedup the computation of Haar features Integral image representation of the image is computed. The method is based on the one described in [10] and general discussion on Haar transforms for 2-d images given in [5]

Chapter 5

Matching using Normalized Cross-Correlation

Template matching can be used for many things such as vehicle tracking, cell identification and (suitably modified to deal with binary images) to search large sequences of typed text.

Cross-correlation (as we have described it so far), is sensitive to absolute variations in the intensity of the image (i.e. to illumination)

To avoid this effect, we can calculate a normalized cross-correlation function [11],[12]

5.1 Brief of Cross-Correlation

In signal processing, cross-correlation is a measure of similarity of two waveforms as a function of a time-lag applied to one of them. This is also known as a sliding dot product or inner-product. It is commonly used to search a long duration signal for a shorter, known feature. It also has applications in pattern recognition, single particle analysis, electron tomographic averaging, and cryptanalysis.

For continuous functions, f and g, the cross-correlation is defined as [11]:

$$(f * g) = \int_{-\infty}^{+\infty} f^*(\tau * (g(t+\tau))d\tau \tag{5.1}$$

The cross-correlation is similar in nature to the convolution of two functions. Whereas convolution involves reversing a signal, then shifting it and multiplying by another signal, correlation only involves shifting it and multiplying (no reversing).

In an Autocorrelation, which is the cross-correlation of a signal with itself, there will always be a peak at a lag of zero.

If X and Y are two independent random variables with probability distributions f and g, respectively, then the probability distribution of the difference X - Y is given by the cross-correlation f * g. In contrast, the convolution f * g gives the probability distribution of the sum X + Y.

In probability theory and statistics, the term cross-correlation is also sometimes used to refer to the covariance cov(X, Y) between two random vectors X and Y, in order to distinguish that concept from the ”covariance” of a random vector X, which is understood to be the matrix of covariances between the scalar components of X.

Explanation For example, consider two real valued functions f and g that differ only by a shift along the x-axis. One can calculate the cross-correlation to figure out how much g must be shifted along the x-axis to make it identical to f. The formula essentially slides the g function along the x-axis, calculating the integral of their product for each possible amount of sliding. When the functions match, the value of (f*g) is maximized. The reason for this is that when lumps (positives areas) are aligned, they contribute to making the integral larger. Also, when the troughs (negative areas) align, they also make a positive contribution to the integral because the product of two negative numbers is positive.

With complex-valued functions f and g, taking the conjugate of f ensures that aligned lumps (or aligned troughs) with imaginary components will contribute positively to the integral.

In econometrics, lagged cross-correlation is sometimes referred to as cross-autocorrelation

5.2 Normalized Cross-Correlation

For image-processing applications in which the brightness of the image and template can vary due to lighting and exposure conditions, the images can be first normalized. This is typically done at every step by subtracting the mean and dividing by the standard deviation. That is, the cross-correlation of a template, t(x,y) with a subimage f(x,y) is [11],[12]

$$NCC = \frac{1}{n-1}\sum_{x,y}\frac{(f(x,y)-\mu_f)(t(x,y)-\mu_t)}{\sigma_f \sigma_t} \tag{5.2}$$

where n is the number of pixels in t(x,y) and f(x,y). In functional analysis terms, this can be thought of as the dot product of two normalized vectors. That is, if

$$\text{F(x,y)=f(x,y)-}\mu_f$$

and

$$\text{T(x,y)=t(x,y)-}\mu_t$$

5.3 Computational Aspect

In this section we will discuss the complexity part of the matching and what is the impact of it on the system.

- In this approach the complexity of the matching the video is O(mn); where m is the number of the frames in the video and n is the number of entry in the database.
- If we increase the size of the database the searching time will increase significantly. But as well as it give better performance.
- Basically system has tested on the approximate 3000 number of frame and for template matching its take more then 8 hours of time. Because if we consider the worst case for the frame then its has to search for all the n templates for each and every frame so we consider these way we have to perform the NCC for all n template with the input frame , so this is very time consuming procedure.
- So we need efficent trade off between the database size and the searching technique.

5.4 Matching

After Background subtraction system has generated the template database its include the foreground object . On that we perform the haar transfrom [12].

Now it take one by one frame perform haar transform on it and then template is search on to the input image by using NCC.

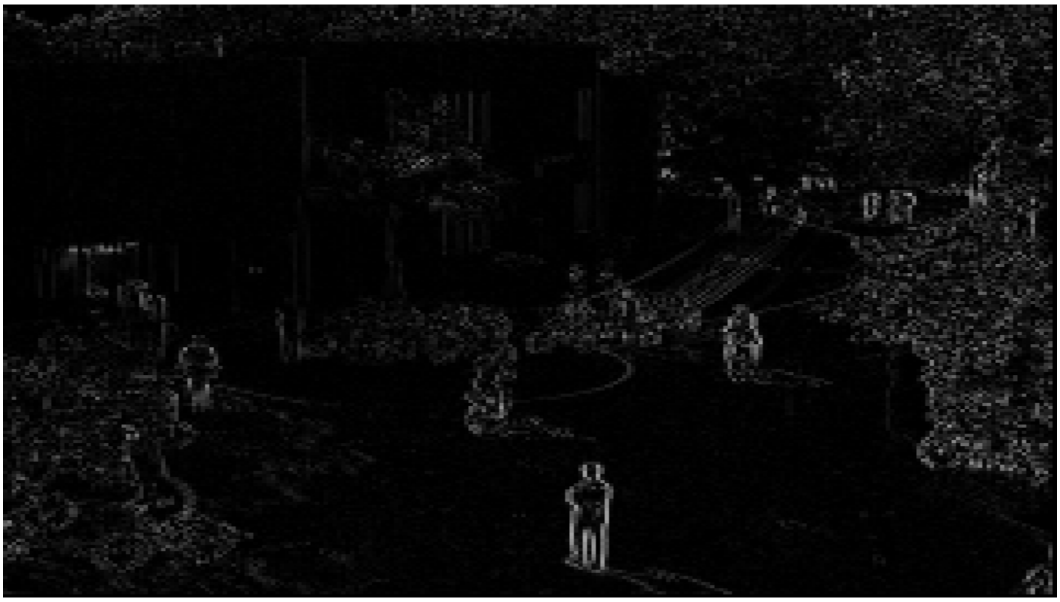

Figure 5.1: Haar Transform of Input Image

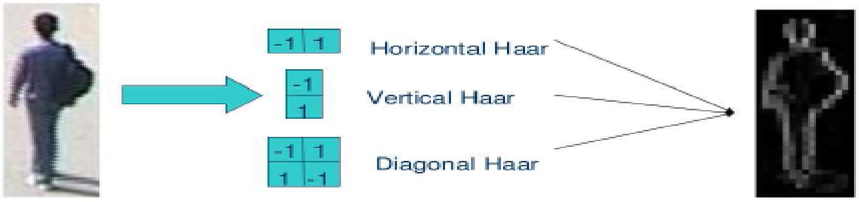

Figure 5.2: One of the template from the database

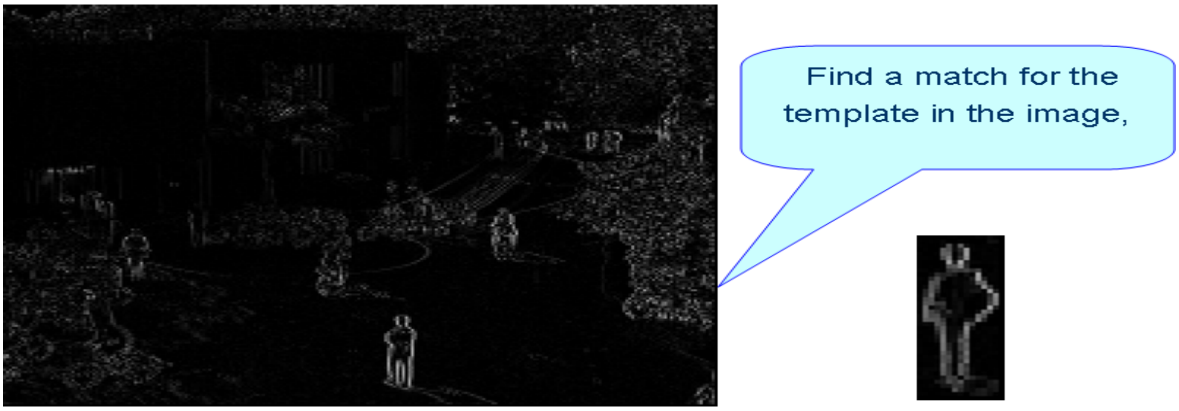

Figure 5.3: Search in the Image

Chapter 6

System Implementation

6.1 Basic steps to Implement the System

- Extract all frames from the video (e.g. avi or mpeg). Find moving objects from each and every frame by applying Gaussain mixture model.

- Details of Gaussain Mixture Model is given in Chapter 3. Implementation aspect is addressed here. Assign Gaussain Mixture Model to every pixels of base frame. Each Gaussain Mixture Model consists of array of clusters. Each cluster has three parameters. For each frame update the parameters as discussed in Chapter 3.

- Check each and every pixel falls into which cluster. If a pixel does not fall in any of clusters then check maximum cluster size is achieved or not. If it is not achieved then assign the new cluster to it. If the maximum cluster size is achieved then, merge two clusters with the lowest membership component, and penalized the rest of the clusters. Assign newly empty cluster to that pixel.

- Create rectangle by finding connected component using a simply 8-way connected strategics. Once the blob is obtained find the x$_{min}$, x$_{max}$, y$_{min}$ and y$_{max}$ respectively and draw rectangle from it.

- Find the Haar Transform details given in the Chapter4. Stored the Haar Transform coefficents of the moving objects in the database, which is collection of the different types of objects.

- Matching is done using Normalized cross Correlation. To match each frame, find Haar Transform and compare it with template stored in the database using Normalized Cross Correlation. Searching time will increase as number of entry in the database is increases. So the trade-off between the database entry any searching has to been found.

- **Background Subtraction**

  Here GMM is used for performing background subtraction. A Foreground Image Mask is created which in turn applied on every frame to identify the foreground objects. Each pixel of the Foreground Image Mask is a GMM. Further, as mentioned earlier, each GMM consists of certain number of clusters. A cluster consists of mean, variance and mixing parameter. Since image is of RGB format there is a need for 3 different Foreground Image Mask, one for each channel. Instead, just to have a single Foreground Image Mask for the sake of simplicity, the Clusters are implemented as structure in which each variable is an array of 3 elements to represent each channel (RGB).

Clusters

Mean [3];

Variance [3][3];

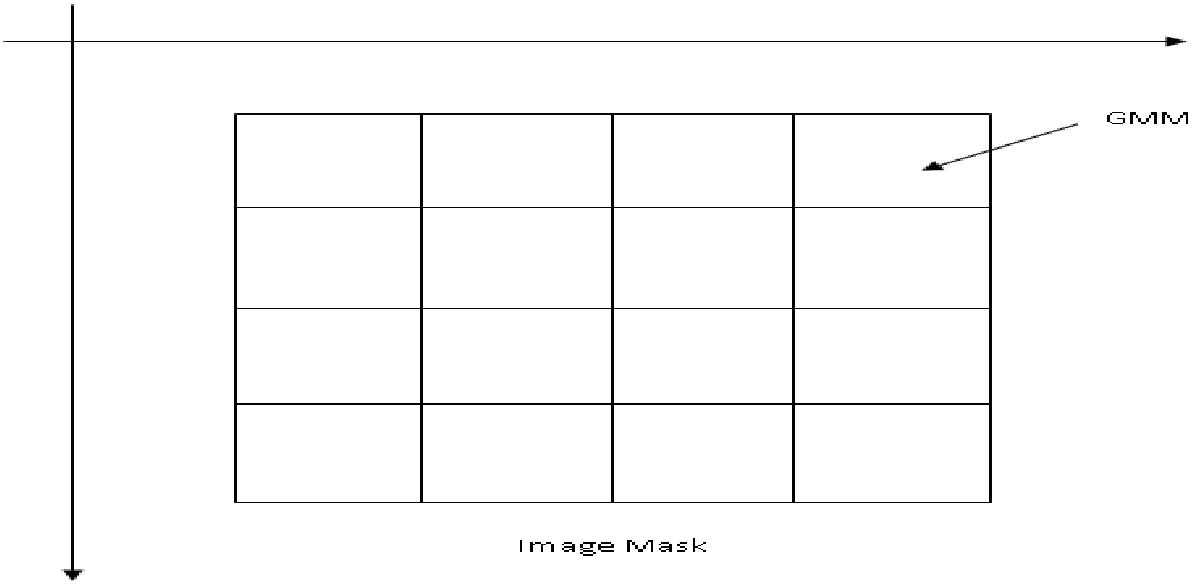

Figure 6.1: Image-Mask

MixingParameter [3];

While Initializing, Mean is assigned the value of pixel and Mixing Parameter is always initialized by 1, but for variance there is a need for some predefined constant. Hence we take INITIALTHRESHOLD as our initial value of variance. This value governs the merging of pixels in the clusters. Hence this value must be carefully chosen after performing sufficiently many test with different values under different scenarios.

Similarly, GMM is also defined as a Structure which is as follows

GMM

Clusters c[MAXCLUSTERS];

ClustersUsed;

Pixel;

PixelCopy;

In simple terms Foreground Image Mask is used as two dimensional array of GMM (GMM[HEIGHT][WIDTH], where HEIGHT and WIDTH are the height and width of the video frames respectively).

The ClustersUsed variable keeps the counts of the total clusters which are used in the GMM. The Pixel variable identifies whether the current pixel is foreground or background. If Pixel is 1 it indicates that it is part of background. The PixelCopy is just a copy variable which is required to create a temporary copy of the current Pixel. The MAXCLUSTER is a constant predefined value. It has a lot of impact on the performance of system, hence its value must be properly chosen after sufficiently testing it under different scenario.

An abstract algorithm of Background subtraction is described in the earlier chapter. Here detailed algorithm for Background subtraction is explained. The details of the equation used in the following algorithms were explained in the Background subtraction chapter and are omitted here for the sake of simplicity.

BackgroundSubtraction

```
a. for(n=0;n<=TOTAL-FRAMES;n++)
b. for(i=0;i<HEIGHT;i++)
c. forj=0;j<WIDTH;j++
d. extract pixel.
e. for(k=1;k<ClustersUsed; k++)
   Check whether the pixel lies within this cluster If(Pixel lies inside the cluster)
   { Set MacthedClusterIndex to the cluster which is more closer to pixel.
```

```
f. If( a Matched Cluster is Found)
   Update that Cluster; Penalize the Rest of the Clusters;
g. Else
   /* The pixel does not lie in any cluster Hence we need to add a new cluster */
h. if(ClustersUsed¡MAXCLUSTER) { Initalize the Cluster with the current pixel value;
i. Increment ClustersUsed.
j. Else { Find the Cluster with the minimum mixing Parameter and replace it with a newly created cluster initialized from the current pixel value }
```

Once the Background subtraction is over it is time to extract the foreground objects. The erosion and dilation makes sure that the pixels that were connected get connected which due to background subtraction may have been disconnected. Erosion and Dilation are the fundamental operations in Morphological image processing. The details on these operations can be referred from any Image processing text book. Now in order to perform these operations, a copy of image must be preserved. Hence now the PixelCopy variable of GMM structure comes into picture. Essentially, this is just same as creating a new copy of image. It is just that the PixelCopy variable takes advantage of Locality of reference i.e. for the current pixel it is usually required to make the change to current location in the copy of image.

The Algorithm for Erosion is explained below.

```
a. Erosion(distance d)
b. for(i=0;i<HEIGHT;i++)
c. for(j=0;j<WIDTH;j++)
```

```
   /*Check if the current pixel is set*/
d. If(pixel(i,j) =0)
   /* Assign the same value to neighboring d pixels */
e. for(k= -d ; k<d; k++)
f. for(l= -d; l<d ;l++)
g. PixelCopy(i+k,j+l)=0
```

The erosion the structure element is a matrix of size d.. Basically, If the current pixel is 0 then set the neighboring d pixels to 0. The dilation is dual of Erosion. i.e erosion on 0 is equivalent to dilation on 1. Hence in the above algorithm if we change the 0 to 1 the operation becomes dilation.

Once the Erosion Dilation is over then extraction of the blob needs to be done. For the extraction the eight-way connected strategy is used recursively. A custom stack is implemented for the same. The algorithm for the same is given as follows.

- **Extract Blob**

```
a. for(i=0;i<HEIGHT;i++)
b. for(j=0;j<WIDTH;j++)
c. If(PixelCopy(i,j)==1)
d. PixelCopy(i,j)==0 Put the PixelCopy(i,j) on stack
   While(stack not empty)
   Bring the stacktop pixel; Check the 8-neighbors and put the one which are set on stack
   Also Keep track of min and max x,y cordinates
```

e. Draw rectangle around the min and may x,y coordinates.

- **Haar Transform**

The Blob thus obtained are then labeled and stored in the database. A varied dataset is required before the system could be put to real test. The Blob's Haar Transform is stored in the database. In order to avoid redundancy, it is necessary to generate a generic template for different objects to be counted. The different blob's of a similar object are used to generate a template for it. Essentially, the template is just the average of the Haar Transform of blobs of similar objects.

The Algorithms for Haar Transforms are as follows.

```
a. Haar1D(Width w)
b. While(w>1)
c. w=w/2;
d. for(k=1;k<w;k++)
   Ary[k]=OrgAry[2*k]+OrgAry[2*k+1]; Ary[k+w]=OrgAry[2*k]-OrgAry[2*k+1];
e. for(k=1;k<w*2;k++)
   OrgAry[k]=Ary[k]
f. Haar2D(Height ,Width ) h=Height; w=Width;
g. While(w>1 || h>1)
h. If(w>1)
i. for(i=0;i<Height;i++)
   Copy into OrignalRow to TempRow Call Haar1D on TempRow; Copy the
   Haar1D of TempRow to OrignalRow;
```

j. If(h>1)

k. for(i=0;i<width;i++)

Copy into OrignalCol to TempCol Call Haar1D on TempCol; Copy the Haar1D of TempCol to OrignalCol;

- **Matching.**

To count the object in the frame, it is first required to identify the object in the frame. This step is done by matching the stored templates against the Haar Transform of the current frame. The Normalized Cross Correlation is used for it. It is required that we need to test match all the levels of Haar of the Templates with the current frame.

Once identified the objects it is just required to increment its count.

Chapter 7

Test Analisys & Result

7.1 Analisys of Result

- Here system is the tested on the basically two video , 1^{st} the system-clock video and the 2^{nd} is the Pedestrain video.

- From these video the template is generated and stored in the database.In these system it have around 43 template in the database.Which is the collection of the different view of the pedestrain in it. But in these database there is no such template with the side view of the objects, because here in the example video it not clearly getting the side view of the objects.

- **Computational Aspect** Here the Computation time basically depend on the two basic things. 1^{st} is the size of the database and 2^{nd} is the searching method that going to use for it.Here for these system it created about the 43 different template there are in the database, and in the video it have around about 2900 frames. Complexity of the system in the worst case is O(43*2900) in generalize terms it would be O(nm), where n is the number of template in the database and m is the number of frames in the video.

- Here for the Pedestrain video the time is taken to execute the whole system is

Total-Frame	Obj. Detected	Neg. Detection	Actual-Object	Percentage-Detection
2900	45	30	108	43

Table I: Total Count of Object

around about 26 hours INTEL CORE-DUO (1.60 GHz,2 GRAM).In Finding the moving object it take around 16 hours and for the matching purpose it take around 7-8 hours.Here these timing is base on the observation.

- From the observation it could suggest the finding the moving object takes more time to compute it parameters, though Gaussian Mixture Model is useful for real time video. So for that it have to sample the certain frame, because in subsequence frame the object does not change drastically. So suggestion is that every 50_{th} or whatever user want frames selected from the video.
- In these video I did not get the side view template perfectly so that type of template is not there in the database.
- According to that on an average it got 43 percentage of accuracy.

Chapter 8

Conclusion and Future Scope

8.1 Conclusion

The basic working of Haar based matching and Object counting was implemented, however the accuracy was found to be very low. This can be improved by increasing number of templates. The code for matching needs to be speeded up, based on heuristic approach or a probabilistic model.Here the accuracy is to low So for that it need sufficient amount of templates in the database and also here for matching the NCC is use, but here there are certain different technique can be used for the classification purpose.

8.2 Future Scope

- Need to build a larger database of templates.
- Efficient matching algorithm is required,(e.g. Support vector machine ,Artificial Neural Netwrok..etc).
- In a video frame sequence the number of object do not change drastically in adjacent frames, therefore the search space could be reduced by tracking people across frames

References

[1] P. Sinha, "Pedestrain detection using wavelet template," *IEEE Computer Society Conference*, pp. 193–199,, june 1997.

[2] T.Tsukiyama and Y.Shirai, "Detection of the movements of persons from a sparse sequence of tv images pattern recognition," *ACM SIGCOMM Computer Communication Review*, April 1985.

[3] K.Rohr, "Incremental recognition of pedestrains from image sequences," *Computer Vision and Pattern Recognition*, pp. 8–13, 1993.

[4] M.Leung and Y. Tang, "A region based approach for human body analisys.pattern recognition," july 1989.

[5] J. S. Walker, *A Primer on Wavelets and Their Scientific Application*, vol. 2. 2008. http://www.taylorandfrancis.com.

[6] F.Riesz and B.Sz-NAgy, "Funtinal analisys," 1955.

[7] P.Sinha, "Object recognition via image invariants ,a case study," may 1994.

[8] M. Piccardi, "Background subtraction techniques : A review," April 2003. `www-staff.it.uts.edu.au/~massimo/BackgroundSubtractionReview-Piccardi.pdf`.

[9] S. Deane, "A comparison of background subtraction techniques," *IEEE*, April 2003. http://portfolio.ecs.soton.ac.uk/20/3/irp-report-ieee.pdf.

[10] S.Mallat, "A theory of multiresolution decomposition: The wavelet representation," *IEEE Transaction on Pattern Analisys and Machine Intelligence*, july 1989.

[11] Website. `http://en.wikipedia.org/wiki/Cross-correlation`.

[12] Website. `http://www-cs-students.stanford.edu/~robles/ee368/matching.html`.

Index