Chapter 1
INTRODUCTION

1.1 Automated Surveillance System for people tracking

In recent years, automated surveillance has gained much attention. It is of utmost importance to know what is happening around in house front, public places and everywhere. Possible threats and suspicious activities have to be detected in order to prevent vandalism [21]. Continuous monitoring of a place of interest is needed and this involves deploying a human to watch videos all day long, which is really a daunting task. So there is a need to assist him in monitoring people. Although it is much difficult to fully automate without human intervention, certain simple questions like “How many people are there in the scene?” , “What is the identity of each person?” , “Were this person been here before?” , “If so when is the last time he was here?” and “ With whom has he interacted?“ can be answered to assist security personnel. An attempt to develop a system for people monitoring in public places has been tried with a view to answer these vital questions.

A simple framework for surveillance has been proposed with various functionalities like tracking multiple objects in the scene, keeping track of objects even under occlusion, activity recognition of the objects. First the foreground is extracted using the background subtraction methods discussed in [2] [6] [16] [18] [27] [28] [32]. The blobs are extracted from the background and only the blobs, which are above an experimentally tested threshold, are considered for further processing. Tracking of objects is achieved by using the color distribution histogram models [5] where each person is visually interpreted as a color histogram model based on the pixel color of the blob. This is stored as a model and the blobs coming in from successive frames are checked for similarity. If there is a high match, the blob is assigned to the corresponding model and if that blob does not match with any models available then it is stored as a separate model. This process is repeated for all frames in the live feed. Thus tracking of objects in the scene i.e., maintaining correspondence between blobs of two frames is achieved using color histograms.

The problem of detecting the objects even under occlusion is tried to be solved using histogram distance model. Occlusion refers to the state of where an object of interest's view is blocked partially or fully by some other object or even a background. In this, if the current blob is larger in size i.e., above a fixed threshold than the previous frame, the pixel classifier that classifies each pixel to a model starts functioning. Histogram is computed for each pixel along with its 8 distance pixel neighbours and checked with histograms of all the models for similarity. If there is a best match, the pixel is assigned the colour of that model pixel. After all the pixels in the joined blob are classified in this fashion, the resultant coloured blob will be the individual objects with the corresponding model colours taken from the knowledge base that is created. Thus tracking and labelling of objects even under occlusion is achieved.

A framework for recognising the activities of the detected objects is tried out by using skeletonisation and star skeleton models. A distance plot from centroid to each of boundary pixels is plotted with distance of the pixel from centroid along y-axis and boundary pixels along x-axis. Features like mean, median, zero crossings, energy, RMS value etc., are extracted from this two dimensional signal and are classified accordingly for distinguishing activities like walking, running & idle.

1.2 Overview of the thesis

The remainder of this thesis is categorised into six different chapters.

Chapter 2 deals with the **related work** involved in the real-time surveillance domain. It discusses the previous work of the researchers and throws light on the various methods, concepts adopted by the researchers.

Chapter 3 involves **designing the system** of how the problem is tackled with the different modules and flowcharts explaining them.

Chapter 4 elaborates on the **motion detection** part of the system and it explains how actually the blobs are detected and the pre-processing involved beforehand like background subtraction, Thresholding and noise removal.

Chapter 5 explains about the **tracking** of the objects detected from motion detection module. It gives away how the blob is tracked and how the motion correspondence is established across frames using the colour histogram model.

Chapter 6 discusses the **activity recognition** module of how the activities like walking and running are identified and what are the results of achieving the discrimination. It involves the explanation of the contour-distance signal method, used here in the system, for activity recognition.

Chapter 7 discusses the results & inferences obtained in each module while conducting experiments with various dataset videos. **Future work and research** that can be done further is also discussed in a broader sense.

Chapter – 2
RELATED WORK

2.1 Concepts and Terminology

In this thesis, a number of computer vision terms will be used which are either not commonly well defined or which are sometimes used in different ways by different people. The following definitions specify how these terms are used.

2.1.1 Definition (blob): A blob is a connected set of pixels in an image. The pixels contained in a blob must be locally distinguishable from pixels that are not part of the blob.

2.1.2 Definition (region): A region is the rectangular subimage defined by the minimal enclosing bounding box of an image area containing one or more blobs.

2.2 Related Work

Real time automated visual surveillance has been a popular area for scientific and industrial research since it was pioneered by Badler and Hogg [10][26]. People tracking naturally play a key role in any visual surveillance system. Although this field has been explored for quite a few years, it only contains fewer numbers of contributions in research. Only in this decade it started to gain much attention to make sure everything is in place as far as security is concerned. A number of tracking algorithms for different applications have been presented here (Baumberg, 1995; Bremond and Thonnat, 1997; Cai et al., 1995; Gavrila and Davis, 1996; Haritaoglu et al., 2000; Johnson, 1998; Khan et al., 2001; Lipton et al., 1998; Sidenbladh et al., 2000; Wren et al., 1995).

One example for a background model is a pixel wise median filter of the video images over time. There are number of other background modeling methods available. One simple method is this: a constant image of the scene is taken when there were no people in the image and considered as a template image for subtraction. In some situations such a simple method is not sufficient, e.g. in outdoor scenes where the lighting conditions change over time, and where the background itself can contain movement, as it happens with trees in the wind. More complex methods address these difficulties using elaborate statistical models for each pixel. Widely used examples are the multi-modal statistical motion detector from MIT that models each pixel with a mixture of 3 to 5 Gaussian distributions [6] and the non-parametric statistical method [2] from the University of Maryland. These methods have the disadvantage that they require a significant amount of computational time, which at present limits their use in real time systems. Some methods [33] try to overcome this by carrying out the computationally expensive background modeling process in an offline training stage. This means, however, that the system cannot adapt to changes in background or lighting conditions while it is operating.

2.2.1 Background models

Several models have been put forward for background maintenance and subtraction in the literature [2] [6] [16] [18] [27] [28] [32]. All the models for background can be roughly divided into pixel-level and nonpixel-level. Pixel-level background models in fact are models of pixel process. The value of a particular pixel over time is called pixel process, i.e. pixel process is a time series of pixel values. For a particular pixel *{x,y}*, the pixel process can be written as:

$$\left\{X_1, X_2, \ldots X_t\right\} = \left\{I(x, y; t) : 1 \le i \le t\right\} \qquad \text{.........(2.1)}$$

It was known that the background image could be obtained by exposing a film for long enough period of time. It is said that the background image is average of a image sequence that is long enough. For pixel *{x,y}*, if $B(x,y;\ t)$ simplified as B_t stands for estimate of background value at time t.

$$B_t = \frac{1}{t}\sum_{i=1}^{t} X_i \quad \text{or} \quad B_t = \frac{t-1}{t} B_{t-1} + \frac{1}{t} X_t \qquad \text{.........(2.2)}$$

Moving objects can be identified simply by thresholding the distance between B_t and X_t. To handle the change of lighting condition, a moving-window average method is proposed. At one instant, exponential forgetting is used. The background update equation is:

$$B_{t+1} = (1-\alpha)B_t + \alpha X_{t+1} \quad \text{………(2.3)}$$

An obvious problem with this equation is that all information coming from both background and foreground is used to update the background model. If some objects move slowly, these algorithms will fail. The solution to this problem is that only those pixels not identified as moving objects are used to update background model. This is implemented by updating equation:

$$B_{t+1} = B_t + (\alpha_1(1-M_t) + \alpha_2 M_t)D_t \quad \text{………(2.4)}$$

where D_t is the difference between present frame and background model, and M_t is the binary moving objects hypothesis mask.

A generalization of the previous approaches was presented in the following paper. Each pixel process was modelled with mixture of K Gaussian distributions. Problems like "tree moving", "moved objects" were solved in a reasonable speed. The probability that a certain pixel has intensity x_t at time t is estimated as:

$$P(X_t) = \sum_{j=1}^{k} \frac{w_j}{(2\pi)^{\frac{d}{2}} \left|\sum_j\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-\mu_j)^r \sum_j^{-1} (X_t-\mu_j)} \quad \text{……..(2.5)}$$

$$B = \arg\min_b \left[\left(\sum_{j=1}^{b} w_j/\sigma_j \right) / \left(\sum_{j=1}^{k} w_j/\sigma_j \right) > T \right] \quad \text{……..(2.6)}$$

where w_j is the weight, μ_j is the mean and Σj is the covariance for the jth Gaussian distribution. The k distributions are ordered based on w_j / σ_j and the first B distributions is the model of background. In all of the probability models above, pixel processes are treated as data set without order. Temporal context of data is not used explicitly. A general Topology Free HMM is also available, and several states splitting criterions are compared. A three-states HMM without adaptation is introduced to model background. Nonpixel-level models include Eigen-background, Wallflower. Principle component analysis is used to determine means and variances over entire sequence (whole image as vectors) in Eigen-background. But, being not adaptive is a shortcoming of this algorithm. The idea of Wallflower processes images at various spatial scales: pixel level (linear prediction), region level (filling) and frame level (model switch). He adopts linear prediction algorithm to model pixel process, and by updating the stored previous image data, background maintenance is accomplished, but more storage space is need. Some nonpixel-level model can solve the problems that are thought to be inherent in pixel model.

2.2.2 Model Complexity

One important issue when modeling the visual appearance of people or other objects is the complexity of the model. The two foremost problems are the facts that humans are complex articulated bodies, and their appearance changes depending on the camera view and on any occlusions. Simple models of humans are usually easy to implement and fast enough to be used in real time systems. However, they often lack the ability to adapt to the changing visual appearances of humans under changes of posture and viewing angle. These models might also have difficulties in the presence of occlusion. More complex models have a greater ability to adapt. This makes them more general and more powerful, however, the increase in complexity has some disadvantages, too. Apart from being difficult to realize, their computational demands can make them unsuitable for real time applications. Moreover, the high dimensionality of the "state" of the human model produces the need for some means to handle difficulties like visual ambiguities. For instance, if the tracker models the position and posture of a person's hand, this aspect might be undefined when the hand is occluded. Complex models are also more likely to fail in situations that are not expected by the system (for example, missing markers on objects, sitting people as opposed to walking people, change in lighting situations, moving background etc).

These issues make the process of choosing a suitable model an important issue in the design of a system. The trade-off between the advantages and disadvantages discussed above is usually decided by the demands of the respective application such as speed and detail of required output.

2.2.3 Classification of Human Models

While the level of complexity in human modeling for tracking varies over a wide range, the methods can be classified into three main categories of increasing model complexity:

Category 1: Methods using region- or blob-based tracking, sometimes with additional classification schemes based on colour, texture or other local image properties. Sometimes these methods are called “model-free”, although the computer program used for tracking can be said to implicitly define a model.

Category 2: Methods using a 2D appearance model of a human being. The dimensionality “2D” relates to the fact that the model represents the human’s appearance in the two-dimensional image as opposed to the 3D space it is moving in.

Category 3: Methods using an articulated 3D model of a human being, that is, a model of a person in 3D space. Another aspect that can be modeled is the way in which the appearance of an object or its representation (e.g. model parameters) varies over time. An example of a temporal appearance model of a person has been given by Johnson 1998. The underlying assumption is that the appearance of a moving object varies in a regular manner over time and can therefore be modeled and predicted from frame to frame.

2.3. Discussion

The more complex the model used for people detection and tracking, the better the system can handle the particular situations for which it is trained. However, with the level of complexity in the model the complexity of the computational problem also increases. The use of a 3D model means that the internal representation of the person cannot be taken from the image because an image only gives two-dimensional measurements. One approach to solve this problem involves the use of more than one camera, e.g. a stereo camera system, which in effect approximates 3D measurements of the person. Other approaches use iterative methods with projection from a 3D representation of the person onto the 2D image space. This makes the problem highly nonlinear and thereby often incompletely solvable and computationally expensive. It should be possible to speed up part of this process by using dedicated video hardware (e.g. existing 3D accelerated video graphics cards) for the projection 3D to 2D. However, the nonlinear and complex optimization problem remains. Currently no such accelerated tracker exists.

Chapter 3
SYSTEM DESIGN

3.1 Introduction

This chapter deals with the overall design of the system with the detailed flow control and parameters flow inside the system. The chapter throws light on the system as a whole and gives a sense of the whole working methodology.

3.2 Design of the system

The system is designed in such a way that it integrates several modules which can function independently on its own. Modules are integrated for effective usage of the whole scenario in accomplishing the tasks taken. **Figure 3.1** shows the overall design of the system with different modules.

The different modules includes Motion detector, Head detector, Identity tracker, Activity recogniser and the Integrator. All these modules can run independently giving outputs satisfactorily. But the system is designed to output various functionalities with integration of all outputs.

The next section gives a detailed view of all the modules and its control flow of parameters.

3.2.1 Motion Detector

The motion detector module aims at detecting the motion in the video and extracting the foreground pixels that are in motion. **Figure 3.2** shows the motion detector module in a flowchart fashion. It subtracts the background that is static and extracts the foreground, humans in this case who is moving.

Input: Video image

Output: Blobs of humans who are moving in the scene

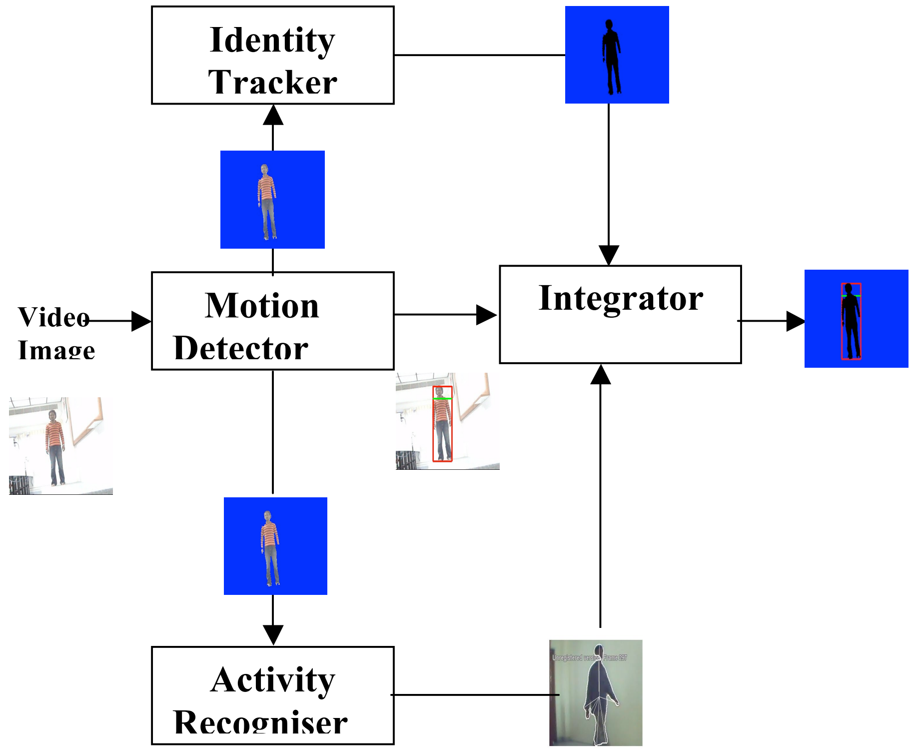


FIGURE 3.1: Overall architecture of the tracking system

3.2.2 Head Detector

The head detector module detects the head from the blob given by the motion detector module. This module is embedded in the motion detector module itself. The bounding box supplied by the motion detector module is taken and with intuition the head position is approximated.

The algorithm of the head detector is given as follows:

Input: Blobs with bounding box

Output: Box bound blobs with head detected

1. Find the length of the bounding box and mark the half of it giving the torso, head in first half and the thigh, leg in other half.
2. Mark one-third of the first half to get the approximate head position.

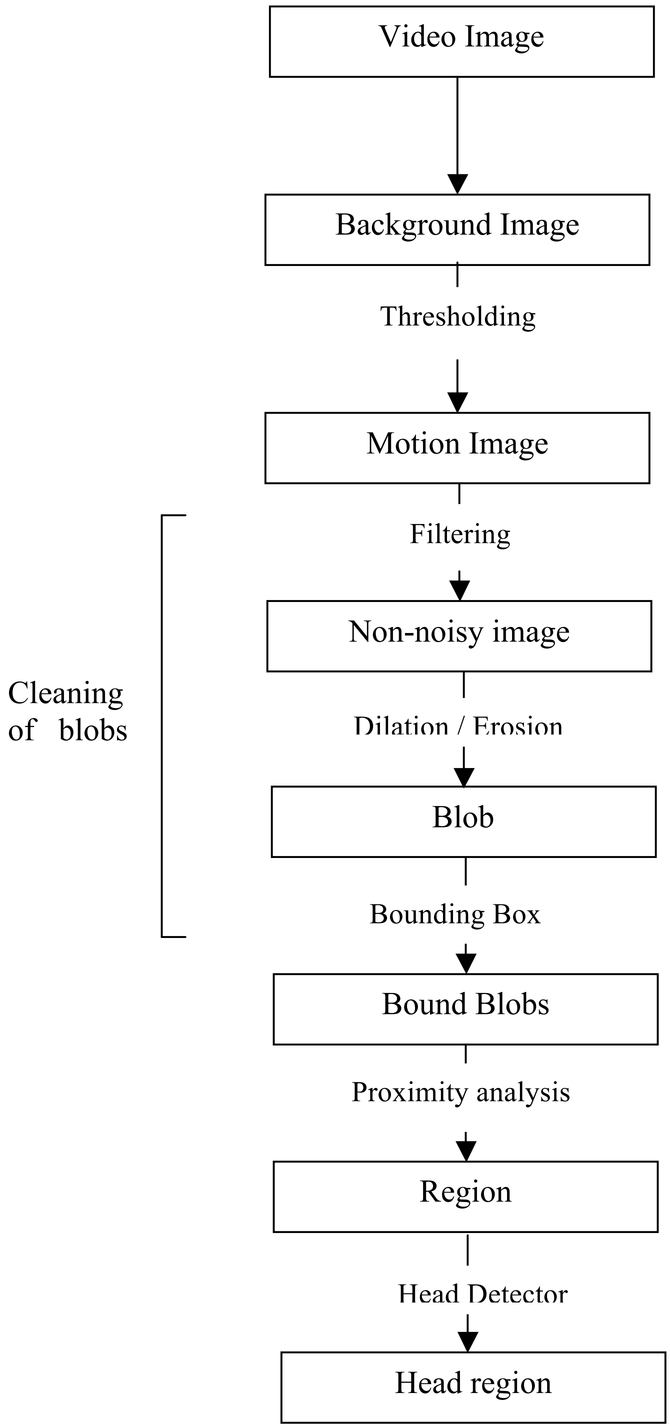


FIGURE 3.2: Process flow of motion detection module

3.2.3 Identity Tracker

The identity tracker module identifies the blobs and tracks it across all frames. This module helps in solving the issues like correspondence of blobs between frames. The algorithm below gives the process flow of the identity tracker module. Tracking is done through the color histogram model that is created on the fly in real time. The histogram model of the blob establishes the correspondence of the blobs in consecutive frames. The centroid of the blob is computed and is stored across all frames for offline processing providing object's trajectory. This histogram model even address the issue of tracking even under occlusion provided atleast half of the object is visible. When the person enters the scene after some time he is correctly identified. So this histogram approach helps in establishing motion correspondence as well as tracking under occlusion.

The algorithm of the identity tracker is given below:

Input: Blobs with bounding box from motion detection module

Output: Tracked blobs with its trajectory

1. Compute histogram of the blob.
2. Correlate with the histograms of all the models available.
3. Find the maximum value of the correlation results.
4. If maximum value is less than a threshold

   Add a new model to the available list with this histogram of the blob.

   Else mark the blob as belonging to the corresponding model having maximum correlation value.
5. Compute the centroid of the blob and keep track of the blobs' trajectory.
6. Get the next frames' blob from motion detector.

3.2.4 Activity Recogniser

This module recognises the activities done by the tracked objects from identity tracker module. This activity recogniser recognises the various activities done by the humans. In this thesis the activities tested are walking, running and idle.

The algorithm below shows the process flow of the activity recogniser module. In this module, the tracked objects are taken from identity tracker module and in these blobs contours are drawn and distance plot is drawn with distance along y-axis and boundary pixels along x-axis. The Centroid of the object is computed first and then distances are calculated from that centroid to each of the boundary pixels and a plot is drawn. This plot is taken as a 2 dimensional signal and parameters like RMS value, mean, median, zero crossings, energy are taken for classifying different activities like walking, running or idle.

The following algorithm gives the whole process flow of the activity recognition module.

Input: Tracked blobs

Output: Tracked Blobs with activities detected

1. Compute the contour of the blob & find boundary pixels.
2. Compute the centroid of the blob.
3. Calculate the distances of the boundary pixels from centroid.
4. Plot this signal as contour-distance signal with distance along y-axis and boundary pixels along x-axis.
5. Extract features from signal like rms, mean, median, zero crossings of the median line, number of peaks, etc.
6. Classify using NN for discriminating activities like walking and running.

3.2.5 Integrator

This module integrates all the modules and outputs the resulting video in different formats. This module is designed to work for showing output in the form of video result format. In the future it will be extended to support XML format written to a website or a mobile phone of a security personnel with a view to alarm any suspicion in the scene.

Input: Results from all modules

Output: Integrated result in video format

Integrator provides the system a unique way of formatting output in video format and it also going to take care of indexing the video or recording it for further offline processing whenever some suspicion will be found in the future. So this integrator is left unfinished for time being and it will be dealt in future with a view to expand this system further.

3.3 Discussion

Thus a real time surveillance system is tried out for achieving the above said objectives in an efficient manner. Lot of questions need to be answered further as no system is perfect in its sense. But this system has tried to address some of the issues successfully.

Chapter 4
MOTION DETECTION

4.1 Introduction

In this chapter individual module motion detector is discussed in detail. It talks about the pre-processing needed for the live video stream. This chapter discusses about how the foreground objects are neatly extracted from the existing background and how these extracted objects are smoothened for any noise present in the video and cleaned.

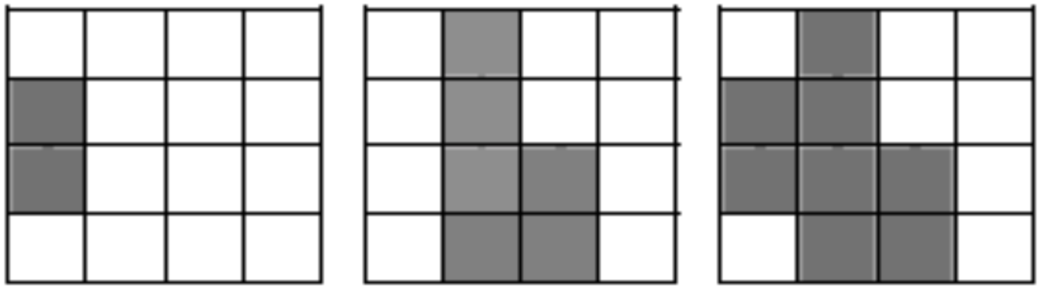
4.2 Concepts & Terminologies

4.2.1 Definition (Difference Image) A Difference Image is a monochromatic (grayscale) image. Its creation involves the pixel wise absolute difference between the Background Image and the Video Image.

4.2.2 Definition (Motion Image) A Motion Image is created by thresholding the Difference Image. Therefore the influence of the noise on the Motion Image is dependent on the noise amplitude.

4.3 Background Subtraction

Generally there are two approaches in image subtraction: (i) background subtraction as discussed in [2], [6], and [18] where a static background is kept as a reference frame and all the incoming live feed frames are subtracted; and (ii) temporal differencing as discussed in [16] and [27] where consecutive frames will be subtracted. Background subtraction is prone to sudden illumination changes, lighting conditions. Temporal differencing gives the perfect picture but it is computationally expensive compared to background subtraction. Motion image that resulted from image subtraction is depicted in figure 4.1. The shaded region shown in the figure illustrates the dynamic pixels.



(a) (b) (c)

FIGURE 4.1: Motion Image (a)*frameA* (b)*frameB* (c)Motion image between *frameA* and *frameB*

Combining these two commonly used methods give rise to another method called BSTD method as discussed in [32] that leads to less RMSE value than these common methods. The figure 4.2 gives the contrast between adaptive and conventional background subtraction methods.

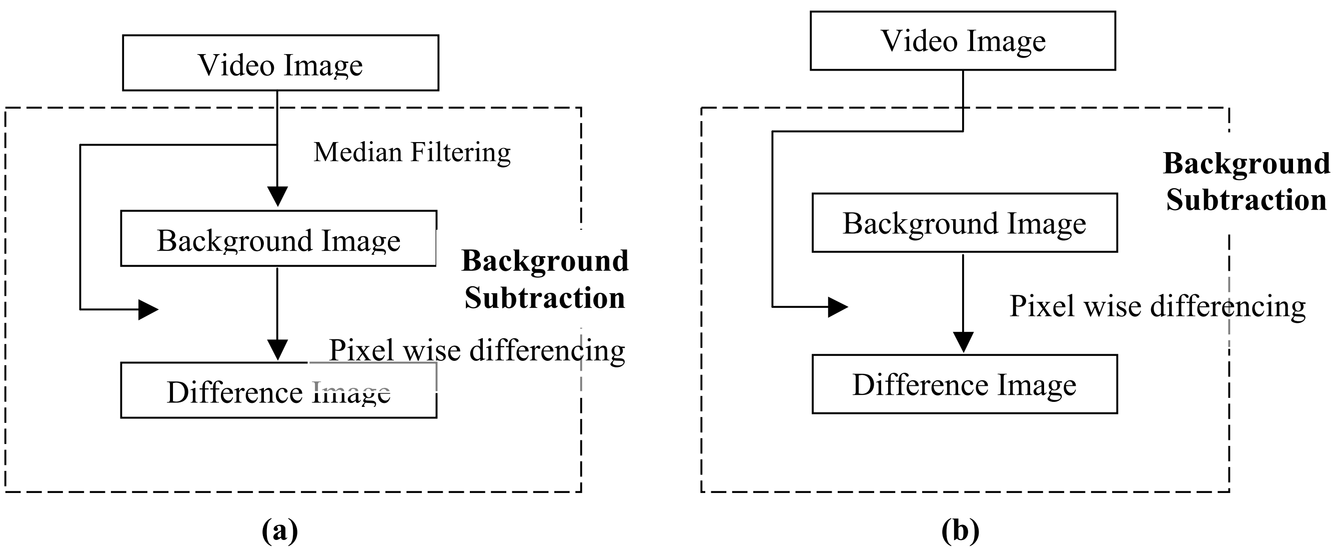


FIGURE 4.2: Background subtraction methods (a) Adaptive background subtraction (b) Conventional background subtraction

4.3.1 BSTD Method

Each gray value *A(x,y)* of *frameA* is subtracted from its corresponding gray value *B(x,y)* of *frameB* , where *w* and *h* is the frame width and height respectively.

$$\{\forall x\ \forall y : A(x,y) \mid x = 1,2,3,...w \text{ and } y = 1,2,3,...,h\}$$
$$\{\forall x\ \forall y : B(x,y) \mid x = 1,2,3,...w \text{ and } y = 1,2,3,...,h\} \qquad ....(4.1)$$

The difference value between two corresponding pixels *A(x,y)* and *B(x,y)*, is converted into absolute value and stored in difference matrix d^{AB} as illustrated in equation 4.2. This is to eliminate negative value after undergoing subtraction [15]. Motion mask $motion_{AB}$ between the two frames is obtained after applying thresholding on the difference matrix d^{AB} .

$$d^{AB} = |(A) - (B)|$$
$$motion_{AB}(x,y) = \begin{cases} 1, \text{if } d^{AB}(x,y) > T_d \\ 0, \text{otherwise} \end{cases} \qquad ....(4.2)$$

where T_d is the difference threshold value.

In order to obtain the motion mask of frame f^k, background subtraction and temporal differencing have been applied. Background subtraction is performed between frame f^k and background frame *B*. The result of this background subtraction is a difference matrix d^B . Temporal differencing is performed between frame f^k and f^{k-1}, and frame f^k and f^{k+1}. The outputs are two difference matrixes and they are referred as d^{k-1} and d^{k+}. Then thresholding with difference threshold value T_d is performed on the difference matrixes; d^{k-1}, d^B and d^{k+1}.

$$d^{k'} = |f^k - f^{k'}|$$
$$where\ K' = k-1\ and\ k+1$$
$$d^B = |f^k - B| \qquad ....(4.3)$$
$$where\ K' = k-1, k\ and\ k+1$$

The process is followed by applying *AND* operator between d^B and d^{k-1}, and d^B and d^{k+1}. The output of the *AND* operations are two motion masks: $motion_{k-1}$ and $motion_{k+1}$. Lastly, motion mask of frame f^k is obtained by applying *OR* operator between these two motion masks and the output is named as $motion_k$. The process flow is illustrated in figure 4.2.

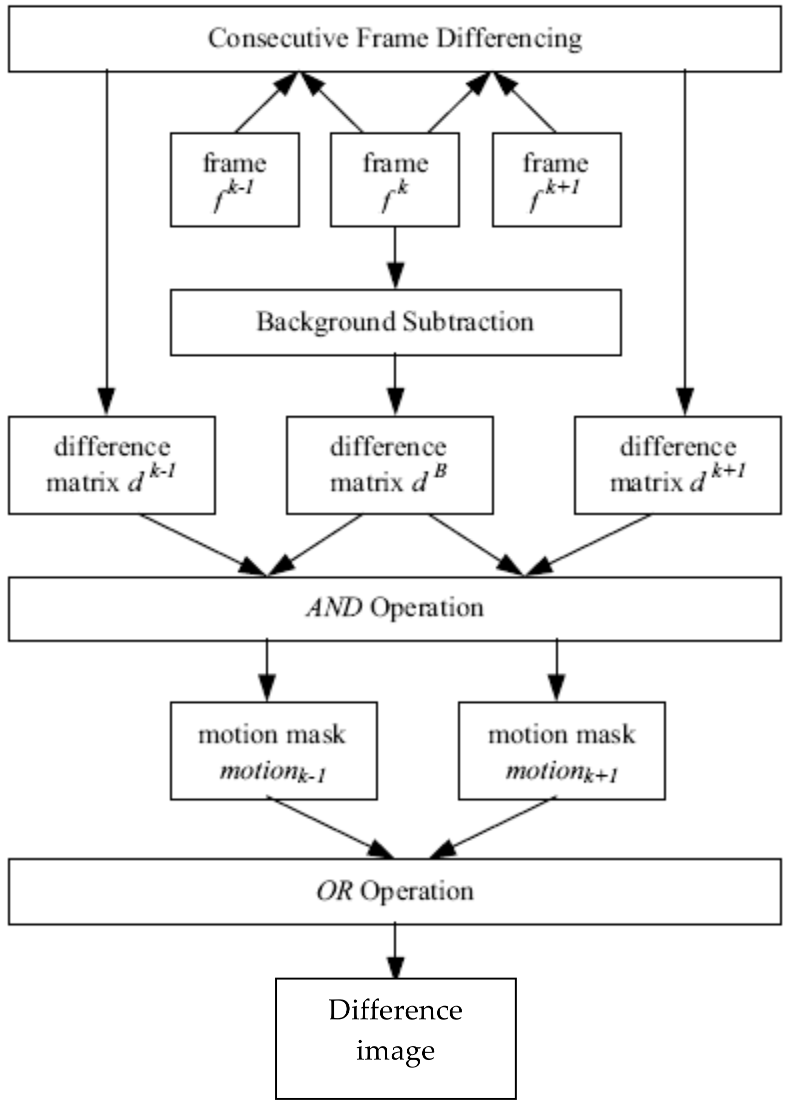


FIGURE 4.2: Process flow of the combined BSTD method.

4.4 Cleaning up of blobs

After the difference image is extracted, the blobs extracted will be noisy. Its advisable to remove those noises present due to both natural artifacts like flag waving in air, trees swaying in wind, illumination changes in scene, etc., and also due to video camera artifacts like less contrast ratio and wrong calibration in the camera.

The best possible option of removing noise from the image is by passing under a median filter that removes the noises present. Median filter is a non-linear filter and it is best used for removal of salt and pepper noise. In this case, it's better to go for Gaussian filter for smoothing the image.

After the smoothing and removal of noise, dilation and erosion are done on the blobs resulting in individual blobs. Some parts of the body may be separated while applying these above said operations on the image. In such cases, they will be regarded as separate blobs; this can be addressed using the proximity decision-making technique that can be illustrated in the following example.

Consider a situation where the whole blob of an individual and his two feet are separated from the body as shown in figure 4.3. Usually this will be treated as three different blobs, but this will result in wrong calculations later. So using proximity technique, this can be reduced to one region (collection of one or more blobs).

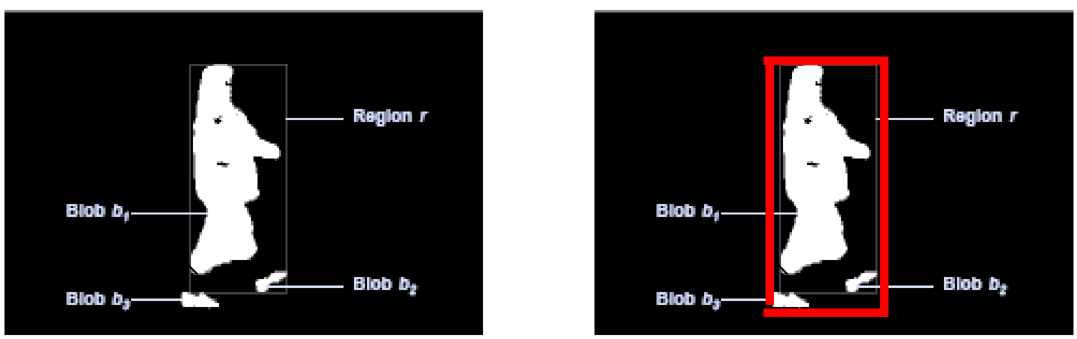


FIGURE 4.3: Proximity analysis technique. (a) Human object with three blobs separated (b) Three blobs joined to form a region comprising one or more blobs.

The algorithm is given as follows:

1. Draw the bounding box of the blobs and store the sizes of each box.
2. Check whether any box of size so small than the present box size exists in a span of 100 pixels across the border of the box.
3. If it exists, add the small blob to the current blob and redraw the bounding box with the small blob included.

The number given, say 100 pixels, is by intuition only. Again there is a wide scope for research in finding a suitable threshold globally by estimating the current scenario.

4.5 Issues encountered

The issues in the motion detection are as follows:

Illumination changes, Flag & trees' waving in wind – An adaptive background subtraction is tried out.

Person wearing the same colour of clothing as that of background – Yet a good solution has to be provided.

Parts of the body separated due to dilation & erosion operations – Proximity analysis is tried out to address this issue.

Two persons occluding each other – Pixel classifier based on histogram color distance is proposed and has to be implemented.

4.6 Experimentation & Analysis

In this module of detecting blobs, the three methods of background subtraction are tried out such as Conventional background subtraction (BS) method [2][6][16][18][27][32], Temporal differencing (TD) [16][27] and the combined BS&TD method [32]. The results of all the methods are shown in this section. Figure 4.4 shows the results of the motion detection module using all the methods. The test sequences used are the videos taken with a web camera that is of not good quality and has got more distortion and another video from a shopping mall in Portugal used for CAVIAR project [31] by Robert Fischer.

Figure 4.5 shows the results using the conventional background subtraction method at a shopping mall in Portugal. The figure shows the blobs detected in the video sequence. As there is a need for the full silhouette of the object for further processing in the identity tracker module, the method that gives the full blob is used in this stage.

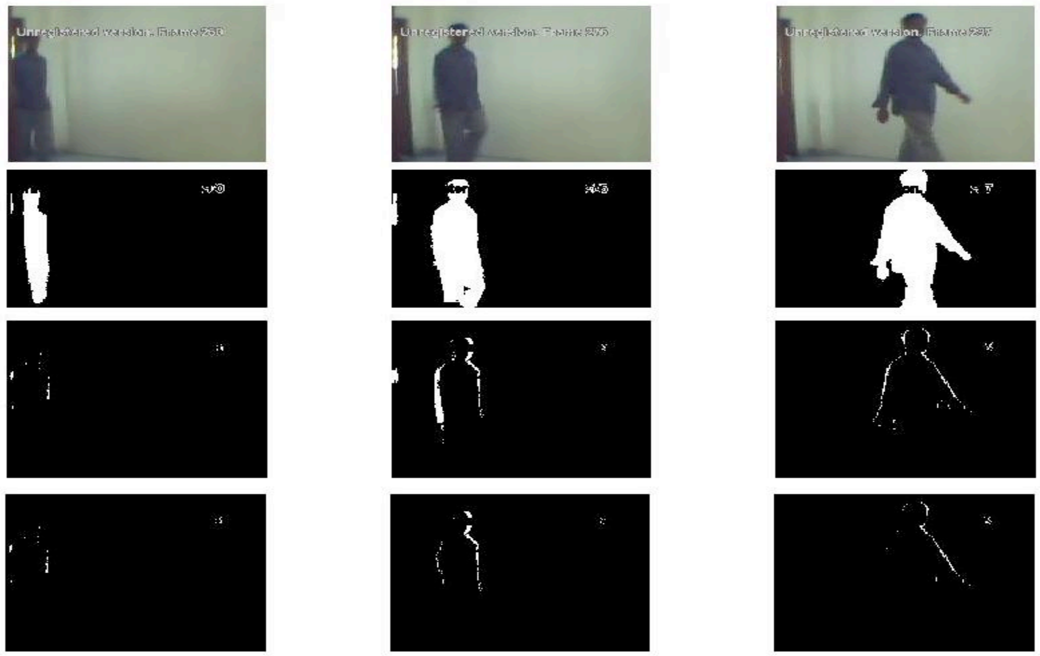


FIGURE 4.4: Blob detection using background subtraction. The first row gives the video sequence used for testing the algorithm with three frames taken in random. The second row shows the blobs detected using the conventional background subtraction method. Third row gives the results for temporal differencing method and the last row has the results using combined BS&TD method.

FIGURE 4.5: Blob detection in shopping mall sequence using conventional BS method

Even though the temporal differencing and the combined BS&TD method give satisfactory results; they cannot be used for further processing. Conventional background subtraction gives very good results in indoor environment with controlled lighting. But it will fail miserably when it is operating outdoors due to illumination changes, objects' movements like flags, trees waving behind the scene.

4.7 Discussion

This section summarizes all the discussions that have already presented before. The blob is detected successfully and certain problems like illumination changes and noise removal are solved partially for the purpose of tracking. The problem of who is present in the scene is solved in the next chapter with a color histogram model. This module works fine for almost every case under ideal lighting conditions inside an indoor environment. This module needs to be tested at outdoors.

Chapter 5
TRACKING

5.1 Introduction

This chapter discusses the tracking of the objects in the scene. It includes the topics of how the objects are tracked and what is the method employed in doing that and how effective is it to other methods.

5.2 Related Work

Tracking is one of the most important research fields for computer vision researchers. There have been a quite a number of contributions in the tracking domain. For the purpose of tracking, several papers have been explored and these are one of the few to name. The works done by numerous researchers [3] [12] [13] [15] [22] [24] [30] are explored. Tracking people in a cluttered scene is proposed by Bremond & Thonnat [12]. Ismail, Harwood & Davis [15] proposed a system for real-time tracking. Nils [21] [22] [23] [24] studied the pedestrians in traffic crossings and a real-time tracking system has been developed.

5.3 Tracking

Tracking refers to the process of keeping track of the objects in scene i.e., establishing the correspondence of the blobs in successive frames. The tracking here is done by the color histogram model [5].

5.3.1 Histogram Model

Histogram. The color histogram *h* of image *I* is defined for I€[m] such that $h_I(c_i)$ gives for any pixel in I , the probability that the color of the pixel is c_i . Given the count

$$H_I(c_i) \equiv \left|\left\{p \in I_{c_i}\right\}\right| \qquad \text{………(5.1)}$$

$$h_I(c_i) = \frac{H_I(c_i)}{|I|} \qquad \text{.........(5.2)}$$

Histogram refers to the distribution of the colours in the blob i.e., how many times does a colour appear in the scene. After the motion detector detects the motion and extracts the foreground and turns into a blob, the frequency of colour distribution of the blob i.e., its histogram is computed and stored into a model for further analysis.

In this method, the histogram model method is affected by some parameters such as the number of bins taken and the method of similarity check used. A true image with 24 bit RGB has got 16 million colours in it with red channel accounting for 256 variations similarly green and blue each varying from 0 to 255. So totally 256*256*256 equalling 16777216 colours which is 16 million approximately. The human eye cannot distinguish all these 16 million colours. This is an advantage for choosing the number of bins. It is practically impossible to take all 16 million colours as bins and checking for similarity. So the reduction of bins is required for effective computation. The following formula reduces the number of bins.

$$\text{Histogram}_{\text{index}} = (\text{red} / 32) + 8 * (\text{green} / 32) + 64 * (\text{blue} / 32) \qquad \text{.......(5.3)}$$

where the $\text{histogram}_{\text{index}}$ is the index value of the histogram bin. It gives the colour of the pixel.

Illustration of grouping of colours: An example where a white pixel and a black pixel is taken. As the white pixel has got all red, green and blue components are all 255, so the value of the histogram index becomes this,

$$\text{Histogram}_{\text{index}} = (255 / 32) + 8 * (255 / 32) + 64 * (255 / 32) \qquad \text{equals 511.}$$

When a black pixel is taken, its red, green and blue are all 0, so the histogram index value is 0. So the value varies from 0 to 511 totalling 512 bins. This implies that in red, green and blue components, say 32 combinations of colours are grouped into one colour. 32*32*32 colours are grouped as one colour and assigned a bin in the histogram. So the 16 million colours have been reduced to just 512 colours.

5.3.2 Working methodology

The objective of the human tracking module is to label and track each individual consistently throughout the video sequence. The first problem to be solved when tracking people is detecting them. For detection of humans, as explained in chapter 4 a background subtraction method [32] is used. Although the idea behind background subtraction is simple, the problem is very challenging. Changes in illumination conditions, movements of tree branches waving in the wind, camera noise, shadows or reflections can all affect the detection performance. After this process, the system expects each isolated person to be segmented into an isolated blob. Thus it assumes that in an indoor environment each detected blob will correspond to a person. For instance, if there is a part of the body that has been segmented into a different blob after the background subtraction stage, and this blob has not been removed during the thresholding process, the system will fail and detect that blob corresponding to a part of the body as a new person. It is also assumed that when a person enters the scene he/she will be isolated. If two or more people enter together and are segmented into the same blob by the background subtraction algorithm, the system will treat them as though they were a single person until they split. At this point the system will begin to keep track of each individual separately, assigning the label of the group to one of the components of the group and new labels for the rest. There would be the possibility for the system to go back in time and segment the people while they were together, but this was not implemented maybe in future it will be tried out. As soon as a person enters the scene, a model for that person is generated and stored. A model consists of histogram information of how the colours are distributed in the image. In the next frame, another model is built for each of the foreground blobs that are detected by the motion detector module. Then the similarity measure between all the histograms of the stored models and all the blobs in the current frame are calculated using one of the distance measures given below.

$$\sum_{i}^{n}(x_i-\mu_x)(y_i-\mu_y) \qquad \ldots\ldots\ldots(5.4)$$

$$\sum_{i}^{n} abs(x_i - y_i) \qquad \ldots\ldots\ldots(5.5)$$

$$\sum_{i}^{n} (x_i - y_i)^2 \qquad \ldots\ldots\ldots(5.6)$$

where n is the required number of bins

Using one of these distance measures given in equations (5.4), (5.5)&(5.6) the similarity check is done and the most similar blobs are matched as long as the similarity measure is above a certain threshold. However, if a blob in the current frame is at a significant distance from all the stored models, a new model is initialized. The fixing of threshold in any operation in images is still a mystery to the researchers. A local threshold and a global threshold can be fixed to fix this problem but the only way to address this issue is by experimentally fixing the threshold. Once the matching stage is completed, all the models are updated using this equation below.

$$h_I(c_i, t) = \alpha h_I(c_i, t-1) + (1-\alpha) h_I^{new}(c_i, t) \qquad \ldots\ldots(5.7)$$

Once the matching and updating of the models are done the next frame is processed and the process continues till the system is stopped.

5.4 Issues encountered

The issues encountered in this module are as follows:

Fixing of thresholds in the similarity check with the histograms of the current blob to that of the available models.

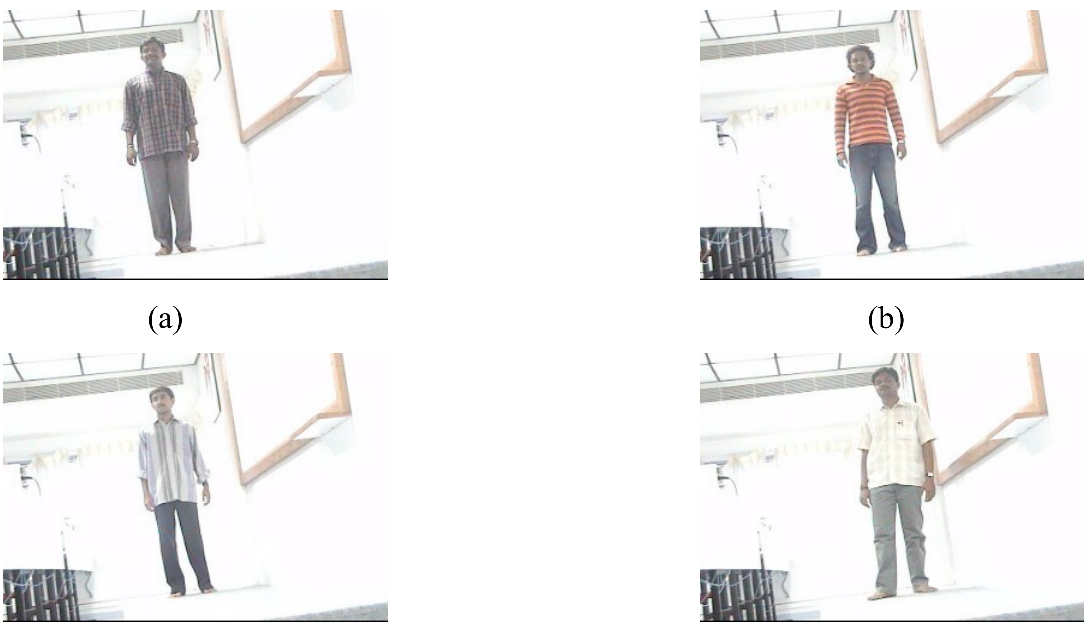
Persons occluding each other in the scene cannot be distinguished, as only one whole blob will be obtained from the motion detection module.

Persons wearing same colour of clothing cannot be discriminated efficiently. So looking for an algorithm that takes spatial information along with the colour information.

Entry and exit of persons in the scene are solved by establishing a criterion where only the blob is take into consideration only when it is fully visible in the scene.

5.5 Experimentation & Analysis

The experiments were done on the system configured with 2.4GHz Athlon uni-processor with 512MB RAM and programmed with C using Intel's open source image library OpenCV. The test sequences consist of four persons with different clothing at an indoor environment. The color histogram model correctly identifies the people as different persons entering the scene. Various histogram correlation distance measures been tested with these sequences and the results are attached with this section and the effect of the histogram bin sizes are also discussed. Basically, more the number of bins, the sparser the array and it will provide wide range of diversity in colour. But it will increase the computation. So there is a trade-off between the computational complexity and the accuracy of the colour dispersion into bins.



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

FIGURE 5.1: Four persons used in test sequences (a) Arjun (b) Subash (c) Vasanth (d) Jayaram

The test sequence consists of 1800 frames that have four people entering and leaving the scene in random fashion. The figure 5.1 shows the four persons used in the test sequences. The system needs to correctly identify the person and label him as the correct person who has already entered the scene or is new to the scene. The four persons are vasanth, arjun, jayaram and subash. The system needs to label them as what they are. From the experiments conducted it is observed that the persons are correctly identified and the false positives are really low. Only 5% false positives are present with these four persons. The following table gives the idea of how each person appearance is closer to each other. The numbers in the cells indicate the level of closeness i.e., similarity. Higher the number in the cell, higher is the similarity. The values of the cells in the table are obtained by correlating the histograms of the blob with that of available model.

Blob / Model	Subash	Vasanth	Arjun	Jayaram
Subash	**13.25**	9.69	6.2	3.38
Vasanth	8.8	**12.08**	5.12	2.78
Arjun	5.88	4.95	**11.99**	5.8
Jayaram	3.57	2.88	5.86	**11.58**

TABLE 5.1: Confusion matrix of the Correlation sum of the histograms

Models	**Colour of the model**
1	Red
2	Black
3	White
4	Green
5	Yellow
6	Cyan
7	Magenta

TABLE 5.2: Lookup table for colouring of the models

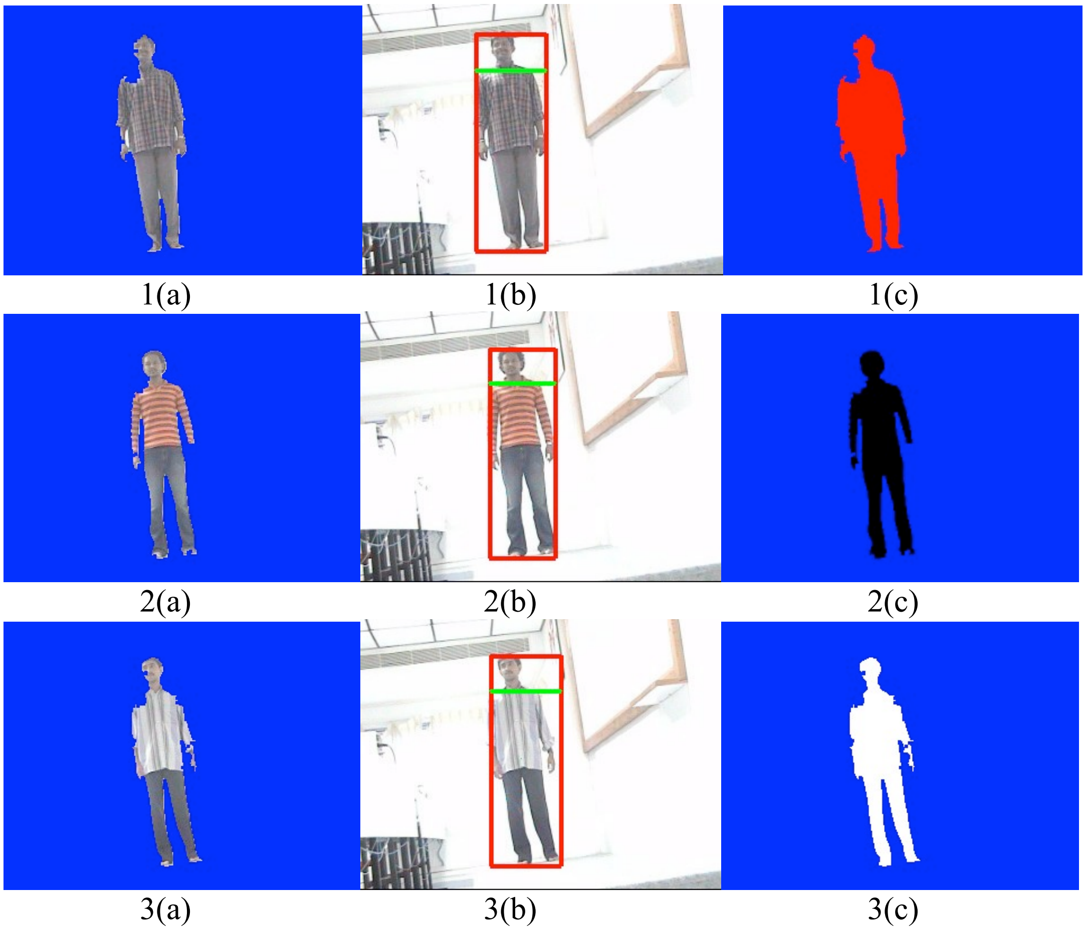


FIGURE 5.2: Results of the identity tracker. The first column gives the colour blob detected from the motion detection module. The second column gives the bounding box enclosing the blob in red and the head tracker with a green horizontal line. The third column shows the output of the identity tracker where each person is different from each other. The different colours used are from the lookup table in Table 5.2 used for colouring the blob and it is mainly used for visual appeal and ease of understanding.

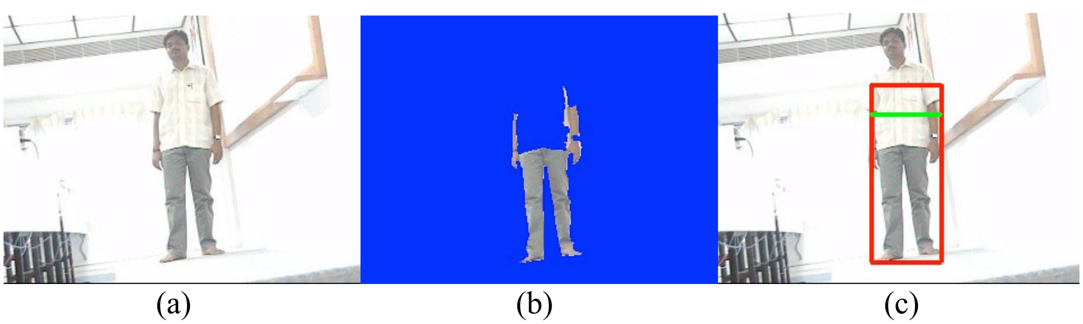


FIGURE 5.3: Error in identity detection due to the effect of the background. (a) Model (b) Blob detected due to the effect of background as he is wearing a shirt with the same colour as that of background. (c) Consequences of wrong blob detection in the bounding box stage with head tracker (green line) showing his chest as the head.

From the table 5.1, it is also observed that the maximum similarity is obtained when histogram is self-correlated with itself. The figure 5.2 shows the tracked results in the test sequence using the color histogram model. The different colours used in the last column of the figure shows the different identities of the person. The following lookup table is made for each model stored in the knowledge base.

When the first person enters the scene he will be coloured in red and whenever he re-appears in the scene, he will be coloured the same colour. And the next new person is identified and coloured as black and so on. To illustrate, only seven colours of the models are shown in the table 5.2.

There are very few scenarios where the algorithm fails miserably in identifying the right person. Figure 5.3 shows this scenario. This is an effect of the accuracy of the blob detected in the motion detector module. When the blob is detected in a crude sense, the results of this algorithm also become miserable. This shows the importance of the pre-processing stage.

5.6 Discussion

This section summarizes the details and the results obtained in the identity tracker module. Various issues were encountered during this module and some of the issues were addressed fully and some partially. Thresholding of the similarity check is one of the main problems in this module while correlation is done. An empirical way of fixing the thresholds is tried out and it is known that it is not the best way. The entry and the exit of persons are taken care of by assuming the blob needs to be processed only when it is away from the image boundary fully. In spite of all these issues, this module works satisfactorily when people comes with different clothing. When feature like shape of the contour is taken along with the color information, the system will improve in efficiency and it discriminates the people wearing same dress.

Chapter 6

ACTIVITY RECOGNITION

6.1 Introduction

This chapter details the module of the work of activity recognition of the tracked objects in the system. It discusses the details of how the blobs' contour is drawn, how the centroid is computed and the contour distance plot is drawn to arrive at the activity done by the object. It provides notes about the star skeleton method already being used in this domain.

6.2 Related Work

Activity Recognition has gained much attention among computer vision researchers in the recent years. There have been few contributions to this field already from great researchers around the world. Recurrent motion image of Omar & Mubarak [23] deals with the movements of the objects by estimating the cylinders around the body and recognising them. Another paper on image skeletonisation by Kanade, Lipton [14] emphasizes on the skeletonisation of the human silhouettes and taking out cyclic cues from them to identify various activities. Human estimation concepts given by Ebihara, Ohya & Yamada [18] involves the estimation of the human joints, head, center of gravity and then making out the activity of the human.

6.3 Activity Recognition

The activity recognition is first tried out using the star skeleton model [14] and also with the contour-distance signal method. The objects' activities are tried with the features like contour shape; and the contour-distance plot method.

6.3.1 Star Skeleton Model

An important cue in determining the internal motion of a moving target is the change in its boundary shape over time and a good way to quantify this is to use skeletonisation. There are many standard techniques for skeletonisation such as thinning and distance transformations. However, these techniques are computationally expensive and moreover, are highly susceptible to noise in the target boundary. This method provides a simple, real-time, robust way of detecting extremal points on the boundary of the target to produce a “star” skeleton. The star skeleton consists of only the extremities of the object joined to its centroid in a “star” fashion.

The algorithm of deriving the “star” skeleton is as follows:

Input: Silhouette of humans i.e., blobs

Output: Star skeletons (sticks) from centroid

1. Detect the edges in the silhouette and obtain a closed contour of the object.
2. Find the coordinates of all boundary pixels.
3. Find the centroid of the object.
4. Compute the distances of all boundary pixels to the centroid.
5. Plot the values of distance along y-axis and boundary pixels along x-axis and find out the local maxima.
6. Plot lines to these local maxima pixels from centroid and these are the sticks of the object.

For finding Centroid of the object the following formula is used.

$$x_c = \frac{1}{N_b}\sum_{i=1}^{N_b} x_i \quad \text{……….} (6.1)$$

$$y_c = \frac{1}{N_b}\sum_{i=1}^{N_b} y_i \quad \text{………..} (6.2)$$

N_b __ Number of boundary pixels

(x_i, y_i) __ Pixel on boundary

(x_c, y_c) __ Centroid pixel

But it only gives the approximation of the centroid and not the exact one. If a hand is stretched in one direction then the centroid too shifts its position considerably, which is not efficient.

To find the local maxima, traverse through the distance array and search for this condition $I > I - 1$ and $I > I + 1$, where I is the index of the array.
Store the position in zerocrossings matrix. Get the exact coordinates from the boundary pixels matrix. Plot lines from centroid to boundary pixels of boundary matrix yielding sticks. Thus the "star" skeleton is achieved.

Once the star skeleton model is computed for an object for a single frame, it is taken for consecutive frames and viewed for cyclic motion analysis [14] where certain periodicity exists. Activities like walking and the running are distinguished with this cyclic motion.

6.3.2 Contour-distance signal Model

The contour-distance signal model is a slight modification to the "star" skeleton approach as discussed in section 3.2.4. In this approach, instead of drawing the star skeleton the contour distance plot is drawn. This is the variant in this model to the star skeleton model. After contour is drawn and the centroid is calculated using the equations (6.1) & (6.2), the distances are computed from the centroid to each of the boundary pixels. Then a plot is drawn with distances along y-axis and the boundary pixels along x-axis. Consider this as a one-dimensional signal and features are extracted from this. Features like the RMS value of the signal, mean, median, zero crossings, with respect to median line, the energy of the signal, no. of boundary pixels, no. of peaks, maximum peak-to-peak distance and minimum peak-to-peak distance, etc.,. These features are extracted with a view to discriminate between walking and running by feeding it to a classifier after training.

6.4 Issues encountered

Skeletonisation leads to more number of branches in the skeleton of the silhouette. This has to be pruned effectively for a single pixel width. This is solved with another method called star skeletonisation [14].

Presence of spurious sticks in the star skeletonisation method is due to the ruggedness of the signal. A solution of smoothing the signal is presented to remove those large numbers of local maxima present.

Satisfactory discrimination between activities is not achieved as expected.

6.4 Experimentation & Analysis

In this activity recogniser module, the tests sequences were taken from the ones used in the motion detection module and this prototype is tested using Matlab on a 512MB RAM, 2.4GHz Athlon uni-processor powered system. The results of this module are shown in this section. Figure 6.1 shows the star skeleton method on the test video.

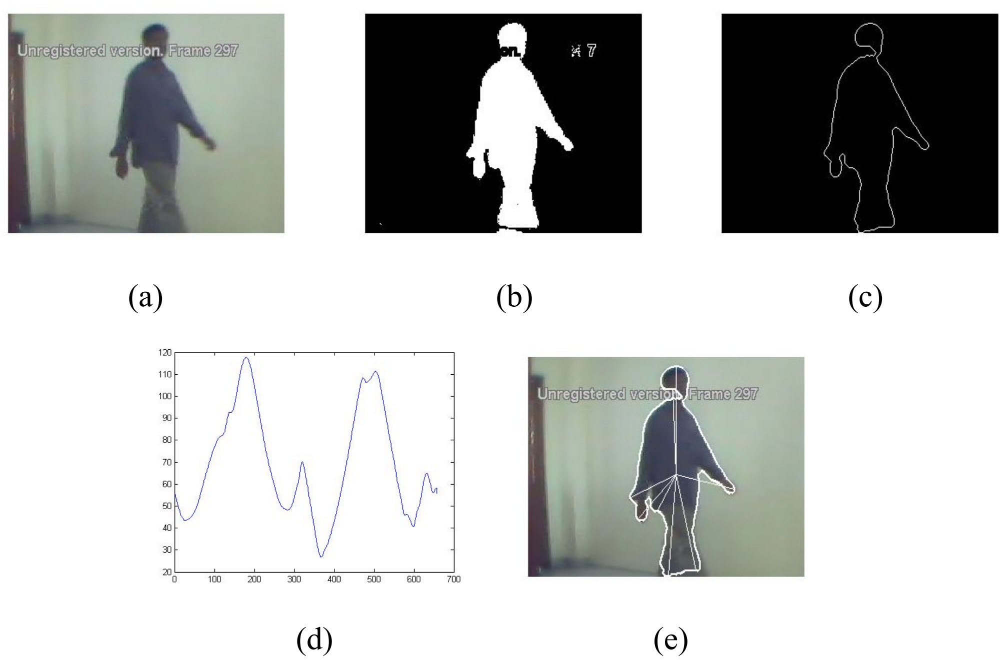
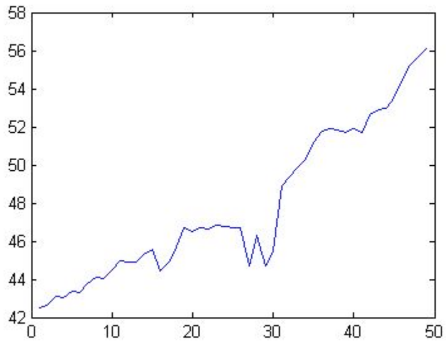
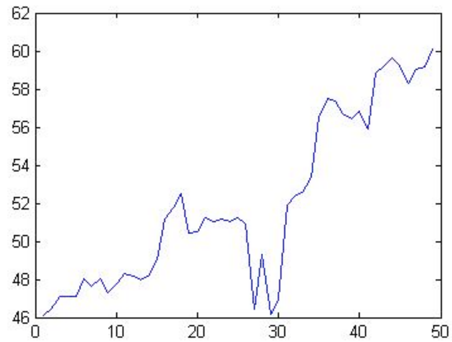


FIGURE 6.1: Star skeleton method. (a) Test video (b) Blob detected from motion detection module (c) Contour of the blob (d) Contour distance plot showing the distances of the boundary pixels to the centroid (e) Star Skeleton with the sticks drawn from centroid to the local maxima in the distance plot.

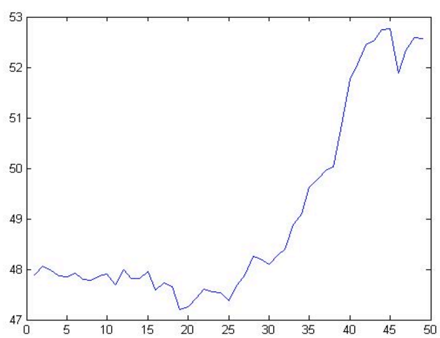
The star skeleton model gives the sticks from centroid to the local maxima in the distance plot. The sticks got by this method are taken for few seconds and checked for periodicity in the cyclic motion. The cyclic motion distinguishes the activities like walking and running. In addition to that, the motion of the arms while running and walking also differs. The issue here is the presence of noise in the signal and the ruggedness of the signal.



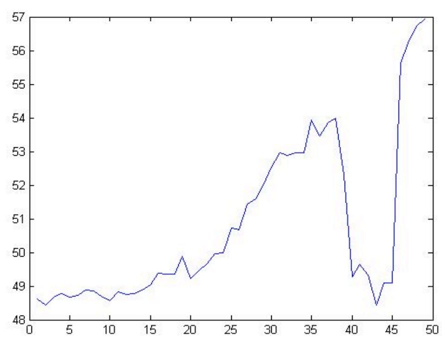
1(a)



1(b)

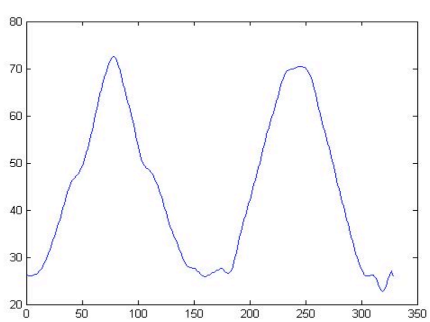


2(a)

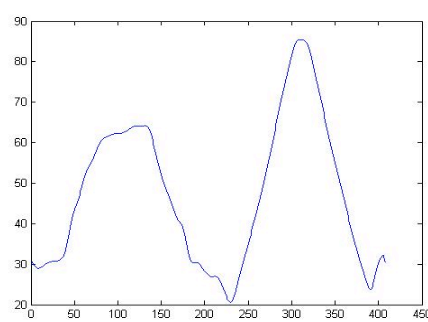


2(b)

FIGURE 6.2: Features from the two test sequences. 1(a) Mean of 50 samples for a test sequence with a person without an umbrella 1(b) Median of the same test sequence 2(a) Mean of 50 samples of the test sequence involving a person with an umbrella 2(b) Median of the same.



(a)



(b)

FIGURE 6.3: Contour-distance signal method (a) Man without umbrella (b)Man with an umbrella

So before finding the local maxima of the signal, it is mandatory to smooth the signal to reduce the sticks. Even after smoothing, the noise still exists. So the number of sticks in this method depends on the smoothness of the signal. If the blob is detected in a slight crude fashion, this effect will eventually affect the number of sticks obtained. This paved the way for finding a new solution to the problem where discrimination is prominent. Instead of smoothing the signal, the signal is taken as it is and features are extracted. This led to the contour-distance signal method. Figure 6.2 shows the results of the contour distance signal method for 50 frames of a man walking with & without an umbrella. Figure 6.3 shows the single frame contour-distance signal plot of a man with & without an umbrella.

Classification done on the features

For the discrimination stage in the activity recognition, the test sequences taken are the man walking with & without an umbrella, persons doing simple activities like walking, running and standing idle and browsing. Features like rms, mean, median and zero crossings are taken from the contour-distance signal and fed to the neural network for classification. To illustrate the procedure, the following example is taken. Consider the scenario of identifying either the person is walking or running, a two class problem. The total number of features used for classification is 4. To identify an activity, collections of frames' data need to be taken. It's not enough if data of only one frame is available. So 50 frames data that lasts almost to 2 seconds in real time is considered. This is one sample for a particular activity say walking. Similarly, a sample for running exists. Some m number of sample videos is taken. In those samples, m/2 samples are used for training & the other half for testing. Once the training is finished, the rest of the samples are used for testing the classification rate. Thus the procedure for the classification using neural networks is done.

Parameters like rms, mean, median, zero crossings are extracted with a view to distinguish various activities. But after analysis with the results, only the direction of the moving object can be identified. Only 47% classification rate is obtained. Inferences from the results are given below:

(i) Objects near the camera and away from the camera can be identified with the number of boundary pixels available in the contour. If the object is close to the camera, it will appear larger as compared to the object far away to the camera.

(ii) The signal is the same for the object walking near the camera and the other object doing another activity far away from the camera.

(iii) Direction of the object is detected from the way the median and the mean is growing or shrinking. If the curve is increasing, then the object is coming towards the camera and vice versa.

6.5 Discussion

The summary of the activity recogniser module is given in this section. The activity recogniser module does not provide satisfactory results. It does not discriminate between the activities like walking and running. The only inference obtained is the direction in which the object moves.

Chapter 7
FUTURE RESEARCH & CONCLUSION

7.1 Introduction

This chapter discusses about the results obtained, inferences made and the future work in this system and enhancement of this system with the conclusion about the whole thesis.

7.2 Future research

There is a strong and penchant desire for research in this domain still left. There are so many issues untouched by many researchers still. This system as a whole needs to be improved quite a lot in terms of the user-friendly interface, the methods used in order to tackle the issues present. Starting from the motion detection module, there have been quite a number of issues to be resolved.

The problem of sudden illumination changes poses a serious threat still after exploring this field for years. It is very important to develop a very good algorithm that can nullify the illumination and remove the noises present in the environment without compromising the quality of the blob detected.

Another issue of the system is that the fixing of thresholds. In future, a nonparametric model needs to be implemented where parameters are no longer required.

Occlusion is the main problem in tracking. While tracking a single individual in a crowd involves lots of occlusion with each other. This has to be taken of by the histogram distance model proposed but for time being it is not implemented.

One important issue to the identity tracker module is the scenario where persons wearing same colour of clothing enters the scene. The algorithm cannot distinguish the two persons. So a means of taking spatial relation and the shape of the object along with colour information is required for the above said problem.

In activity recognition module, the contour-distance signal method does not provide satisfactory results. If it is combined with the features like contour of the object, it will discriminate well.

7.3 Conclusion

As to conclude, the objectives laid down for the thesis is partially solved and all the modules are working fine except for the activity recognition discriminator. Certain issues have been addressed and others are left untouched for want of time. There is a strong penchant for extending this work and taking it into the next level.

BIBLIOGRPAHY

[1] Adam M Baumberg, “Learning Deformable Models for Tracking Human Motion”, PhD thesis, School of Computer Studies, University of Leeds, Leeds, UK, October 1995.

[2] Ahmed Elgammal, David Harwood and Larry Davis, “Non-parametric model for background subtraction” in 6th European Conference on Computer Vision (ECCV 2000), Dublin, Ireland, pages 751–767. Springer Verlag, 2000.

[3] Alan J Lipton, Hironobu Fujiyoshi and Raju S Patil, “Moving target classification and tracking from real-time video” in Proceedings of the DARPA Image Understanding Workshop (IUW’98), Monterey, USA, pages 129–136, November 1998.

[4] Alan Moore and Tony Backwith. Development tools for realtime systems. Real-Time Magazine, 7(1):33–37, Quarter 1 1999.
http://www.realtime-info.com/magazine/99q1/1999q1_p033.pdf.

[5] Balcells-Capellades .M, DeMenthon .Dand Doermann .D, “An appearance based approach for consistent labeling of humans and objects in video” in Pattern Analysis and Applications, pages 1433-7541, November 2004.

[6] Chris Stauffer and Eric L Grimson, “Adaptive background mixture models for real-time tracking” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’99), USA, volume 2, pages 246–252, 1999.

[7] Dariu M Gavrila and Larry S Davis, “Tracking of humans in action: A 3-D modelbased approach” in ARPA Image Understanding Workshop, Palm Springs, USA, pages 737–746, February 1996.

[8] Dariu M. Gavrila, “Vision-based 3D Tracking of Humans in Action”, Ph.D thesis, Department of Computer Science, University of Maryland, College Park, USA, 1996.

[9] David A Forsyth and Jean Ponce. Computer Vision: A Modern Approach. Prentice Hall, New Jersey, USA, 2003.

[10] David Hogg, “Model-based vision: A program to see a walking person” in Image and Vision Computing, 1(1):5–20, February 1983.

[11] EventHelix.com Inc. Issues in real-time system design. Article in an online collection, 2000–2002.
http://www.eventhelix.com/RealtimeMantra/IssuesInRealtimeSystemDesign.htm.

[12] Fran¸cois Br´emond and Monique Thonnat, "Tracking multiple non-rigid objects in a cluttered scene" in Proceedings of the 10th Scandinavian Conference on Image Analysis (SCIA '97), Lappeenranta, Finland, volume 2, pages 643–650, June 1997.

[13] Hedvig Sidenbladh, Michael J Black and David J Fleet, "Stochastic tracking of 3D human figures using 2D image motion" in David Vernon, editor, 6th European Conference on Computer Vision (ECCV 2000), Dublin, Ireland, pages 702–718. Springer Verlag, 2000.

[14] Hirupouba Fujiyoshi, Alan J. Lipton & Tackco Kanade, "Real time human motion analysis by image skeletonisation ", in IEICE Transactions & Information Systems , January 2004.

[15] Ismail Haritaoglu, David Harwood and Larry S Davis, "W4: Real-time surveillance of people and their actions" in IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):809–830, August 2000.

[16] Koay S.Y, Ramli A.R, Lew Y.P, Prakash .Vand Ali.R, "A Motion Region Estimation Technique for Web Camera Application", *Student Conference on Research and Development Proceedings*, pp. 352-355, Shah Alam Malaysia, 2002.

[17] LIU Ya, AI Haizho, XU Guangyou, "Moving Object Detection and Tracking Based on Background Subtraction", 2001.

[18] Masanori Yamaday, Kazuyuki Ebihara, and Jun Ohya , "A New Robust Real-time Method for Extracting Human Silhouettes from Color Images" in Proceedings of the Third International Conference on Automatic Face and Gesture Recognition (FG'98), April 1998.

[19] Micahlis Vazirgiannis & Ouri Wolfson ,"Spatio-temporal Model & Language for moving objects on road networks" in Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases, pages 20--35, 2001.

[20] Neil Johnson, "Learning Object Behaviour Models", PhD thesis, School of Computer Studies, University of Leeds, Leeds, UK, September 1998.

[21] Nicolas Chleq, Fran¸cois Br´emond and Monique Thonnat., "Image understanding for prevention of vandalism in metro stations" in Advanced Video-based Surveillance Systems, volume 488, pages 106–116, USA, November 1998.

[22] Nils T Siebel and Steve Maybank, "Real-time tracking of pedestrians and vehicles" in Proceedings of the 2nd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS'2001), Kauai, USA, December 2001.

[23] Nils T Siebel and Steve Maybank, "The application of colour filtering to real-time person tracking" in Proceedings of the 2nd European Workshop on Advanced Video Based Surveillance Systems, UK, pages 227– 234, September 2001.

[24] Nils T. Siebel, "Design and Implementation of People Tracking Algorithms for Visual Surveillance Applications", Ph.D thesis, University of Reading, March 2003.

[25] Omar Javed & Mubarak Shah, "Tracking And Object Classification For Automated Surveillance" in Proceedings of the 7th European Conference on Computer Vision, Part IV, pages: 343 – 357,2002.

[26] Paolo Remagnino, Adam M Baumberg, Tom Grove, Tieniu Tan, David Hogg, Keith Baker and Anthony Worrall. "An integrated traffic and pedestrian model-based vision system" in Proceedings of the Eighth British Machine Vision Conference (BMVC97), pages 380–389. BMVA Press, 1997.

[27] Paul L. Rosin and Tim Ellis, "Image Difference Threshold Strategies and Shadows Detection", 1995.

[28] Pons .J, Prades-Nebot .P, Albiol .Aand Molina .J, "Fast Motion Detection in Compressed Domain for Video Surveillance", *IEE 2002 Electronics Letters*, Vol.38 No. 29, pp. 409-411, April 2002.

[29] Prabhakar .B and Damodar V.Kadaba, "Automatic Detection and Matching of Moving Objects", *CRL Technical Journal*, Vo.3 No.3, pp.32-37, Dec 2001.

[30] Qin Cai, Amar Mitiche and J K Aggarwal, "Tracking human motion in an indoor environment" in Proceedings of the 2nd International Conference on Image Processing (ICIP'95), pages 215–218, 1995.
[31] Robert Fischer, "CAVIAR – Context Aware Vision using Image-based Active Recognition ", July 11, 2004.
[32] Shabhe Mat Desa & Qussay A. Salih, "Image Subtraction in real time moving object extraction", in Proceedings of the International Conference on Computer Graphics, Imaging and Visualization , January 2004.
[33] Simon Rowe and Andrew Blake. Statistical mosaics for tracking. Image and Vision Computing, 14(8):549–564, August 1996.
[34] Yiwei Wang, Robert E. Van Dyck & John F. Doherty, "Tracking moving objects in video sequences" in Proceedings of IEEE Applied Imagery Pattern Recognition Workshop Washington DC, Oct. 2000.