CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The tracking of objects in video can be used in many applications such as video surveillance [1], human computer interaction [2], e-health care [3], autonomous robot system [4] etc. However, there are many challenges in the object tracking. In real life scenarios, illumination variation can affect the system. Object occlusion is also a very big issue. The occlusion can be partial or full and it can be for short intervals or longer period. The complicated movements of the object also deteriorate the performance of the tracker. The tracking algorithm also faces the problem of clutter environment. Researchers are working continuously to overcome these issues and many algorithms have been suggested so far. Object representation methods are widely used for these problems which are based on object contour, shape, texture and feature etc. These methods are broadly classified into five categories. These categories are based on appearances, feature, contour, model and hybrid.

The methods based on appearance [5] use the visual information of the object. This visual information can be object's color, shape, and texture. These methods are able to handle the small transformation of the object which includes translation and rotation. But there is a limitation with these methods, they are not illumination invariant and special care is required in this scenario.

In Feature based tracking methods [6], some feature points are needed to be calculated and the tracking is performed on the basis of matching of the features from one frame to another. The strength of these methods is capability to deal with partial occlusion but they possess one drawback. The feature based methods are not able to correctly distinguish between the target feature points and background feature points.

Contour based methods [7] are able to track the rigid and non-rigid objects. Contours are represented by snakes and geodesic active contour or meshes [8]. However issue arises when the

objects are occluded. So, occlusion detection techniques have to be incorporated into the system and objects estimation need to be done to track the object.

Tracking methods based on model [9] can deal with partial occlusion and illumination variation. These methods create the object model (2D or 3D). These models exploit the object shape information and are able to deal with change in view of object's angle. These methods store the shape information of the each object in memory and major change in appearance is also stored and matching is performed from one frame to another. The main drawback of such methods is very computationally expensive and if there are multiple objects then the computation requirement is enormous.

Finally, the hybrid methods are those methods which are a combination of above said methods. These methods incorporate the two or more above mentioned methods [10] [14] and utilizes their advantages and effectiveness in such a way that the overall algorithm not able to deal with the various problems of the tracking but they are computationally effective. The major researches are being conducted in this category.

In our approach we are using hybrid modelling which fuses the appearance based and feature based modelling to present a novel approach for the tracking. From appearance based model we are using color, edge and texture and from feature based model we are using FAST features [11].

The proposed method detects the objects in a frame using background subtraction. There are two possibilities available: one the background frame is available and other is the background need to be learned from the video. In the first case, a background frame is generated by taking averages of multiple frames in which the objects are not available. However, this approach cannot be generalized for the practical scenarios because it is difficult to have frames without object in real life. Also, there is change in illumination or light intensity for the whole day so one background frame used for tracking objects in the morning may not be effective in the afternoon.

However, the second approach is more towards realistic scenario where the background frame is learned from the video. In [12] a detailed description of background modelling is presented. Various types of background modelling can be seen in [13] [15] [16] [17] [18] [19]. In our method, we have used 3 Gaussian Mixture Models and the system is trained with 40 frames. The number of GMM can be increased which will result into better accuracy, but it will lead to

computational cost. A mask is generated on the basis of background where pixel with value 1 belongs to the foreground and pixel 0 will represent the background. Morphological operations are used for connecting components. A blob which has an area less than 400 pixels will not belong to foreground.

In appearance based modelling, the color, edge and textures are used for the tracking. In most of the cases the color is sufficient to track the object, but when the object has similar color to the background then it becomes difficult for the system to track. To overcome this problem, texture is used. Texture is a degree of intensity dissimilarity of a surface which enumerates properties such as smoothness and regularity. Compared to the color space model, texture requires a processing step. On the basis of color, the texture features are less sensitive to illumination changes as same as to edge features.

In this work, we have used feature also so as to make the system robust. Invariant features are a very powerful tool that can be used in various applications for robust tracking. A good feature should possess few properties like repeatability, locality, quantity, accuracy etc. There are many features descriptor are available and each descriptor has its pros and cons. A very robust descriptors like SIFT provide very good result but they are not computational efficient. One of the popular corner detector was Harris which is used to detect the corners in an image [20]. The other two popular methods are SIFT [21] and SURF [22]. Both of these methods are computationally expensive and also have been patented. Another feature detector is FAST (*Features from Accelerated Segment Test).* It was introduced by Rosten and Drummond in [23] [24] which is based on SUSAN (Smallest Uni-Value Segment Assimilating Nucleus Test) detector [25] for corner detection. FAST feature extraction has shown a significant performance for real time computer vision applications because of its high repeatability and fastness. Small patches are made across these feature points and matching is performed. Feature based tracking is useful in the partial occlusion scenario.

In the bibliography of tracking algorithm, most of them are based on object representation based on appearance. Earlier methods use the initial object appearance as a base model and uses it throughout the algorithm. Due to which, these methods are not able to handle the severe change in object appearance. Sometime it gives the unsatisfactory result for partial occlusion as well.

Few examples of such methods can be found in [27] [28] in which various versions of mean shift algorithm are used. The problem of partial occlusion can be dealt by fragmenting the object into various fragments and tracking of each fragment individually. The performance of the tracker is affected by the size of the fragments. Too many or big fragments causes high computation while too few fragments can direct the tracker away from its actual path. Adaptive appearance model [29] can deal with change in object view angle. The same thing can be done using multiple hypothesis for object state [30]. These methods are based on Bayesian estimation and approaches to Sequential Monte Carlo methods such as Particle filter [31].

1.2 ORGANIZATION OF THESIS

Chapter 2: This chapter includes the overview of background modelling using Gaussian Mixture Models. It also covers the mathematical formulation of background.

Chapter 3: This chapter is about the Appearance based model. In appearance based model, we generally use color, edge and texture to represent an object. Histogram of color and edges are used for matching and Gabor filter is used for the texture analysis.

Chapter 4: This chapter is about the feature based model in context of object tracking. FAST feature has been discussed for the feature extraction and matching.

Chapter 5; This chapter includes the mathematical formulation of the Kalman filter and how it can be used for the tracking of multiple objects and explains the prediction of an object when the object is not visible.

Chapter 6: This chapter discusses the proposed algorithm for the tracking. It combines the technique of Appearance and Feature based methods and helps the system to track the object in an efficient way.

Chapter 7: Results are mentioned in this chapter and conclusion and future scope is discussed.

CHAPTER-2

BACKGROUND MODELING USING GMM

2.1 INTRODUCTION

In the surveillance domain, change detection has been extensively used in order to segment foreground objects from the background. Foreground objects are associated between frames in order to perform a scene analysis and detect events of interest. The parts of the scene which are normally observed are considered as background. Therefore, it is assumed that the background can be well described by means of a statistical model, the background model. Nevertheless, there are some background characteristics as moving foliage or sudden illumination changes, which make challenging the task of background modelling and maintenance. A comprehensive study of the main challenges and some principles that might be used to tackle them can be found in [12]. The segmentation of foreground objects by means of detecting the changes with reference to a background model is commonly known as background subtraction.

The wide range of issues a background model has to deal with has given rise to a high number of background subtraction approaches [12] [15] [16] [17] Since their first formulation in [13], per-pixel adaptive Gaussian Mixture Models (GMMs) have become a popular choice because of their ability to achieve many of the requirements of a surveillance system, e.g. adaptability and multimodality, in real-time with low memory requirements. Basically, the history of each pixel is modelled by a mixture of K Gaussian distributions, which are updated by means of an Expectation Maximization (EM) -like algorithm. Based on this model, pixels are classified as either background or foreground. The method proposed in [13] has been enhanced in many directions, covering the background model initialization [18], the model maintenance [19] and the processing of video sequences captured with pan-tilt-zoom cameras and even with a mobile observer.

2.1.1 Principle of Background Modelling Using Mixture of Gaussians

Stauffer and Grimson [13] generalized this idea by modelling the recent history of the color features of each pixel {*X*1,...,*Xt}* by a mixture of K Gaussians. It considers the values of a particular pixel over time as a "pixel process". The "pixel process" is a time series of pixel values, e.g. scalars for gray values or vectors for color images. At any time, t, what is known about a particular pixel, $\{x_0, y_0\}$ is its history

$$\{X_1, ...X_t\} = \{I(x_0, y_0, i) : 1 \le i \le t\} \quad \text{.....(2.1)}$$

where I is the image sequence.

First, each pixel is characterized by its intensity in the RGB color space. Then, the probability of observing the current pixel value X_t at time t is considered given by the following formula in the multidimensional case:

$$P(X_t) = \sum_{k=1}^{K} \omega_{k,t} . \eta(X_t, \mu_{k,t}, \Sigma_{k,t}) \quad \text{.....(2.2)}$$

where K is the number of distributions used in GMM, $\omega_{k,t}$ is a weight associated to the k^{th} Gaussian at time t with mean $\mu_{k,t}$ and standard deviation $\Sigma_{k,t}$ which is assumed to be the diagonal matrix $\sigma_{k,t}^2 I$. η is a Gaussian probability density function.

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(X_t - \mu)\Sigma^{-1}(X_t - \mu)} \quad \text{.....(2.3)}$$

For computational reasons, covariance matrix is assumed as $\sigma_{k,t}^2 I$ which represents that RGB color components are independent to each other and have the same variances. So, the covariance matrix can be written as:

$$\Sigma_{k,t} = \sigma_{k,t}^2 I$$

The assumption allows us to avoid a costly matrix inversion at the expense of some accuracy. Thus, the distribution of recently observed values of each pixel in the scene is characterized by a mixture of Gaussians.

So, each pixel is characterized by a mixture of K Gaussian distributions. Once the background model is defined, the different parameters of the mixture of Gaussians must be initialized. The parameters of the MOG's model are the number of Gaussians K, the weight $\omega_{k,t}$ associated to the k^{th} Gaussian at time t with mean $\mu_{k,t}$ and standard deviation $\Sigma_{k,t}$.

K determined the multimodality of the background and by the available memory and computational power. Stauffer and Grimson [13] proposed to set K from 3 to 5. In our work we have used K=3. The model is initialized by EM algorithm and parameter initialized are weight, mean and the covariance matrix

Once the parameter initialization is done, a first foreground detection is done and the parameters are updated for the next frame. Firstly, [13] used the criterion of ratio $r_j = w_j / \sigma_j$ and ordered the K Gaussians following this ratio value in descending order. This ordering supposes that a distribution with high weight and small variance tend to top of the list of all the Gaussian distribution of particular pixel and thus to be a part of the background model.

The first B Gaussian distributions which exceed certain threshold *T* can be considered for a background distribution as:

$$B = \arg\ \min_K \left(\sum_{k=1}^{K} \omega_{k,t} > T \right) \quad \text{.....(2.4)}$$

Where $B \leq K$ and T is predefined threshold indicating the minimum portion of the data that should be assumed to be background.

The other distributions are considered to represent a foreground distribution. Then, when the new frame comes at times t+1, a match test is made for each pixel. A pixel matches a Gaussian distribution if the Mahalanobis distance

$$sqrt\left((X_{t+1} - \mu_{i,t})^T \Sigma_{i,t}^{-1} . (X_{t+1} - \mu_{i,t})\right) < k\sigma_{i,t} \quad \text{.....(2.5)}$$

where *k* is a constant threshold equal to 2.5.

The model is adapted by means of EM-like algorithm by using winner takes all updating strategy. This means that when matching is performed for a particular pixel with value X_t to all of the Gaussian distribution in descending order. The first distribution in descending order (say

m^{th}) whose distance is τ times smaller than its standard deviation is selected as the best match. τ is generally set to be 2-3. So, if the best match lies in first B Gaussian distribution, then the pixel is considered to be as background otherwise a foreground pixel.

Secondly, if none of the Gaussian distribution matches with the current pixel value X_t, a new distribution is created. The new distribution is initialized with the current value of the pixel as mean value and a default value of the variance with a low prior weight. If none of the distributions in the mixture model is free then newly created distribution replaces the lowest r_j ratio distribution. As, no match is found with any of the K Gaussians. In this case, the pixel is classified as foreground.

For the distribution to which the matched component is found can be updated as follows:

$$\omega_{k,t+1} = (1-\alpha)\omega_{k,t} + \alpha M_{k,t} \quad \text{.....(2.6)}$$

where α is a considered to be a constant learning rate and $M_{k,t}$ is a binary function with value 1 for the matched case and 0 otherwise.

Furthermore, $\mu_{m,t+1}$ and $\sigma_{m,t+1}$ parameter of m^{th} distribution (the distribution with highest matching) is updated as:

$$\mu_{m,t+1} = (1-\rho_{m,t+1})\mu_{m,t} + \rho_{m,t+1}X_{t+1} \quad \text{.....(2.7)}$$

$$\sigma^2_{m,t+1} = (1-\rho_{m,t})\sigma^2_{m,t} + \rho_{m,t}(X_{t+1}-\mu_{m,t+1})(X_{t+1}-\mu_{m,t+1})^T \quad \text{.....(2.8)}$$

Where m is the distribution from {1,..K) and $\rho_{m,t}$ is a learning rate $\rho_{m,t} = \alpha.\eta(X_t \mid \mu_m, \Sigma_m)$

Fig 2.1 Foreground detected after background subtraction using GMM

Due to the good compromise between segmentation results, memory and processing time requirements, GMM have been extensively used in the surveillance domain. Nevertheless, there are still some improvement possibilities in the formulation of what is being suggested by Grimson [13].

CHAPTER 3

APPEARANCE BASED MODEL

3.1 COLOR HISTOGRAM

The color of the object can be useful in matching an object from one frame to another frame. In a color image, there are 3 components which are RGB. When an object is detected, its color information is stored in its color histogram.

The object is modelled as m-bin histogram which is defined as:

$$\hat{q} = \{\hat{q}_u\}_{u=1...m} \quad \text{.....(3.1)}$$

The histogram is normalized as

$$\sum_{u=1}^{m} \hat{q}_u = 1$$

A target candidate at location **y** in the subsequent frame is described by its histogram as:

$$\hat{p}(y) = \{\hat{p}_u(y)\}_{u=1...m} \quad \text{.....(3.2)}$$

Where $\sum_{u=1}^{m} \hat{p}_u = 1$ which is sought to be the closest to $\hat{\mathbf{q}}$.

The metric used for measuring the distance between the histogram is:

$$d_{\hat{q}}(y) = \sqrt{1 - \rho[\hat{p}(y), \hat{q}]} \quad \text{.....(3.3)}$$

Where $\rho[\hat{p}(y), \hat{q}]$ is the bhattacharyya distance. Smaller value of bhattacharyya distance represents the high value of matching.

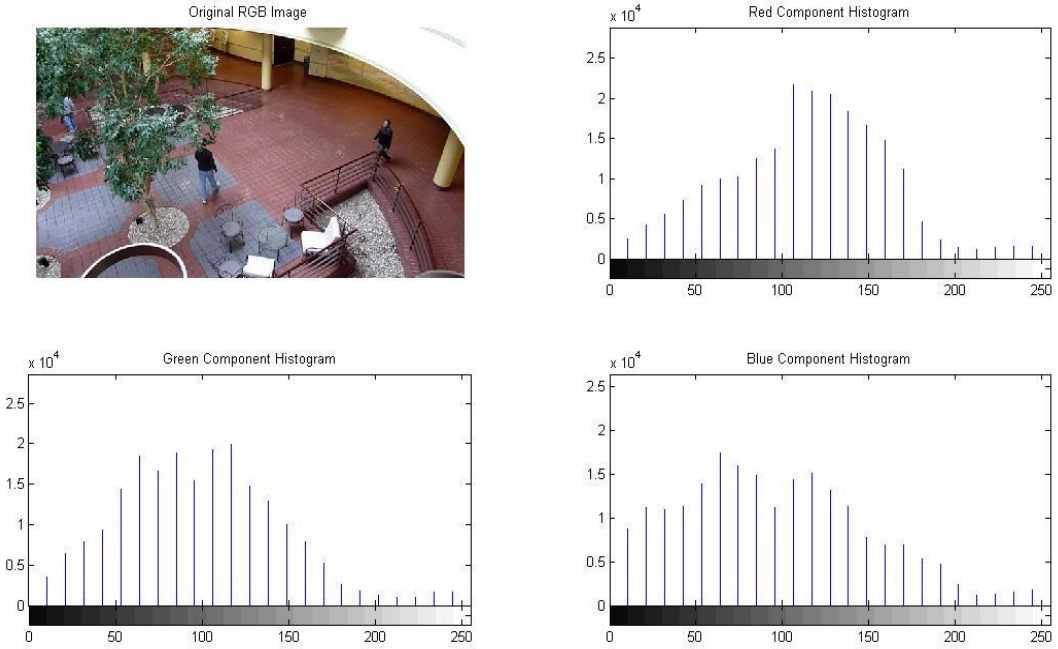


Fig 3.1 Color histogram of whole frame of video

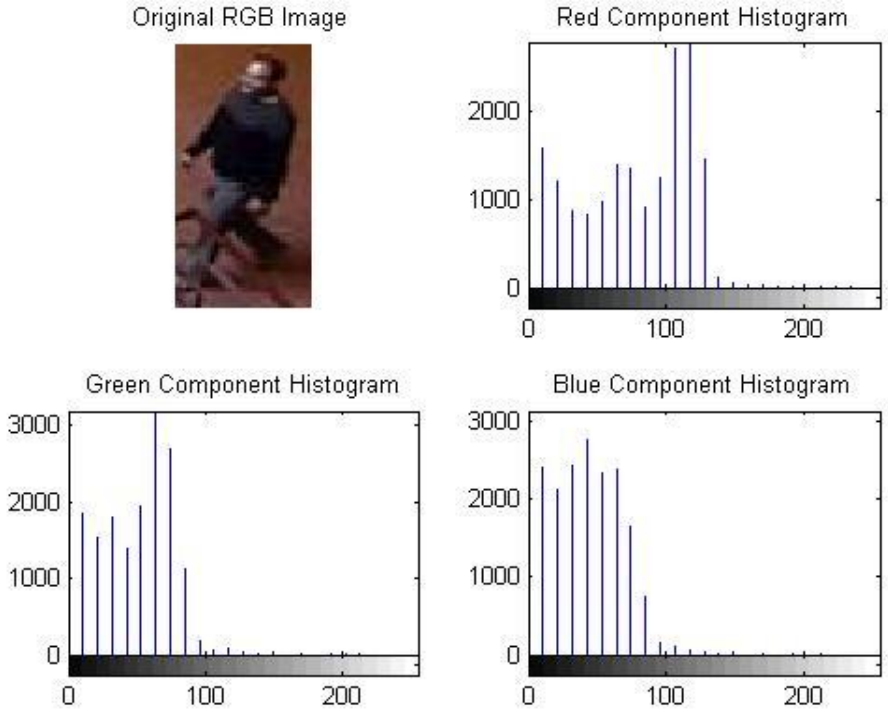


Fig 3.2 Color histogram of one object in that frame

3.2 EDGE HISTOGRAM

In many cases, color histogram features are not sufficient for object description, especially for object of varying size. Edge Histogram can help when the size of objects varies.

Edge contains some of the most valuable information in an image. There are definitely several applications where this information can be used. For instance, the edges can be used to measure the size of an object, or to isolate a specific object from the background, or to recognize and classify objects. An edge can be defined as the boundary where there is a strong change in the intensity or a local discontinuity in the pixel values that exceeds a given threshold value. Thus, edges preserve the shape of the body and reduces the date that need to be processed.

The Canny edge detector has shown a good performance in edge detection. It finds the edges by looking local maxima of the gradient of image. The gradient is calculated using derivative of Gaussian filter. This method uses two thresholds to detect strong and weak edges. An edge pixel with gradient value above the upper threshold will surely be a part of edge. Similarly, a pixel with gradient value lower than lower threshold will not be surely a part of edge. Those pixels whose gradient lies in between the upper and lower threshold are considered as edge points if they are connected to strong edges. This method is therefore less likely to be fooled by noise as compared to other methods like Sobel, Prewitt etc.

The Canny edge detector can be implemented as follows:-

Step 1: Before trying to locate and detect any edges, noise must be filtered out from the input image. Gaussian filter is used extensively in image processing for smoothening of the images, and also it can be computed using a simple mask. The sensitivity of the detector for noise depends upon the size of the Gaussian window, larger the Gaussian mask, lower is the sensitivity of detector towards noise. While with the increase in size of the Gaussian mask, the localization error also increases.

A 5*5 Gaussian filter can be seen as:

1/273	1	4	7	4	1
	4	16	26	16	4
	7	26	41	26	7
	4	16	26	16	4
	1	4	7	4	1

Table 3.1 5*5 Gaussian Filter

Step 2: After smoothening of image, the gradient of image at each pixel is calculated. The gradient [32] is calculated by convolving with gradient operator as seen in table 3.2. The Gx operator provides the gradient in horizontal direction while the other operator calculate the gradient in vertical direction. Then, the gradient magnitude and direction at each pixel can be calculated by equation 3.4

-1	0	1
-2	0	2
-1	0	1

Gx

-1	-2	-1
0	0	0
1	2	1

Gy

Table 3.2 Gradient Operator

$$G = \sqrt{Gx^2 + Gy^2} \quad \text{.....(3.4)}$$

$$\Theta = \arctan(Gy / Gx) \quad \text{.....(3.5)}$$

Step 3: Once the edge direction is known, the next step is the binning of the direction. 4 possible direction will be sufficient for the desired task and these directions are 0 degrees (in the horizontal

direction), 45 degrees (along the positive diagonal), 90 degrees (in the vertical direction), or 135 degrees (along the negative diagonal).

Step 4: After the edge directions are known, non-maximum suppression now has to be applied. Non maximum suppression is used to trace along the edge in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered to be an edge. This will give a thin line in the output image. Two threshold are used in canny edge detector, one is upper threshold which provides the confirmation of any pixel to be edge pixel and the other threshold is lower threshold and it confirms the pixel not to be a part of edge. A decision is required for the pixels value lying in between the upper threshold and lower threshold. If the pixel is a part of connected component then pixel is edge pixel otherwise it is not. Fig 3.4 and 3.5 show the canny edge detector output of the image

Fig. 3.3 Canny edge detector output of whole frame

Fig 3.4 Canny edge detector output of object's image in frame

Once the histogram of color and edge are calculated the next aim is calculate the similarity between candidate and target joint distribution using Bhattacharyya distance. Here, we will explore how marginal histograms can be used for object tracking. Let ρ_1 be the Bhattacharyya coefficient calculated using color features only and let ρ_2 be the Bhattacharyya coefficient calculated using edge features only.

$$\rho_1 = \sum_u \sqrt{q_u p_u}$$

$$\rho_2 = \sum_m \sqrt{s_m t_m}$$

where **q** and **p** are the target color histogram and candidate color histograms and **t** and **s** are the target and candidate edge histograms, respectively. u and m are the bins. Let ρ be the product of ρ_1 and ρ_2

$$\rho(y) = \rho_1(y)\rho_2(y)$$

Whenever there will be matching, the product of Bhattacharyya coefficient will increase.

3.3 GABOR FILTER FOR TEXTURE ANALYSIS

The application of Gabor filter can be widely seen in image description [32]. Gabor filter is made with filter banks which works on different scales and orientation due which the Gabor filter works as scale and orientation invariant. Information can be extracted from visual scene using Gabor filter and these filter shows excellent properties [33]. A common way to texture analysis is convolving the image pixel (x,y) with N filters which result into a N dimensional vector. The local and global features can be extracted from image using Gabor filter by selecting the some parameters like bandwidth, frequency (f_0) and orientation (θ).

To extract the spatial and frequency information at the same time leads to popularity of wavelet transform. Gabor wavelet provides the good resolution in time and frequency domain. The main objective of Gabor filter is to capture the visual properties like spatial frequency characteristics, orientation selectivity and spatial locality [34] and leads to utilization of Gabor in various applications as feature vector. These feature vectors are formed based on different scales and orientation. The Gabor filter can be mathematically expressed as [33][34]:

$$F(x, y) = \exp(-(x_0^2 + \gamma^2 y^2) / 2\sigma^2) \times \cos(2\pi x_0 / \lambda) \quad \text{.....(3.6)}$$

$$x_0 = x\cos\Theta + y\sin\Theta \quad \text{.....(3.7)}$$

$$y_0 = -x\sin\Theta + y\cos\Theta \quad \text{.....(3.8)}$$

Where λ is the wavelength, σ is the standard deviation of the Gaussian factor. Spatial frequency bandwidth is represented as σ/λ, and used 0.5 for this work. Ψ- Phase offset to get the symmetry of the kernel in terms of origin. Γ is the aspect ratio which provides the ellipticity to the receptive field. There can variation in scale and orientation selection. In our method, we have selected $\sigma = \{1, 2, 3\}$ and $\theta = \{0^o, 45^o, 90^o, 135^o\}$.

These steps can be easily understood as:

(1) An input image is feed to the Gabor filter bank and it results into different intensity profile. After it, smoothening is done using Gaussian filter which brings the intensities in the discrete levels.

(2) Now as the intensity are differentiated so we calculate the features vectors which are used for texture representation and hence a texture based segmented image is generated.

Some properties of Gabor filters:

- It is a tunable band pass filter and similar to Short Time Fourier Transform
- Satisfies the lower-most bound of the time-spectrum resolution (uncertainty principle)
- It‘s a multi-scale, multi-resolution filter
- Has selectivity for orientation, spectral bandwidth and spatial extent.
- It has response similar to that of the Human visual cortex (first few layers of brain cells)
- Its drawbacks is high computational cost because of selection of large filter banks. The selection of number of filter banks can vary from application to application.

As in case of object tracking, Gabor filter is used for the feature extraction and its output vector has high dimensionality. If the dimension of input image is given 160x120 pixels then after the convolution of the image with Gabor filter bank of 3 scales and 4 orientations the dimensionality will become 12x160x120=230400. The 2D Gabor filtered images is converted into a pattern vector this process is repeated for all other 12 responses. At last all the pattern vectors for 12 responses are arranged either in rows or columns as features.

CHAPTER 4

FEATURES BASED MODEL

4.1 INTRODUCTION

Feature detection has been used widely in computer vision applications e.g. tracking of object based on matching, registration or reconstruction of 3D model, stereo correspondence and motion tracking. For the purpose of object tracking, feature points are extracted from the object body and they are matched with the features of the candidate object region. On the basis of matching, the locations of the objects are decided and objects are tracked. Feature based trackers are very useful for dealing with partial occlusion. When the object is partially occluded, some of its body parts are visible and feature points on the visible part of the object will match with its previous frame. Corners can be considered as a very good feature point. Because of corner, the system can be made rotation invariant and scale invariant up to some extent.

The basic properties of good feature detection are as follows:

- If two images of the same objects are taken at a different angle, a high percentage of feature detected on the scene part visible in both images should be found in the same image.
- The features should be local, so as to reduce the probability of occlusion and to allow simple model approximations of the geometric and photometric deformations between two images taken under different viewing conditions.
- The number of detected features should be sufficiently large, such that a reasonable number of features are detected even on small objects
- The detected features should be accurately localized, both in image location, as with respect to scale and possibly shape.

Several different approaches to corner detection are known in literature. The Harris corner detection algorithm has been widely used in various earlier methods [20].

The global operator SIFT (Scale Invariant Feature Transform) [21] extracts the feature points in the image which are invariant to illumination change, scale and rotation. Although SIFT is very robust method, but it is computationally expensive as well. The alternate to the SIFT which become very popular was SURF (speeded up robust features). More recently, a new feature detector (FAST) is becoming popular because of its computational efficiency.

4.2 FAST DETECTOR

The FAST (*Features from Accelerated Segment Test)* detector, introduced by Rosten and Drummond in [23, 24] builds on the SUSAN detector [25]. SUSAN computes the corner based on a circular region around the centre pixel. It computes the self-similarity with the pixels in that disc area. The pixels which are closer to the centre pixel are assigned stronger weights strength. This measure is known as the USAN (Univalue Segment Assimilating Nucleus). A high value of USAN indicates the centre pixel is having more similarity with neighbouring pixels and it should not be treated as corner pixel while high value of it will assign the centre pixel as corner pixel.

Self-dissimilarity can be measured using the pixels on the segment of the discretized circle instead of the whole circular area and this idea is taken further by FAST, which compares pixels only on a circle of fixed radius around the point. The test criterion operates by taking a ring of 16 pixels around the centre pixel or nucleus. Each of the circle pixels is classified into 3 categories: brighter, darker and similar. These 3 categories are built based on threshold as shown in eq. (4.1)

$$S_{p\to x}=\begin{cases} d, & \mathrm{I}_{p\to x}\le \mathrm{I}_p - t & \text{(darker)} \\ s, & \mathrm{I}_p - t < \mathrm{I}_{p\to x} < \mathrm{I}_p + t & \text{(similar)} \\ b, & \mathrm{I}_p + t \le \mathrm{I}_{p\to x} & \text{(brighter)} \end{cases} \quad \text{.....(4.1)}$$

FAST classifies p as corner if there exists a segment of 12 contiguous pixels which are all brighter or darker than the nucleus pixel p. The selection of t should be carefully performed. A larger value of t will reduce the number of corners in the image, but it will provide the strong

corners while low value of t will generate a large number of corners with the inclusion of weak points.

The test for this condition can be optimized by considering only 4 pixels at the beginning which are 1, 5, 9, 13 see fig (4.2). If 3 out of 4 pixels are neither brighter nor darker than centre pixel by some threshold then the candidate pixel cannot be corner. This optimization reduces the computation cost. Based on above approach, 16 pixel's intensity values are treated as feature vector and it is classified as positive or negative based on 12 contiguous pixels. This portioning helps in reducing the matching of features, as, positive features are not required to be matched with negative ones. So, the computation can be further reduced.

There are two major weaknesses with this method which are identified by [24] [25] which are: First, the detector is not generalized well for arc length less than 12. And secondly, the speed of operation depends upon the order in which comparison is performed. In [24] [25], these issues have been considered.

First, the set of all central pixels in all training images are labeled as corner or non-corner using segment test criteria for a given arch length L and threshold T. Second, for each of the 16 pixels surrounding the centre pixel, one of the three states is assigned. Finally, the ID3 [26] algorithm is applied to this data set to find the best ternary decision tree classifier. The ID3 algorithm is used to select the pixels which yield the most information about whether the candidate pixel is a corner. This is measured by the entropy of the positive and negative corner classification responses based on this pixel.

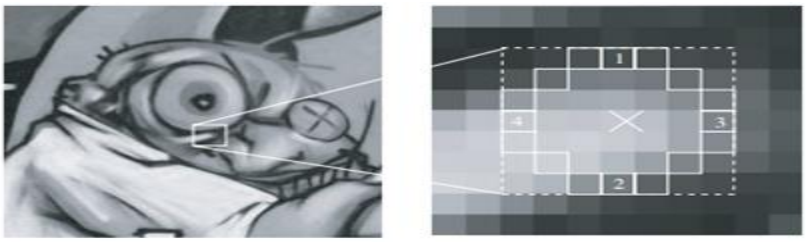


Fig 4.1 Illustrations of pixel examined by the FAST detector

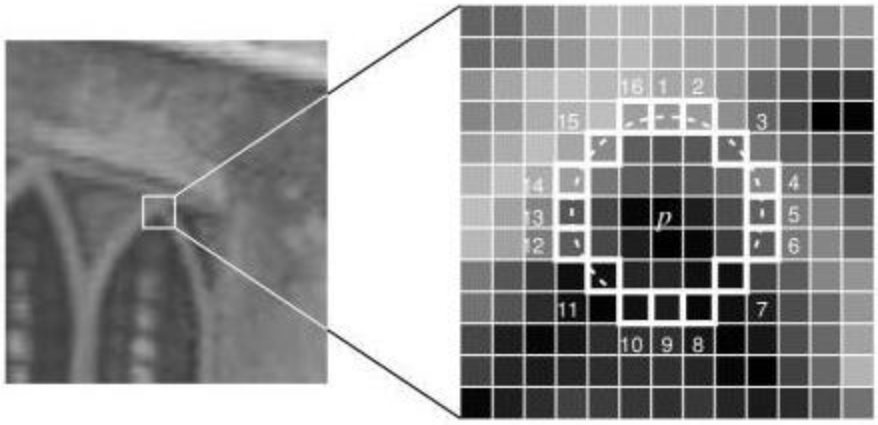


Fig 4.2 a circle of 16 pixels around the centre pixel p

The process is applied recursively on all three subsets and terminates when the entropy of a subset is zero. The decision tree resulting from this partitioning is then converted into C-code, creating a long string of nested if-then-else statements which is compiled and used as a corner detector. Finally a non-maxima suppression is applied to the sum of the absolute difference between the pixels in the circle and the centre pixel. This results in a very efficient detector which is up to 30 times faster than the DoG detector.

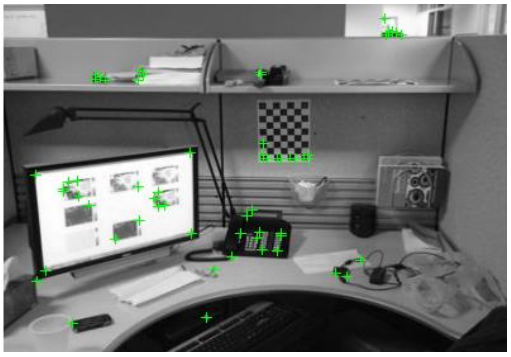
FAST is many times faster than the other existing detectors like SIFT, SURF etc. Due to its high speed, it can be used for real time applications. Although it has good computation efficiency, but it does not include the orientation factor which other methods like SIFT and SURF are using.

(a) (b) (c)

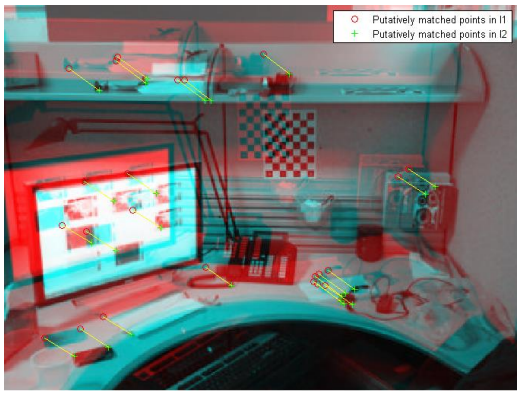
Fig 4.3 Detection of FAST Features. (a) Original image, (b) FAST features extracted, (c) strongest 50 FAST features

(a)

(b)



(c)



(d)

Fig 4.4 Matching of FAST Feature in a stereo image pair (a) left and right stereo image, (b-c) FAST features detected on both the images, (d) Correspondence between image pair

CHAPTER 5

KALMAN FILTER

5.1 INTRODUCTION

The Kalman filter is a mathematical tool developed by Rudolph E. Kalman in 1960 to make an estimation of an observed variable using a predictor-update set of Equations. If the noise is characterized as Gaussian and the problem where the filter is used can be represented as linear, the Kalman filter estimation is considered to be the optimal solution or the minimum squared error (MSE). In the nonlinear case, the Kalman filter has to be linearized and the solution is considered an ad hoc state estimator.

The Kalman Filter is one of the most widely used methods for tracking and estimation of the state of a linear dynamic system due to its simplicity, optimality, tractability, and robustness.

5.1.1 Mathematical Mechanism

The Kalman Filter is a recursive linear estimator, applying only to Gaussian densities, of a more general probability density propagation process. The fundamental formulation of the Kalman filter would be understood as a Bayesian filtering framework.

$$\text{prediction}: \quad p(x_{t+1} \mid y_{1:t}) = \int p(x_{t+1} \mid x_t)\, p(x_t \mid y_{1:t})\, dx_t \qquad \text{.....(5.1)}$$

$$\text{update}: \quad p(x_{t+1} \mid y_{1:t+1}) = \frac{p(y_{t+1} \mid x_{t+1})\, p(x_{t+1} \mid y_{1:t})}{\int p(y_{t+1} \mid x_{t+1})\, p(x_{t+1} \mid y_{1:t})\, dx_{t+1}} \qquad \text{.....(5.2)}$$

Here the prediction equation predicts the next state based on all previous observations while the update equation represents the corrected state with inclusion of current observation.

The state model equation of Kalman filter can be written as

$$x_k = F_{k-1}x_{k-1} + G_k u_k + v_{k-1} \quad \text{.....(5.3)}$$

$$z_k = H_k x_k + w_k \quad \text{.....(5.4)}$$

Where,

x_k is the state vector, which is unknown in nature

F_{k-1} is the motion model matrix,

G_k is the control matrix,

u_k is the control input,

v_{k-1} is the motion noise which is a Gaussian noise $v_{k-1} \sim N(0, Q_{k-1})$

z_k is the measurement vector,

H_k is the measurement matrix, and

w_k is the measurement noise which is also Gaussian in nature $\mathrm{w}_k \sim N(0, R_k)$

where Q_k and R_k are known as the process and measurement noise covariance matrices at kth instant respectively. Both the noises are considered to be Gaussian in nature and they assumed to have zero mean. The filtering algorithm can be divided in two sections, the prediction and the update. The ultimate goal of the Kalman filter is reduce the error covariance matrix P which is defined as below:

$$P_{k|k-1} = E[(\mathrm{x}_k - \hat{x}_{k|k-1})(\mathrm{x}_k - \hat{x}_{k|k-1})^T] \quad \text{.....(5.5)}$$

The above equation represents the error covariance matrix at kth instant based on previous k-1 instant. $\hat{x}_{k|k-1}$ is the state estimate at kth instant based on k-1 instant which is evaluated as :

$$\hat{x}_{k|k-1} = F_{k-1}\hat{x}_{k-1|k-1} + G_k u_{k|k-1} \quad \text{.....(5.6)}$$

Hence, covariance matrix can be written as

$$P_{k|k-1} = F_{k-1}P_{k-1|k-1}F_{k-1}^T + Q_{k-1} \quad \text{.....(5.7)}$$

The update equations are:

$$K_k = P_{k|k-1}H_k^T(H_k P_{K|k-1}H_k^T + R_k)^{-1} \quad \text{.....(5.8)}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - H_k\hat{x}_{k|k-1})$$
$$P_{k|k} = (I - K_k H_k)P_{k|k-1} \quad \text{.....(5.9)}$$

where K_k is the gain used to weight the measurement against the prediction at kth instant. The prediction equations find a priori distribution by predicting the mean and variance of the next value of the observed variable. In the case of the update equations, they calculate the posterior distribution by using the new weighted measurements.

If the process is measured perfectly and the measurement error approaches zero, i.e., $R_k \rightarrow 0$, then

$$\lim_{R_k \to 0} \mathrm{K}_k = H_k^{-1}$$

The update equation become:

$$\begin{aligned}\hat{x}_{k|k} &= \hat{x}_{k|k-1} + H_k^{-1}(z_k - H_k\hat{x}_{k|k-1}) \\ &= \hat{x}_{k|k-1} + H_k^{-1}z_k - \hat{x}_{k|k-1} \\ &= H_k^{-1}z_k\end{aligned}$$

Conversely, if the state is estimated perfectly, then a priori covariance matrix $P_{k|k-1}$ approaches to zero and Kalman gain becomes zero

$$\lim_{P_{k\,|\,k-1} \to 0} \mathrm{K}_k = 0$$

the gain ignores the residual, in favour of the estimate $\hat{x}_{k|k}$

$$\begin{aligned}\hat{x}_{k|k} &= \hat{x}_{k|k-1} + 0(z_k - H_k\hat{x}_{k|k-1}) \\ &= \hat{x}_{k|k-1}\end{aligned}$$

5.2 KALMAN FILTER FOR THE OBJECT TRACKING

Kalman filter in the reference of object tracking can be used in the prediction of the location and the centroid of the object and its size. If the camera sampling rate is 30 frames per second then this frame rate is sufficient to make us an assumption of little change in the movement of the object in the adjacent frame. So the change in the size of the object and object

movement will not be abrupt and hence the centroid and the size of the tracking window can be considered as a feature vector for the tracking of objects. In terms of operation, the Kalman filter tracking model can be divided into three sub modules which are: motion estimation model, matching of features and model update.

5.2.1 Motion Estimation Model

Kalman filter used for tracking is defined in terms of its states, motion model, and measurement equations.

The State Equation is given by

$$x_{k+1} = Ax_k + v_k$$

The Observation Equation is given by

$$z_k = Hx_k + w_k$$

Matrix x_k is a stathe positionr which contains pthe sizeion of the centroid, size of the tracking window and their velocities. It is an eight-dimensional state vector, which can be expressed as:

$$x_k = [x_{0,k}, y_{0,k}, l_k, h_k, v_{x,k}, v_{y,k}, v_{l,k}, v_{h,k}] \quad \text{.....(5.10)}$$

Where, $x_{0,k}, y_{0,k}$ represent c, ntroid coordinates in (x,y) direction, l_k, h_k represent half-width and half-height of the tracking window, $v_{x,k}, v_{y,k}, v_{l,k}, v_{h,k}$ represents their speed respectively.

The measurement vector of the system adopts the following form:

$$z_k = [x_{0,k}, y_{0,k}, l_k, h_k]^T \quad \text{.....(5.11)}$$

A is the transition matrix which relates the previous state to the next state without noise. *H* is the measurement matrix which provides the measurement of the state at that particular instant. v_k and measurement noise w_k are the Gaussian noises which are independent to each other. These noise values are entirely dependent on the system that is being tracked and adjusted empirically.

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Observation matrix H can be described as

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Once the state and measurement equations are defined, estimation of the object in the next frame can be initiated. Kalman filter faces a practical issue that it can be used effectively when the change in object size or location is small.

5.2.2 Feature matching

Feature matching is performed between two frames. The i^{th} object in k^{th} frame can be described by its centroid x^i_k, y^i_k and tracking window area as $\mathbf{S}^i_k$.

First, a distance between the centroid of i[th] object in the k^{th} frame is calculated with the jth object in the k+1th frame. So, a distance matrix is formed which has all the distance of one object with all objects in the next frame. This distance can be calculated as below:

$$D(i,j)=\frac{|\sqrt{(x_k^i-x_{k+1}^j)^2+(y_k^i-y_{k+1}^j)^2}|}{\mathrm{Max}_n|\sqrt{(x_k^i-x_{k+1}^n)^2+(y_k^i-y_{k+1}^n)^2}|} \quad \text{.....(5.12)}$$

Second, area similarity matrix is also formed by calculating the area difference between the *ith* object in the *k* frame with *jth* object in the *k*+1*th* frame and it is defined as:

$$A(i,j)=\frac{|S_k^i-S_{k+1}^i|}{Max_n|S_k^n-S_{k+1}^n|} \quad \text{.....(5.13)}$$

Where, $S_k^i=4*l_k^i*\mathrm{h}_k^i$.

With these definitions we define cost function is:

$$V(i,j)=\alpha D(i,j)+\beta A(i,j) \quad \text{.....(5.14)}$$

Where $\alpha+\beta=1$. The smaller the cost function's value is, the two objects are more likely have correspondence. α and β are user tunable. In our method, we have given distance matrix more weightage as compared to area matrix.

5.2.3 Model Update

When the minimum value of cost function is found, we use the *k*+1*th* frame features to update parameters of kalman filter motion model, and use them as the input in the next frame. Repeatedly doing this to finish the model update until the moving objects disappeared.

5.3 PROS AND CONS OF KALMAN FILTER

- If the object is fully occluded by background or by any foreground object in that case, it will keep predicting the object in linear motion.

- The main disadvantage of the Kalman filter is that it can be used only for linear and Gaussian model. To incorporate some extent of non-linearity, Extended Kalman filter can be used. If the system requirement is for non-linear and non-Gaussian model then Particle filter can be best utilized.

CHAPTER 6

PROPOSED METHOD

6.1 FLOW CHART OF PROPOSED METHOD

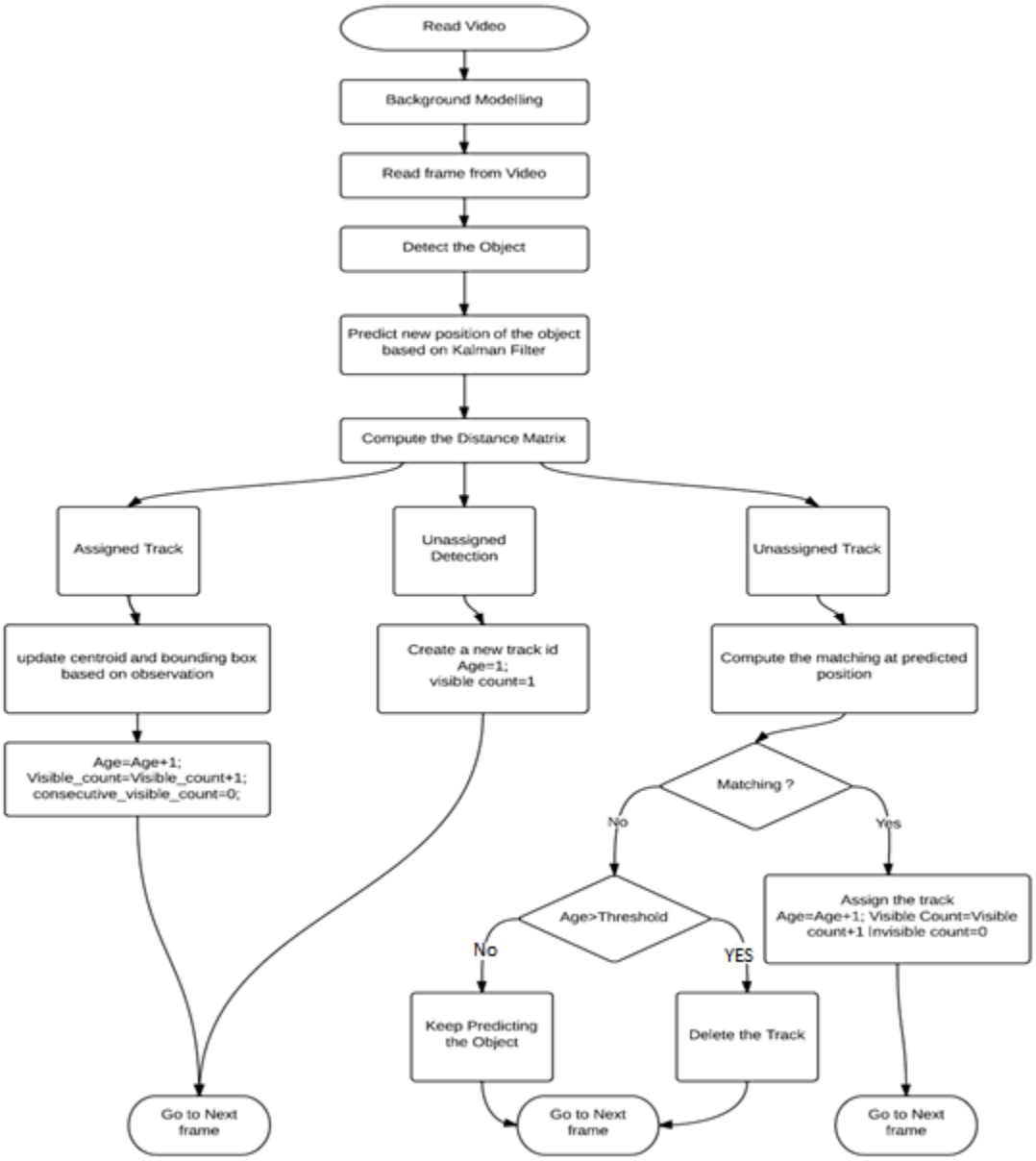


Fig 6.1 Flow chart of proposed method

6.2 IMPLEMENTATION

The Proposed method works on the principal of background subtraction where background is estimated on the basis of Gaussian Mixture model. For the tracking, the algorithm uses color, edge, texture and FAST features. Fusion of all the features will give a robust result. The algorithm suggests a novel approach to reduce the computation for the tracking.

The most of the computation is spent in evaluating the features such as edge, texture, FAST features etc. and their matching. If there is some certainty about the object location in the frame, then computation can be reduced substantially. The certainty of objected can be commented on the basis of Euclidean distance. If the frame rate is 30 fps then it can be commented that the object movement is restricted in next frame. So the Euclidean distance of particular object from one frame to another frame will be very low as compared to its Euclidean distance with other objects.

The proposed method does not perform the matching at every frame. Let us consider a case of a non-occluded scenario, there are multiple objects are available. As the objects are not occluded, they can be easily differentiated by the camera after background differencing. There will be substantial gap among the objects. So, it is not required to perform the feature matching. The objects with lower Euclidean distance will correspond to each other. Now let's consider another case. The objects are very close or they are partially occluded. If background subtraction is performed then it will merge both the objects and system will not be able to distinguish between objects. In that case Euclidean distance approach is not sufficient. To improvise the result, we perform the feature matching. For feature matching, we are using color, edge, texture and FAST feature. Hence, even though the objects are occluding partially, we can track them.

Also, we are tracking the objects in Bayesian framework. If an object is fully occluded by any other object or the background then the system keep on predicting the object based on its past observations.

If the object does not appear after certain number of frames then it can be considered that object is gone away from the frame and the id of that particular object is deleted.

In this algorithm, when the system detects a new object, it assigns a new track id to this object. The track id is basically a identifier which identifies an object among many other objects. So the system gives a integer number to the object and this id remains along with the object throughout the video. Along with track id, the system also assigns age, visible count, consecutive_invisible_count as counters which helps in explaining the detail of object in video. For instance, the age represents the number of frames that the object is available in video after first time detection. visible_count counts the number of frames in which the object was identified (not occluded). In every coming frame if the object is visible this counter increases. consecutive_invisible_count counts the number of consecutive frames in which the object is not visible. The system keep on predicting the object based on kalman filter. If this count value is greater than a particular threshold then it can be considered as the object is lost and all the details of the objects are deleted. If object appears again within time limit then consecutive_invisble_count is reset to zero and visible_count is increased by 1.

6.2.1 ALGORITHM OF PROPOSED METHOD

The method proceeds as below:

1. A video file is read and background modelling is done on it. Here we are using a Gaussian Mixture model (GMM) for background modelling. The number of GMM used is 3. The Number of frame for training is considered to be 40.
2. Now a frame is read from the video and background subtraction is performed on it. Morphological operations are applied on it and objects are extracted from it. An object is considered to be valid if it appears for 8 consecutive frames.
3. Color, Edge, Texture and FAST features are extracted from the objects and put in a stack.
4. Objects centroids and bounding boxes are calculated using blob analysis. Here an area of minimum 400 pixels is considered for objects consideration. Connected components with size less than 400 pixels are not considered. This property is tuneable.

5. Every new object detected is provided a track id, age, visible_count and consecutive_invisble_count. Once the object is visible after some frame, consecutive_invisble_count counter is reset.
6. The new position of the object is predicted using Kalman filter.
7. A distance matrix is computed for each object detected with each track and correspondence of each tack with each detection is done on the basis of Hungarian Algorithm which works on the principle of cost minimization.
8. Based on Hungarian Algorithm, the result is divided into 3 categories which are assigned tracks, unassigned tracks and unassigned detection. Assigned tracks are those tracks which are associated with a particular object. So it's a one to one mapping. The unassigned track indicates about those objects which are not visible. It may be because of object occlusion (partially or fully). In that case the object location is predicted for some frames. If the object doesn't show up within the threshold period then object is assumed to gone from the camera and that object track id is closed. Unassigned detection indicates the new object is arrived into the frame. So for the new object, a new track id is assigned

For Assigned Tracks:

In this category, the objects are visible and correspondence with previous frame is identified. The object position is corrected based on the observations and counter value of age,. Visible_Count and consecutive_invisble_count are updated as below:

Age=Age+1;

Visible_Count= Visible_Count+1;

consecutive_invisble_count=0;

For Unassigned Tracks:

Age=Age+1;

Visible Count= Visible Count;

consecutive_invisble_count= consecutive_invisble_count +1;

For Unassigned detection:

Age=1;

Visible Count= 1;

consecutive_invisble_count=0;

9. For the unassigned track, we perform the feature matching. So, if an object is partially occluded even then it can be identified. If the matching is above a threshold limit, then the unassigned track is assigned to corresponding object.
10. Repeat the step 2 to 9 until the last frame is processed.

Condition to assign a track to be lost:

An unassigned track is deleted when the object does not appear consecutively for few frames.In our method, we have calculated the visibility ratio which is the ratio of age and visible count. If an object is consecutively not visible for 15 frames and visibility ratio is less than 0.7 then that particular object can be deleted.

CHAPTER 7

RESULT

The proposed method has been applied on various test videos and results are satisfactory. The algorithm estimate the foreground using background subtraction and tracked using Kalman filter with matching of color, edge, texture and FAST features. The proposed method is able to track the object in medium and bright intensity video. The tracker is able to track the multiple objects and able to deal with partial and full occlusion. The system is also able to justify the scale and rotation variation of the object in the video frame.

7.1 Tracking Results with Various Videos

Test Video 1

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Fig 7.1 Tracking scenario of objects in hostel compound (a) detection of objects in video frame, (b) Prediction of 2nd object, (c) Disappearing of 1st object, (d) 1st object is out of frame and its prediction, (e) Reappearance of 1st object, (f) Object is closer to the camera, (g) Change in appearance of object, (h) Object comes to normal position.

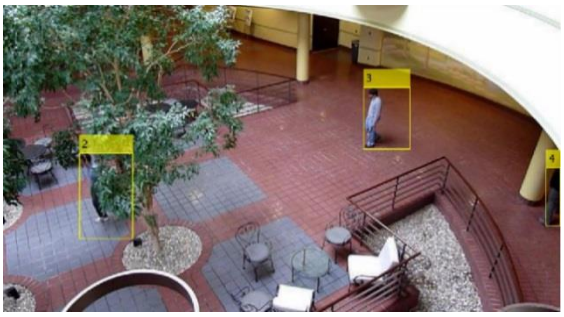
Test Video 2

(a)

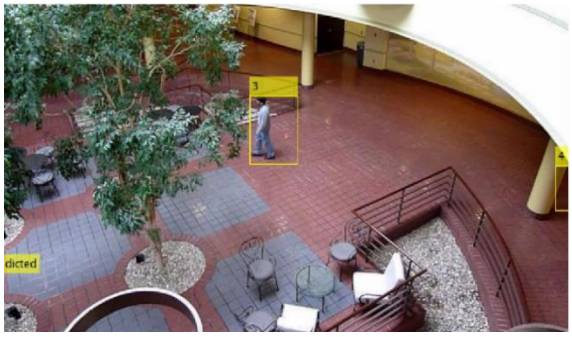
(b)

(c)

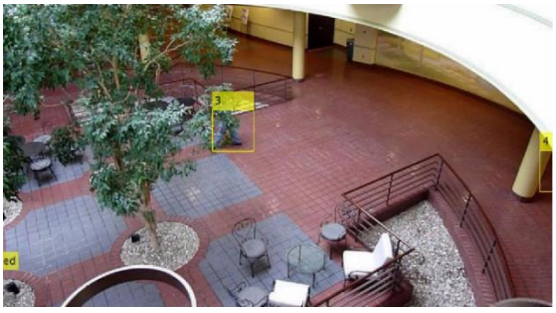
(d)



(e)



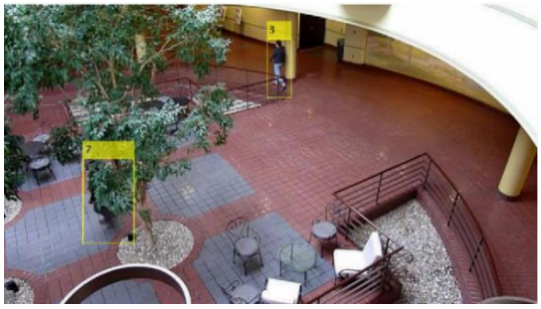
(f)



(g)

(h)

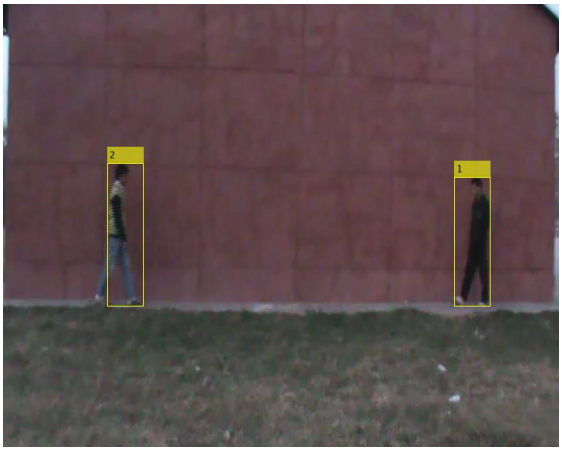
(i)



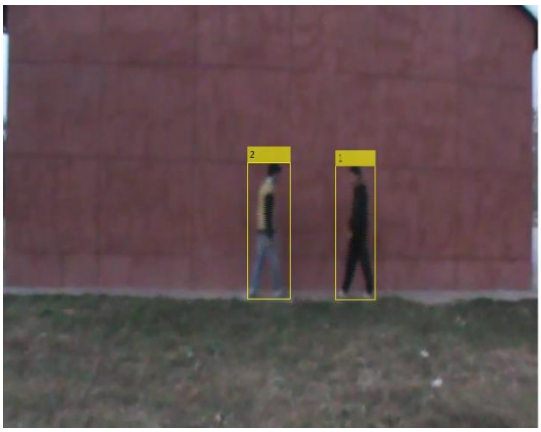
(j)

Fig 7.2 Tracking Scenario of multiple objects in good lightening condition (a) Object no. 2 is in the frame and another object is appearing, (b) object 3 also detected after continuous visible of 8 frames, (c) Object 2 is being partially occluded, (d) object 2 is heavily occluded while another object is appearing in frame, (e) Object 2 reappears and object 4 is confirmed, (f) object 2 disappears and system predicts the position, (g-j) few objects are partially occluded and few are visible

Test Video 3



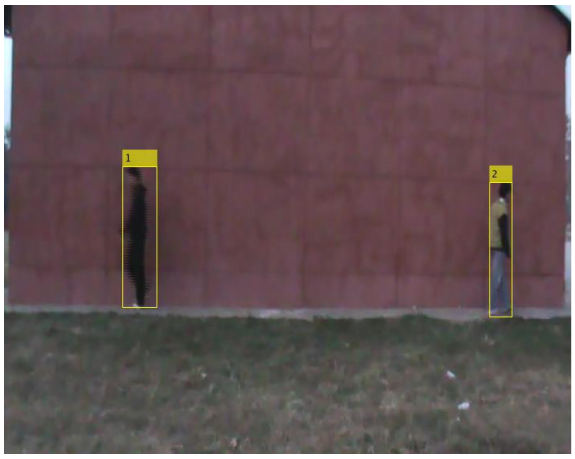
(a)



(b)

(c)

(d)



(e)

Fig 7.3 Tracking Scenario of two objects with occlusion, (a-b) Object 1 and 2 are moving towards each other, (c) Objects are occluded, (d-e) objects are parted away and moving

7.2 Conclusion and Future Work

In this work, a tracking method is proposed for visual object tracking. The tracker uses the visual information of color histogram and edge histogram along with texture and FAST Features at t-1 frame and tries to find the object in frame t. The tracker tracks the objects based on Euclidean distance if they are not occluded so as to save the computation and it performs the matching of Texture and FAST features when the object is partially occluded. In case of full occlusion, the tracker predicts the location of objects based on Kalman filter. The tracker has been applied to

various data which includes the scenario of single and multiple objects with or without the occlusion.

However this method has certain limitations: First, the camera system should be static as the system learns the background using GMM and extraction of object works on the principal of background subtraction so the system should be static. Moreover, Kalman filter cannot predict the sudden change in object direction and speed. Also, the method is not able to deal with shadows of the object and while subtraction of background they become the part of the object.

I conclude the dissertation by providing some directions for the future work. To deal efficiently with occlusion, multi camera system can be used. Stereo camera system can be helpful to the system by providing the depth detail of object in image. To predict the sudden change in object speed or direction, particle filter can be used instead of Kalman filter. By using adaptive GMM model, the system can be used for moving camera instead of static system.

REFERENCES

[1] J. Wang, G. Bebis, and R. Miller, “Robust video-based surveillance by integrating target detection with tracking,” in Proc. Conf. CVPRW OTCBVS, Jun. 2006, pp. 137–145.

[2] G. R. Bradski, “Computer vision face tracking for use in a perceptual user interface,” in Proc. IEEE Workshop Appl. Comput. Vision, Oct. 1998, pp. 214–219.

[3] H. Luo, S. Ci, D. Wu, N. Stergiou, and K. Siu, “A remote marker less human gait tracking for e-healthcare based on content-aware wireless multimedia communications,” IEEE Wireless Commun., vol. 17, no. 1, pp. 44–50, Feb. 2010.

[4] N. Papanikolopoulos, P. Khosla, and T. Kanade, “Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision,” IEEE Trans. Robot. Autom., vol. 9, no. 1, pp. 14–35, Feb. 1993.

[5] C. Yang, R. Duraiswami, and L. Davis, “Efficient mean-shift tracking via a new similarity measure ,” in Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recogn., vol. 1, Jun. 2005, pp. 176–183

[6] L. Fan, M. Riihimaki, and I. Kunttu, “A feature-based object tracking approach for realtime image processing on mobile devices,” in Proc. 17th IEEE ICIP, Sep. 2010, pp. 3921–3924.

[7] J. Wang, G. Bebis, and R. Miller, “Robust video-based surveillance by integrating target detection with tracking,” in Proc. Conf. CVPRW OTCBVS, Jun. 2006, pp. 137–145.

[8] N. Papanikolopoulos, P. Khosla, and T. Kanade, “Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision,” IEEE Trans. Robot. Autom., vol. 9, no. 1, pp. 14–35, Feb. 1993.

[9] G. R. Bradski, “Computer vision face tracking for use in a perceptual user interface,” in Proc. IEEE Workshop Appl. Comput. Vision, Oct. 1998, pp. 214–219.

[10] T. Okuma, “A natural feature based 3d object tracking method for wearable augmented reality,” in Proc. AMC, 2004, pp. 451–456.

[11] H. Luo, S. Ci, D. Wu, N. Stergiou, and K. Siu, “A remote marker less human gait tracking for e-healthcare based on content-aware wireless multimedia communications,” IEEE Wireless Commun., vol. 17, no. 1, pp. 44–50, Feb. 2010.

[12] T. Bouwmans, F. El Baf, B. Vachon,"Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey", Recent Patents on Computer Science, 1(3):219-237, 2008.

[13] Stauffer C, Grimson W. “Adaptive background mixture models for real-time tracking”. Proc IEEE Conf on Comp Vision and Patt Recog (CVPR 1999) 1999; 246-252.

[14] O.Zoidi, A.Tefas, I.Pitas, “Visual object tracking based on local steering kernels and color histograms” ,IEEE Trans. Circuits Syst. Video Technol. 23(5) (2013) 870–882.

[15] M. Karaman, L. Goldmann, D. Yu, and T. Sikora, “Comparison of static background segmentation methods,” Vis. Commun. Image Process., vol. 5960, p. 596069, Jul. 2005.

[16] M. Cristani, M. Farenzena, D. Bloisi, and V. Murino, “Background subtraction for automated multisensor surveillance: A comprehensive review,” EURASIP J. Adv. Signal Proc., vol. 2010, no. 1, p. 343057, 2010.

[17] S. Brutzer, B. Hoferlin, and G. Heidemann, “Evaluation of background subtraction techniques for video surveillance,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit., Jun. 2011, pp. 1937–1944.

[18] P. Kaewtrakulpong and R. Bowden, “An improved adaptive background mixture model for real-time tracking with shadow detection,” in Proc. 2nd Eur. Workshop Adv. Video Based Surveillance Syst., Sep. 2001, pp. 1–5.

[19] Z. Zivkovic, “Improved adaptive Gausian mixture model for background subtraction,” in Proc. Int. Conf. Pattern Recognit., 2004, p. 28–31.

[20] Harris, C., Stephens, M., “A combined corner and edge detection”, In: Proceedings of The Fourth Alvey Vision Conference. (1988) 147–151.

[21] Lowe, D.G., “Object recognition from local scale-invariant features”, International Journal of Computer Vision 60 (2004) 91–110.

[22] Bay, H., Tuytelaars, T., Gool, L.V., "Surf: Speeded up robust features", In: European Conference on Computer Vision (ECCV'06). (2006) 404–417.

[23] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in Proceedings of the International Conference on Computer Vision, pp. 1508–1511, 2005.

[24] Edward Rosten and Tom Drummond, "Machine learning for high speed corner detection" in 9th European Conference on Computer Vision, vol. 1, 2006, pp. 430–443.

[25] S. M. Smith and J. M. Brady, "SUSAN — A new approach to low level image processing", International Journal of Computer Vision, vol. 23, no. 34, pp. 45–78, 1997.

[26] J. R. Quinlan, "Induction of decision trees," Machine Learning, vol. 1, pp. 81–106, 1986.

[27] C. Yang, R. Duraiswami, and L. Davis, "Efficient mean-shift tracking via a new similarity measure," in Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recogn., vol. 1, Jun. 2005, pp. 176–183.

[28] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 5, pp. 564–577, May 2003.

[29] S. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance: Adaptive models in particle filters," IEEE Trans. Image Process., vol. 13, no. 11, pp. 1434–1456, Nov. 2004.

[30] C. Yang, R. Duraiswami, and L. Davis, "Fast multiple object tracking via a hierarchical particle filter," in Proc. 10th IEEE ICCV, vol. 1, Oct. 2005, pp. 212–219.

[31] A. Doucet, N. D. Freitas, and N. Gordon, Eds., Sequential Monte Carlo Methods in Practice. Berlin, Germany: Springer, 2001.

[32] Timo and Matti, "Image description using joint distribution of filter bank responses", Pattern Recognition Letters 30 (2009) 368–376

[33] Pradeep and Jayant, "Rotation and intensity invariant shoeprint matching using Gabor transform with application to forensic science", Pattern Recognition 42 (2009) 1308 – 1317.

[34] Xuewen, Xiaoqing and Changsong, "Gabor filters-based feature extraction for character recognition", Pattern Recognition 38 (2005) 369 – 379