An Approach for Object Tracking in Video Sequences

Kalyan Kumar Hati

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, India

An Approach for Object Tracking in Video Sequences

Dissertation submitted in

March 2013

to the department of

Computer Science and Engineering

of

National Institute of Technology Rourkela

in partial fulfillment of the requirements

for the degree of

Master of Technology (Research)

by

Kalyan Kumar Hati

(Roll 610CS101)

under the supervision of

Dr. Pankaj Kumar Sa

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, India

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, India. www.nitrkl.ac.in

Dr. Pankaj Kumar Sa
Assistant Professor

March 27, 2013

Certificate

This is to certify that the work in the thesis entitled *An Approach for Object Tracking in Video Sequences* by *Kalyan Kumar Hati*, bearing roll number 610CS101, is a record of original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of *Master of Technology (Research)* in *Computer Science and Engineering.* Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Pankaj K. Sa

Acknowledgment

This dissertation, though an individual work, has benefited in various ways from several people. Whilst it would be simple to name them all, it would not be easy to thank them enough.

The enthusiastic guidance and support of *Prof. Pankaj Kumar Sa* inspired me to stretch beyond my limits. His profound insight has guided my thinking to improve the final product. My solemnest gratefulness to him.

My humble acknowledgment to *Prof. Banshidhar Majhi* for his constructive criticism during entire span of research.

My sincere thanks to *Prof. S. K. Jena*, *Prof. A. K. Turuk*, *Prof. G. K. Panda*, and *Prof. D. Patra* for their continuous encouragement and invaluable advice.

It is indeed a privilege to be associated with people like *Prof. B. D. Sahoo*, *Prof. R. K. Dash*, and *Prof. K. S. Babu.* They have made available their support in a number of ways.

Overwhelming thanks to all members of the Department of Computer Science and Engineering, NIT Rourkela for their encouragement and co-operations throughout.

Many thanks to my comrades and fellow research colleagues at *Intelligent Computing and Computer Vision Laboratory.* It gives me a sense of happiness to be with you all.

Finally, my heartfelt thanks to my family for their unconditional love and support. Words fail me to express my gratitude to my beloved parents, who sacrificed their comfort for my betterment.

Kalyan Kumar Hati

Abstract

In recent past there has been a significant increase in number of applications effectively utilizing digital videos because of less costly but superior devices. This upsurge in video acquisition has led to huge augmentation of data, which are quite impossible to handle manually. Therefore, an automated means of processing these videos is indispensable. In this thesis one such attempt has been made to track objects in videos. Object tracking comprises two closely related processes; object detection followed by tracking of the detected objects. Algorithms on these two processes are proposed in this thesis.

Simple object detection algorithms compare a static background frame at pixel level with the current frame in a video. Existing methods in this domain first try to detect objects and then remove shadows associated with them, which is a two-stage process. The proposed approach combines both the stages into a single stage. Two different algorithms are proposed on object detection. First one to model the background and the next to extract the objects and remove shadows from them. Initially, from first few frames the nature of each pixel is determined as stationary or non-stationary and considering only the stationary pixels a background model is developed. Subsequently, a local thresholding technique is used to extract objects and discard shadows.

After successfully detecting all the foreground objects, two different algorithms are proposed for tracking the objects and updating the background model. The first algorithm suggests a centroid searching technique, where a centroid in current frame is estimated from the previous frame. Its accuracy is verified by comparing the entropy of dual-tree complex wavelet coefficients in the bounding boxes of both the frames. If estimation becomes inaccurate, a dynamic window is utilized to search for accurate centroid. The second algorithm updates the background using a randomized updating scheme.

Both stages of the proposed tracking model is simulated with various recorded videos. Simulation results are compared with the recent schemes to show the superiority of the model.

***Keywords*:** Vision and scene understanding, background modeling, background subtraction, dual-tree complex wavelet transform, Shannon entropy, object kinematics.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Humans are the most blessed creatures in the universe. The presence of all the five sensory organs distinguishes them from other living beings. The sight sensory organ helps them in receiving visual information. This visual information, otherwise known as *scene* can be captured as an image by a camera and stored for future use. A single image is inadequate enough to represent a scene with motion information. Such scenes are recorded by capturing a sequence of images at regular intervals. Each image of the sequence is known as *frame*. When successive frames are projected with the progress of time, we call it as *video*. Projection of successive frames at a particular rate creates an illusion, which convey a sense of motion in the scene.

Digital video processing refers to processing of video by a digital computer [1]. In the memory of a digital computer, video storage can be viewed as stacking of frames along the time axis (t) with spatial information of each frame being represented by the (x, y) dimension. Figure 1.1 depicts a pictorial illustration of the same.

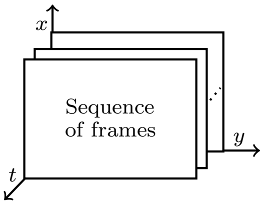


Figure 1.1: Representation of video in memory of a digital computer.

Mathematically each frame is a matrix of order $h \times w$, and the t^{th} frame may be expressed as —

$$f(x,y,t) = \begin{bmatrix} f(0,0,t) & f(0,1,t) & \cdots & f(0,w-1,t) \\ f(1,0,t) & f(1,1,t) & \cdots & f(1,w-1,t) \\ \vdots & \vdots & \ddots & \vdots \\ f(h-1,0,t) & f(h-1,1,t) & \cdots & f(h-1,w-1,t) \end{bmatrix} \tag{1.1}$$

where h and w refer to the height and width of the frame respectively. The intensity or gray level at pixel location (x,y) at projection t is denoted by (x,y,t).

Some of the subareas of digital video processing are listed as —

(i) Frame-rate conversion

(ii) Super-resolution

(iii) Restoration and noise reduction

(iv) Segmentation

(v) Watermarking

Among the above subareas, investigation in this thesis has been confined to video segmentation.

The remainder of this chapter is organized as follows. Section 1.1 starts with an introduction to video segmentation followed by some of its applications. A concise review on various approaches adopted on two applications are also outlined in this section. The generic tracking model and its drawbacks are presented in Section 1.2. Research goals are discussed in Section 1.3 followed by the proposed tracking model in Section 1.4. Finally, the Section 1.5 outlines the layout of the thesis.

1.1 Video Segmentation

Video segmentation or layer extraction, is a classical problem in computer vision that involves the extraction of foreground objects from a set of images. In image segmentation the goal is to segment an image into spatially coherent regions, whereas in video segmentation frames are segmented into temporally coherent regions.
Some of the practical applications of video segmentation are —

(i) Indexing,

(ii) Compression,

(iii) Object detection,

(iv) Object tracking, and

(v) Shot boundary detection.

In this thesis we have concentrated on object detection and object tracking.

1.1.1 Object Detection

Object detection involves locating object in the frames of a video sequence when it first appears in the video [2]. A common approach is to use information from single frame. However, some detection methods make use of the temporal information computed from a sequence of frames to reduce the number of false detection. This temporal information is usually in the form of frame differencing, which highlights changing regions in consecutive frames. Given the object regions in the frame, it is then the tracker's task to perform object correspondence from one frame to the next to generate the tracks.

Review on Detection Algorithms

Researchers have contributed several detection algorithms using various approaches. Algorithms reported in literature may broadly be categorized into the following four groups based on the approaches used.

(i) **Point detectors:** Point detectors are used to find interest points in frames which have an expressive texture in their respective localities. Interest points have been long used in the context of motion, stereo, and tracking problems. A desirable quality of an interest point is its invariance to changes in illumination and camera viewpoint. In the literature, commonly used point detectors include Moravec's interest operator [3], Harris interest point detector [4], KLT detector [5], and SIFT detector [6].

(ii) **Segmentation:** Segmentation algorithms tries to segment the frame into perceptually similar regions. Some of the segmentation method used are —

  (a) ***Mean-Shift clustering:*** Comaniciu and Meer have proposed Mean-Shift clustering which tries to find clusters in the joint spatial and color space (l, u, v, x, y), where (l, u, v) represents the color and (x, y) represents the spatial location [7].

  (b) ***Graph-Cuts algorithm:*** Segmentation problem can be formulated as a graph partitioning problem, where the vertices (pixels) $V = \{u, v, \cdots\}$, of a graph (frame) G, are partitioned into N disjoint subgraphs (regions), by pruning the weighted edges of the graph. The total weight of the pruned edges between two subgraphs is called a *cut*. The weight is typically computed by color, brightness, or texture similarity between the nodes.

  (c) ***Active contours:*** In an active contour framework, object segmentation is achieved by evolving a closed contour to the object's boundary, such that the contour tightly encloses the object region.

(iii) **Background subtraction:** Object detection can also be achieved by background subtraction technique. The basic principle is to compare a static background frame with the current frame of the video pixel by pixel. This technique involves building a model of the background and any frame then can be compared with the model to detect zones where a significant difference occurs. The above process is called as background subtraction [2].

(iv) **Supervised classifiers:** Object detection can be performed by learning different object views automatically from a set of examples by means of a supervised learning mechanism. Learning of different object views waives the requirement of storing a complete set of templates. Given a set of learning examples, supervised learning methods generate a function that maps inputs to desired outputs.

In this dissertation object detection is achieved using background subtraction approach.

1.1.2 Object Tracking

Object tracking determines the motion of the projection of one or more objects in the image plane. This motion is induced by the relative motion between the camera and the observed scene. It is literally defined as, “Locating a moving object or multiple objects over a period of time using camera” and technically as, “Problem of estimating the trajectory or path of an object in the image plane as it moves around a scene.” Object tracking can be applied in many areas like automated surveillance, traffic monitoring, human computer interaction etc. Challenges in the same area include noise in frames, complex object motion and shape, occlusion, change in illumination etc.

Review on Tracking Algorithms

Methods for object tracking can be classified into following four categories according to the tools used during tracking.

(i) **Region-based methods:** These methods provide an efficient way to interpret and analyze motion in a video sequence. An image region can be defined as a set of pixels having homogeneous characteristics. It can be derived by image segmentation, which can be based on distinctive object features like color, edges etc. Essentially, a region would be the image area covered by the projection of the object of interest onto the image plane. Alternatively, a region can be the bounding box of the projected object under examination.

(ii) **Contour-based methods:** An alternative way of devising an object tracking algorithm is by representing the object using outline contour information and tracking it over time, thus retrieving both its position and shape. Such a modeling method is more complicated than modeling entire regions. However, contour-based tracking are usually more robust than region-based object tracking algorithms, because it can be adapted to cope with partial occlusions. Also the outline information is insensitive to illumination variations.

(iii) **Feature point-based methods:** Feature point-based object tracking can be defined as, the attempt to recover the motion parameters of a feature point in a video sequence. More formally, let $f = \{f_0, f_1, \cdots, f_N\}$ denotes the N frames of a video sequence and $p_i\left(x_i, y_i\right), i = 0, 1, \cdots, N$ denote the positions of the same feature point in those frames. The task at hand is to determine a motion vector $d_i\left(d_{x,i}, d_{y,i}\right)$ that best determines the position of the feature point in the next frame, $m_{i+1}\left(x_{i+1}, y_{i+1}\right)$, that is: $m_{i+1} = m_i + d_i$. The object to be tracked is usually defined by the bounding box or the convex hull of the tracked feature points.

(iv) **Template-based methods:** Template-matching techniques are used by many researchers to perform object tracking. Template-based tracking is closely related to region-based tracking because a template is essentially a model of the image region to be tracked. These methods involve two steps for tracking, initialization step followed by matching step. In the first step template can be initialized by various on-line and off-line methods. During matching, it involves the process of searching the target image to determine the image region that resembles the template, based on a similarity or distance measure.

In present contribution object tracking is achieved using feature point-based method.

1.2 Generic Tracking Model

While recording a movie by a camera one need to consider camera position and scene dynamics. Camera position can be fixed or variable. Like wise scene can also be static or dynamic. Considering all four aspects, a movie can be captured in either of the four situations mentioned below,

(i) *Fixed* camera position and *static* background,

(ii) *Fixed* camera position and *non static* background,

(iii) *Variable* camera position and *static* background, and

(iv) *Variable* camera position and *non static* background.

In this work videos are obtained using fixed camera position and static background scene.

Video obtained from a static camera and a fixed background gives a clue for the object detection by background subtraction technique. In this approach, initially a background is modeled using the first frame or a combination of the first few frames of the video. Any frame of the video then can be compared pixel by pixel with the model developed to extract foreground objects. Shadows being an integral part of the scene are very often detected as foreground objects. Shadow suppression methods are employed to suppress the shadows. The detected objects are tracked in subsequent frames using any categories of algorithm presented in Section 1.1.2. In order to accommodate changes in the background scene, model developed is monitored and updated in due course of time. The entire model is depicted in Figure 1.2.

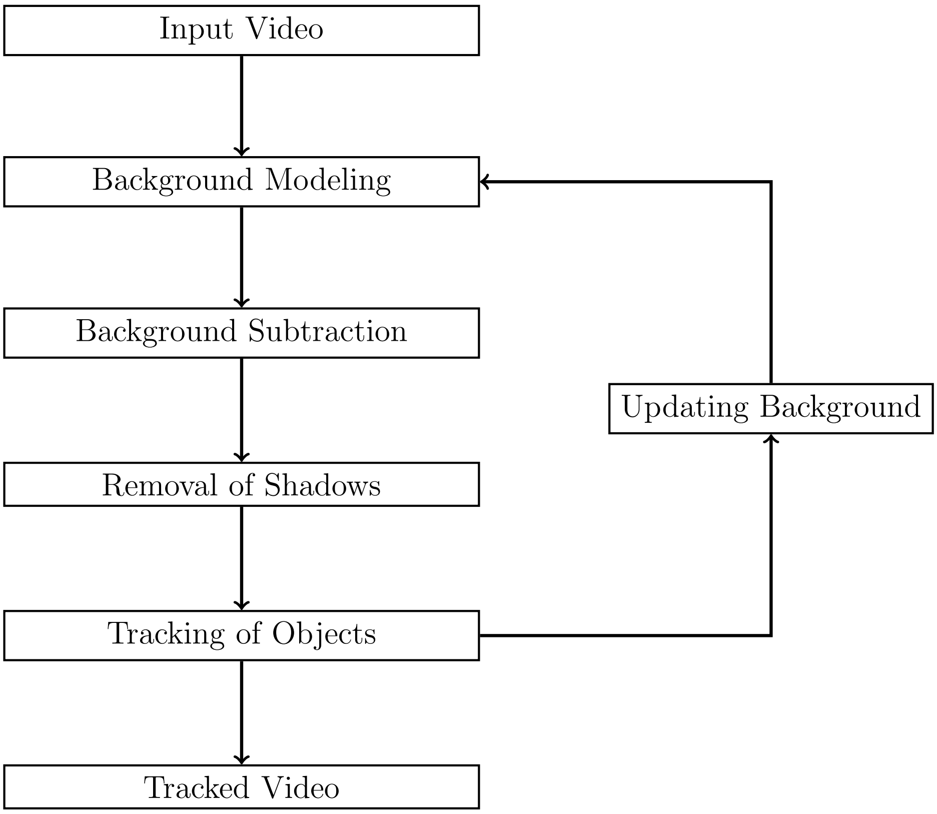


Figure 1.2: The Generic Tracking Model.

The generic tracking model may deliver miserable performance under the following situations.

(i) Waving of leaves,

(ii) Scene illumination variations,

(iii) Uneven lighting,

(iv) Use of global thresholding for object detection,

(v) Shadows identified as object,

(vi) An additional step required to remove shadows,

(vii) Complex object motion and shapes,

(viii) Shift variant feature used during tracking,

(ix) Computationally inefficient centroid searching technique, and

(x) Updating background model frequently.

1.3 Research Goal

Considering the drawbacks of the generic tracking model, our research goals are framed as —

1. Development of a background model, which alleviates the problem of waving of leaves, scene illumination variations, and uneven lighting.
2. Formulation of a thresholding technique, that discards shadows while detecting objects and thereby eliminating the shadow removal step of the generic tracking model.
3. Use of a feature that is invariant to complex object motion and shape.
4. Development of a computationally efficient centroid searching technique.
5. Designing of an effective background updating scheme that updates the minimum information in the model to accommodate maximum changes.

1.4 Proposed Tracking Model

The proposed tracking model takes few initial frames to model the background. The foreground objects can be detected in any subsequent frame by comparing it with the developed model. The proposed model is capable enough to handle any shadow associated with the object without the help of any additional shadow removal step. The detected objects are then tracked in the subsequent frames using their features. In order to make the background model adaptable to changes occurring in the scene, we update the background model in due time. The proposed tracking model is depicted in Figure 1.3.

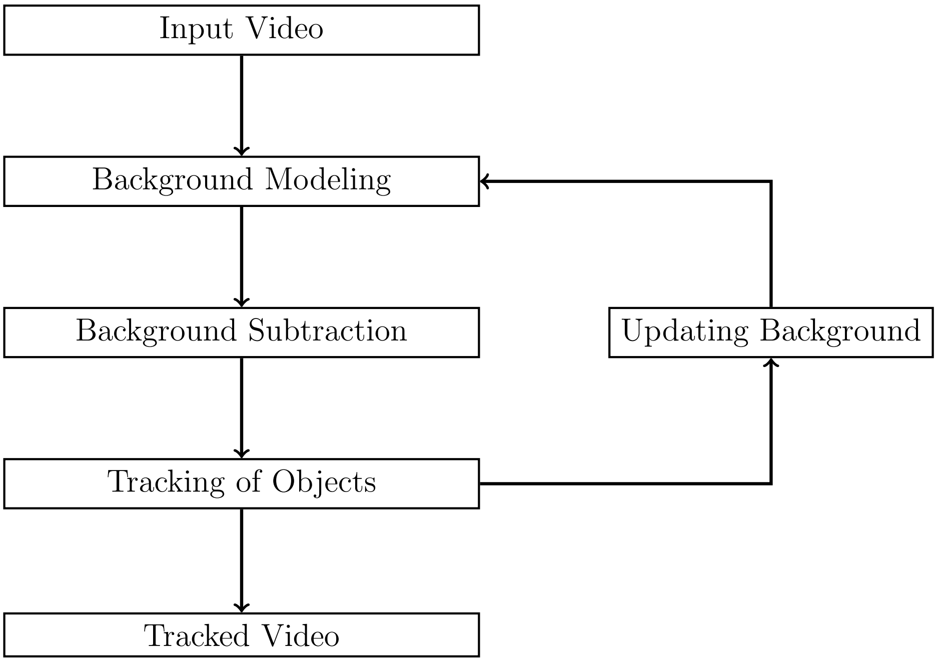


Figure 1.3: The Proposed Tracking Model.

Background Modeling

In background modeling few initial frames are considered for the development of background model. Pixels in these frames are classified as stationary or non-stationary by analyzing their deviations from the mean. The background is then modeled taking all the stationary pixels into account. Background model thus developed defines a range of values for each background pixel location.

Background Subtraction

A local thresholding based background subtraction is used to find the foreground object. Two local threshold namely, local lower threshold and local upper threshold are defined for each background pixel considering the pixel range obtained in modeling step. These local thresholds help in successful detection of objects suppressing shadows. The increase and decrease in the intensity level of the background pixels is taken care by upper and lower part of the predefined intensity range respectively.

Object Tracking

Detected objects are tracked in subsequent frames of the video using two parameters derived from each object. The first parameter is velocity and the second parameter is entropy of dual-tree complex wavelet transform. Dual-tree complex wavelet transform is applied on the detected foreground objects. There after entropy of the resultant coefficients are calculated. Considering velocity, centroid, and entropy of an object in segmented frames, object centroid is calculated using Euclidean distance for subsequent frames.

Updating Background

In order to accommodate changes in the background scene and suppress ghost an *object's zero velocity* concept has been introduced. Moreover, the background subtraction is performed in a random interval of time.

1.5 Thesis Layout

One algorithm is proposed for each of the above four steps of object tracking model. These algorithms are organized in two separate chapters. An outline of the thesis is as follows —

Chapter 2: Intensity Range based Background Subtraction for Object Detection In this chapter, an object detection scheme is presented. It produces objects without any shadow and has the capability to eliminate the shadow removal step of object tracking. The scheme suggests two different algorithms, the first one to model the background from initial few frames and the second one to extract the objects based on local thresholding. The strength of the scheme lies in the fact that it accommodates illumination variation as well as motion variation in background.

Chapter 3: Object Tracking Using Dual-Tree Complex Wavelet Transform Object tracking step of our tracking model is presented in this chapter. Proposed approach tracks objects in subsequent frames of the video using object's velocity and entropy of the object's dual-tree complex wavelet transform coefficient's. Object centroid in subsequent frames are then calculated using Euclidean distance. A background updating algorithm is also included in this chapter to update background model.

Chapter 4: Conclusion This chapter provides the concluding remarks with a stress on achievements and limitations of the proposed schemes. The scopes for further research are outlined at the end.

The contributions made in each chapter are discussed in sequel, which include proposed schemes, their simulation results, and comparative analysis.

Chapter 2

Intensity Range based Background Subtraction for Object Detection

Object detection deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. It has applications in many areas of computer vision, including image retrieval, pose estimation, and video surveillance etc. Object detection, in videos obtained from static camera and fixed background, is achieved through *background subtraction* technique. In this approach moving objects in a scene can be obtained by comparing any frame of the video with the model of the background [2].

In most of the suggested schemes, the object detected is accompanied with misclassified foreground objects due to illumination variation or motion in the background. In many cases, shadows are falsely detected as foreground objects during object extraction. Presently, an additional step is needed to remove these misclassified objects and shadows for effective object detection. To alleviate these problems, we propose a simple but effective object detection technique, which is invariant to change in illumination and motion in the background. The proposed approach also neutralizes the presence of shadows in detected objects.

The suggested background model initially determines the nature of each pixel as stationary or non-stationary and considers only the stationary pixels for background model formation. In the background model, for each pixel location a range of values are defined. Subsequently, in object extraction phase our scheme employs a local threshold, unlike the use of global threshold in conventional schemes.

Rest of this chapter is organized as follows. Some of the related work are reviewed in Section 2.1. The next two sections propose two algorithms in sequel; *Background Modeling* in Section 2.2 and *Background Subtraction* in Section 2.3. Simulation results are discussed in Section 2.4. Finally, Section 2.5 summarizes the chapter.

2.1 Related Work

Initially, filters were used for background modeling and subtraction. One such method is described by Koller *et al.*, which addresses the problem of multiple car tracking with occlusion reasoning [8]. They have employed a contour tracker, based on intensity and motion of boundaries. In order to achieve this, they have used linear Kalman filter in two ways, one for estimating the motion parameters and another for estimating the shape of the contour of the car. Maintenance of background model being an important aspects of background modeling and subtraction, Toyama *et al.* developed a three component system for background maintenance namely, pixel level component, region-level component, and frame-level component [9]. The first component performs Wiener filtering to make probabilistic predictions of the expected background. The second component fills in homogeneous regions of foreground objects. Finally, the third component detects abrupt and global changes.

Wren *et al.* have proposed to model the background independently at each pixel location (i, j) [10]. The model is based on computation of Gaussian probability density function (pdf) on the last n pixel values. In order to avoid the pdf calculation from beginning at each new frame, a running average at time t is computed as follows,

$$\mu_t = \alpha I_t + (1 - \alpha)\mu_{t-1} \tag{2.1}$$

where I_t is the pixel's present value, μ_{t-1} is the previous average, and α is an empirical weight. The other parameter of the Gaussian probability density function, the standard deviation σ_t, can be computed similarly. In addition to speed, the advantage of the running average is given by the low memory requirement for each pixel. Here each pixel consists of two parameters (μ_t, σ_t) instead of the buffer with the last n pixels values. At each t frame time, the I_t, pixel's value can then be classified as a foreground pixel if the inequality in following equation holds;

$$|I_t - \mu_t| > k\sigma_t \tag{2.2}$$

Koller *et al.*, in their work [11] have identified that the equation (2.1) of [10] is more often updated and therefore modified the model as —

$$\mu_t = M\mu_t + (1 - M)(\alpha I_t + (1 - \alpha)\mu_{t-1}) \tag{2.3}$$

where the binary value M is 1 in correspondence of a foreground value, and 0 otherwise.

Lo and Velastin proposed to use median value of the last n frames as the background model [12]. Cucchiara *et al.* corroborated that such a median value provides an adequate background model even though the n frames are subsampled with respect to the original frame rate by a factor of 10 [13]. The main disadvantage of a median-based approach is that, its computation requires a memory with the recent pixels values.

Stauffer and Grimson developed a complex procedure to accommodate permanent changes in the background scene [14]. The procedure is named as Mixture of Gaussian. Here each pixel is modeled separately by a mixture of K Gaussian,

$$P(I_t) = \sum_{i=1}^{K} \omega_{i,t} \times \mathcal{N}\left(I_t; \mu_{i,t}, \Sigma_{i,t}\right) \tag{2.4}$$

where $K \in [3, 5]$.

Elgammal *et al.* proposed to model the background distribution by a non-parametric model based on Kernel Density Estimation (KDE) on the buffer of the last n background values [15]. According to [16] KDE guarantees a smooth, continuous version of the histogram. In [15] the background pdf is given as a sum of Gaussian kernels centered in the most recent n background values, x_i —

$$P(x_t) = \frac{1}{n}\sum_{i=1}^{n}(x_t - x_i, \Sigma_t) \tag{2.5}$$

The method described by Seki *et al.* is based on the assumption that, neighboring blocks of background pixels should follow similar variations over time [17]. While this assumption holds most of the time especially for pixels belonging to the same background object, it becomes problematic for neighboring pixels located at the border of multiple background objects.

Few samples are collected over time and used to train a principal component analysis (PCA) model. A block of a new video frame is classified as background if the observed image pattern is close to its reconstructions using PCA projection coefficients of eight-neighbouring blocks. Such a technique is also described by Power and Schoonees, but it lacks an update mechanism to adapt the block models over time [18]. Oliver *et al.* focused on the PCA reconstruction error [19]. A similar approach, the independent component analysis (ICA) of serialized images from a training sequence, is described by Tsai and Lai for training of an ICA model [20]. The resulting demixing vector is then computed and compared to that of a new image in order to separate the foreground from a reference background image. The method is said to be highly robust to indoor illumination changes.

A two-level mechanism based on a classifier was introduced by Lin *et al.* [21]. This classifier first determines whether an image block belongs to the background or foreground. Appropriate block wise updates of the background image are then carried out in the second stage, depending upon the results of the classification. The scheme proposed by Maddalena and Petrosino also works on the basis of classification, where the background model learns its motion patterns by self organization through artificial neural networks [22].

The W4 model presented by Haritaoglu *et al.* is a simple and effective method [23]. It uses three values to represent each pixel in the background image: the minimum and maximum intensity values, and the maximum intensity difference between consecutive images of the training sequence. Gutchess *et al.* proposed a background model in which multiple hypotheses of the background value at each pixel were generated by locating periods of stable intensity in the sequence [24]. The likelihood of each hypothesis is then evaluated using optical flow information from the neighbourhood around the pixel, and the most likely hypothesis is chosen to represent the background. Jacques *et al.* brought a small improvement to the W4 model together with the incorporation of a technique for shadow detection and removal [25]. C.R. Jung proposed a new background subtraction algorithm with shadow identification [26]. In the training stage, robust estimators are used to model the background, and a fast test is used to detect foreground pixels in the evaluation stage. A statistical model is combined with expected geometrical properties for shadow identification and removal. Finally, morphological operators are applied to remove isolated foreground pixels.

Barnich and Droogenbroeck proposed a universal background subtraction algorithm called ViBe for video sequences [27]. In ViBe, each pixel in the background can take values from its preceding frames in same location or its neighbor. Then it compares this set to the current pixel value in order to determine whether that pixel belongs to the background, and adapts the model by choosing randomly which values to substitute from the background model. Kim and Kim introduced a novel background subtraction algorithm for temporally dynamic texture scenes [28]. The scheme adopts a clustering-based feature, called fuzzy color histogram (FCH), which has an ability of greatly attenuating color variations generated by background motions while still highlighting moving objects. Instead of segmenting a frame pixel-by-pixel, Reddy *et al.* used an overlapping block-by-block approach for detection of foreground objects [29]. The scheme passes the texture information from each block through three cascading classifiers to classify them as background or foreground. The results are then integrated with a probabilistic voting scheme at pixel level for final segmentation. This scheme is very effective due to the presence of three different classifiers.

From the existing literature, it is observed that most of the schemes perform three operations in sequel namely, background modeling, foreground object extraction, and finally removal of misclassified objects and shadow from the detected objects. Further, due to the use of global threshold in object detection, the complexity is more. Moreover, it is observed that most of the simple schemes are ineffective on videos with illumination variations, motion in background, and dynamically textured indoor and outdoor environment etc. On the other hand, such videos are well handled by complex schemes with higher computational cost. Keeping this in mind, we suggest here an intensity range based object detection scheme which models the background considering a set of initial frames of the sequence followed by a local thresholding approach for object extraction. Simulation has been carried out on standard videos and comparative analysis has been performed with competent schemes.

2.2 Background Modeling

The proposed detection scheme consists of two stages. The first stage deals with developing background model. This stage consists of two steps. First step is *background model intilization.* This step tries to classify each pixel as stationary or non-stationary in the frames required for background modeling. Next step of this stage is *development of background model.* Here a background model is developed considering stationary information of the pixel. In the second stage a local threshold based background subtraction method tries to find the objects by comparing any frame with the established background. Proposed scheme uses two parameters namely, window size W (an odd length window) and a constant C for its computation. The optimal values are selected experimentally. The stages and the parameter selection process of proposed scheme are described below in sequel.

2.2.1 Background Model Intilization

Conventionally, the first frame or the combination of first few frames is considered as the background model. However, this model is susceptible to illumination variation, uneven lighting etc., and also to small changes in the background like waving of leaves. A number of solutions to such problems are reported, where the background model is frequently updated at higher computational cost and thereby making them unsuitable for real time deployment. In the proposed scheme few initial frames are considered for background modeling. Pixels in these frames are classified as stationary or non-stationary by analyzing their deviations from the mean.

Background model initilization algorithm starts with consideration of n initial frames as $\{f_1, f_2, \cdots, f_n\}$, where $20 \leq n \leq 30$. From any pixel location (i, j) in all n initial frames, elements are collected and put into an vector $\overrightarrow{U}$. A window of size $W < n$ is slide from $U(1)$ to $U(n)$. Let $\overrightarrow{V}$ be a vector of dimension W. In each pass following operations are performed—

1. $\sigma \leftarrow$ standard deviation of $\overrightarrow{V}$
2. $\mathbf{D}(p) \leftarrow |V(\lfloor W \div 2 \rfloor) - V(p)|$,
   for each value of $p = 0, \cdots, (W-1)$ and $p \neq \lfloor W \div 2 \rfloor$.
3. $\mathbf{S} \leftarrow$ sum of least $\lfloor W \div 2 \rfloor$ magnitudes of $\overrightarrow{\mathbf{D}}$
4. **If** $\mathbf{S} \leq \lfloor W \div 2 \rfloor \times \sigma$ **is true**
   $V(\lceil W \div 2 \rceil)$ `is stationary`
   else
   $V(\lceil W \div 2 \rceil)$ `is non stationary`.

After all elements of $\overrightarrow{U}$ are traversed, the pixels from $U(\lceil W \div 2 \rceil)$ to $U(n - (\lfloor W \div 2 \rfloor))$ are labelled as either stationary or non-stationary. The entire process followed at pixel location (i, j) is repeated for all pixel locations in the frame. Finally, frames $f_{\lceil W \div 2 \rceil}$ to $f_{n - \lfloor W \div 2 \rfloor}$ will have pixels classified as either stationary or non-stationary.

From the above description it can be inferred that, for background model initialization, initial frames are required as input and at the end of the process pixels in these frames are classified as stationary or non-stationary as output. The steps of the *background model initilization* algorithm are presented in *Algorithm* 1.

```
input : Initial frames from input video
output: Frames having pixels classified as stationary or non-stationary
Consider $n$ initial frames as $\{f_1, f_2, \cdots, f_n\}$, where $20 \leq n \leq 30$.
for $k \leftarrow 1$ to $n - (W - 1)$ do
    for $i \leftarrow 1$ to height of frame do
        for $j \leftarrow 1$ to width of frame do
            $\overrightarrow{V} \leftarrow \left[f_k(i,j), f_{k+1}(i,j), \ldots, f_{k+(W-1)}(i,j)\right]$
            $\sigma \leftarrow$ standard deviation of $\overrightarrow{V}$
            $\mathbf{D}(p) \leftarrow |V\left(k + (\lfloor W \div 2 \rfloor)\right) - V\left(p\right)|$, for each value of
            $p = k + l$, where $l = 0, \cdots, (W - 1)$ and $l \neq \lfloor W \div 2 \rfloor$
            $\mathbf{S} \leftarrow$ sum of least $\lfloor W \div 2 \rfloor$ magnitudes of $\overrightarrow{\mathbf{D}}$
            if $\mathbf{S} \leq \lfloor W \div 2 \rfloor \times \sigma$ then
            Label $f_{k+(\lfloor W \div 2 \rfloor)}(i,j)$ as stationary
            else
            Label $f_{k+(\lfloor W \div 2 \rfloor)}(i,j)$ as non stationary
        end
    end
end
```

Algorithm 1: Background model initilization

2.2.2 Development of Background Model

The background is then modeled taking all the stationary pixels into account. The developed background model defines a range of values for each background pixel location around its true intensity.

In *development of background model* algorithm, stationary pixels at any pixel location (i, j) in the frames form $f_{\lceil W \div 2 \rceil}$ to $f_{n - \lfloor W \div 2 \rfloor}$ are put into a vector $\vec{\mathcal{R}}$. Minimum and maximum value from it are determined and kept in two two-dimensional vector $M(i, j)$ and $N(i, j)$ respectively. The entire process is repeated for each pixel location in the frame. Finally, $M(i, j)$ and $N(i, j)$ will contain the minimum and maximum value of the stationary pixels from frames produced as output of *Algorithm* 1 at respective pixel location (i, j). $M(i, j)$ and $N(i, j)$ represent the background model, defining a range of values for each background pixel location.

From the above description it can be concluded that, for development of background model, frames having pixels as stationary or non-stationary are taken as input and at the end of the process min and max frames are produced in the form of background model as output. The steps of the *development of background model* algorithm are presented in *Algorithm* 2.

```
input  : Frames having pixels as stationary or non-stationary
output: Background model consisting of min and max frame
for $i \leftarrow 1$ to height of frame do
    for $j \leftarrow 1$ to width of frame do
        $M(i, j) = \min [f_s(i, j)]$ and $N(i, j) = \max [f_s(i, j)]$, where
        $s = \lceil W \div 2 \rceil, \cdots, n - (\lfloor W \div 2 \rfloor)$ and $f_s(i, j)$ is stationary
    end
end
```

Algorithm 2: Development of background model

2.3 Background Subtraction

After successfully developing the background model, a local thresholding based background subtraction is used to find the foreground object. A constant C is considered that helps to calculate the local lower threshold $\mathbf{T_L}$ and the local upper threshold $\mathbf{T_U}$. These local thresholds help in successful detection of objects, removal of misclassified objects, and suppressing shadows if any.

Background subtraction algorithm takes the developed background model and a frame f as its input. It produces a segmented frame as its output consisting of foreground object if any with shadow suppressed. Algorithm is repeated for each location in the frame. At each pixel location threshold $\mathbf{T}(i,j)$ is calculated as $\mathbf{T}(i,j) = \frac{1}{C}[M(i,j) + N(i,j)]$ where, C is a constant. Considering $\mathbf{T}(i,j)$ local thresholds are calculated as —

- Local lower threshold: $\mathbf{T_L}(i,j) = M(i,j) - \mathbf{T}(i,j)$
- Local upper threshold: $\mathbf{T_U}(i,j) = N(i,j) + \mathbf{T}(i,j)$

If $f(i,j)$ value lies in between $\mathbf{T_L}$ and $\mathbf{T_U}$, then it is a *background pixel* else a *foreground pixel.* The steps of the *background subtraction* algorithm are outlined in *Algorithm* 3.

2.4 Results

To show the efficacy of proposed detection scheme, simulation has been carried out on different recorded video sequences. Different video sequences used are —

(i) **Single Man Indoor (SMI):** This video was captured inside a hall where a person walks into the center of the scene, gives few poses and walks out. The sequence was recorded with only one fluorescent lamp switched on, which was not sufficient enough to light the entire hall and thereby ensuring the illumination variation. This video also has the property of pose variations. This scenario presents single man tracking in an indoor environment.

(ii) **Single Man Outdoor (SMO):** This movie was recorded outdoor in a partly cloudy day. A person walks from one end of the scene to another end. This scenario illustrates single man tracking in an outdoor environment.

```
input  : Background model and a frame $f$
output: Detected objects in frame $f$
for $i \leftarrow 1$ to height of frame do
    for $j \leftarrow 1$ to width of frame do
        Threshold $\mathbf{T}(i,j) = \frac{1}{C}(M(i,j) + N(i,j))$
        $\mathbf{T_L}(i,j) = M(i,j) - \mathbf{T}(i,j)$
        $\mathbf{T_U}(i,j) = N(i,j) + \mathbf{T}(i,j)$
        if $\mathbf{T_L}(i,j) \leq f(i,j) \leq \mathbf{T_U}(i,j)$ then
        Segmented Frame $\mathbf{S}_f(i,j) = 0$ // Background pixel
        else
        Segmented Frame $\mathbf{S}_f(i,j) = 1$ // Foreground pixel
    end
end
```

Algorithm 3: Background subtraction

(iii) **Multiple Man Outdoor (MMO):** The video was taken in the same environment where SMO video was taken. In this video two persons moves in the scene, resulting a scenario of multiple person tracking.

(iv) **Left Bag (LB):** This video sequence has been chosen from *you tube*. In this sequence a person walks into the scene with a bag in his hand. He leaves his bag and walks back. Again he reappears in the scene empty handed, picks up the bag and moves out of the scene. This scenario presents a situation, where background updating is necessary.

(v) **Hall Monitor (HM):** This sequence is from Center for Image Processing Research (CIPR) unit of Rensselaer Polytechnic Institute, New York, USA.

The above sequences, considering their attributes, are the most suitable candidates for validation of generalized behavior of the proposed scheme.

For comparative analysis the above five video sequences are processed with the proposed scheme and three other existing models namely, Gaussian mixture model (GMM) [30], expected Gaussian mixture model (EGMM) [31], and model of Reddy *et al.* [29]. Percentage of correct classification for object detection (PCC_{OD}) is used as the metric for comparison, which is defined as —

$$PCC_{OD} = \frac{TP + TN}{TPF} \times 100 \tag{2.6}$$

where TP is true positive, which represents the number of correctly detected foreground pixels and TN is true negative, which represents the number of correctly detected background pixels. TPF represents total number of pixels in the frame. TP and TN are measured from a predefined ground truth frame.

Further, the window size (W) used during classification of a pixel as stationary or non-stationary is chosen experimentally by varying $W = 5, 7, 9, 11, 13$. Similarly, for each window the constant C to calculate the local threshold is varied between 3 and 13 in a step of 1. For each combination of W and C, the PCC_{OD} is computed. A graphical observation among these three parameters is shown in Fig. 2.1 considering the "SMI" video sequence. It may be seen that for $W = 9$ and $C = 7$, the PCC_{OD} achieved maximum of 99.55%. Similar observations are also obtained for other four video sequences. The objects detected from different frames are depicted in Figs. 2.2 – 2.11. It may be observed that, object detection performance of proposed scheme is superior to GMM and EGMM schemes, however it has similar performance with Reddy *et al.*'s scheme. But, present scheme is computationally efficient compared to Reddy *et al.*'s scheme as the latter uses three cascading classifiers followed by a probabilistic voting scheme.

The PCC_{OD} obtained in each case is listed in Table 2.1. The higher accuracy of PCC_{OD} is achieved due to the intensity range defined for each background pixel around its true intensity. The increase and decrease in the intensity level of the background pixels due to illumination variation is taken care by upper and lower part of the predefined intensity range respectively. Such increase or decrease in intensity may be caused by switching on or off of additional light sources, movement

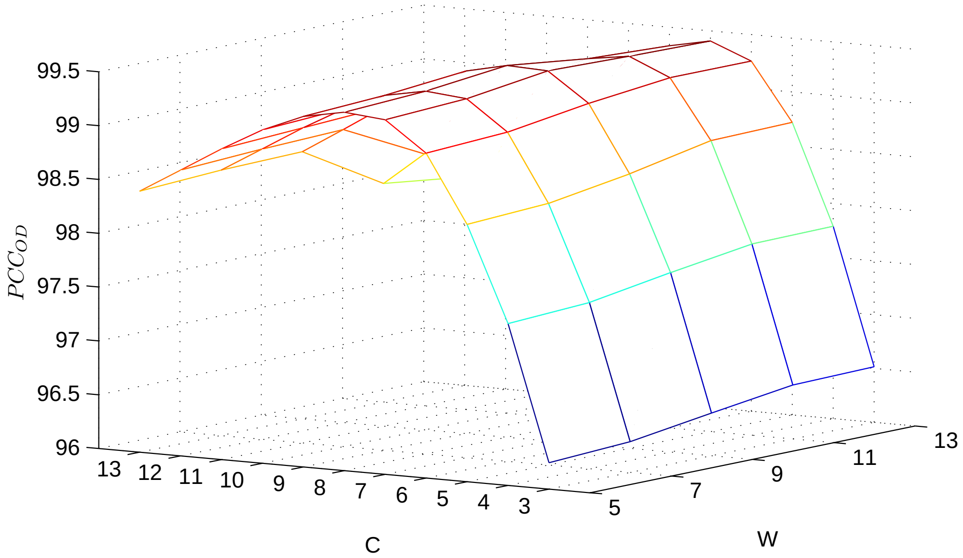


Figure 2.1: Variation of percentage of correct classification for object detection (PCC_{OD}) with window size (W) and constant (C)

of clouds in the sky etc. Moreover, as shadows have low intensity value when falls on any surface, decreases its intensity by some factor. Therefore, the proposed scheme has an advantage of removing the shadows if any, at the time of detecting the objects.

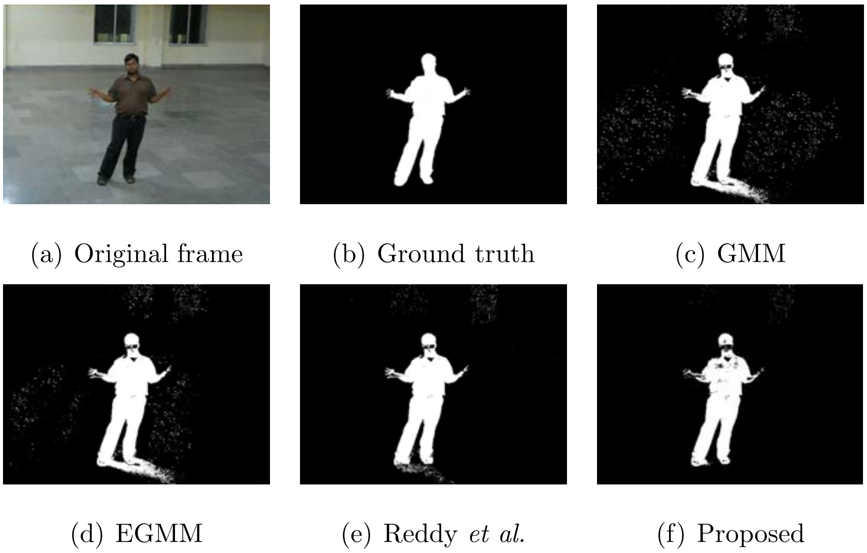
2.5 Summary

In this chapter a simple but robust scheme of background modeling and local threshold based object detection is proposed. Videos with low illumination background, illumination variant background, and low motion background are considered for simulation to test the generalized behavior of the scheme. Recent schemes are compared with the proposed scheme, both qualitatively and quantitatively. It is, in general, observed that the suggested scheme outperforms

Table 2.1: Comparative analysis of PCC_{OD}

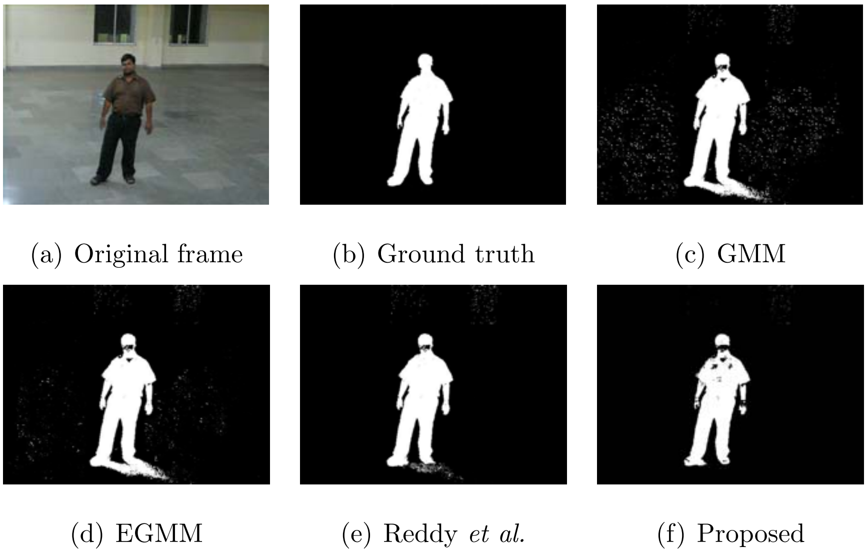
Method	Frame Numbers									
	SMI		SMO		MMO		LB		HM	
	135	182	83	135	141	197	116	164	83	178
GMM	97.78	97.64	97.52	97.83	98.03	97.89	97.64	98.12	97.63	97.37
EGMM	98.16	98.31	98.23	98.27	98.34	98.56	98.31	98.27	98.32	98.14
Reddy *et al.*	99.23	99.26	99.03	98.82	99.13	99.07	98.86	98.66	98.87	98.94
Proposed	99.40	99.55	99.19	98.97	99.02	98.93	99.16	99.03	99.41	99.13

others and detects objects free of shadows in all possible scenarios considered.


(a) Original frame (b) Ground truth (c) GMM

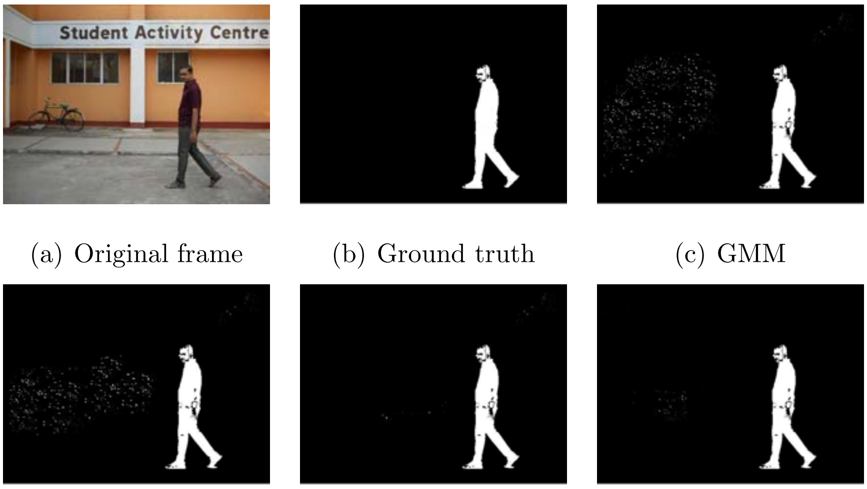
(d) EGMM (e) Reddy *et al.* (f) Proposed

Figure 2.2: Objects detected in frame 135 of "SMI" sequence.


(a) Original frame (b) Ground truth (c) GMM

(d) EGMM (e) Reddy *et al.* (f) Proposed

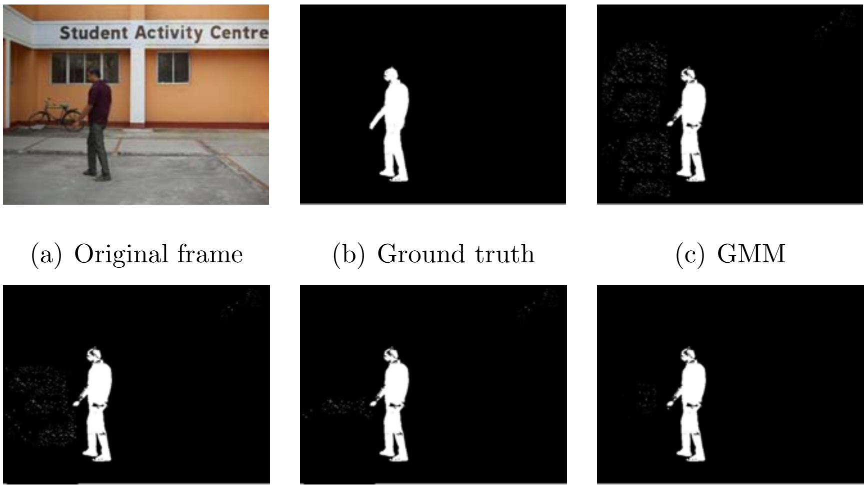
Figure 2.3: Objects detected in frame 182 of "SMI" sequence.



(a) Original frame (b) Ground truth (c) GMM

(d) EGMM (e) Reddy *et al.* (f) Proposed

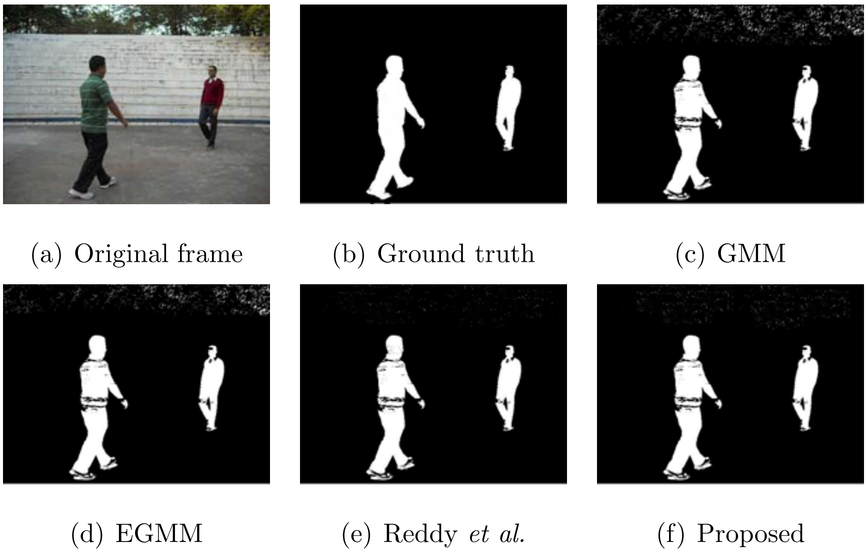
Figure 2.4: Objects detected in frame 83 of "SMO" sequence.



(a) Original frame (b) Ground truth (c) GMM

(d) EGMM (e) Reddy *et al.* (f) Proposed

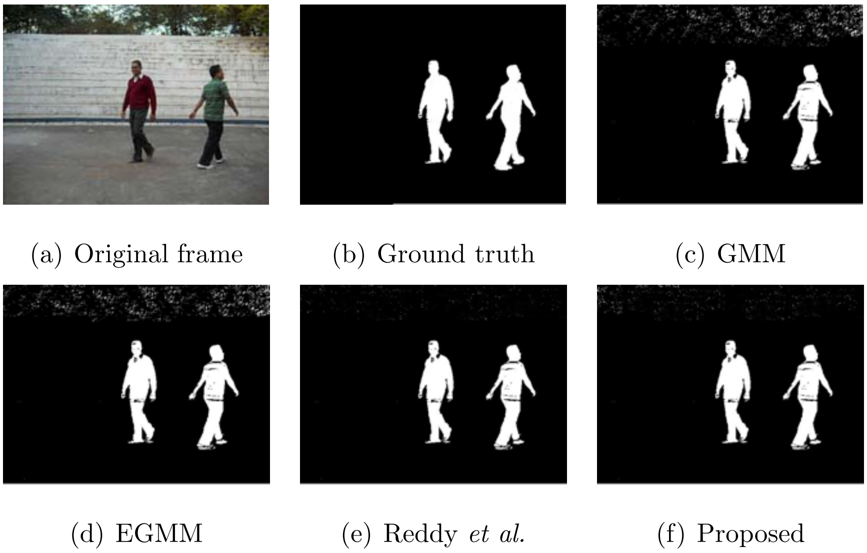
Figure 2.5: Objects detected in frame 135 of "SMO" sequence.



(a) Original frame (b) Ground truth (c) GMM

(d) EGMM (e) Reddy *et al.* (f) Proposed

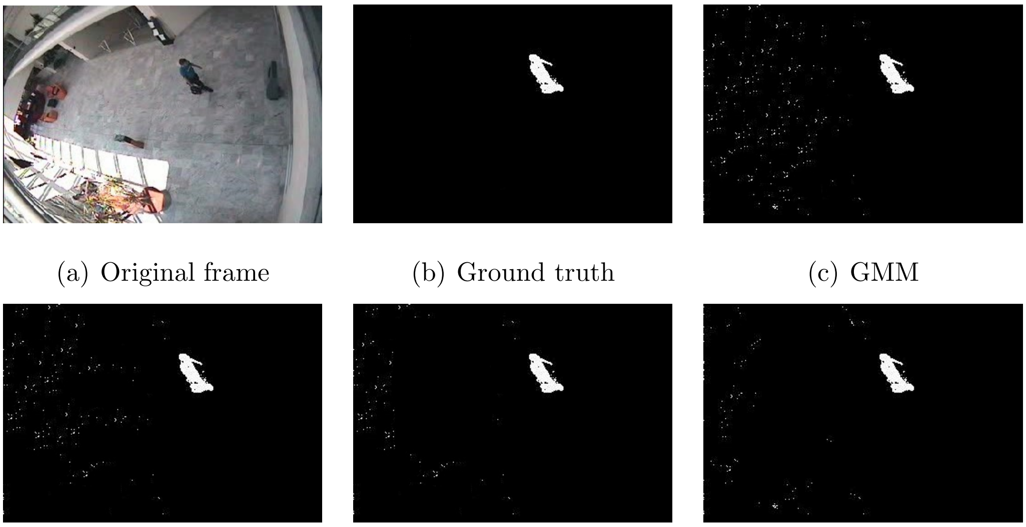
Figure 2.6: Objects detected in frame 141 of “MMO” sequence.



(a) Original frame (b) Ground truth (c) GMM

(d) EGMM (e) Reddy *et al.* (f) Proposed

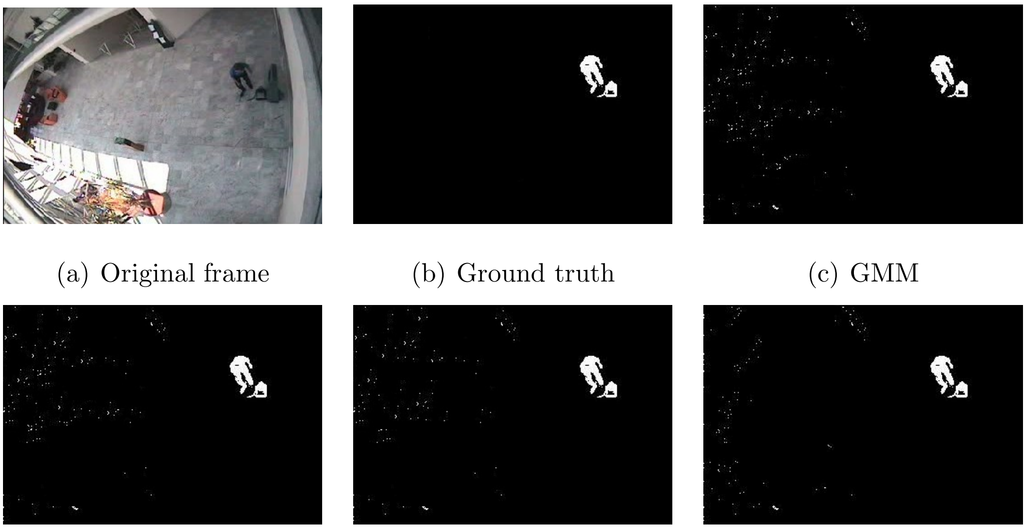
Figure 2.7: Objects detected in frame 197 of “MMO” sequence.



(a) Original frame (b) Ground truth (c) GMM

(d) EGMM (e) Reddy *et al.* (f) Proposed

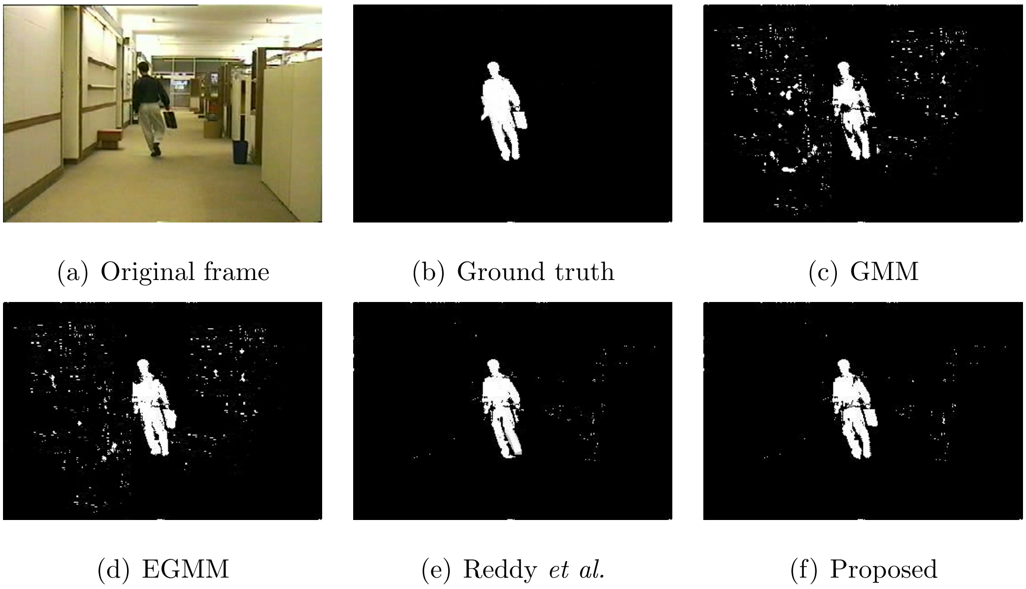
Figure 2.8: Objects detected in frame 116 of "LB" sequence.



(a) Original frame (b) Ground truth (c) GMM

(d) EGMM (e) Reddy *et al.* (f) Proposed

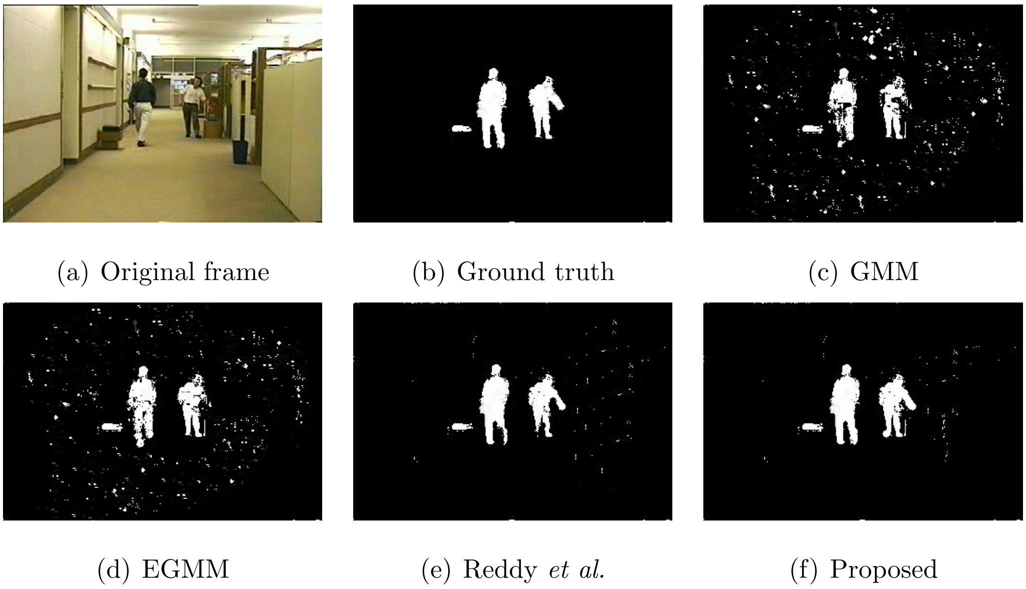
Figure 2.9: Objects detected in frame 164 of "LB" sequence.



(a) Original frame (b) Ground truth (c) GMM

(d) EGMM (e) Reddy *et al.* (f) Proposed

Figure 2.10: Objects detected in frame 83 of “HM” sequence.



(a) Original frame (b) Ground truth (c) GMM

(d) EGMM (e) Reddy *et al.* (f) Proposed

Figure 2.11: Objects detected in frame 178 of “HM” sequence.

Chapter 3

Object Tracking using Dual-Tree Complex Wavelet Transform

This chapter deals with tracking of the detected objects. Tracking can be defined as the problem of estimating the trajectory of an object in the image plane as it moves around a scene [2]. The popular methods for tracking are generally based on moving object regions [32]. In these methods, a bounding box is identified and tracked, which is calculated for connected components of moving objects in two-dimensional space. The disadvantages of this method includes, the dependency of different object properties like size, shape, color, etc. In order to avoid the above shortcomings, researchers moved towards feature based tracking [33]. Both real-valued as well as complex-valued wavelet coefficients can be used as object feature in feature based tracking. However, real-valued wavelet transform suffers from shift invariance and lack of directional selectivity [34]. Hence, complex-valued wavelet coefficients are used as object feature for tracking to devoid such limitations.

Complex wavelets have not been popularly used in image processing due to difficulty in designing complex filters, which needs to satisfy a perfect reconstruction property [35]. To overcome the above property, N. G. Kingsbury proposed a dual-tree implementation of the complex wavelet transform (CWT) called as dual-tree complex wavelet transform (DTCWT) [35]. It uses two trees of real filters to generate the real and imaginary parts of wavelet coefficients.

In this chapter two different algorithms for tracking of objects detected in video sequences by Algorithm 3 of last chapter are proposed. The first algorithm suggests searching of centroid of an object in successive frames. Initially, centroid is estimated from the previous frame. Its accuracy is verified by comparing the entropy of dual-tree complex wavelet coefficients in the bounding boxes in two frames. If estimation is found to be inaccurate, a dynamic window is utilized to search for accurate centroid. The second algorithm tries to suppress ghost using an efficient background updating model.

The rest of the chapter is organized as follows: A survey on the related work is presented in Section 3.1. Basic concepts related to proposed tracking technique are briefed in Section 3.2. Section 3.3 describes the proposed object centroid identification algorithm for object tracking. Simulation results are presented in Section 3.4. Finally, summary of the chapter is provided in Section 3.5.

3.1 Related Work

Khare *et al.* proposed a method in which, object is tracked in subsequent frames based on DTCWT. It computes the energy of dual-tree complex wavelet coefficients corresponding to the object area and matches it with the energy computed in the neighborhood area [36]. This scheme is simple and and does not require any other parameter except complex wavelet coefficients. During searching of objects in subsequent frames, a trivial matching algorithm with more computational complexities is performed. Subsequently, Singh *et al.* proposed a modified algorithm, but it fails to accurately estimate the objects [37]. In order to track non-rigid object in complex wavelet domain Prakash *et al.* proposed an approach in which the object is assumed to be deformable under limit, that is, it may change its shape from one frame to another [38]. The basic idea in their method is to decompose the image into two components namely, a two dimensional motion and a two dimensional shape change. The motion component is factored out while the shape is explicitly represented by storing a sequence of two dimensional models. Each model corresponds to an image frame. The proposed method performs well only when the change in the shape in the consecutive frames is small.

It is observed that the existing schemes in the object tracking are computationally inefficient and mostly use a fixed size window while searching for centroid. In addition, the accuracy is also limited due to non use of object kinematics in successive frames. To alleviate these limitations, a dual-tree complex wavelet transform based tracking scheme is proposed, which utilizes a variable size window during centroid identification. It also takes object velocity information into consideration.

3.2 Basic Concepts

The proposed tracking scheme is based on two fundamental concepts, namely dual-tree complex wavelet transform and Shannon entropy. For better understanding of the suggested scheme both the concepts are discussed in nutshell prior to the proposed centroid searching algorithm for tracking.

3.2.1 Dual-Tree Complex Wavelet Transform

Nick G. Kingsbury proposed a dual-tree implementation of the complex wavelet transform (CWT) called as dual-tree complex wavelet transform (DTCWT) [35]. It uses two trees of real filters to generate the real and imaginary parts of wavelet coefficients. It comprises two parallel wavelet filter bank trees, tree A for real and tree B for imaginary or vice versa, which contain carefully designed filters of different delays that minimizes the aliasing effects due to down sampling. It should be noted that the two trees are independent, which makes them easy to implement in parallel. It is having properties like shift invariance, directional selectivity, and perfect reconstruction [35]. Figure 3.1 shows the dual-tree complex wavelet transform of a 1-D signal "$x\,(n)$".

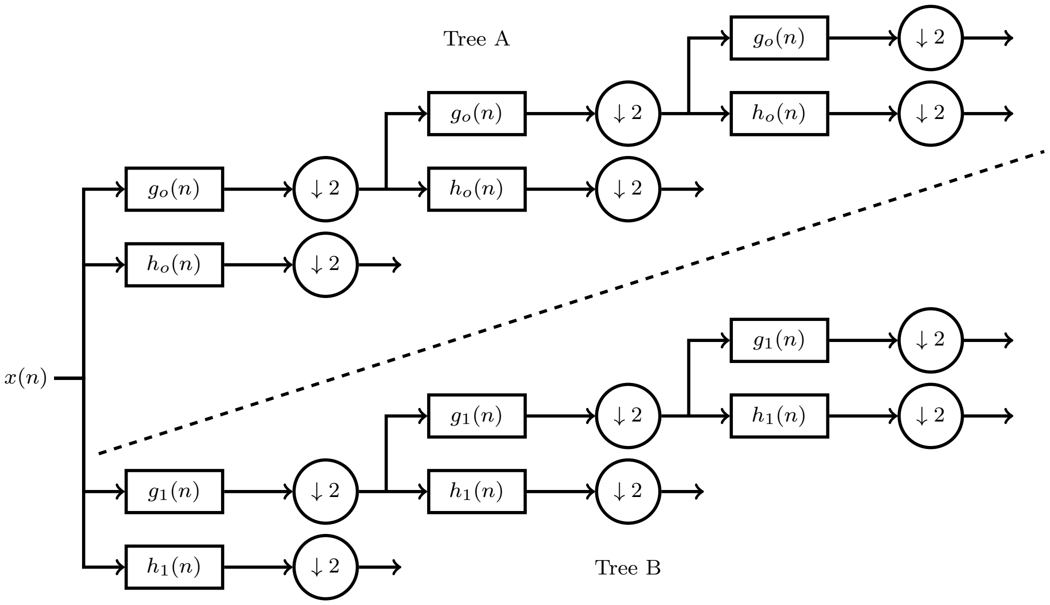


Figure 3.1: Dual-tree complex wavelet transform of "$x\,(n)$".

3.2.2 Shannon Entropy

Shannon entropy is the average unpredictability in a random variable, which is equivalent to its information content. The concept was introduced by Claude E. Shannon [39]. He denoted the entropy H of a discrete random variable X with possible values $\{x_1, x_2, x_3, \cdots, x_n\}$ having the probability mass function $P(X)$ as —

$$H(X) = E[I(X)] = E[ln(P(X))]. \tag{3.1}$$

Here E is the expected value, and I is the information content of X. $I(X)$ is itself a random variable. The entropy can explicitly be written as —

$$H(X) = \sum_{i=1}^{n} P(x_i)I(x_i) = \sum_{i=1}^{n} P(x_i)\log_2\left(\frac{1}{P(x_i)}\right) = -\sum_{i=1}^{n} P(x_i)\log_2\left(P(x_i)\right) \tag{3.2}$$

3.3 Proposed Tracking Technique

Proposed tracking technique mainly aims at finding the centroid of the object in subsequent frames of the video. It is calculated using DTCWT, Shannon entropy, and object velocity. If the velocity of an object in subsequent frames remain the same then the new centroid is found in best case. In contrast if the velocity changes then the new centroid is searched in a dynamically created window. A concept called *object's zero velocity* has been introduced to update the background and suppress ghost.

Let the video to be processed contains a total of N frames as $\{f_1, f_2, \cdots, f_n, f_{n+1}, \cdots, f_N\}$. Algorithm 1 in Chapter 2 takes n initial frames as its input. Frames f_{n+1} and f_{n+2} are given as input to Algorithm 3 to detect objects. The leftover frames $\{f_{n+3}, f_{n+4}, \cdots, f_N\}$ are used for tracking the detected objects. So, it can be concluded that proposed tracking algorithm takes $\{f_{n+3}, f_{n+4}, \cdots, f_N\}$ original frames and f_{n+1} and f_{n+2} segmented frames as input. The output produced are tracked frames. The assumptions and the terminologies used in the proposed algorithm are given below for clear understanding of centorid finding algorithm given in Algorithm 4 .

Preconditions and terminologies used are —

(i) Let m objects are detected in segmented frames f_{n+1} and f_{n+2}. The objects detected are represented as $\{O^1, O^2, \cdots, O^m\}$.

(ii) $D^X_{a,b}$ and $V^X_{a,b}$ represents Euclidean distance and velocity of an object X in between frame a and b respectively.

(iii) $\left(C^X_{i,z}, C^X_{j,z}\right)$ and $\mathcal{ED}^X_z$ represent the centroid and entropy of the dual-tree complex wavelet transform of bounding box surrounding the object X in z^{th} frame respectively.

(iv) Let t be the time between two successive frames.

Initially the velocity between the two known centroids are calculated. Assuming that the object is moving with constant velocity, the unknown centroid $\left(\mathbf{C^X_{i,z}}, \mathbf{C^X_{j,z}}\right)$ in the z^{th} frame for the X object is calculated using the equations given in step 4 and 5 of the Algorithm 4. The correctness of the estimated centroid is determined by comparing the entropies of the DTCWT for the same object in the z^{th} and $(z-1)^{th}$ frames. If they are same, we confirm $\left(\mathbf{C^X_{i,z}}, \mathbf{C^X_{j,z}}\right)$ as our new centroid, else a window is constructed to find the new centroid. Equations in steps $10-13$ of Algorithm 4 demonstrate the construction of window. Each location from top-left corner to bottom-right corner of the new window thus formed, is considered as the centroid of the z^{th} frame. Entropy of the DTCWT for each centroid is then compared with that of the centroid in the previous frame. The searching is stopped when a match is found. The detailed steps are given in Algorithm 4.

Updating the Background Model

In order to accommodate changes in the background scene and suppress ghost an *object's zero velocity* concept has been introduced. If velocity of an object remains zero for a time period T seconds, then background model associated with the object boundary is remodeled to accommodate objects in the background. Moreover, the background subtraction is performed in a random interval of time. In each frame while searching the centroid of objects a random number Z is generated and if Z is found to be one then a background subtraction for two subsequent frames, from the frame where Z is found to be one is performed. Z is calculated as —

$$Z = K \times T \tag{3.3}$$

where K is frame rate. Similarly, if a stationary object gains velocity, then ghost is suppressed in the same way. In the proposed approach only the background model associated with the object boundary is updated, where as in existing methods like, ViBE [27] and W4 [23] entire background model is updated.

```
1  for $z \leftarrow k+3$ to $N$ do
2      for $X \leftarrow 1$ to $m$ do
3          $V^X_{(z-2),(z-1)} = \frac{D^X_{(z-2),(z-1)}}{t}$
4          $V^X_{(z-1),z} = \frac{D^X_{(z-1),z}}{t}$
5          $V^X_{(z-2),z} = \frac{D^X_{(z-2),z}}{2t}$
6          Calculate $\left(\mathbf{C^X_{i,z}}, \mathbf{C^X_{j,z}}\right)$ using equations in step 3 to 5.
7          if $(\mathcal{ED}^X_z = \mathcal{ED}^X_{z-1})$ then
8          $\left(\mathbf{C^X_{i,z}}, \mathbf{C^X_{j,z}}\right)$ is confirmed as centroid.
9          else
10         $h = |C^X_{i,z} - C^X_{i,(z-1)}|$
11         $w = |C^X_{j,z} - C^X_{j,(z-1)}|$
12         $D_h = 2h + 3$
13         $D_w = 2w + 3$
14         $\epsilon_h \leftarrow \lfloor D_h \div 2 \rfloor$
15         $\epsilon_w \leftarrow \lfloor D_w \div 2 \rfloor$
16         for $m \leftarrow (C^X_{i,z} - \epsilon_w)$ to $(C^X_{i,z} + \epsilon_w)$ do
17             for $n \leftarrow (C^X_{j,z} + \epsilon_h)$ to $(C^X_{j,z} - \epsilon_h)$ do
18                 $\left(\mathbf{C^X_{i,z}}, \mathbf{C^X_{j,z}}\right) = (m, n)$
19                 if $(\mathcal{ED}^X_z = \mathcal{ED}^X_{z-1})$ then
20                 $\left(\mathbf{C^X_{i,z}}, \mathbf{C^X_{j,z}}\right)$ is confirmed as centroid.
21                 Break.
22             end
23         end
24     end
25 end
```

Algorithm 4: Centroid searching

3.4 Experimental Results

To show the efficacy of the proposed tracking techniques, simulation has been carried out on video sequences used for object detection in Chapter 2. Person in “SMI” and “SMO” is named as “O1.” In “MMO” person with green shirt is named as “O1” and other one as “O2.” “LB” sequence consists of two movable objects. Person carrying the bag is named as “O1” and the bag is named as “O2.” As the person leaves the bag and walks out, it is required to update the background for accurate tracking. The number of movable objects in “HM” is four. Initially a person comes into field of view (FOV) of camera with a briefcase in his hand. The person is named as “O1” and briefcase as “O2”. “O1” keeps “O2” on a desk and moves away from FOV of camera. This describes a scenario of object in motion changes to a stationary object from “O2” point of view. Meanwhile, another person comes to the FOV of the camera empty handed. This person in our simulation has been identified as “O3”. “O3” picks up a television set from another desk. Television is described as “O4”. This presents a scenario of stationary object changes to object in motion from “O4” point of view. Hence it is needed to update the background to correctly identify the objects and suppress the ghosts.

For comparative analysis, video sequences are processed with the proposed tracking technique and two other existing models namely, method by Khare *et al.* [36] and Prakash *et al.* [38]. Percentage of correct classification for object tracking (PCC_{OT}) is used as the metric for comparison, and is defined as —

$$PCC_{OT} = \frac{TDC}{TNC} \times 100 \tag{3.4}$$

where TDC represents the number of truly detected centroids for each object in the frame and TNC represents the total number of centroids for each object individually. TDC is measured from a predefined ground truth frame. The comparative performance analysis of PCC_{OT} is in Table 3.1.

In “HM” sequence during the calculation of PCC_{OT}, “O2” and “O4” are not considered as they appear for very small amount of time but in simulation results of the frames in all situations are clearly shown with indication to background updating and ghost suppression concept. It may be clearly observed that proposed tracking technique has an upper hand as compared to other schemes with respect to PCC_{OT}.

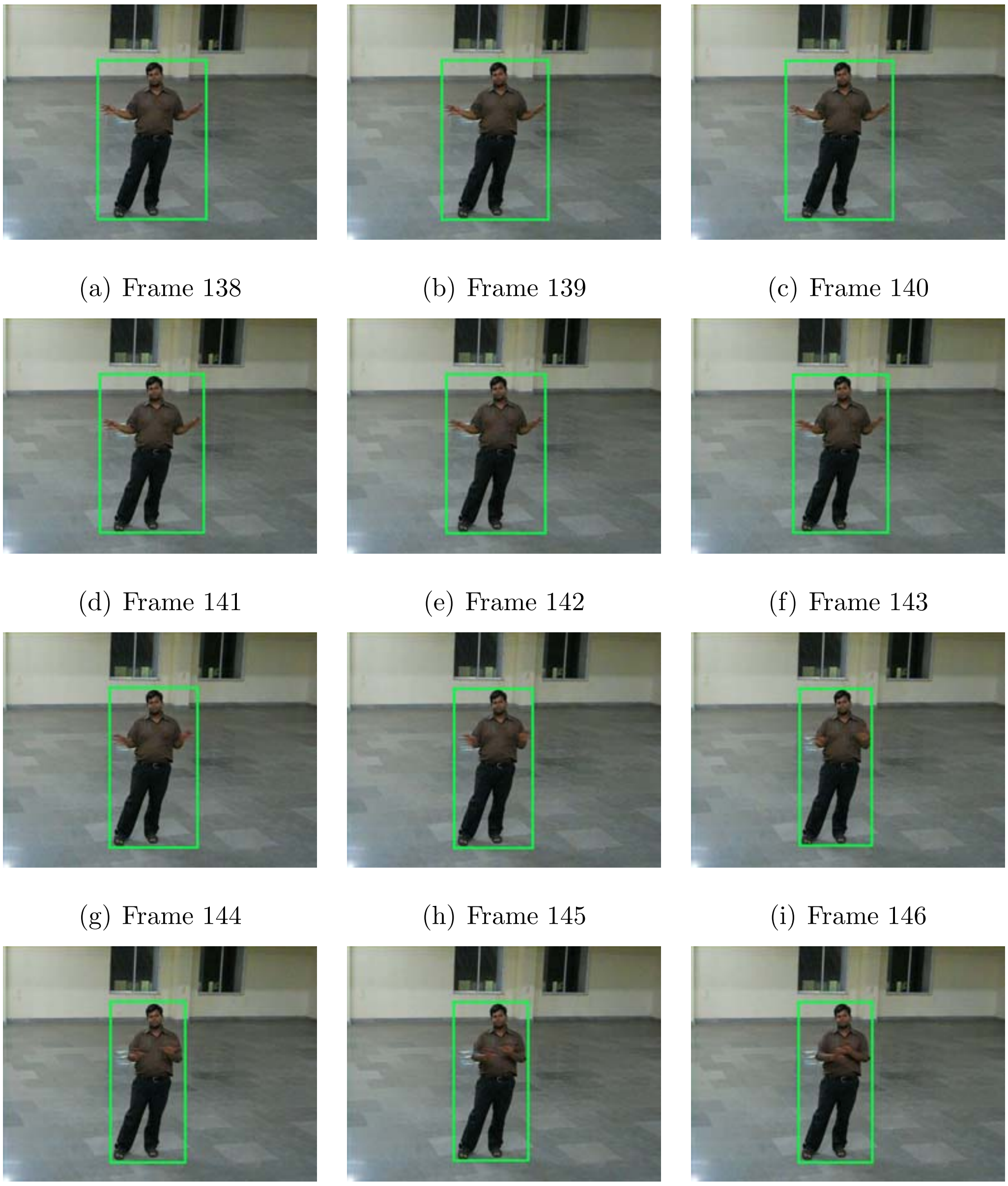
Table 3.1: Comparative analysis of PCC_{OT}

Method	Objects							
	SMI	SMO	MMO		LB		HM	
	O1	O1	O1	O2	O1	O2	O1	O3
Khare *et al.* [36]	91.79	91.53	91.48	91.83	92.08	91.87	91.67	91.52
Singh *et al.* [37]	93.59	93.84	92.67	93.04	93.52	93.15	92.98	93.78
Proposed Technique	96.79	96.15	95.58	95.27	95.89	95.93	96.05	95.48

DTCWT is implemented using a ten tap filter as given in [35]. Figs. 3.2 - 3.6 show the simulation results of all the sequences. Figs. 3.5 and 3.6 clearly indicate that the "bag" and "briefcase", are detected as foreground object in Frames 176 to 180 of "LB" sequence and Frames 174 to 176 of "HM" because of its motion, is suppressed in later sequences of frame as a result of updating the background model respectively. Similarly, ghost created in Frames 223 to 225 of "HM" is also suppressed in later sequences in Figure 3.6.

3.5 Summary

In this chapter a centroid searching algorithm using a dynamic window and an efficient background updating model for suppression of ghost in subsequent frames are proposed. The objects which remain static for a predefined time duration are updated as background for subsequent object detection. The centroid detection utilizes entropy information of DTCWT coefficients in a bounding box of an object in two successive frames. The suggested scheme is simulated on video sequences of different properties and comparative analysis is performed with traditional methods. The improved PCC_{OT} value for centroid detection justifies the superiority of the



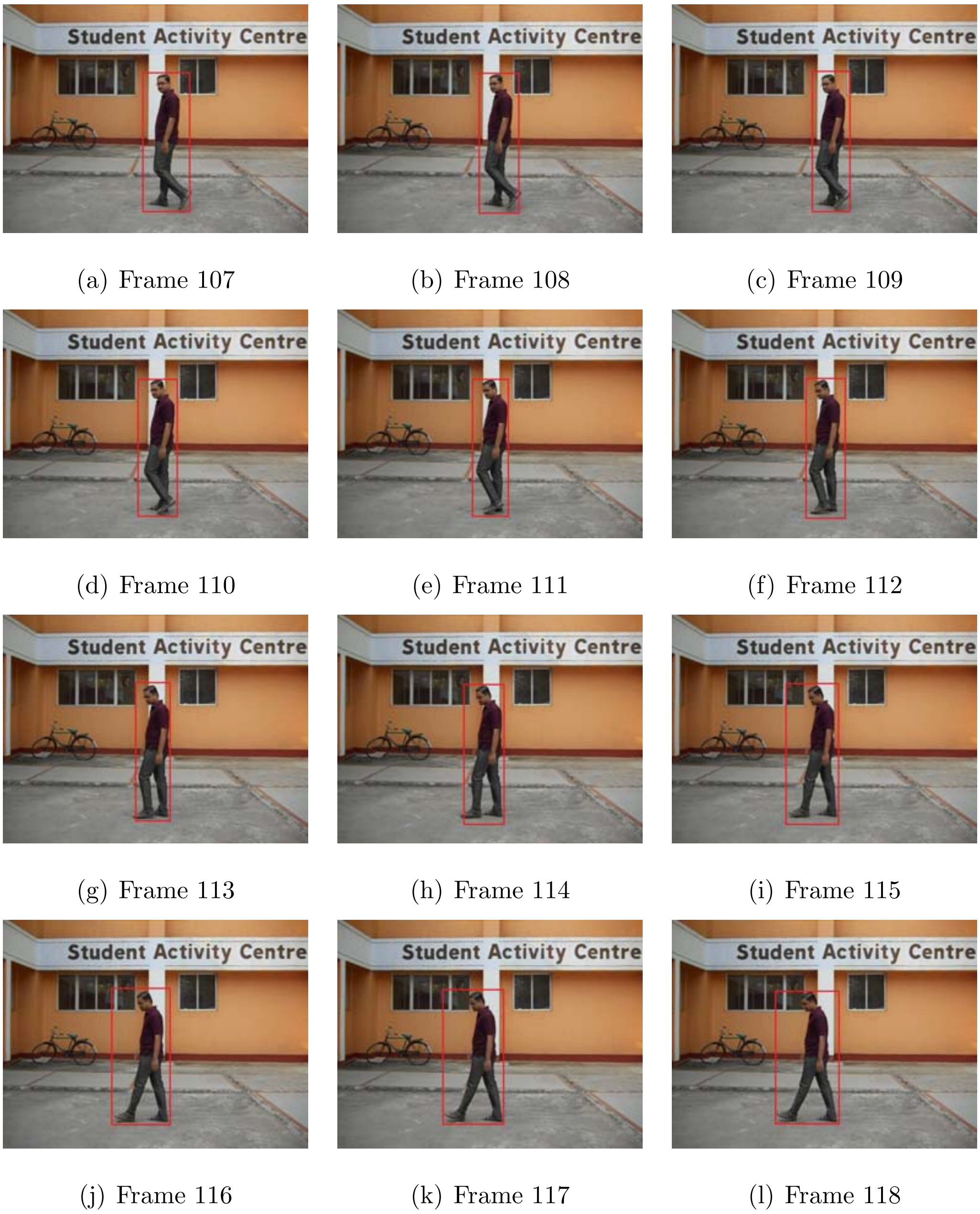
(a) Frame 138 (b) Frame 139 (c) Frame 140

(d) Frame 141 (e) Frame 142 (f) Frame 143

(g) Frame 144 (h) Frame 145 (i) Frame 146

(j) Frame 147 (k) Frame 148 (l) Frame 149

Figure 3.2: Tracked frames of “SMI” sequence.



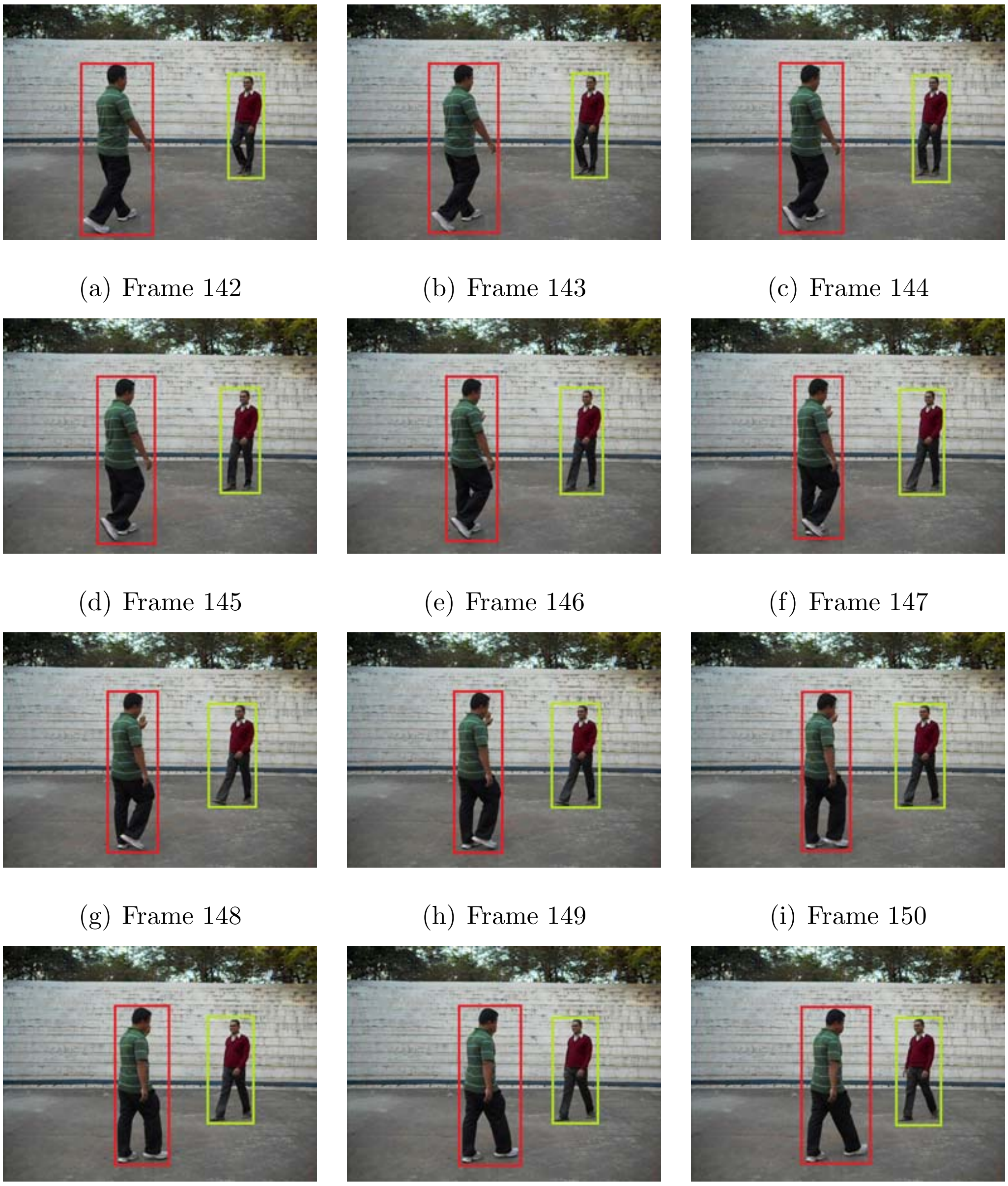
(a) Frame 107 (b) Frame 108 (c) Frame 109

(d) Frame 110 (e) Frame 111 (f) Frame 112

(g) Frame 113 (h) Frame 114 (i) Frame 115

(j) Frame 116 (k) Frame 117 (l) Frame 118

Figure 3.3: Tracked frames of “SMO” sequence.



(a) Frame 142

(b) Frame 143

(c) Frame 144

(d) Frame 145

(e) Frame 146

(f) Frame 147

(g) Frame 148

(h) Frame 149

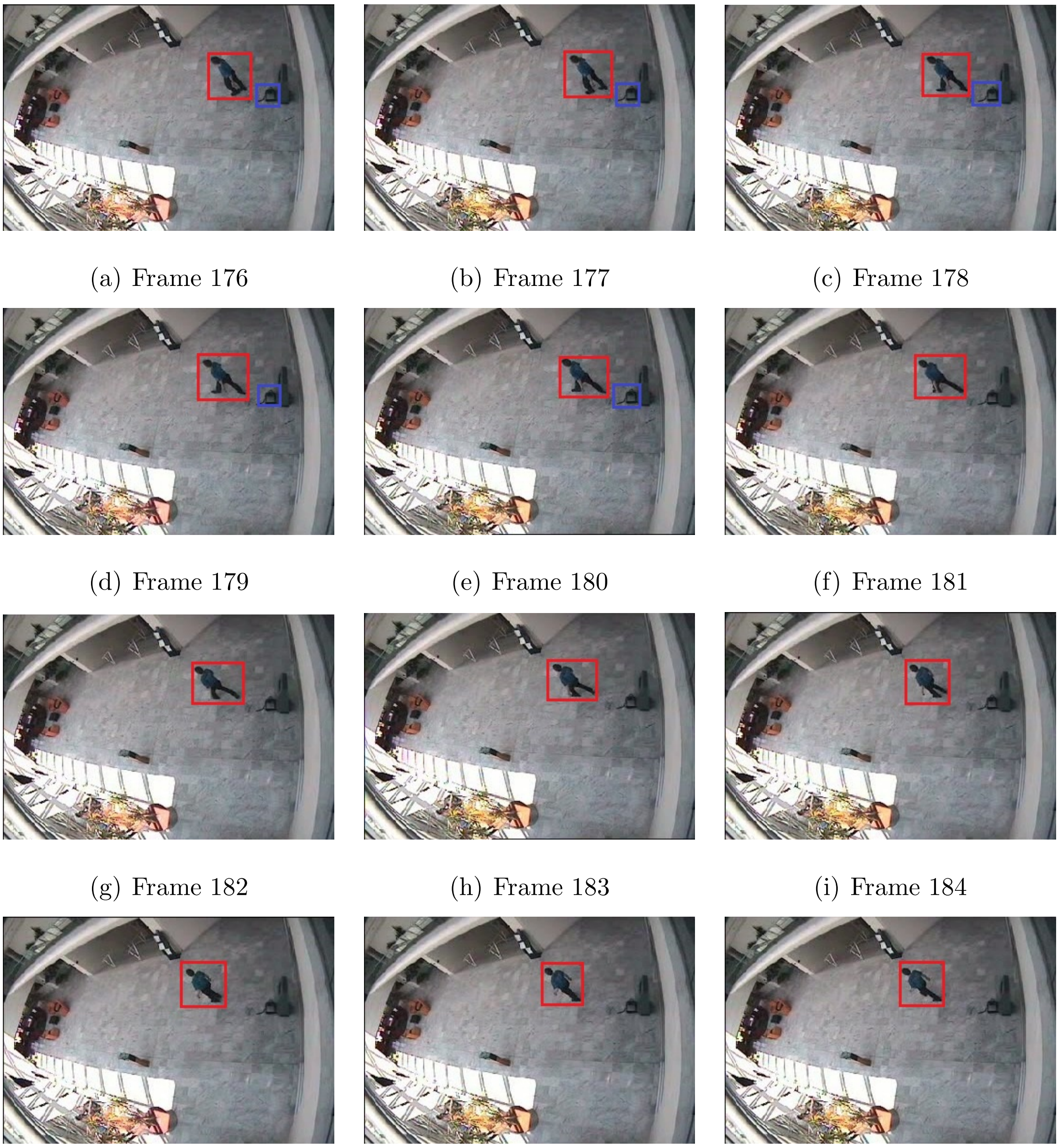
(i) Frame 150

(j) Frame 151

(k) Frame 152

(l) Frame 153

Figure 3.4: Tracked frames of “MMO” sequence.



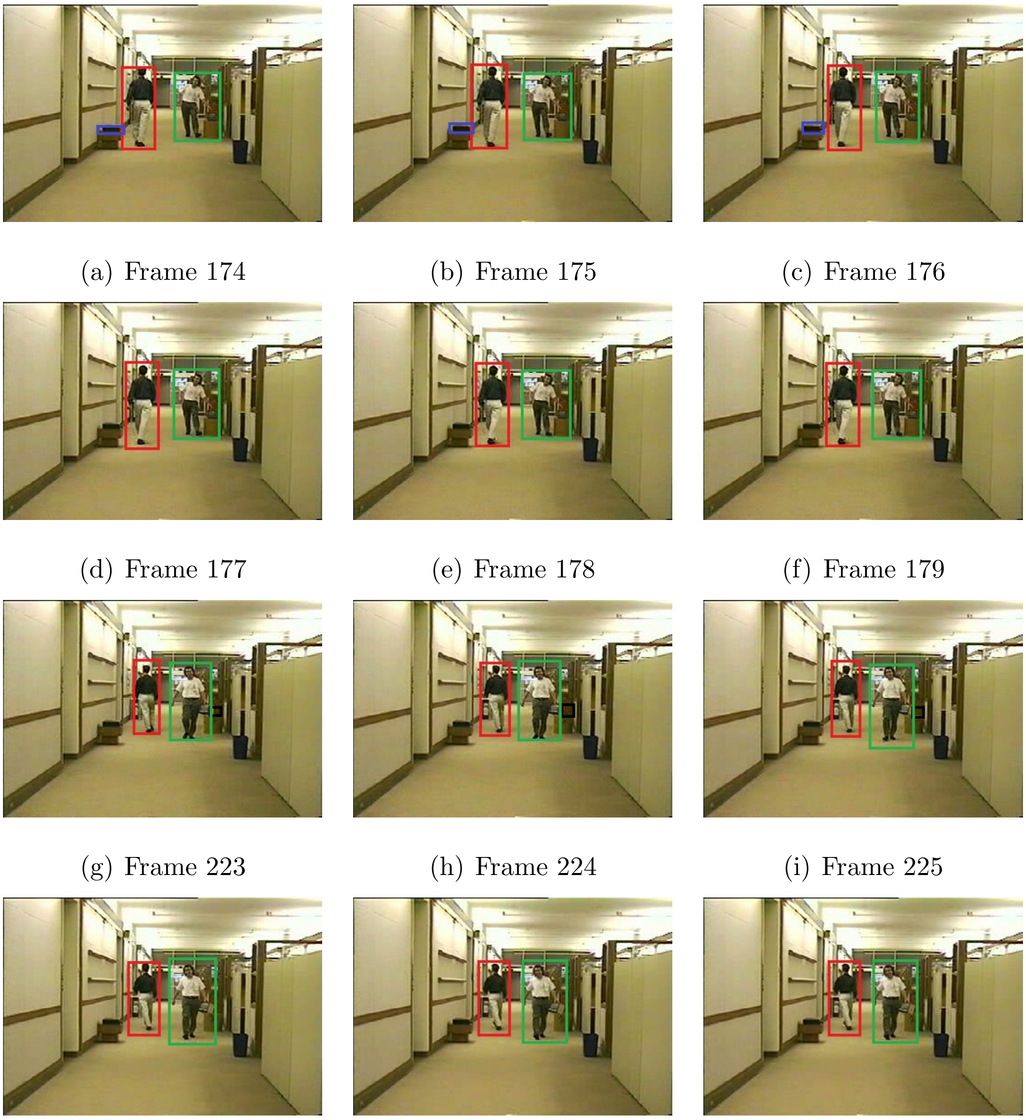
(a) Frame 176 (b) Frame 177 (c) Frame 178

(d) Frame 179 (e) Frame 180 (f) Frame 181

(g) Frame 182 (h) Frame 183 (i) Frame 184

(j) Frame 185 (k) Frame 186 (l) Frame 187

Figure 3.5: Tracked frames of “LB” sequence.



(a) Frame 174 (b) Frame 175 (c) Frame 176

(d) Frame 177 (e) Frame 178 (f) Frame 179

(g) Frame 223 (h) Frame 224 (i) Frame 225

(j) Frame 226 (k) Frame 227 (l) Frame 228

Figure 3.6: Tracked frames of “HM” sequence.

proposed centroid identification algorithm. The visual results are given to show the capability of ghost suppression by the suggested background updating model.

Chapter 4

Conclusion

Object tracking is an important computer vision application which consists of two closely related processes; object detection and tracking of the detected objects. Object detection in videos obtained from static camera and fixed background is achieved through *background subtraction* approach. In this approach a background model is developed considering the first frame or first few frames. Subsequently, a thresholding technique is utilized to extract foreground objects. Shadows are very often misclassified as foreground objects, which needs an additional step to remove before the detected objects can be tracked. Object tracks are computed by various approaches. Centroid in subsequent frames are searched in a fixed size window, which makes the algorithm more complex. Inorder to accommodate changes in the background scene, updating background model plays a vital role. Frequent and entire updating of the background model makes the method computationally inefficient.

For the last two decades, researchers across the globe have been working towards object detection and tracking as well. Significant volumes of literature are available in this domain. Real time deployment of the algorithm demands higher accuracy with less complexity, which makes the problem still open and needs significant research. In this thesis, efforts have been made to detect and track objects and evaluations are made experimentally.

An approach for object detection is presented in Chapter 2. Proposed detection scheme starts with considering first few frames of the video. Pixels in these frames are classified as stationary or non-stationary according to their intensity along temporal axis. Considering the stationary pixel information, background model is developed. A local thresholding technique tries to extract foreground objects and suppresses shadows at low computational cost. Comparative analysis demonstrates the efficacy of the proposed detection scheme.

Chapter 3 presents a method to track the detected objects. In this chapter two algorithms are presented. The first algorithm suggests searching of centroid of each object in successive frames. Initially, centroids are estimated from the previous frame. Its accuracy is verified by comparing the entropy of dual-tree complex wavelet coefficients in the bounding boxes in two frames. If estimation is found to be inaccurate, a dynamic window is utilized to search for accurate centroid. The second algorithm tries to suppress ghost using an efficient background updating model. Simulation results and comparative analysis with the traditional methods show the superior performance of the proposed scheme.

The proposed tracking model suffers from few limitations like, occlusion, presence of object in first frame etc.

Scope for Further Research

The research findings made out of this thesis gives a scope to go beyond tracking. The proposed tracking model can be extended for object recognition. Features utilized to track objects in subsequent frames of the video can be stored in data base for recognition. Video segmentation in general and object tracking in particular have immense potential, which if used in constructive ways can be boon to the mankind.

Bibliography

[1] Alan C. Bovik. *Handbook of Image and Video Processing.* Elsevier Academic Press, United States, 2nd edition, 2005.

[2] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Computing Survey*, 38, Dec 2006.

[3] H. Moravec. Visual mapping by a robot rover. In *International Joint Conference on Artificial Intelligence*, pages 598 – 600, 1979.

[4] C. Harris and M. Stephens. A combined corner and edge detector. In *Fourth Alvey Vision Conference*, pages 147 – 151, 1988.

[5] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593 – 600, 1994.

[6] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91 – 110, 2004.

[7] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603 – 619, 2002.

[8] Dieter Koller, Joseph Weber, and Jitendra Malik. Robust multiple car tracking with occlusion reasoning. In *Proceedings of European Conference on Computer Vision*, pages 189 – 196. Springer-Verlag, May 1994.

[9] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 255 – 261, Sep 1999.

[10] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780 – 785, Jul 1997.

[11] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. In *Proceedings of the Twelveth IAPR International Conference on Computer Vision Image Processing*, volume 1, pages 126 – 131, Oct 1994.

[12] B.P.L. Lo and S.A. Velastin. Automatic congestion detection system for underground platforms. In *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 158 – 161, 2001.

[13] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1337 – 1342, Oct 2003.

[14] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on CVPR*, pages 246 – 252, 1999.

[15] Ahmed M. Elgammal, David Harwood, and Larry S. Davis. Non-parametric model for background subtraction. In *Proceedings of the Sixthth European Conference on Computer Vision-PartII*, ECCV '00, pages 751 – 767, 2000.

[16] M. Piccardi. Background subtraction techniques: a review. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 4, Oct 2004.

[17] M. Seki, T. Wada, H. Fujiwara, and K. Sumi. Background subtraction based on cooccurrence of image variations. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 65 – 72, Jun 2003.

[18] P. Power and A. Schoonees. Understanding background mixture models for foreground segmentation. In *Proceedings Image and Vision Computing*, pages 267 – 271, Nov 2002.

[19] N.M. Oliver, B. Rosario, and A.P. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831 – 843, Aug 2000.

[20] D.M. Tsai and S.C. Lai. Independent component analysis-based background subtraction for indoor surveillance. *IEEE Transactions on Image Processing*, 18(1):158 – 167, Jan 2009.

[21] H.H. Lin, T.H. Liu, and J.H. Chuang. Learning a scene background model via classification. *IEEE Transactions on Signal Processing*, 57(5):1641 – 1654, May 2009.

[22] L. Maddalena and A. Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, 17(7):1168 – 1177, Jul 2008.

[23] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809 – 830, Aug 2000.

[24] D. Gutchess, M. Trajkovics, E. Cohen-Solal, D. Lyons, and A.K. Jain. A background model initialization algorithm for video surveillance. In *Proceedings of the Eighth IEEE International Conference on Computer Vision*, volume 1, pages 733 – 740, Jul 2001.

[25] J.C.S. Jacques, C.R. Jung, and S.R. Musse. Background subtraction and shadow detection in grayscale video sequences. In *Eighteenth Brazilian Symposium on Computer Graphics and Image Processing*, pages 189 – 196, Oct 2005.

[26] C.R. Jung. Efficient background subtraction and shadow removal for monochromatic video sequences. *IEEE Transactions on Multimedia*, 11(3):571 – 577, Apr 2009.

[27] O. Barnich and M. Van Droogenbroeck. ViBe: A Universal Background Subtraction Algorithm for Video Sequences. *IEEE Transactions on Image Processing*, 20(6):1709 – 1724, Jun 2011.

[28] Wonjun Kim and C. Kim. Background subtraction for dynamic texture scenes using fuzzy color histograms. *IEEE Signal Processing Letters*, 19(3):127 – 130, Mar 2012.

[29] V. Reddy, C. Sanderson, and B.C. Lovell. Improved foreground detection via block-based classifier cascade with probabilistic decision integration. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(1):83 – 93, Jan 2013.

[30] Liyuan Li, Weimin Huang, Irene Y. H. Gu, and Qi Tian. Foreground object detection from videos containing complex background. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 2 – 10. ACM Press, Nov 2003.

[31] Z. Zivkovic. Improved adaptive gausian mixture model for background subtraction. In *Proceedings IEEE International Conference on Pattern Recognition*, pages 28 – 31, Aug 2004.

[32] V. Sonka, M. Hlavac and R. Boyle. *Image Processing Analysis and Machine Vision.* Thomson, 3rd edition, 2007.

[33] A. Khare and U.S. Tiwary. Daubechies complex wavelet transform based moving object tracking. In *IEEE Symposium on Computational Intelligence in Image and Signal Processing*, pages 36 – 40, Apr 2007.

[34] I.W. Selesnick, R.G. Baraniuk, and N.G. Kingsbury. The dual-tree complex wavelet transform. *IEEE Signal Processing Magazine*, 22(6):123 – 151, Nov 2005.

[35] N.G. Kingsbury. Image processing with complex wavelets. *Philosophical Transactions of the Royal Society London A*, 357(1760):2543 – 2560, 1999.

[36] Manish Khare, Tushar Patnaik, and Ashish Khare. Dual tree complex wavelet transform based video object tracking. In *International Conference on Information and Communication Technologies*, pages 281 – 286, Sep 2010.

[37] R. Singh, R.K. Purwar, and N. Rajpal. A better approach for object tracking using dual-tree complex wavelet transform. In *International Conference on Image Information Processing*, pages 1 – 5, Nov 2011.

[38] Om Prakash and Ashish Khare. Tracking of non-rigid object in complex wavelet domain. *Journal of Signal and Information Processing*, 2(2):105 – 111, 2011.

[39] Claude E Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379 – 423, Jul 1948.

Dissemination

1. **Kalyan Kumar Hati**, Pankaj Kumar Sa, and Banshidhar Majhi. Intensity Range based Background Subtraction for Effective Object Detection. *IEEE Signal Processing Letters*, 20(8):759 - 762, August 2013.

2. **Kalyan Kumar Hati** and Pankaj Kumar Sa. Object Tracking Neutralizing the Effect of Shadows. *Research Scholars' Day 2012*, Indian Institute of Space Science and Technology, Thiruvananthapuram, India, December 2012.

   This paper received **Best Paper Award** in the engineering track.

3. **Kalyan Kumar Hati**, Pankaj Kumar Sa, and Banshidhar Majhi. LOBS: LOcal Background Subtracter for Video Surveillance. *2012 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics*, pages 29 – 34, Hyderabad, India, December 2012.

   This paper has been ranked in the **First Decile** of the 48 papers presented at the conference.

Kalyan Kumar Hati

Computer Science and Engineering Department,
National Institute of Technology Rourkela,
Rourkela – 769 008, India.

+91 94399 16832.

KalyanKumarHati@nitrkl.ac.in

Qualification

- M.Tech. (Research) (CSE) (Continuing)
  National Institute of Technology Rourkela.
- B.Tech. (CSE)
  Biju Pattanaik University of Technology, Rourkela, [8.30 CGPA]
- 12th
  Central of Board of Secondary Education, New Delhi, [70.60%]
- 10th
  Central of Board of Secondary Education, New Delhi, [76.80%]

Permanent Address

At – Nagapal
Po – Udala
Dist – Mayurbhanj
Pin – 757 041 (India)

Date of Birth

August 19, 1987