B.Sc. in Computer Science and Engineering Thesis

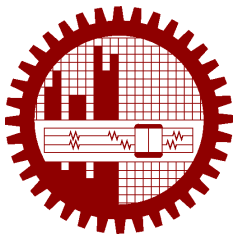
Submitted by

Md. Aminul Islam
201305087

Md. Anwar Hossain
201305119

Supervised by

Dr. Md. Monirul Islam



Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
Dhaka, Bangladesh

October 21, 2018

CANDIDATES' DECLARATION

This is to certify that the work presented in this thesis, titled, **'Scene Reconstruction From Multiple Static CCTV Cameras"**, is the outcome of the investigation and research carried out by us under the supervision of Dr. Md. Monirul Islam.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Md. Aminul Islam
201305087

Md. Anwar Hossain
201305119

CERTIFICATION

This thesis titled, **'Scene Reconstruction From Multiple Static CCTV Cameras"**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in October 21, 2018.

Group Members:

Md. Aminul Islam

Md. Anwar Hossain

Supervisor:

Dr. Md. Monirul Islam
Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology

ACKNOWLEDGEMENT

At First, we like to thank the Almighty for being able to complete our thesis. Then, we like to express our gratitude to our supervisor, Prof. Dr. Md. Monirul Islam for his helpful direction for the thesis. He listens us, helps us in our work, corrects us if we makes mistakes. He gives proper directions in all cases of our thesis. He gives us ideas when we need to implement tasks of our thesis. He also gives us some of his source codes to help us. His constant supervision, direct guidance, detection of flaws in our works as well as words of encouragement makes this thesis a success.

Dhaka
October 21, 2018

Md. Aminul Islam

Md. Anwar Hossain

Contents

List of Figures

List of Algorithms

ABSTRACT

Scene reconstruction from multiple CCTV cameras has a great impact on visual surveillance system, representing crime scene, and many other cases of observation. There are some works on scene reconstruction but actually a little work on scene reconstruction from multiple CCTV cameras. As visual surveillance systems collect a huge amount of data from many different scenes or surveillance cameras, people expect the total activities of an object with little human labeling effort as possible. We propose an automated system that can analyze videos sent from multiple cameras simultaneously, and present the information of an object to user in convenient and easy understandable form. Moreover, if there is no camera at a place, we propose an algorithm to predict an object's movement in that place using his past behaviours. The persons who are in supervision of monitoring will get the work of them easier and more convenient than finding the object from lots of videos manually by using this automated system.

Chapter 1

Introduction

Scene reconstruction is the process of reconstructing a digital version of a real world object from video sequences or scans of the object. Scene reconstruction occupies a major area in computer vision. In visual surveillance, a key task is to monitor activities in the videos or scenes. People have much interest in discovering typical and abnormal activities, detecting activities of some objects, and knowing some structures of the objects, such as paths commonly taken by objects, sources etc , where objects appear and disappear. As visual surveillance systems collect a huge amount of data from many different scenes or surveillance cameras, people expect the total activities of an object with little human labeling effort as possible. Because of the limited view, and limited operating angle of cameras, often is necessary to use multiple cameras with partially overlapping views, which combined can monitor whole area of the interest. Such systems usually generates collection of separated videos, each covering a certain part of the supervised area. A person responsible for supervision must watch and analyze all videos in order to monitor persons moving in the controlled area. The human capacity for simultaneous multiple video monitoring is limited. Therefore, there is a need to develop computer systems that can analyze videos sent from multiple cameras simultaneously, and present the information to user in convenient and easy understandable form.

Sometimes we need monitoring of an specific object and its movement but its movement cannot be tracked in a single CCTV camera if its movement space is too large. Then we need several CCTV cameras in an environment. If we want to monitor of an object in the total environment then we need the videos of that object where it has been. Therefore, we have to search that object in those videos and then combine those to form total monitoring of that object. Moreover if the object is in an area where there is no camera or the camera is damaged there, we want to predict the movement of the object in that area from its previous or later characteristics.

In this chapter we introduce our topics and briefly discuss about motivation behind this, summary of existing works, limitations and objectives.

1.1 Problem Statement and Motivation

Our problem involves scene reconstruction. Here, scene reconstruction means part of scenes of interested object or scene from different cameras is combined as a sequential movement or incidents to form a unified video or scene as a whole. Videos from multiple CCTV cameras are taken as input of the time of the incident which we want to monitor. Besides we want to take account where camera is not available or damaged. We predict the characteristics of the object in that area using its past and future behaviours. Thus we can get the total sequential scene or video of the interested object using the predicted and actual data.

Visual surveillance system has a lot of data from different cameras. If we want to track an object from those lots of data , it takes a lot of time and sometimes may be almost impossible. A person who is in supervision may need a lot of effort to do this. But if we have an automated computer system of tracking that object in those videos and reconstruct the videos as a single scene or video then it takes too much less effort for the person who is in supervision. So total activities of an object can be get with minimum effort through this automated system.

Sometimes it may need to represent a crime scene if occurred by a person or a group of person. We can collect data from multiple cameras at the time of the incident and track the suspected person or group of person and then reconstruct the activities of the person or a group of person in a single scene from those data and represent it for the judgement of the incident. It takes less effort and less time for the representation of the crime scene. The layers can use these automated system to give logic to or against a person in the court. If the suspected person is not in videos for a time being then we can also guess the person's movement by prediction using his later or future characteristics. Thus the lawyers can give reasons using this predictions. Moreover, we can use it in traffic monitoring of the movement of a specific vehicle.

Much is not done in the scene reconstruction from multiple CCTV cameras. In this era of computer, people want to reduce their work and try to done it by computer system. If a computer system of the problem described earlier can be established, people who are involved of supervision will get their work easier from their previous work of monitoring a person manually from a lots of data and they now need a minimum effort. Moreover, lawyers can also get a benefit to represent the incident of the crime to or against a suspected person.

1.2 Summary of Existing Works and Limitations

Several ideas of scene reconstruction, scene modeling, moving object tracking have been proposed in past. In [1] authors have proposed a solution for segmenting out moving object from background using Gaussian mixture models. In [2] authors have detected moving objects in a video using multi-channel kernel fuzzy correlogram based background subtraction method.

In [3] authors have made a solution for detecting object from multiple camera view. In [4, 5] authors have proposed a solution for scene modeling from multiple views of camera. Besides, some scene reconstruction process have also been introduced. This literature review is discussed elaborately in chapter 3.

The main limitation of the existing work is that there is no complete process of scene reconstruction from multiple CCTV cameras to monitor a object . There are processes of tracking moving object in a video, there are processes of scene modeling, scene reconstruction, tracking the velocity of moving objects etc. But for full monitoring of an object in a large environment covered by multiple cameras, there is no unified computer system which will reconstruct a sequential video or scene of the movement or characteristics of the interested object in the total incident from the multiple cameras. Besides, if a camera is missing from a place, there is no way to get the object's movement in that place. There is actually no work on missing scenes regarding this.

1.3 Objectives

The human capacity for simultaneous multiple video monitoring is limited. Therefore, we propose to develop a computer system that can analyze videos sent from multiple cameras simultaneously, and present the information to user in convenient and easy understandable form. We take the videos from multiple CCTV cameras of the time of the incidents that we want to explore. Then, we search the interested object in the videos and find a sequential movement or characteristics of the object from the videos to form a single video or scene of the movement or characteristics of the object. If the object is not found for the time being in a video , then it may have been a place where no camera is available or the camera of that area may be damaged. We want to predict the movement of the object in that area by using its previous or later characteristics. But for this, we surely need the background image of that area where we are predicting. Thus we get the total human or object activities in an environment in a minimum human effort. The persons who are in supervision of monitoring get the work of them more easy and convenient than finding the object from lots of videos manually.

1.4 Scope of the Thesis

Computer vision is a vast area which includes scene reconstruction, event detection, video tracking, object recognition, 3D pose estimation, learning, indexing, motion estimation, image segmentation and image matting. We limit our work to scene reconstruction, image recognition, image matting, image segmentation and motion estimation. Image segmentation is needed to segment out moving objects from background. Motion estimation of the objects is needed to

predict the objects movement in a place where there is no camera. Image recognition recognizes an object in the multiple CCTV cameras. For this, we use SURF features. Video tracking tracks the objects in a video. Scene reconstruction reconstructs the scene of an object from different CCTV cameras. Moreover foreground detector, blob analysis is needed. Foreground detector is used to segment moving objects from the background. Thresholding is needed for this. It outputs a binary mask, where the pixel value of 1 corresponds to the foreground and the value of 0 corresponds to the background. Blob analysis detects connected components from a video frame. Image matting separates foreground image from the background.

1.5 Contributions

There is no complete process of scene reconstruction from multiple CCTV cameras to monitor a object. We develop a model that can take videos from multiple CCTV cameras of a fixed time and can extract the scenes only containing the interested object. Previously, there is no work for monitoring an object from multiple CCTV cameras automatically. If there is no camera at a place, then we develop a prediction algorithm based on the object's previous characteristics which predicts the path and movement of the object in a place where camera is missing. Normally, distance in meter unit between two pixels is considered as constant. But actually when the pixels goes apart from image center, distance in meter unit between two pixels gradually increases. Here, we develop a polynomial approximation between the pixel distance and the distance in meters so that the distance in meters between two pixels increases when they goes far from image center. This gives better approximation in our prediction algorithm. Besides, we develop a linear relationship between object size and its position with reference to camera. This makes our prediction algorithm more realistic. We also insert image in a background and move it with its various views according to need. This makes the prediction algorithm more realistic. These all are our new contributions to the thesis. We complete all the work in MATLAB. We use some previous works like tracking moving object, object recognition, image segmentation, image matting, SURF features and we have also our own contribution to the thesis described above. The combination of these two make our thesis a success.

1.6 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 1 is introduction which contains our topics and brief discussion about motivation behind the topics, summary of existing works, limitations and objectives. Chapter 2 includes some terminologies regarding our thesis. In chapter 3 , we describe the existing works in this field and limitations of the existing works. We discuss proposed methodology and related theory of our thesis in chapter 4. In chapter 5, we describe

the works done by us according to proposed methods and their experimental results. Finally, we make a conclusion of our thesis and describe our future works in chapter 6.

Chapter 2

Background

2.1 Introduction

We need to learn some things for reconstructing scenes from multiple videos. We need tracking moving objects and their velocity in each video files. We need blob analysis to find connected component in a video. We need to recognize an object into video files and track them. Image segmentation is needed to segment moving object from images. Foreground detection, features extraction etc are needed for video processing. We need image matting to separate foreground from background image. We need to learn this things to complete our thesis. These terms are discussed throughout the chapter.

2.2 Scene Reconstruction

Scene reconstruction is the process of reconstructing a digital version of a real world object from video sequences or scans of the object. It is a sub-domain of computer vision. Scene reconstruction may be of many variety like 3D scene reconstruction from multiple views of camera, scene reconstruction from multiple sequence images or videos. In our thesis, we define scene reconstruction as a combined scene or video from multiple videos from different cameras.

2.3 Video Tracking

Video tracking is the process of locating a moving object (or multiple objects) over time using a camera. It has a variety of uses, some of which are: human-computer interaction, security and surveillance, video communication and compression, augmented reality, traffic control, medical imaging and video editing. Video tracking can be a time consuming process due to the amount

of data that is contained in video. Adding further to the complexity is the possible need to use object recognition techniques for tracking, a challenging problem in its own right. Figure 2.1 shows tracking a moving object in a video frame. The bounding box shows the moving object in the figure. The moving object detection in the figure uses foreground detection method. [1]

Figure 2.1: Boundary box shows a moving object in a video frame.

2.4 Object Recognition

Object recognition is technology in the field of computer vision for finding and identifying objects in an image or video sequence. Humans recognize a multitude of objects in images with little effort, despite the fact that the image of the objects may vary somewhat in different view points, in many different sizes and scales or even when they are translated or rotated. Objects can even be recognized when they are partially obstructed from view. This task is still a challenge for computer vision systems. Many approaches to the task have been implemented over multiple decades.

2.5 Image Segmentation

In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super-pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. [6] Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics. In our thesis we need image segmentation to segment out moving object and to segment foreground object from background. There are many methods for image segmentaion like Ostu's

method, watershed segmentaion, K-means clustering etc. Figure 2.2 shows an original image and its segmented image . Watershed segmentation method is used to segment the image in the figure.

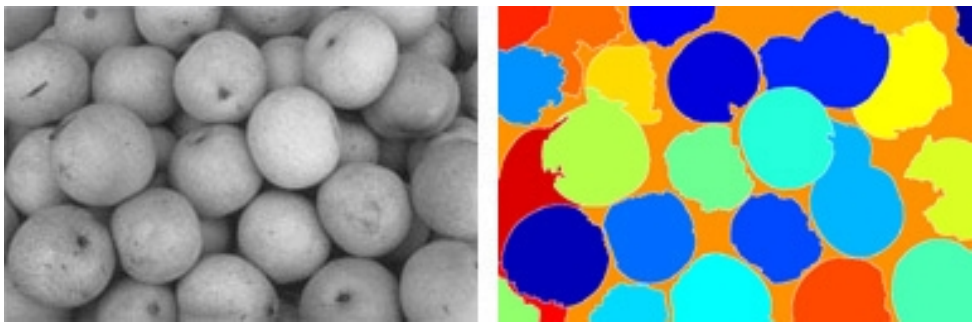


Figure 2.2: Left one shows the original image and right one is the segmented image.

2.6 Different Types of Image

Digital images are made of picture elements called pixels. Typically, pixels are organized in an ordered rectangular array. The size of an image is determined by the dimensions of this pixel array. The image width is the number of columns, and the image height is the number of rows in the array. Thus the pixel array is a matrix of M columns x N rows. To refer to a specific pixel within the image matrix, we define its coordinate at x and y. The coordinate system of image matrices defines x as increasing from left to right and y as increasing from top to bottom.Having defined the number of pixels, MxN, only provides a rectangular shape for our image. One more parameter, intensity, is needed to truly define an image. Each pixel has its own intensity value, or brightness. If all the pixels have the same value, the image will be a uniform shade; all black, white, gray, or some other shade. It is in the type of intensity used for each pixel that image types vary.This may many types like Gray Scale Image, RGB Image, Binary Image etc.Color images have intensity from the darkest and lightest of three different colors, Red, Green, and Blue. The various mixtures of these color intensities produces a color image. Black and white images only have intensity from the darkest gray (black) to lightest gray (white). This is called gray scale image. In addition to the intensity type of each pixel, the range of intensity values also varies. A binary image is a digital image that has only two possible values for each pixel. Typically, the two colors used for a binary image are black and white. The color used for the object(s) in the image is the foreground color while the rest of the image is the background color.

2.7 Thresholding

Thresholding is the simplest method of image segmentation. From a gray scale image, thresholding can be used to create binary images. The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity $I_{i,j}$ is less than some fixed constant T (that is, $I_{i,j} < T$), or a white pixel if the image intensity is greater than that constant. Thresholding is done on gray scale image. There are many variations of thresholding like contrast thresholding, multi-level threshlding etc. Figure 2.3 shows a binary thresholding example. The pixels which contribute to foreground image are white and the background pixels are black.

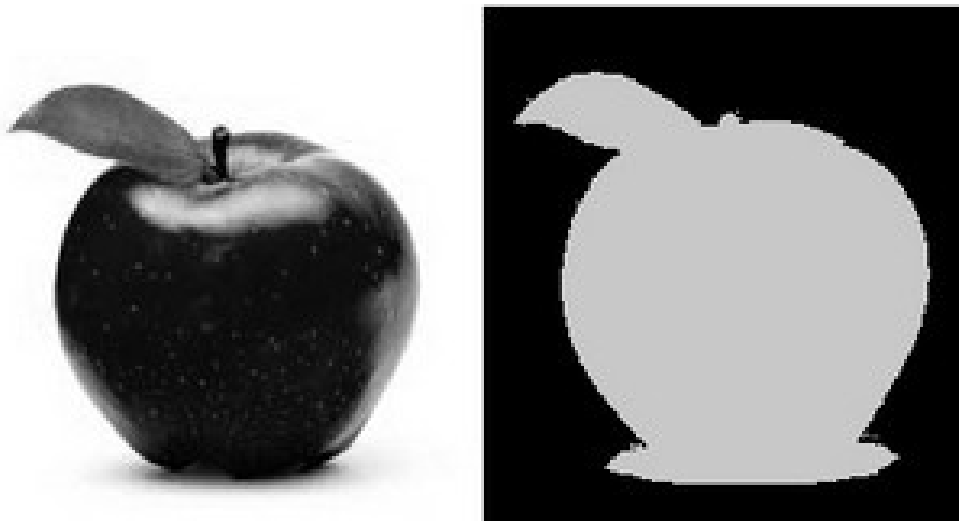


Figure 2.3: Left one indicates the original image and right one is its binary image.

2.8 Video Frame and Frame Rate

A single frame of video is literally an image, little different to any photo or graphic. But there's one difference, which is that the compression system is used to store video information, often includes a temporal factor, a time factor etc. They can compare two frames or more and only store the differences.

In motion pictures, television, and in computer video displays, the frame rate is the number of frames or images that are projected or displayed per second.

2.9 Mask Image

In image processing, a kernel, convolution matrix, or mask is a small matrix. It is used for blurring, sharpening, embossing, edge detection, image substraction, addition and more. This

is accomplished by doing a convolution between a kernel and an image. Masking involves setting some of the pixel values in an image to zero, or some other ”background” value. A mask image is simply an image where some of the pixel intensity values are zero, and others are non-zero.

2.10 Blob Detection

In computer vision, blob detection methods are aimed at detecting regions in a digital image that differ in properties, such as brightness or color, compared to surrounding regions. Informally, a blob is a region of an image in which some properties are constant or approximately constant; all the points in a blob can be considered in some sense to be similar to each other.

2.11 Foreground Detection

Foreground detection is one of the major tasks in the field of computer vision whose aim is to detect changes in image sequences. Many applications do not need to know everything about the evolution of movement in a video sequence, but only require the information of changes in the scene. Detecting foreground to separate these changes taking place in the foreground of the background. It is a set of techniques that typically analyze the video sequences in real time and are recorded with a stationary camera.

In our thesis, the foreground detector is used to segment moving objects from the background. It outputs a binary mask, where the pixel value of 1 corresponds to the foreground and the value of 0 corresponds to the background. Figure 2.4 shows a foreground detection example. Here, foreground detection is done by thresholding.

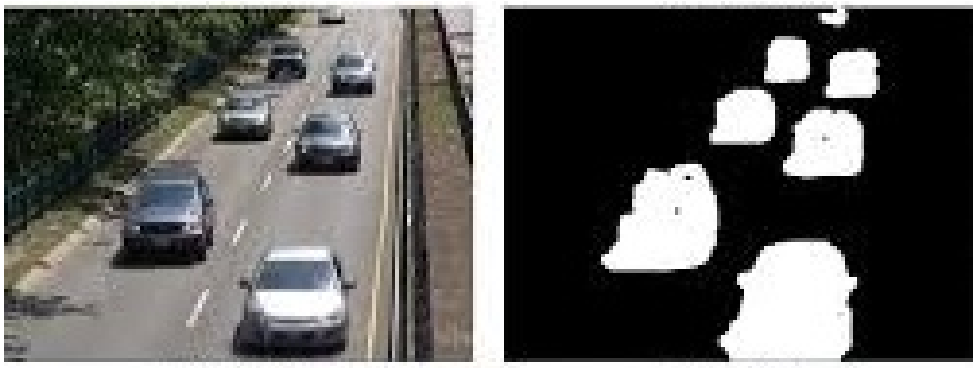


Figure 2.4: Left one is the original image and right one is its foreground image.

2.12 Feature Extraction

In computer vision, a feature is a piece of information which is relevant for solving the computational task related to a certain application. In simple words features are nothing but the unique signatures of the given image or unique properties that defines an image.

Feature extraction is the process of transforming the raw pixel values from an image, to a more meaningful and useful information that can be used in other techniques, such as point matching or machine learning. In Image processing, algorithms are used to detect and isolate various desired portions or shapes (features) of a digitized Image or video stream. It is particularly important in the area of optical character recognition.

Feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. When the input data to an algorithm is too large to be processed and it is suspected to be redundant (e.g. the same measurement in both feet and meters, or the repetitiveness of images presented as pixels), then it can be transformed into a reduced set of features (also named a feature vector). Determining a subset of the initial features is called feature selection.The selected features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data.

There are a quite lot of methods for performing feature extraction in image processing and they are domain dependent. Histogram of an image can also be used as feature of images but that is very naive technique. There are a few standard techniques like HOG, SIFT and SURF.

2.12.1 SURF Features

In computer vision, Speeded Up Robust Features (SURF) is a patented local feature detector and descriptor. It can be used for tasks such as object recognition, image registration, classification or 3D reconstruction. This patented algorithm is mostly used in computer vision tasks and tied to object detection purposes. SURF fall in the category of feature descriptors by extracting keypoints from different regions of a given image and thus is very useful in finding similarity between images.It is advertised to perform faster compared to previously proposed schemes like SIFT.

In [7], square-shaped filters are used as an approximation of Gaussian smoothing. (The SIFT approach uses cascaded filters to detect scale-invariant characteristic points, where the difference of Gaussians (DoG) is calculated on rescaled images progressively.) Filtering the image with a square is much faster if the integral image is used:

$$S(x,y) = \sum_{i=0}^{x} \sum_{j=o}^{y} I(i,j)$$

The sum of the original image within a rectangle can be evaluated quickly using the integral image requiring evaluation at the rectangleâs four corners. In SURF blob detector based on the Hessian matrix used to find points of interest. The determinant calculated from Hessian matrix is the measure of change around the point and with the maximum value of determinant points are selected.

According to [8], The steps of SURF algorithm contain three sections: intersect points detection, intersect points describing and intersect points matching. They are as follows.

- intersect points detection
- intersect points describing and intersect
- points matching

Intersect Points Detecting

Find features(keypoints) that are likely to be found in different images of the same object. Those features should be scale and rotation invariant if possible. corners, blobs etc are good and most often searched in multiple scales.

Intersect Points Describing

Find the right *orientation* of that point so that if the image is rotated according to that *orientation*, both images are aligned in regard to that single keypoint.

Intersect Points Matching

Computation of a descriptor that has information of how the neighborhood of the keypoint looks like (after *orientation*) in the right scale.

Figure 2.5 shows surf point in an image. The green plus points are some of surf points.

2.13 Image Matting

Image matting is an important technique in image and video editing, which was originally developed for film and video production. The task of image matting is to extract the foreground

Figure 2.5: The green plus point indicates surf points in the image.

object from an image by estimating a color and an opacity for each pixel of the foreground object. Figure 2.6 and Figure 2.7 shows the foreground object making the background to black. The foreground pixels are made white and the background image are made black.

Figure 2.6: Input image with strokes

Figure 2.7: Image matting result

2.14 Summary

We need these terminologies described above in our thesis. Image segmentation is needed to segment out objects in a frame. Image recognition recognizes an object in the multiple CCTV cameras. Blob analysis will detect connected components from a video frame. Foreground detector is used to segment moving objects from the background. Thresholding is needed for this. Video tracking tracks the objects in a video. Surf features is used to recognize the input object. Image matting is used to separate foreground from background. We need to extract features from the object to match the object. All these terms are needed to understand our thesis works easily.

Chapter 3

Literature Review

3.1 Introduction

In this chapter, we discuss about the research works we have been following so far. We also describe the significance and limitations of methods we use in our research. Our goal is to reconstruct a scene using videos from multiple static CCTV cameras. Accomplishing this, we detect and track moving objects in multi-camera supervision system. We also detect and show the path that the objects used. For these, we use the most acceptable methods which are described throughout this chapter.

Tackling the situation like missing of any part of the scene due to technical problems, we try to construct that missing part. But we do not find any significant work related to object's movement prediction. So, we use our own method for missing scene reconstruction.

3.2 Moving Object Detection

Moving object detection is a fundamental step in several applications such as human behavior analysis for visual surveillance, human computer interaction, and model based action recognition. One of the significant intuitions in our research is tracking and analyzing human behavior. So, at first we detect moving object. There are several methods for moving object detection. Background subtraction is one of the popular approaches. In our research, we use background subtraction based method [2].

In [2], authors use region based background subtraction. It performs well in presence of dynamic backgrounds without any noticeable loss in the shape of the moving objects. The approach does not require ideal background frames for background model initialization. So, it is very helpful in circumstance like multiple moving objects in a frame.

3.3 Moving Object Tracking

The goal of a video surveillance system is to be able to detect and track moving entities (e.g. people, vehicles), to characterize typical behaviors. We want to achieve that. Moreover an automatic surveillance system is able to learn typical behaviors from unsupervised video data. It does not involve specific knowledge about the actions performed by the humans.

In [9], authors research on multiple motion fields estimated from trajectories. They find it as an efficient tool to describe the movement of objects. It also allows an automatic classification of activities in the scene. The proposed methodology is reliable for a fully automatic extraction of multiple motion fields. This method is really helpful to track objects in denser environments such as crowds of moving people.

In [10], authors develop an algorithm for detecting moving objects and tracking in video sequence using color feature. The proposed algorithm improves the tracking accuracy. It decreases the tracking failures in the video tracking process when the tracked object changes orientation over time. This method also works in denser environments such as crowds of moving people.

3.3.1 Multi-camera Person Tracking

The human capacity for simultaneous multiple video monitoring is limited. Moreover, an object located at the point with overlapping visual fields of two or more cameras appear simultaneously on several different views. Thus it complicates tracking. Therefore, authors of [11] try to develop a computer system that can analyze videos sent from multiple cameras simultaneously. After analyzing videos, it can present the information to user in convenient form. But it fails to track objects if it changes orientation over time.

In [12], authors propose a better method for multi-camera person tracking. This method requires necessary tasks such as pre-processing recordings, detecting a person in motion and finding the contact point between person and floor. Authors consider the person located in the folded multi-camera field of view. They make homographic transformation of a point where person contacts the ground in observed space. In this way, they determine the person's position in 2D space plan. Implemented application runs fast enough in real time operating system. But, this method is not too much useful in noisy and crowded environment.

Authors of [11] propose a suitable method for crowded environments which is more like to real life situation. Reliably tracking people throughout a camera network is an important capability in areas such as law enforcement, homeland protection, and health-care. Authors provide and evaluate GE Global Researchâs tracking system against a subset of [13]. Unfortunately, robust and scalable tracking remains elusive. Automatic surveillance of whole cities and specific urban

sites is challenging for reasons such as site size, occlusion, clutter, and population density.

[11] presents the results of the GE Global Research video tracking system applied to [13]. With its real-time running, system performs well on the tasks of counting and density estimation. The tracking system is able to track individuals across multiple camera views but performed best on view 001 of [13]. Authors use appearance information to track selected individuals through severe crowds. The system performs well in all but the most severe conditions.

3.3.2 Multi-camera Multi-Person Tracking

Automatic initialization and tracking of multiple, potentially changing number of persons in real situations are a classic but challenging topic in computer vision. Along with the development of object detection approaches, the tracking-by-detection framework is adopted widely for multi-object tracking scenarios.

Authors of [14] implement single-camera based multi-object tracking system. But tracking performance is not so good. Reasons behind limitations are large portion of false positive and missing detection caused by severe occlusions or bad lightness conditions. By contrast, tracking with multiple cameras, one view can change information with others and compensate the data scarcity. However, data association from multiple cameras occurs an extra difficulty known as ghost effect [15] caused by triangulation of objects in 3D space.

In [16], authors propose a global optimization approach. They use two graphs for multi-person tracking in multiple calibrated camera systems. Authors adopt hypothesis on the ground plane to reconstruct from 2D detection from all available views. Afterwards, they extract track fragments (Tracklets [17]) by finding the min-cost paths in a so-called hypothesis graph. Finally, complete tracks are generated by linking those tracklets through a so-called tracklet graph. The impact of calibration and object detection errors limit the result of this method. In this situation, more restrict constraints can improve the precision of recognizing of identical objects.

3.4 Path Detection

We track a person closely observed by multiple static video cameras and then reconstruct his full path under this video surveillance system. So, path detection is a crucial part of our research. Authors of [18] consider the problem of learning the routes or paths taken by pedestrians walking through outdoor scenes. The routes and paths are represented by a spatial model.

The motivation for building such models is threefold. **Firstly**, constructing a log of movement patterns over long periods of time (i.e. weeks) requires an efficient method to encode and annotate individual tracks. **Secondly**, accumulating tracks over a long time period establishes a

norm of typical movements. This norm support the recognition of atypical or unusual movement and patterns of behaviour. **Finally**, the information can be used to support the tracking process and enhancing the predictive capabilities of the system to correspond objects over successive image frames. This gives the system the opportunity to predict forward many frames, based on the current location and direction. We use this method in our research.

3.5 Scene Reconstruction

There are many effective approaches to reconstruct dense dynamic shape in controlled environments with static backgrounds and illumination. A common assumption of widely used visual-hull based reconstruction approaches is prior foreground/background segmentation.

Authors of [19] introduce a general approach to dynamic scene reconstruction from multiple moving cameras without prior knowledge or limiting constraints on the scene structure, appearance, or illumination. Existing techniques for dynamic scene reconstruction assume fixed camera and calibrated background. These approaches are not robust for general dynamic scenes captured with sparse moving cameras. Tackling these limitations, authors introduce an automatic method for initial coarse dynamic scene segmentation and reconstruction without prior knowledge of background appearance or structure. The proposed approach also eliminates the requirement for prior knowledge of scene structure and appearance.

All the existing works show the reconstruction of a single image using images from multiple static/dynamic camera. But, we need to construct a single video scene using the data from multiple static camera. So, we use our own algorithm for scene reconstruction.

3.6 Movement Prediction

Predicting human motion using 3D Cartesian description is a challenging problem in human movement science. A major difficulty is prediction of a large number of degrees of freedom. It simultaneously optimizes various human performance measures with respect to physical and physiological constraints. In our research, we predict the movement of the given object in the missing scenes.

The ability to recognize humans and their activities by vision is key for a machine to interact intelligently and effortlessly with a human-inhabited environment. In [20], authors conduct a survey on human movement. They use this data to make a satisfactory result on movement prediction.But the scope of this survey is limited to work on whole-body or hand motion. They emphasis on discussing the various methodologies. This methodologies are grouped in 2-D approaches with or without explicit shape models and 3-D approaches. They conclude with

some thoughts about future directions.

Nowadays, Many researchers are working on path prediction to find efficient network resource management schemes in the context of mobile and wireless computing. Path prediction allows the network and services to further enhance the quality of service levels that the user enjoys. In [21], authors present a path prediction algorithm that exploits human creatures habits. They try to present a novel hybrid Bayesian neural network model for predicting locations on Cellular Networks. They compare experimental results of the proposed Bayesian Neural Network with 5 standard neural network techniques in predicting both next location and next service to request. The result of Bayesian training is a posterior distribution over network weights using Markov chain Monte Carlo methods (MCMC).

Authors of [22] propose a method for predicting pedestrian movement on the basis of a mixed Markov-chain model. MMM takes into account a pedestrian's personality as an unobservable parameter. It also takes into account the effects of the pedestrian's previous status. In comparison with methods based on a Markov-chain model (MM), the proposed MMM-based prediction method is substantially more accurate. We take motivation from these path predicting methods to make our own movement prediction algorithm.

After reviewing on the previous works on *Movement Prediction*, we can come to a decision that there is no mention-worthy successful previous work on human movement prediction rather than they are only experimental surveys. Some are insignificant attempts using Markov Model.

3.7 Limitations

In this chapter, we describe the existing knowledge available for our research. But, this is not enough. We perform more complex works such as, velocity detection, combining scenes from multiple cameras and reconstructing missing scenes. Reconstructing parts of scene unavailable due to technical problems, we predict object's movement using its past behaviour. In real life, movement does not depend only on past behaviour. It depends on many other factors such as object's personality, physical and psychological state etc. So, there are so many degree of freedoms and random variables to predict the movement. We interpolate the object's possible location in the scene under reconstruction. As there is no existing significant work on movement prediction, it is the main challenging part in our research.

For completing our research, we amalgamate the existing works that are required for our work and then accumulate all these scattered exiting works to support our research.

3.8 Summary

Throughout our research, we use scenes from multiple static cameras, recognize the object, extract scene containing only the object, combine scenes, detect velocity of the objects, reconstruct missing scenes and, for that we predict the movement of the person tracked.

In this chapter, we introduce the exiting works that is required for our research and also showed the limitations of those works that we try to overcome for completing our research. We describe some existing techniques for moving object detection, moving object tracking, path detection and scene reconstruction, movement prediction. These are the most challenging issues in our research.

Chapter 4

The Proposed Methodology

4.1 Introduction

The human capacity for simultaneous multiple video monitoring is limited. A person responsible for supervision must watch and analyze all videos in order to monitor persons moving in the controlled area. Because of the limited view, and limited operating angle of cameras, often is necessary to use multiple cameras with partially overlapping views or no overlapping, which combined can monitor whole area of the interest. Therefore, there is a need to develop computer systems that can analyze videos sent from multiple cameras simultaneously, and present the information to user in convenient and easy understandable form. Moreover if the object is in an area where there is no camera or the camera is damaged there, we want to predict the movement of the object in that area from its previous or later characteristics.

We propose to a method to build the system. In this chapter, we describe problem formulation, our work plan, and details of it.

4.2 Problem Description

We build a computer systems that can analyze videos sent from multiple cameras simultaneously, and present the information to user in convenient and easy understandable form. We will take all the video clips from static CCTV cameras at the time of incident we want to explore. Each CCTV camera covers an area. We will take those CCTV cameras in which areas we want to explore the object of interest. Suppose there need N cameras to cover the whole areas and the incident starts at time t and duration of the incident is d. So the incident will end at time $t + d$. We want the movement or characteristics of an object in this time and made a total scene of the object in that time. Let the object id is i at camera 1. When the object is out of scope

from camera 1 then we have to search it in camera 2. Thus from camera $c = 2$ to $N - 1$ when the object is out of scope of camera c we have to search the object in camera $c - 1$ and $c + 1$. Thus by searching all the cameras , we track the object and collect all the frames containing the object. It may be happened that an object may appear in a place where there is no camera available or the camera is damaged there. We predict the movement in that area for the object using its previous or later characteristics. So, we build a prediction model for the object in this case. Thus by collecting actual and predicted data , we can get the total movement and characteristics of the object in the interested area of that time. This is what we want to do. Figure 4.1 is the flowchart of our total computer system.

At first, when we start the process, we take videos as input sent from multiple static CCTV cameras at a fixed time. Then, we track every moving object in the first camera and their velocities in meter per second unit. We take input the object which movement we want to explore. We search the object in all the cameras and output them into separate video files only containing the object. It may be happened that there is a place where there is no camera or the camera is damaged, then we need to predict the object's movement in that place. If there is any missing camera, then we predict the movement of the object in that place using its previous velocity. Then, we combine all the actual and predicted data to form them as a total sequencial scenes of the interested object. Finally, we present the scenes to user. This is our whole process.

4.3 Work Plan

We divide our work to several steps. They are described below :

1. tracking moving objects in a video file
2. tracking velocity of each object
3. determining approximate time needed for every possible path where camera not available or damaged
4. recognizing the input object over the first video file
5. recognizing the input object over the other video files
6. predicting movement where no camera is available or camera damaged
7. collecting all the actual and predicted data containing the object
8. combining all the actual and predicted data to make the total scene

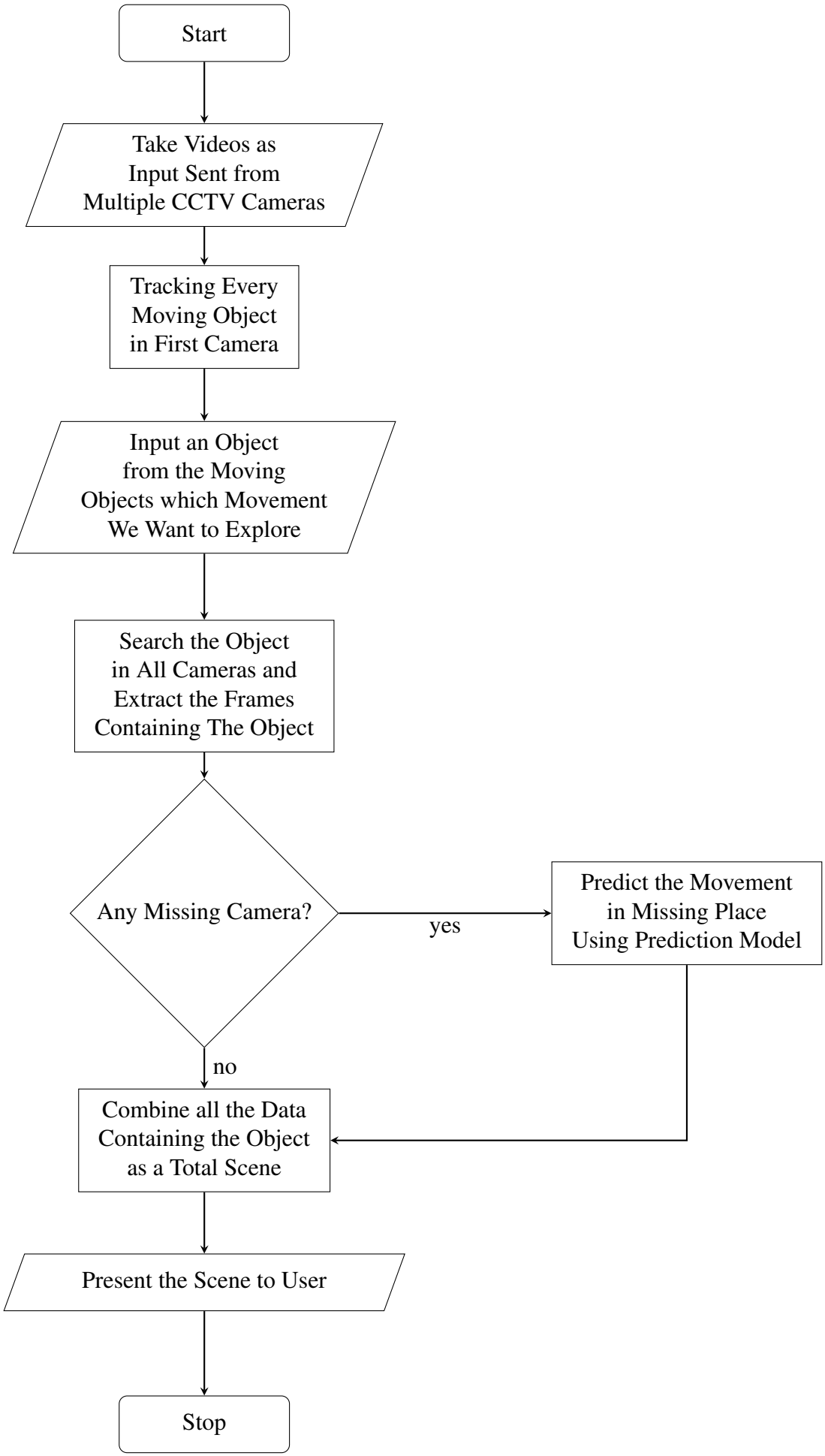


Figure 4.1: This is the flow chart of our total system

4.4 Descriptions of Work Plan

In this section, we give descriptions of our proposed working methodology and related theory. In chapter 5, we give experimental results of these methods.

4.4.1 Tracking Moving Objects in a Video

First we simply need tracking moving objects in a video file and giving an identity of each moving object. We need to separate of the each moving objects to separate the interested object which movement we want to explore. From these separated moving objects, we then need to input an object which movement we want to explore for all the given cameras. We have showed how to perform automatic detection and motion-based tracking of moving objects in a video from a stationary camera. The problem of motion-based object tracking can be divided into two parts:

1. Detecting moving objects in each frame
2. Associating the detections corresponding to the same object over time

The detection of moving objects uses a background subtraction algorithm based on Gaussian mixture models. [1] Morphological operations are applied to the resulting foreground mask to eliminate noise. Finally, blob analysis detects groups of connected pixels, which are likely to correspond to moving objects.

The association of detections to the same object is based solely on motion. The motion of each track is estimated by a Kalman filter. The filter is used to predict the track's location in each frame, and determine the likelihood of each detection being assigned to each track.

Track maintenance becomes an important aspect of this procedure. In any given frame, some detections may be assigned to tracks, while other detections and tracks may remain unassigned. The assigned tracks are updated using the corresponding detections. The unassigned tracks are marked invisible. An unassigned detection begins a new track.

Each track keeps count of the number of consecutive frames, where it remained unassigned. If the count exceeds a specified threshold, the example assumes that the object left the field of view and it deletes the track.

Figure 4.2 is the flowchart of tracking moving objects in a video frame.

A short description of the total procedure is given below :

1. **Create System Object** : This is used for reading the video frames, detecting foreground objects, and displaying results.

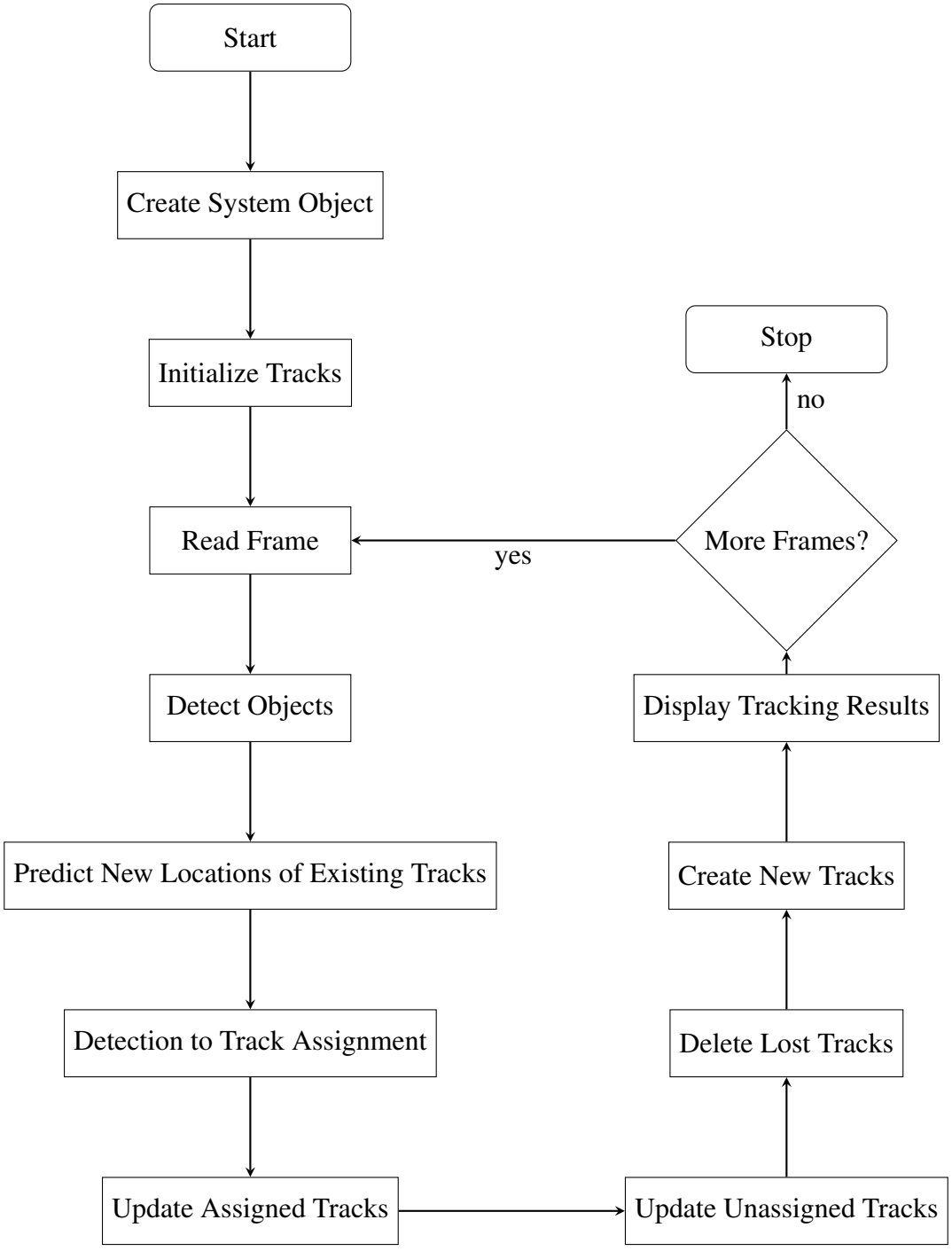


Figure 4.2: This is the flow chart for tracking Moving objects in a video file

2. **Initialize Tracks** : This creates an array of tracks, where each track is a structure representing a moving object in the video. The purpose of the structure is to maintain the state of a tracked object. The state consists of information used for detection to track assignment, track termination, and display.

3. **Read Frame** : This reads the next video frame from the video file.

4. **Detect Objects** : The function returns the centroids and the bounding boxes of the detected objects. It also returns the binary mask, which has the same size as the input frame. Pixels with a value of 1 correspond to the foreground, and pixels with a value of 0 correspond to the background. The function performs motion segmentation using the foreground detector. [1] It then performs morphological operations on the resulting binary mask to remove noisy pixels and to fill the holes in the remaining blobs.

5. **Predict New Locations of Existing Tracks**: This uses the Kalman filter to predict the centroid of each track in the current frame, and update its bounding box accordingly.

6. **Assign Detections to Tracks** : This assigns object detections in the current frame to existing tracks is done by minimizing cost. [23, 24]

7. **Update Assigned Tracks** : The procedure updates each assigned track with the corresponding detection.

8. **Update Unassigned Tracks** : This marks each unassigned track as invisible, and increases its age by 1.

9. **Delete Lost Tracks** : This deletes tracks that have been invisible for too many consecutive frames. It also deletes recently created tracks that have been invisible for too many frames overall.

10. **Create New Tracks** : This creates new tracks from unassigned detections assuming that any unassigned detection is a start of a new track.

11. **Display Tracking Results** : This draws a bounding box and label ID for each track on the video frame and the foreground mask. It then displays the frame and the mask in their respective video players.

12. **Are More Frames Available** : This checks if more video frame is available in the video clip. If it is yes, then it goes to read frame procedure and continues. If it is no then , stop the process.

This is the procedure of detecting moving objects in a video files and tracking them in consecutive video frames.

4.4.2 Tracking Velocity of Each Moving Object

After tracking moving objects, we track velocity of each object. The velocity is needed when we want to predict an object's future movement where there is no camera available or camera is damaged. If we know the previous velocity of an object then we can give an approximation of the object's movement in the missing cameras. We have a boundary box associated with each object which have a centroid. In every frame, we calculate the displacement of the centroid from the previous frame of the object. This value is velocity in pixel per frame for that object in that moment. But we need velocity in meter per second unit. To get this, we need to know the distance between two pixels in meter. But the distance between two pixels in meter is not always equal for all places in a video frame. The distance between two pixels are less in center. The more we go far from the center, the more the distance between two pixels increases. So, the relation among them is not linear. So, their relation cannot be expressed in terms of linear equation. We build a polynomial equation of x. Here, we use four degree polynomial for simplicity. We can use higher degree polynomial but for simplicity and less computational complexity we use four degree polynomial.

$$y = ax^4 + bx^3 + cx^2 + dx + e \tag{4.1}$$

Here,x is the distance of the pixel from the center of the video frame in pixels and y is the distance of the pixel from the center of the video frame in meters.But now, we have to estimate the values of a, b, c, d and e. For this, we need some values of the x and y. We have to collect some real life data points (x,y) . Then by using these data points , we can calculate the values of a, b, c, d and e. This can easily be done with MATLAB ***polyfit*** function. Then, x can be calculated as follows.

$$x = \sqrt{(centerX - centroidX)^2 + (centerY - centroidY)^2} \tag{4.2}$$

Here, $centerX$ and $centerY$ are the center of the frame of the video and $centroidX$ and $centroidY$ are the center of the bounding box of the moving object. Now, let $x1$ and $x2$ be the distances of the centroid of the current frame and previous frame of a moving object which can be calculated by equation 4.2. From the $x1$ and $x2$, we can calculate $y1$ and $y2$ using equation 4.1. The difference of $y1$ and $y2$ is the actual distance covered in meters by the object in that frame. Let z be the actual distance covered in meters by the object in that frame.

$$z = |x_1 - x_2| \tag{4.3}$$

So, z is in meter/frame unit. If we multiply the frame rate per second r by z, then we get the velocity v_i in meter/second unit in that frame.

$$v_i = z \times r \tag{4.4}$$

Now, if we sum up of all the values v_i over the n number of frames in that video file for the object, then we can get total distance d_t covered by the object in that video file.

$$d_t = v_1 + v_2 + v_3 + ... + v_n \tag{4.5}$$

Then, finally, if we divide d_t by n, then we can get the average velocity v_{avg} of the object in that video file. This is how the average velocity of all the moving objects can be determined in a video file.

$$v_{avg} = \frac{d_t}{n} \tag{4.6}$$

4.4.3 Determining Approximate Time Needed for Every Possible Path Where Camera not Available or Damaged

If there is no camera in a place or the camera is damaged then we need to predict the objectâs movement in that place. For this, we need a background image of that place and all the possible path specifications where the object can move. If given average velocity of the object which we have calculated in 4.4.2, then we can calculate the time needed for each path by the input object. These times will be used later to determine the path covered in our prediction model to predict the movement which we will show later in **??**.

Total distance d_t of a path can be calculated using equation 4.5 and we know the average velocity v_{avg} of the object from 4.4.2. So, the approximate time t needed for the object for this path is given as follows.

$$t = \frac{d_t}{v_{avg}} \tag{4.7}$$

4.4.4 Recognizing the Input Object over the First Video File

Now, we have the input object which movement we want to explore in the whole system. So, we need to search the specified object in multiple cameras and we want to track the object's total movement from multiple CCTV cameras. After we track it in first CCTV camera, we have to track the object in all cameras.So, we need a object recognition algorithm.
For recognizing the input object in a video file , we use SURF(Speeded Up Robust Features) feature. Speeded up robust features (SURF) is a patented local feature detector and descriptor. It can be used for as object recognition. It is partly inspired by the scale-invariant feature transform (SIFT) descriptor. The standard version of SURF is several times faster than SIFT and claimed by its authors to be more robust against different image transformations than SIFT. That is why we used it in our thesis for recognizing object.
SURF algorithm is based on the same principles and steps as SIFT; but details in each step are different. The algorithm has three main parts: detecting surf features, extracting surf features and matching features. We have used MATLAB built-in function for SURF. The main parts of the algorithm are described below.

1. **Detecting SURF Features** : We use detectSURFFeatures(I) function of MATLAB which returns a SURFPoints object, points, containing information about SURF features detected in the 2-D grayscale input image I. The detectSURFFeatures function implements the Speeded-Up Robust Features (SURF) algorithm to find blob features. [25]

2. **Extracting SURF Features** : It extracts feature vectors, also known as descriptors, from a binary or intensity image. Descriptors are derived from pixels surrounding an interest point. They are needed to describe and match features specified by a single point location. MATLAB uses extractFeatures(I, POINTS) functions which returns features, an M-by-N matrix of M feature vectors, also known as descriptors. Features can also be a binaryFeatures object. Each descriptor is of length N. The function also returns M number of valid points corresponding to each descriptor. [25–27]

3. **Matching Features** : We detect and extract SURF features of the input object image and the video frame image in each video frame. Now the task is to match the features between these two images to recognize the objects. The matchFeatures(input_object, video_frame) function in MATLAB returns P-by-2 matrix, indexPairs, containing indices to the features most likely to correspond between the two input feature matrices. [28–30]

So, we get the matching features from the above procedure. Now, we draw a boundary box around the detected image in the video frame . Thus for every video frame , we repeat this and extract those frames which have the input object. Then finally we make a video file from those extracted frames which contains only the input image. Here, an important point is that when we detect the input image in a video frame, then the detected image becomes the new input image. This is because in a video file, the detected image may be changed from the input image because of the movement of the object in the video. So every time, we make the current recognized object as the new input object.

4.4.5 Recognizing the Input Object over the Other Video Files

We need to track the specified object in the multiple cameras. If there are N cameras where we want to explore , we have to search the object in camera 1. This is done by object recognition described in 4.4.4. Then we have to search the object from 2 to N cameras and track its movement in those cameras. Every time we repeat the same process, extract the frame containing the object and output them into a new video file.

4.4.6 Movement Prediction

Prediction is one of most important part of our thesis. We predict the movement of the input object where there is no camera available or the camera is damaged. For this, we need a prediction algorithm to give prediction of the movement of the object using its past behaviours, in this case its past velocity. In step 4.4.3, we calculate the approximate time given the background image of the area where camera being not available from the average velocity of the object in its past or future video camera for all the possible paths. This information is needed in our algorithm. But before giving a prediction algorithm, we need some pre-processing steps. Then, we develop the algorithm. So, the prediction steps involves two part. They are as follows.

- pre-processing
- prediction algorithm

They are described below elaborately.

Pre-processing

The pre-processing steps are given below :

1. **Background Image** : First of all, we need a background image of the place where camera is not available. In this background image, we have to insert the specified object and move it.

2. **Removing Background** : We have to place the object in a background and move it. But the object contains in a bounding box of rectangle shape. So, there remain background pixels with the object. When we place the object, we need to remove the background from the bounding box. Therefore, we will set the background color of the bounding box to zero so that only the foreground object appears in it and other pixels are black. For this, we have used several methods.

    - **Image Matting** : Matting refers to the process of extracting foreground object from an image. We have got several codes on the internet for image matting and they can successfully separate the foreground from the background but they all uses a trimap image as input. But we don't get any source for generating trimap image of an image. So, we don't have the trimap input of our object. Therefore, the image matting procedure for getting the foreground image fails.
    - **Image Segmentation** : Image segmentation can be used to get the foreground image from the bounding box. Firstly, we get the segmented regions in the image. Then, from the segmented regions we take only the foreground part by clicking on the regions by the cursor. Experimental results of this is discussed in chapter 5.
    - **Image Cutting** : Another procedure can be used by foreground image cutting. We write a code which takes the image as input and then draw the region by mouse and outputs the foreground image containing the regions.

3. **Multiple Images** : We have to place the object onto a background and move it. But if we move the same image over time then it is not realistic. Suppose we have a front view of the image but the object is moving in far away then we need the back view of the image. This may happen for side views also. Again we need at least two or three positions of the image for every view. We can easily get this by taking some images of the input image from his previous movements and remove background from these images using the procedure described above.

4. **Path Specifications** : We need to know in which paths an object can move and we have to find out the optimal path to predict the movement . So , the path specifications have to be given up . We write a code in which we can specify the all the possible paths in a background image. These will be needed in our prediction algorithm later. We can also

the calculate the distance in meters covered by a path. The distance in meters between two points can be found can be found by using equation 4.1, 4.2 and 4.3. And the total distance can be found by summing up for all points across a path.

5. **Inserting the Object and Movement** : At this point, we need to insert the foreground object in the background image. We insert it in a point and move it along a path. In this regard , how it will move to a path will be discussed in the prediction algorithm.

6. **Reshaping the Object** : The object size is not always same. Suppose if it moves towards the camera then object size will increase, if it goes far away from camera then object size will be greater. If their are sideways movement then object size will also change. We write a code to adjust the object shape according to its positions with reference to camera. Let, r_1 and r_2 is the ratio of object size change per pixel. w, Δw is the image width and change along width from the previous pixel to current pixel.

$$\Delta w = r_1 \times \text{change of pixels along width} + \text{r2} \times \text{change of pixels along height} \tag{4.8}$$

$$w = w + \Delta w \tag{4.9}$$

This is the change along width. In MATLAB change along height can be changed proportional to width.

Movement Prediction Algorithm

Now we have come to a point where we need an algorithm for predicting the optimal path of an object and moving the object across the path where there is no camera available or the camera is damaged. For this, we develop an algorithm which is shown in algorithm 1. All the inputs of the algorithm are made in the pre-processing steps. Now in the algorithm, we search for an optimal path which gives a minimum time error with respect to the missing time in any place where camera not available. Then , we insert the object in the background in the initial position. After that, we calculate total frame number with respect to the missing time and calculate actual distance in pixel per frame for each frame. Now, for each frame, we move the object same to the distance, reshape the object, update the object if view changes, insert it onto the background and save the frame to video file object. Thus we got the approximate movement of the object in the place where there is no camera or the camera is damaged.

Algorithm 1 Path Prediction and Object Movement Algorithm

Input: Initial Pixel x,y**; All Paths,** $P = \{P_1, P_2, ..., P_m\}$**; Object's Previous Velocity,** v**; Missing Time,** t**; Frame Rate,** r**; Background Image,** $background$**; Object Foreground Images,** $objectArray$

Output: Video File Object, $videoObject$

$timeError \leftarrow infinity$
$optimalPath \leftarrow -1$
$optimalPathLength \leftarrow -1$
$videoObject \leftarrow null$
for $i = 1$ to m **do**
  $pathLength \leftarrow path\ length\ of\ P[i]$
  $time \leftarrow pathLength/v$
  if $|time - t| < timeError$ **then**
    $timeError \leftarrow time$
    $optimalPath \leftarrow i$
    $optimalPathLength \leftarrow pathLength$
  end if
end for
$noOfFrames \leftarrow t \times r$
$movingVelocity \leftarrow optimalPathLength/t$
$movingVelocityInPixel \leftarrow calculate\ movingVelocity\ in\ pixels\ per\ frame\ unit$
$object \leftarrow objectArray[1]$
$background \leftarrow insert\ the\ object\ in\ background\ at\ pixel\ (x, y)$
$videoObject[i] \leftarrow background$
for $i = 2$ to $noOfFrames$ **do**
  if $an\ update\ needed\ of\ the\ object$ **then**
    $object \leftarrow find\ suitable\ object\ from\ objectArray$
  end if
  $object \leftarrow reshape\ the\ object$
  $background \leftarrow move\ object\ movingVelocityInPixel\ units\ along\ optimalPath$
  $videoObject[i] \leftarrow background$
end for
return $videoObject$

4.4.7 Collecting All Actual and Predicted Data Containing the Object

So far, we recognize object, search the object in the given N cameras using 4.4.4, 4.4.5. We also build a prediction model to predict the movement of the object where camera not available or damaged using 4.4.6. We track them and give them output to N separate video files. Now our job is to collect all the separate video files those contains the input object so that these video files can be combined later to make a single video of the input object for all the cameras and for prediction.

4.4.8 Combining All the Actual and Predicted Data

Now, we have all the videos containing the object and the prediction model videos file if we need to predict the object's movement in a area where there is no camera or the camera is damaged. We then have to combine all the N video files to make a sequencial scene or video of that object and add the prediction data for the object where there is no camera with the actual data. In that point , we can get the total movement and characteristics of the object in the interested area with actual data and predicted data if needed which we want to acquire.

4.5 Summary

As visual surveillance system collect a huge amount of data from many different scenes or surveillance cameras, people expect the total activities of an object with little human labeling effort as possible. So, we have proposed a computer systems that can analyze videos sent from multiple cameras simultaneously, and present the information to user in convenient and easy understandable form. We make a plan or schedule and all related theory for this purpose in this chapter for completing the tasks so that we can achieve our desired objectives.

Chapter 5

Experimental Results

5.1 Introduction

We propose an automated computer system of tracking object in multiple videos from CCTV cameras and reconstruct the videos as a single scene or video so that it takes too much less effort for the person who is in supervision. Then we make a work plan to accomplish our tasks to get the desired objectives. Thus we can get the total human or object activities in an environment in a minimum human effort if we can complete our our tasks successfully.

In this chapter we describe experimental results of our proposed method. Experimental results is shown for each of the tasks descriped 4. But before that we shortly describe how we collect our data for experiment and the experimental tools we use.

5.2 Data Collection

We collect some videos from the CCTV cameras from Department of Computer Science & Engineering, BUET. We do all our experiments based on these videos collected from the CCTV cameras. From now on, our experimental results is shown on these videos.

5.3 Experimental Tool

We complete all the programming and simulations in MATLAB. Computer Vision System Toolbox of MATLAB provides algorithms and tools for video processing work flows. We can read and write from common video formats, perform common video processing algorithms, and display results with text and graphics burnt in to the video. Video processing in MATLAB uses

System objects, which avoids excessive memory use by streaming data to and from video files. Besides we need object detection of the interested objects over multiple cameras. Built-in object detection algorithms is available in MATLAB. That is why, use MATLAB for our experiment.

5.4 Experimental Results

We do our research works according to our plan we propose earlier. Their experimental results are shown in the following subsections.

5.4.1 Experimental Results of Tracking Moving Objects in a Video

We do our experiment and simulation in MATLAB. Figure 5.1- 5.3 show our experimental result of tracking moving objects with a boundary box in a video frame described in 4.4.1 Here, we show three video frames. In each frame, there are boundary boxes with yellow colour indicating the moving objects. We insert a boundary box in each of the moving objects in each frame in a video file by our code.

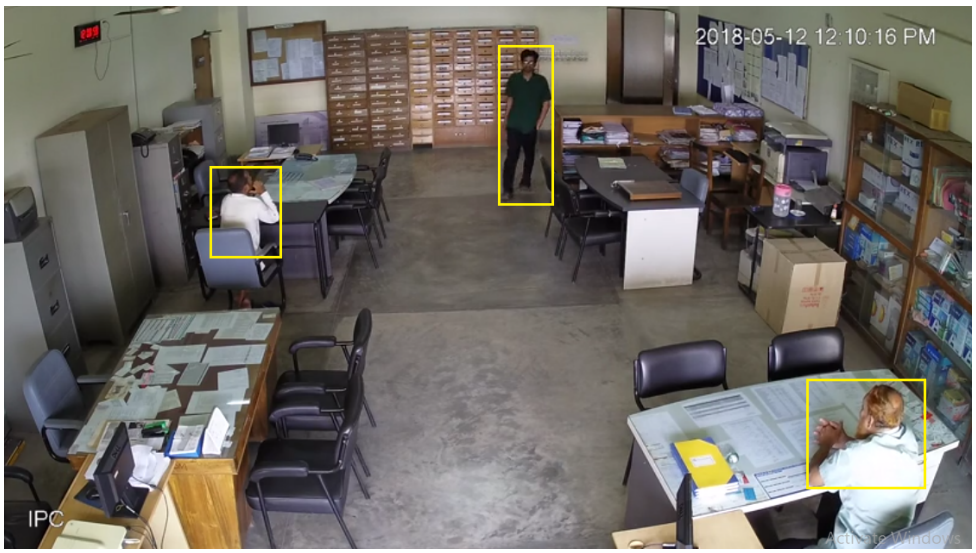


Figure 5.1: Boundary boxes with yellow colour show the moving objects.

Our experiment which is done in MATLAB shows good result of tracking moving objects with a boundary box in video frames.

5.4.2 Experimental Results of Tracking Velocity of Each Moving Object

Figure 5.4- 5.6 shows our experimental result of tracking moving objects with a boundary box in a video frame with instantaneous velocity in meter per second unit which is described in

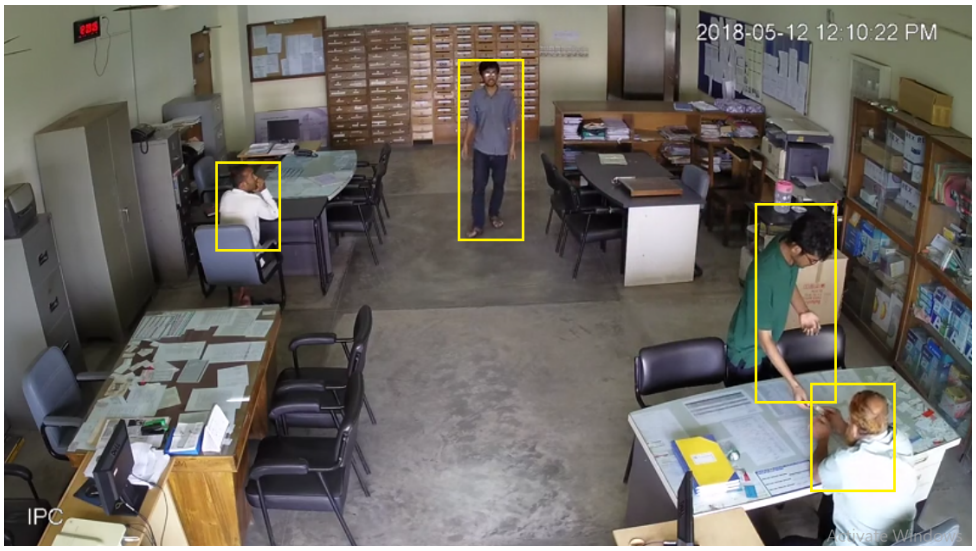


Figure 5.2: Boundary boxes with yellow colour show the moving objects.

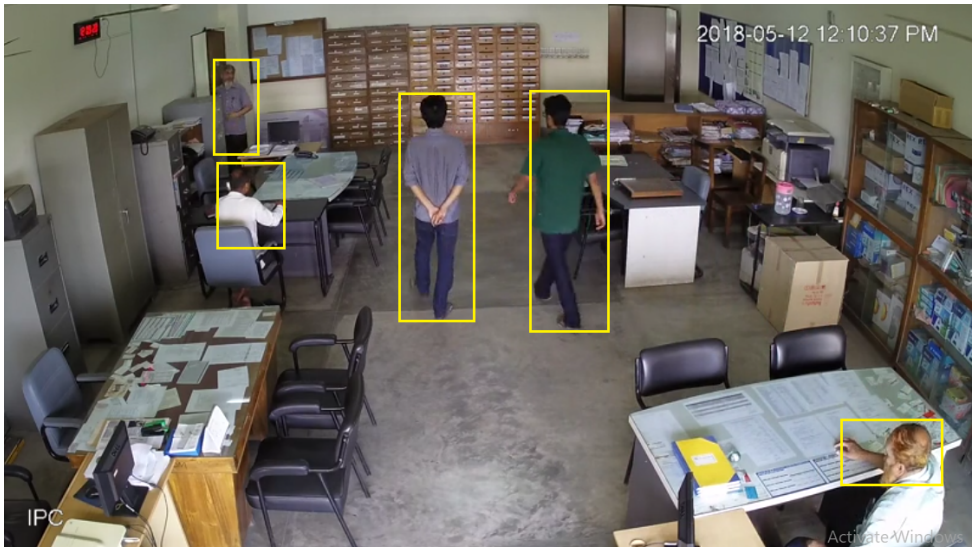


Figure 5.3: Boundary boxes with yellow colour show the moving objects.

4.4.2. We also calculate average velocity of all the moving objects in the video file.

5.4.3 Experimental Results of Path Specification with Approximate Time

Figure 5.7 shows a background image with a possible specified path where camera not available. Thus so many paths can be identified in the background image for the prediction model. And we can calculate the time needed to cover the path by the input object by using equation 4.7 described in 4.4.3.

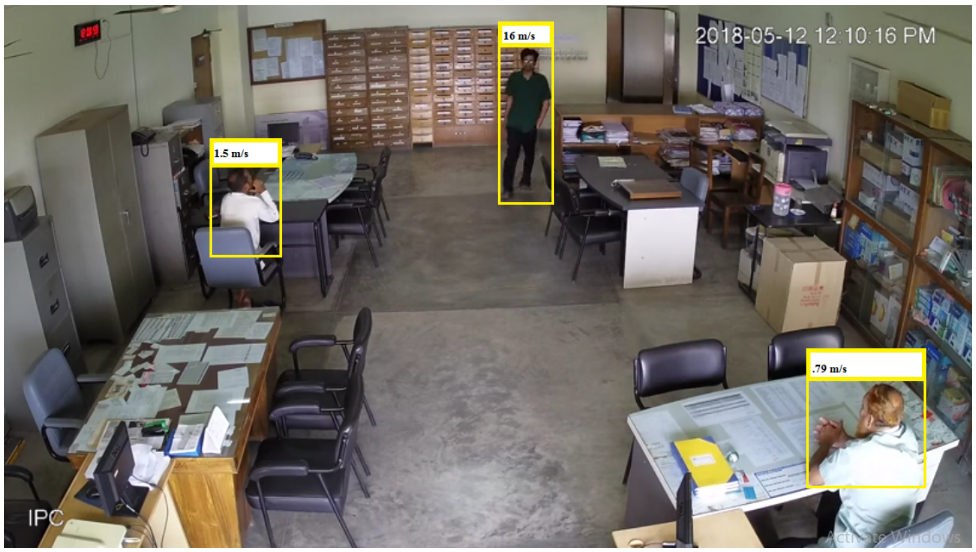


Figure 5.4: Boundary boxes with yellow colour show the moving objects with velocity in meter per second unit.

5.4.4 Experimental Results of Recognizing the Input Object over the First Video File

Figure 5.8 shows an input object which movement we want to explore in the all cameras.

Now our task is to recognize the input object over the video files using SURF features, track the object with a boundary box and extract those frames containing the object described in 4.4.4. Figure 5.9- 5.12 shows our experimental result of recognizing input object with a boundary box in a video file. Here only four frames are showed. We insert a yellow boundary box of the recognized object. Figure 5.12 does not contain the input object , so there is no boundary box in it.

Now, we extract all the video frames from the video containing the object and output it to a new video file.

5.4.5 Experimental Results of Recognizing the Input Object over the Other Video Files

Figure 5.13- 5.15 shows our experimental result of recognizing the input object with a boundary box in video file 2 described in 4.4.5. Here only three frames are showed. We insert a yellow boundary box of the recognized object.

Thus we repeat the same process over the other video files.

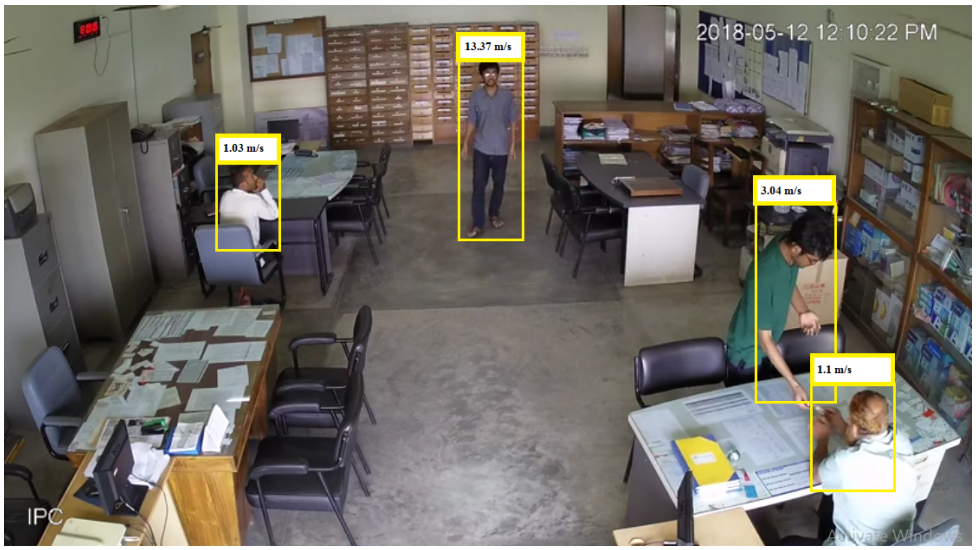


Figure 5.5: Boundary boxes with yellow colour show the moving objects with velocity in meter per second unit.

5.4.6 Experimental Results of Movement Prediction Pre-processing Steps

Figure 5.16 shows a segmented image from the background object which contains the object by image segmentation method. We use image segmentation C++ codes given by our respective thesis supervisor Prof. Dr. Md. Monirul Islam. This code gives the segmented region in the input image and also gives a .map file which contains each pixels in which region.

Then we write a code which takes the segmented image as input and we choose the regions by clicking on the regions of the foreground object and output an image containing only the foreground image. Figure 5.17 shows the foreground image making background as black and keeping the foreground by image segmentation method. Figure 5.18 shows the foreground image by image cutting method by the cursor.

We need different views and different positions of the same object to make it more realistic in the prediction algorithm. Figure 5.19- 5.22 show different views and different positions of the same object which we insert in the background in the prediction algorithm.

Figure 5.23 shows a only a possible path specifications and the distance of paths in meters. There are also many possible paths but only one is shown.

5.4.7 Experimental Results of the Movement Prediction Algorithm

Figure 5.24 shows a background image where we insert our interested object and move the object.

Figure 5.25- 5.28 shows our experimental result of predicting object movement algorithm where there is no camera. Here, only four frames are shown.

Figure 5.6: Boundary boxes with yellow colour show the moving objects with velocity in meter per second unit.

Figure 5.7: A background image shows a specified path where camera not available.

After that , we collect all the actual and predicted video files and combine them to make a total movement of the object in the interested area.

5.5 Summary

We complete our works as we plan for it. We complete each of the tasks almost successfully. We show the detailed procedures and experimental result of each of our task. We combine existing work with our own contributions. These combination of the two make our thesis a success.

Figure 5.8: This is the input object which we want to track.

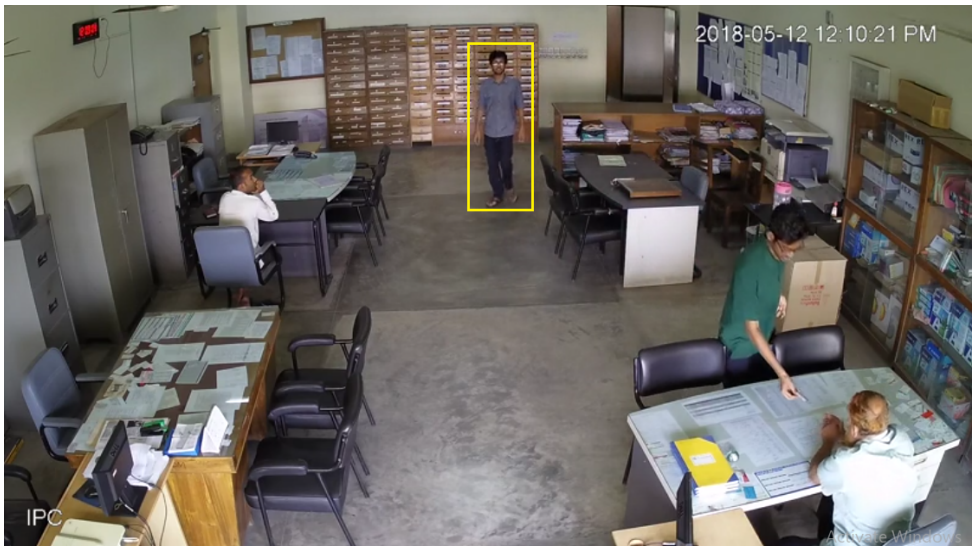


Figure 5.9: Boundary box recognizes the input object in video file 1.

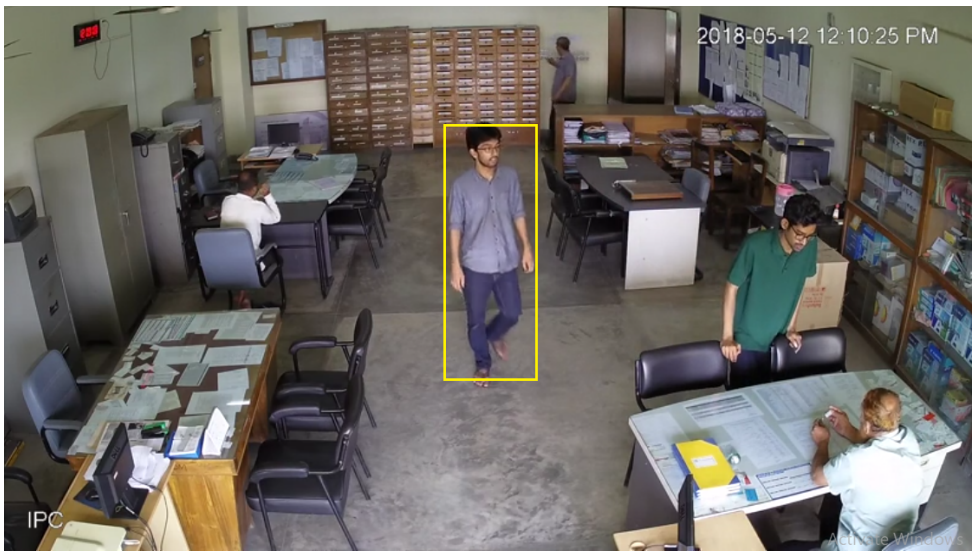


Figure 5.10: Boundary box recognizes the input object in video file 1.

Figure 5.11: Boundary box recognizes the input object in video file 1.

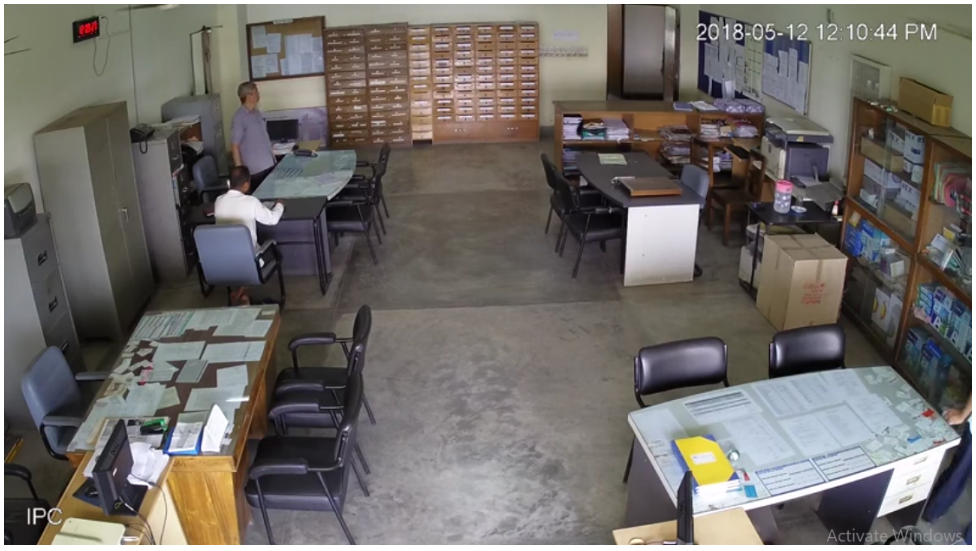


Figure 5.12: There is no input object recognized in this frame.

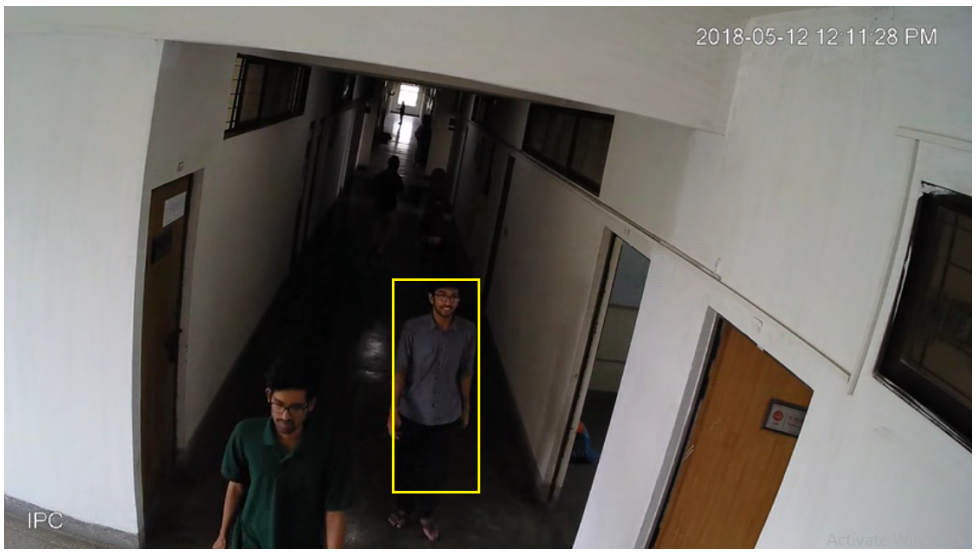


Figure 5.13: Boundary box recognizes the input object in video file 2.

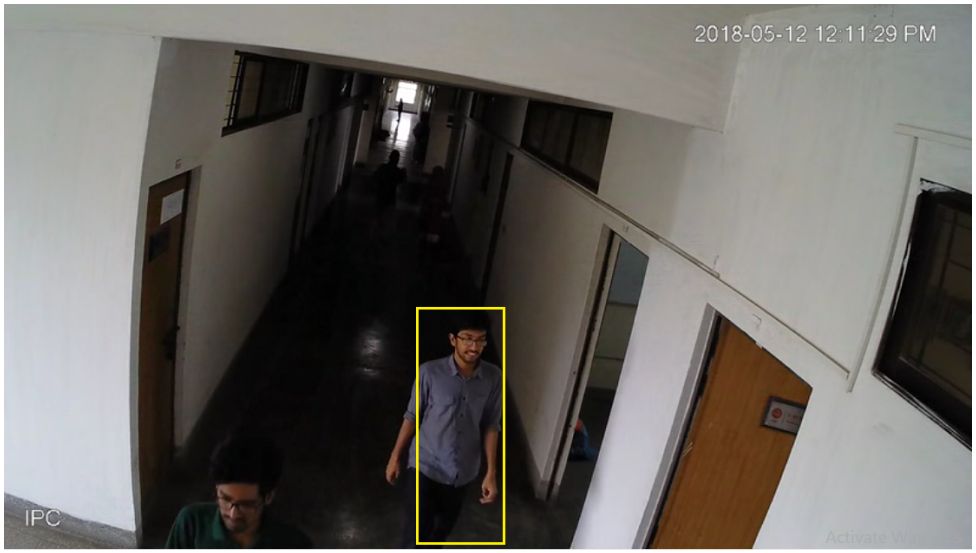


Figure 5.14: Boundary box recognizes the input object in video file 2.

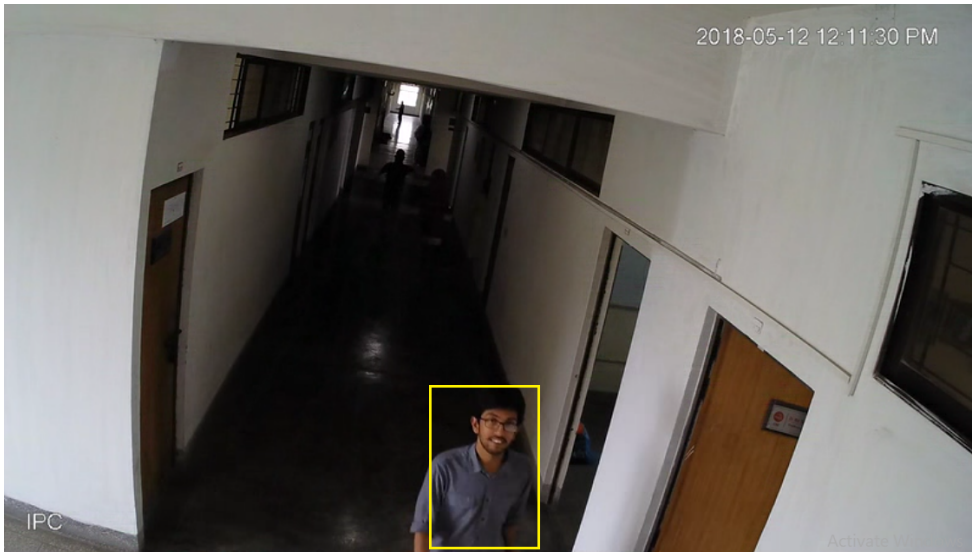


Figure 5.15: Boundary box recognizes the input object in video file 2.

Figure 5.16: This is a segmented image containing the object.

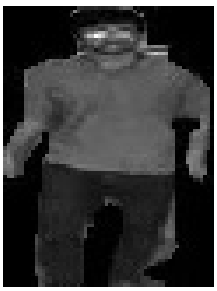


Figure 5.17: This is a foreground image of the segmented image.

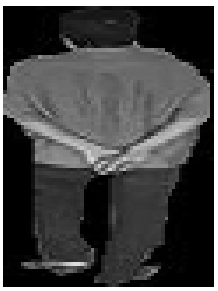


Figure 5.18: This is a foreground image by Image cutting method.

Figure 5.19: This is a front side view of the object.

Figure 5.20: This is another front side view in different position.

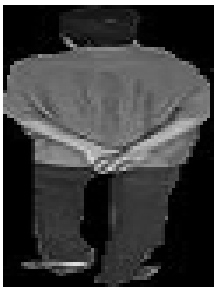


Figure 5.21: This is a back side view of the same object.

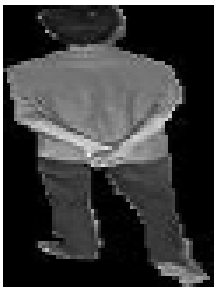


Figure 5.22: This is another back side view in different position.

Figure 5.23: Background image shows specified path with distance in meters.

Figure 5.24: This is a background image of the missing scene.

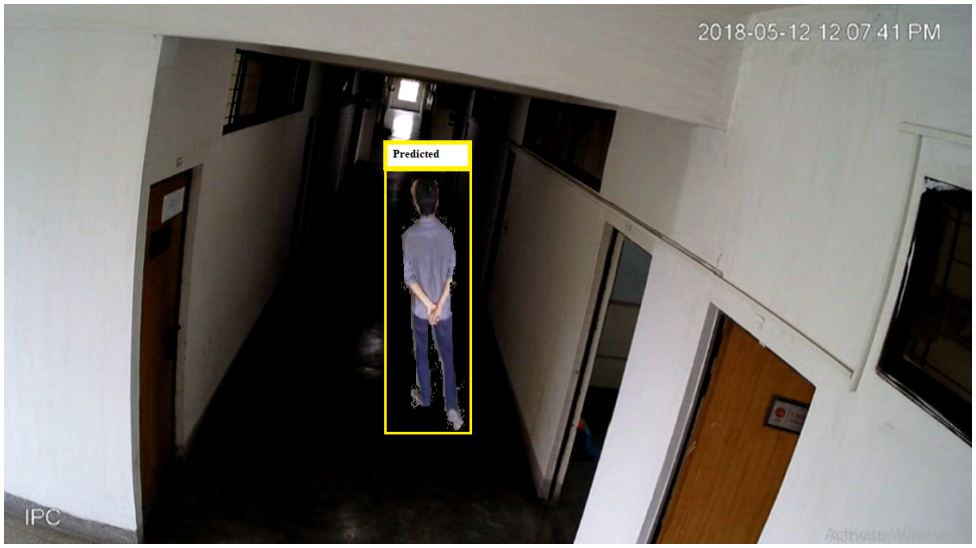


Figure 5.25: Object takes an optimal path using prediction algorithm.

Figure 5.26: Object get smaller when it goes far from camera using prediction algorithm.

Figure 5.27: View and shape change when object comes towards camera using prediction algorithm.

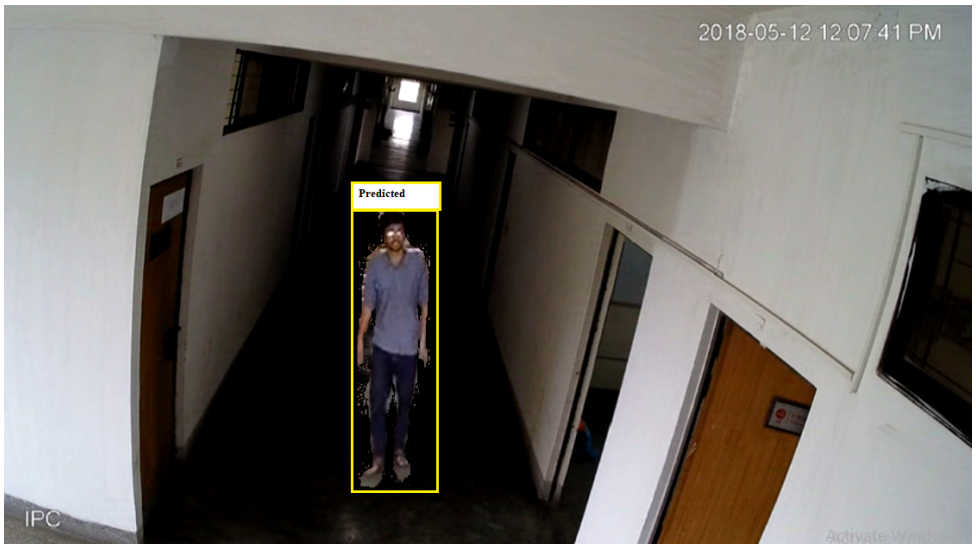


Figure 5.28: Object gets bigger when it comes closer using prediction algorithm.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

Visual surveillance system collects a lot of data and it is almost impossible for a person to track a person's movement manually from these huge collection of data. People have much interest in discovering typical and abnormal activities, detecting activities of some objects, and knowing some structures of the objects, such as paths commonly taken by objects, sources etc , where objects appear and disappear. Moreover, for representing a crime scene, scene reconstruction from multiple CCTV camera's may be useful. Besides, lawyers can also get a benefit to represent the incident of the crime to or against a suspected person. So, we build a automated system which will take videos from multiple CCTV cameras' as input at a fixed time and shows the user a total movement of a specified object in that time. Besides, if there is a missing camera in any place, we have built an algorithm that predicts the movement of that object in that place. For this, firstly, we detect moving objects in a video and tracked them. Then, we track velocity of each moving objects. Here, distance in meter unit between two pixels is considered as constant. But actually when the pixels goes apart from image center, distance in meter unit between two pixels gradually increases. Here, we develop a polynomial approximation between the pixel distance and the distance in meters so that the distance in meters between two pixels increases when they goes far from image center. This give better approximation in our prediction algorithm. Then we calculate the time needed to cover a path by an object giving its previous velocity. This is also needed later in our prediction algorithm. We then took the input of our interested object and recognized the object along all the videos of CCTV cameras and save them as separate output videos only containing the input object. For this ,we use SURF features to recognize the input object. Now, the one of the most important part of our thesis is predicting the object's movement in a place where there is no camera or missing camera. First, we make some pre-processing steps to prepare it for the prediction algorithm. We develope a linear relationship between object size and its position with reference to camera. This make

our prediction algorithm more realistic. We also insert image in a background and move it with its various views according to need. This make the prediction algorithm more realistic. We extract foreground images to insert it on the background. Then finally we build a prediction algorithm which uses previous constant average velocity. We find an optimal path for the object and move the object towards the path with calculated velocity. Thus we give an approximate movement of the object where there is no camera. Finally, we collect and combine all the actual and predicted data. These data are the total movement of the object in the area for a fixed time period. We use some of the previous techniques and we also have our own contribution in the thesis. Combining the two, we complete our thesis work successfully in spite of having some limitations.

6.2 Future Works

There are some limitations of our thesis. We try to overcome them in our future works. We use an object tracking algorithm which can track only movable object. We try to improve the algorithm in future such that it can detect static object too. SURF features which we use cannot recognize the object properly if the object shape totally change at a time or noise in the object. Besides, we use different input of the same object for different cameras to recognize the object in multiple cameras. Future work will include such a method so that an input object can detect the object in all the cameras. Our prediction algorithm uses constant velocity but the object usually does not move in constant velocity but as it is an approximation, we can approximate movement by using constant velocity. Our future work will also include such a movement prediction algorithm that can predict the human movement more accurately in stead of only considering constant velocity.

References

[1] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *1999 Conference on Computer Vision and Pattern Recognition (CVPR '99), 23-25 June 1999, Ft. Collins, CO, USA*, pp. 2246–2252, 1999.

[2] P. Chiranjeevi and S. Sengupta, "Detection of moving objects using multi-channel kernel fuzzy correlogram based background subtraction," *IEEE Trans. Cybernetics*, vol. 44, no. 6, pp. 870–881, 2014.

[3] D. Komlen, T. Lombarovic, M. Ogrizek-Tomas, D. Petek, and A. Petkovic, "Multi-camera person tracking system," in *2012 Proceedings of the 35th International Convention, MIPRO 2012, Opatija, Croatia, May 21-25, 2012*, pp. 1798–1802, 2012.

[4] X. Wang, K. Tieu, and W. E. L. Grimson, "Correspondence-free activity analysis and scene modeling in multiple camera views," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 56–71, 2010.

[5] S. Wang, Y. Wang, and S. Zhu, "Learning hierarchical space tiling for scene modeling, parsing and attribute tagging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2478–2491, 2015.

[6] L. Shapiro, *Computer vision*. Upper Saddle River, NJ: Prentice Hall, 2001.

[7] M. K. Bhure and M. J. Dhande, "Object detection using surf features," 2017.

[8] H. Shuo, W. Na, and S. Huajun, "Object tracking method based on surf," *AASRI Procedia*, vol. 3, pp. 351–356, 2012.

[9] M. V. Afonso, J. C. Nascimento, and J. S. Marques, "Automatic estimation of multiple motion fields from video sequences using a region matching based approach," *IEEE Transactions on Multimedia*, vol. 16, no. 1, pp. 1–14, 2014.

[10] H. A. Abdelali, F. Essannouni, L. Essannouni, and D. Aboutajdine, "Algorithm for moving object detection and tracking in video sequence using color feature," in *Complex Systems (WCCS), 2014 Second World Conference on*, pp. 690–693, IEEE, 2014.

[11] N. Krahnstoever, T. Yu, K. Patwardhan, and D. Gao, "Multi-camera person tracking in crowded environments," in *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pp. 1–7, IEEE, 2009.

[12] D. Komlen, T. Lombarović, M. Ogrizek-Tomaš, D. Petek, and A. Petković, "Multi-camera person tracking system," in *MIPRO, 2012 Proceedings of the 35th International Convention*, pp. 1798–1802, IEEE, 2012.

[13] J. Ferryman and A. Shahrokni, "Pets2009: Dataset and challenge," in *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pp. 1–6, IEEE, 2009.

[14] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 9, pp. 1820–1833, 2011.

[15] A. Wu and Z. Zeng, "Dynamic behaviors of memristor-based recurrent neural networks with time-varying delays," *Neural Networks*, vol. 36, pp. 1–10, 2012.

[16] X. Jiang, M. Körner, D. Haase, and J. Denzler, "A graph-based map solution for multi-person tracking using multi-camera systems," in *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, vol. 3, pp. 343–350, IEEE, 2014.

[17] J. Prokaj, M. Duchaineau, and G. Medioni, "Inferring tracklets for multi-object tracking," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pp. 37–44, IEEE, 2011.

[18] D. Makris and T. Ellis, "Path detection in video surveillance," *Image and Vision Computing*, vol. 20, no. 12, pp. 895–903, 2002.

[19] A. Mustafa, H. Kim, J.-Y. Guillemaut, and A. Hilton, "General dynamic scene reconstruction from multiple view video," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 900–908, 2015.

[20] D. M. Gavrila, "The visual analysis of human movement: A survey," *Computer vision and image understanding*, vol. 73, no. 1, pp. 82–98, 1999.

[21] S. Akoush and A. Sameh, "Mobile user movement prediction using bayesian learning for neural networks," in *Proceedings of the 2007 international conference on Wireless communications and mobile computing*, pp. 191–196, ACM, 2007.

[22] A. Asahara, K. Maruyama, A. Sato, and K. Seto, "Pedestrian-movement prediction based on mixed markov-chain model," in *Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems*, pp. 25–33, ACM, 2011.

[23] M. L. Miller, H. S. Stone, , I. J. Cox, and I. J. Cox, "Optimizing murty's ranked assignment method," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, pp. 851–862, 1997.

[24] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[25] H. Bay, A. Ess, T. Tuytelaars, and L. J. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[26] S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: binary robust invariant scalable keypoints," in *ICCV*, pp. 2548–2555, IEEE Computer Society, 2011.

[27] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: fast retina keypoint," in *CVPR*, pp. 510–517, IEEE Computer Society, 2012.

[28] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[29] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *VISAPP (1)*, pp. 331–340, INSTICC Press, 2009.

[30] M. Muja and D. G. Lowe, "Fast matching of binary features," in *CRV*, pp. 404–410, IEEE Computer Society, 2012.

Generated using Undegraduate Thesis LaTeX Template, Version 1.4. Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh.

This thesis was generated on Friday 16th October, 2020 at 9:50am.