Classifier based incremental reconstruction of human object trajectory in live video streams

PhD report

BY

Muhammad Majid Afzal

Date: 15 - September - 2019

Faculty of Computing

London Metropolitan University

166-220 Holloway Road, London, N7 8DB

Acknowledgments

I would like to thank number of the members of the faculty of university. Primarily, my supervisors, Prof. Karim Ouazzane and Dr. Vassil Vassilev, whose enthusiasm and encouragement were a great inspiration to me. They were not only a source of advice on various topics but also a great companion and helped in all sectors of research.

I am also indebted to other faculty members, especially Pawel Gasiorowski and Ramzi Djemai with whom I spent countless hours on exchanging thoughts, brainstorming and experimenting with innovating technologies so that I could gain inspiration for finding a solution to that little problem that was preventing me to progress further. I also owe a lot to other faculty members who helped me a lot during my research process. I have learnt a lot from all of you.

I would also like to thank my parents and all family members for their continuous support. Finally, I would like to thank all my friends, who were understanding enough to forgive my absences from various events due to my works on research.

Abstract

A common practice for solving problems in video analytics is to consider each problem (e.g. processing speed, complexity of data analysis or tracking of moving object) in isolation, starting with constructing of computational models for each specific data processing task. This is dictated by the need to address the complexity of the overall problem, which typically can be broken into multiple computational tasks – detection, recognition, classification, analysis, prediction, etc. Each of these tasks has its own computational model and uses different data. This approach has been the main focus of research in video processing for a long time and had some success, but it leads to the significant requirement for computational power and is not feasible for real-time video analytics tasks because of the time constraints.

An alternative to this strategy is to link the video analytics tasks in order to leverage the related data sets and computational models. By doing that we can decrease the amount of data and computations necessary and thus, reach satisfactory performance. Consequently, we can improve the accuracy of the solution obtained by simplification through utilizing the previous experience through learning. Our aim in this research is to exploit the variety of parameters of this process – development of the data processing model, features extracted from the data and algorithms of the machine learning.

This research addresses the problem of reconstructing moving object trajectories for real-time video analytics based on continuous processing of input video streams with application to a wide range of tasks in video surveillance. Our approach uses a combination of classifiers for different features of the data, such as shape and colour, as well as other parameters (e.g. the position of human body parts) which can be used for detecting various profiles of interest for further processing of the original video stream. The custom-tailored model of moving object is created by experimenting with different models and algorithms specifically applicable to the task for trajectory reconstruction and selected to make the best use of data and to achieve the best result in terms of speed of data processing which is further discussed in chapter 6.

On the basis of the experimental work conducted in this research we have developed a software system, which can track and identify moving human objects of interest within live video streams through a combination of visual pre-processing (discussed in chapter 5 in video transformation module), machine learning and video post-processing (discussed in chapter 5 in ‘rotation’ activity) methods. It uses a novel object detection algorithm which significantly improves the speed of processing during the tracking and localization steps. This algorithm makes use of a number of comparing analytics method by classifying the shapes and sizes parameters as shown in chapter 4 with different cases.

Although specifically designed to serve the purpose of the research programme for behaviour analysis of moving objects in live video streams this part of our framework can be considered as a standalone problem solving module for software systems which involve object detection, identification and tracking based on live video input. It can be used as an open source software in many areas ranging from public safety management and video surveillance to computer games and animation.

List of Publications

M. Afzal, K. Ouazzane, V. Vassilev and Y. Patel, "Incremental Reconstruction of Moving Object Trajectory", The First International Conference on Applications and Systems of Visual Paradigms, VISUAL 2016, Barcelona, Spain

M. Afzal, K. Ouazzane and V. Vassilev, "K-Nearest Neighbours Based Classifiers for Moving Object Trajectories Reconstruction", The Third International Conference on Applications and Systems of Visual Paradigms, VISUAL 2018, Venice, Italy

Table of Contents

Figures

Tables

Dictionary

Image recognition	Image recognition is organising a detected object into different groups like recognition of humans in a video of people walking. In other words, image recognition is an automated process of recognition of patterns and regularities in data to detect the target object. This is closely related to artificial intelligence and machine learning
Pattern recognition	This is an activity which uses various methods to mine information from given data in general, mainly based on statistical approaches or using different artificial intelligence methods (like the position of human body parts' help in identifying the human object).
Trajectory recognition	Trajectory is an activity which uses the signature of trajectory to compare it with information extracted with the help of different image processing algorithms.
Trajectory reconstruction	This is the main task of this research and it is the reconstruction of the trajectory of moving object from the given input video stream.

1. Introduction

In humans, more than 35% of the brain is devoted to visual processing to allow us to interpret and behave intelligently as part of our daily lives [1]. Intelligent and real time moving object tracking is a difficult issue in computer vision research area. Multiple objects tracking has many useful applications in scene analysis for computerised surveillance. If we can track different objects in an environment of multiple moving objects and reconstruct their trajectories, then there will be a variety of applications. This research is focused on the reconstructing the trajectory of human movements in online signals from video camera for the purpose of further analysis of their behaviour.

1.1. Research problem

The key research point which is addressed in this research is to develop efficient algorithms for intelligent and efficient human object tracking and reconstruction of trajectories of movement from the given input video stream. These reconstructed trajectories must have information about the location and the position of the body and its movements through the space of a visual scene, observed by surveillance cameras. This is necessary for the purpose of motion detection/tracking in secure areas, controlling the flow of crowd movements, analysis of the pattern of movements etc. for various software applications.

In order to process the data swiftly in video analysis application, classifiers are used to track the moving object in the live video stream. This methods makes the data analysis process less computational demanding.

1.2. Intelligent Security and Safety Planner

This research is part of the research program for Visual Analysis of Individual and Group Dynamic Behaviour carried out within the Cyber Security Research Centre of London Metropolitan University. The research group is interested in real-time video analytics with applicability to surveillance in security, disaster recovery and safety management, and customer insight. The ultimate goal of this research program is to construct an efficient framework for visual analytics in real time as presented in the Figure 1-1.

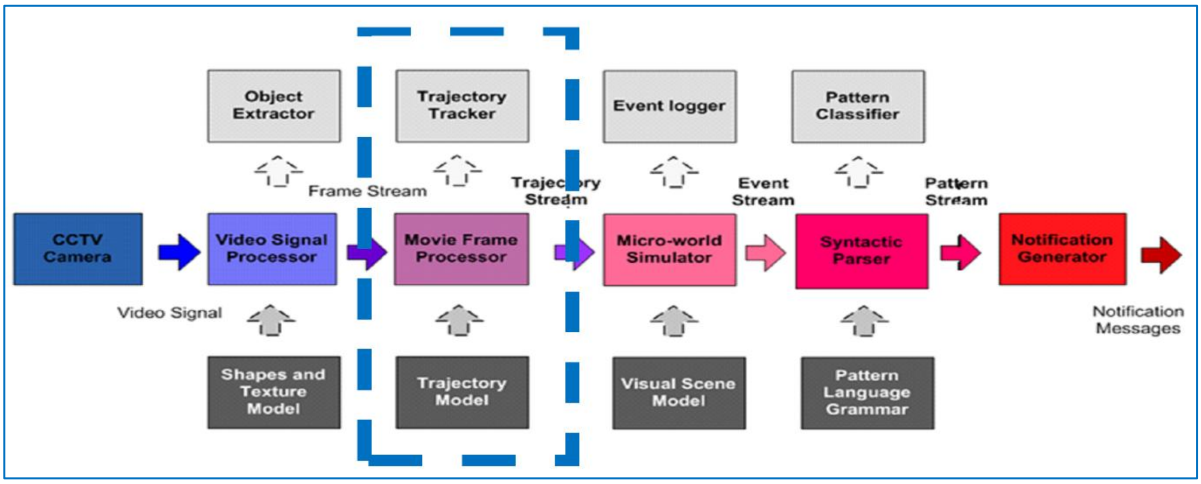


Figure 1-1: General workflow of the overall framework of Visual Analysis of Individual and Group Dynamic Behaviour [2]

This research is focused on analysing and reconstruction of moving objects trajectories within enclosed spaces (rooms, corridors, staircases, floors and open spaces) of big buildings, such as shopping malls and transport stations, as well as in large transport vehicles, such as cruiser ships or in outdoor environment. This thesis is a part of the research of whole framework and it is shown with dashed rectangle in complete framework diagram Figure 1-1.

Moving object tracking in this approach is based on the object-centric representation of the position which forms a tube-like model of the spatial navigation and allows isolated manipulation of the video objects within the focus and performing motions. This can be achieved through an algorithm for processing the flow of incremental information as illustrated in the Figure 1-2.

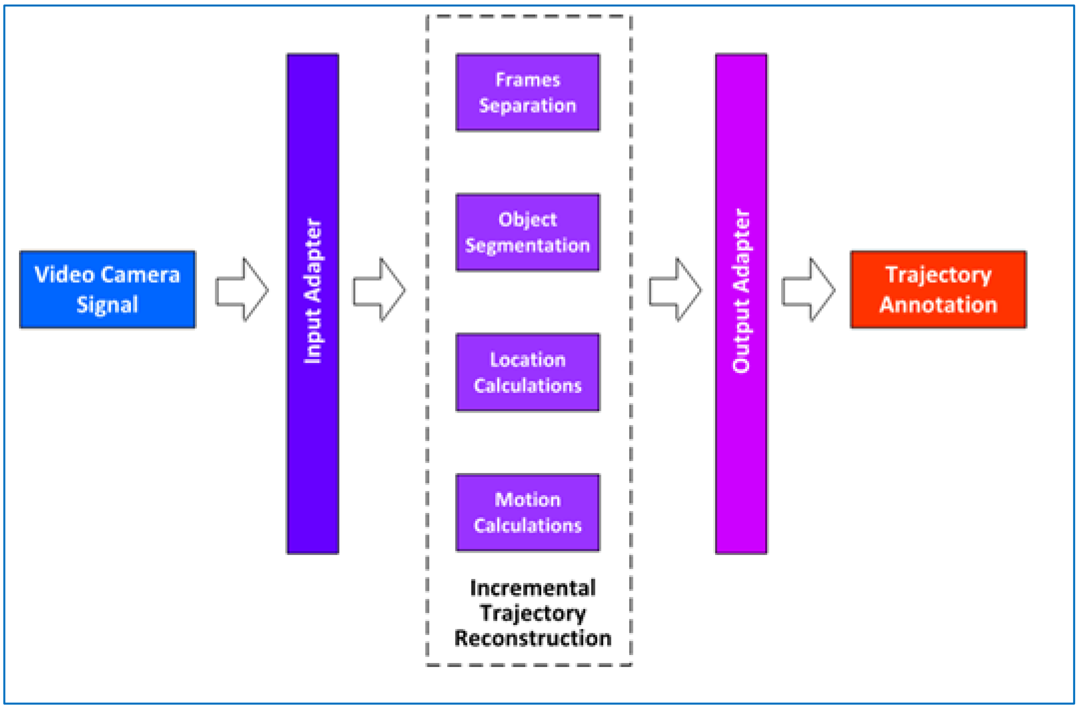


Figure 1-2: Incremental trajectory reconstruction

Figure 1-2 shows the flow of the data in the framework. The dashed line rectangle shows the main processing modules where input data is converted into the required trajectory information.

1.3. The Aim and Objectives of the Proposed Research

1.3.1. Aim

Smart surveillance systems have two essential building blocks namely object segmentation and tracking. However, there are few other things such as background estimation etc. will provide the useful information when integrated with above important parts. The main research element of this research lies on developing a novel framework for reconstructing a trajectory of moving human object in video based on the use of an object segmentation system with the help of classifiers.

1.3.2. Objectives

1. To investigate the current state of research in the area of tracking human objects trajectory and to identify the main problems, existing approaches and available methods

for modelling physical movements of individuals and groups of individuals within enclosed spaces over a limited period.

2. To investigate and compare the methods for object tagging and modelling in order to select appropriate representation of the initial data.

3. To compare and select the modelling languages, which can be used for modelling of moving human object trajectory (e.g. VRML- Virtual Reality Modelling Language, X3D - Extensible 3D Graphics or XML - Extensible Mark-up Language etc.).

4. To experiment with limited dynamic micro-worlds with moving individuals and groups for the purpose of analysing the individual's or group's movements.

5. To select appropriate object segmentation technique for tracking human object trajectory and construct a framework for online video processing based on trajectory analysis using suitable algorithms.

6. To validate the developed framework through comparison of experimented micro-world and recorded real-world video signal.

1.4. Research Hypothesis

The study of visual attention in humans is very much the design of control mechanisms to limit complexity. Using a rather coarse categorization one might divide visual processing into data and model/goal-driven processing. In data-driven processing, the areas of an image to be processed are selected based on their saliency and offered to other modules in a system for higher-level tasks, for example, recognition and description. The selection of which regions are to be processed and how to fuse different image/video descriptors is then performed according to criteria of optimality in the sense of discrimination. One assumes that, based on such preliminary categorization it would be possible to construct an efficient algorithm for trajectory reconstruction, which are recurrent and can then process the video stream in an incremental manner.

1.5. Methodology

Bottom-up approach is undertaken in this research in order to build the theory and framework. Different sections of computer technologies are involved and those can be defined as below,

1.5.1. Event-driven simulation of visual scenes

Moving object tracking is based on the object centric representation for easy and intuitive object tube model of a video navigation and temporal manipulation of video objects. The moving human object in the video is modelled as a collection of spatiotemporal object volumes (object tubes) placed in a 3D grid like structure. This approached has been adopted in the framework in order to reconstruct the trajectories of moving objects.

1.5.2. Reconstructing the trajectory of moving object

This involved extraction of motion information from the video and representation of object trajectories in a 3D interaction like grid. Motion based video representations are used in other video navigation and annotation systems. The focus of these systems is mainly on providing an in-scene direct moving object trajectory from the video. Then object motion information is used in other systems. It is anticipated that for the reconstruction of the trajectory analytical methods are required for connecting the spatial locations of the identified objects across the frames. This is pursued based on incremental approximation of the spatial locations across the video frames using different computational techniques.

1.5.3. Experimental prototyping

As the result of the research it is planned to produce an intelligent framework, capable of reconstructing the moving human object trajectory. The developed software is aimed to be based on the component approach and object-oriented methodology for software engineering. This will allow incorporation of the component into various applications, smart rooms, video surveillance, object tracking, and motion analysis etc. which requires spatial modelling and agent behaviour analysis.

1.6. Research Plan

After discussing the relatively large variety of methods in the section above, it was planned to conduct the research into key stages, where each stage was incorporated its own research study to account the current state of art in the respective area.

1.7. The structure of this report

The content presented in this report has been divided into seven chapters, where each chapter is the basis for the next one. This reflects the aforementioned bottom-up approach development strategy and the structure is intended to present how the workflow of the research looked like.

Chapter 2: Literature Survey

In this chapter, different computational models of moving objects have been reviewed and studied in context of the problem domain specified by the researchers in their papers.

The nature of the existing researches, methodologies and proposed solutions to problems closely related to segmentation, detection and analysis have been presented in a cohesive way so that the literature survey is organized by ideas produced during the reading sessions.

Chapter 3: Conceptualization of moving objects trajectories framework

Focus of this chapter is based on conceptualisation of the architecture of the model used by this framework for re-construction of moving object trajectories. Model should be rich enough to hold the sufficient and efficient information for smart re-presentation of moving object trajectories.

Chapter 4: Classifiers for moving object trajectories reconstruction

In this chapter, the core concept and mathematical theory behind classifiers and implementation has been studied in order to build the fundamental knowledge required for development of framework

Chapter 5: Implementation of moving objects trajectories framework

This chapter describes the implementation of framework using the underlying mathematical concepts presented in previous chapter. The text presents the component and working mechanism of simple framework.

Chapter 6: Comparative analysis of classifiers algorithm

In this chapter, comparison is made between research in this thesis and previous researches using different data-sets.

Chapter 7: Conclusion and recommendation for future work

This chapter focuses on evaluating the present state of the research and presenting the proposal of future work. The description itself is based on the material that was produced through the duration of this research. This chapter has also detailed description about the contribution to knowledge and different strategies developed during the research

1.8. Conclusion

This chapter summarises the entire research, claiming some original contributions and makes recommendations for further exploration of the developed framework. Complete details of this research are presented in following chapters.

2. Literature Survey

2.1. Introduction

The aim of computer vision is to explore one image of a scene by recognizing objects, their building, spatial arrangements and tracking the movements of the objects. This has been stated to as an image understanding. With the passage of time, computer vision has progressively been making the evolution away from understanding single images to investigating images sequences, or video sequence. The main shift in the traditional pattern has been from the identifying of static objects in the scene to the motion-based recognition of actions and events like trajectories of the moving objects.

These trajectories' details of moving objects are important requirements for topics like visual modelling, simulation and analysis of dynamic behaviour. These topics are currently becoming an important direction of research in video surveillance due to its importance for safety management, security and disasters recovery. Many researchers [5][6][9] tried to address issues related to the range of complexity in derivation of behaviour patterns from live video stream. The research in this area includes extraction of image features, identification of shapes, detecting movements and modelling human behaviour for further analysis. Ultimately, these trajectories information will be used for the following tasks:

- Identification of moving objects and moving parts required by different case scenarios.
- Recognition of the key poses of the moving object that are important for the recognizing of the patterns of behaviour in entire input stream.
- Manipulation (estimation) of the individual body part location, rotation and scale to achieve the desired pose.
- Trajectories detail helps in marking key frames in data processing stage for specific frames and encapsulating individual input stream as a data block.
- Reporting the entire moving object trajectory along with the already stored data block information to a format that will be helpful in behaviour analysis.

2.2. Human body motion

Human body movement trajectory's information is needed by algorithm for dynamic patterns processing and these details consist of extensive set of movements constituting the dynamic activity of a moving object. Let's consider a scenario in which a person needs to pick up an object from the floor. In order to show such a behaviour, the person needs to bend or kneel-down, stretch an arm into the direction of the object and then return to the original position holding the object. In this scenario, one can identify three key poses of the body, namely kneeling down, stretching an arm and standing up. To complete the whole motion, the three key poses need to be combined to simulate the movements. In order to simulate such a behaviour with armature one must specify first the key frames, i.e. the starting and ending transition points of the involved movements. The remaining frames in the set represent the intermediate positions of the body in transition between the different poses. The Figure 2-1 and Figure 2-2 show the different armature positions and states.

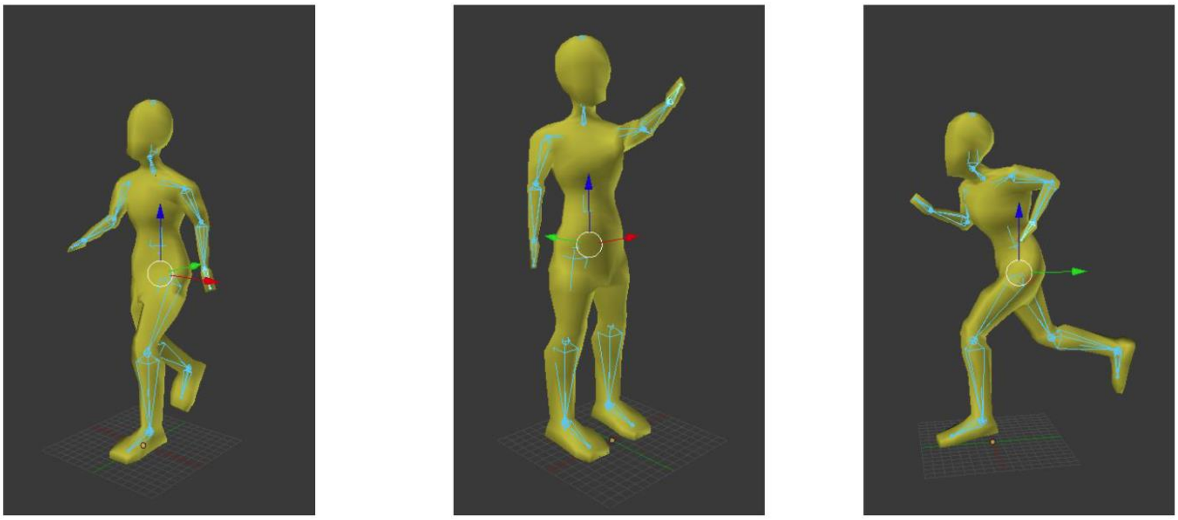


Figure 2-1: Movement of human object (part 1) [2]

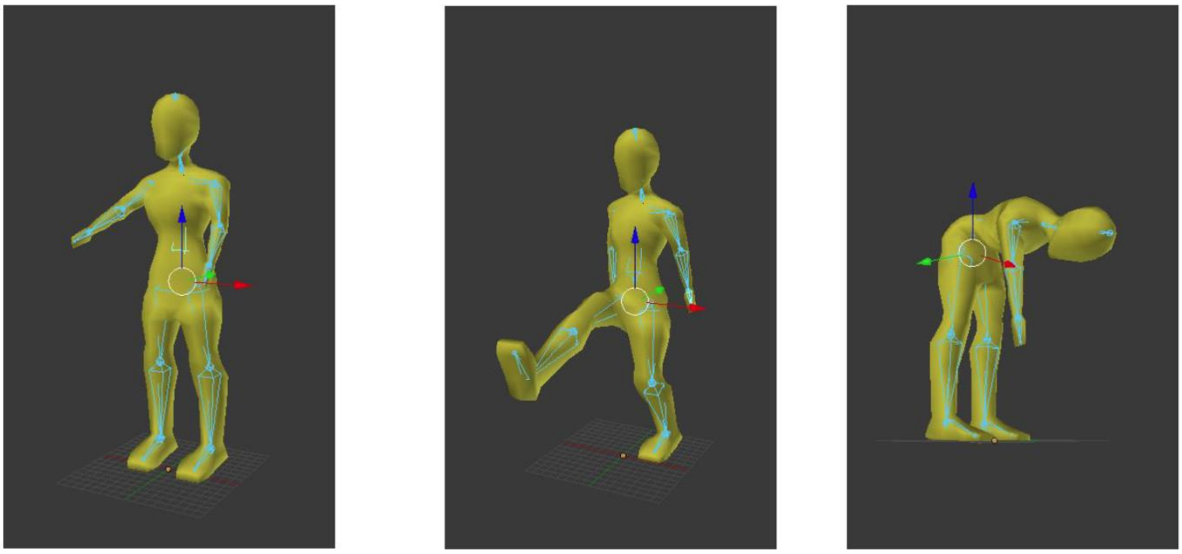


Figure 2-2: Movement of human object (part 2) [2]

The above armature states are used in different human movements. These movements are mentioned in the table below.

Table 2-1: Human body movements and armature positions

Movement Type	**Armature movements**
Fast hand movements	All three states of Figure 2-1 and first state of Figure 2-2 are involved in this movement
Playing music indoor	Mainly first state of Figure 2-2 are utilized in this movement if person is playing music in standing state
Throwing items	Second state of Figure 2-1 and last state of Figure 2-2 are involved in this movement
Walking indoor	Mainly first two states of Figure 2-1 are involved in this movement. For few persons, second state of Figure 2-2 involved in this movement
Two persons walking outdoor	This has same states as above moment. Sometime third state of Fig 2-1 is used in this movement for running people
Picking up stuff	Mainly third state of Figure 2-2 is involved in this movement

2.3. Object detection methods

Different video objects detection methods have been proposed by various researches. This section has raised the awareness of available methods and helps in taking the decision in chapter 4 during different decisions taking moments. Each method has its own benefits for the targeting applications [3]. These object detection methods are needed for tracking of the moving objects in the videos. Many methods for object detection highlight a main area of research in computer vision since many applications require the determination of the existence and the positions of objects in images [4]. Few of the object detection methods types are described below.

2.3.1. Feature-based object detection

There are many applications using feature-based object detection methods. Colour is one of the features used in these types of applications [5]. These methods are most suitable for videos with standing cameras and without any alteration of the background clutter. The spatial histograms consist of the marginal distribution of image over local areas, object texture and shape are concurrently conserved by the spatial histogram representation [6]. A simple histogram is obtained by taking a region of an image, assigning a label to each pixel (with the help of some mapping function), and then computing a histogram of the labels [7]. The histogram counts, for each possible label, how many pixels received that label. This obtains a feature vector (a vector of counts), one count per possible label. Such a histogram does not capture any information about where the labels appear. A spatial histogram addresses that shortcoming by dividing the image up into several smaller patches, computing a histogram for each patch, and concatenating those histograms. For instance, this could take a 32x32 image and break it up into sixteen 8x8 patches, compute a histogram for each patch, and concatenate them to get a feature vector [7]. The Figure 2-3 shows an example of feature-based object detection:

Figure 2-3: Feature based object detection [8]

In the Figure 2-3, "spatial histogram" values are used to draw the squares around the objects detected. Objects present in the Figure 2-3 are consists of distinguishable parts such as wheels, car doors, and car windows etc. These parts are arranged in a relatively fixed spatial configuration and algorithm is detecting these objects with the help of spatial histogram values.

2.3.2. Line-based object detection

Different geometric shapes are formed by multiple points joined and formed enclosed area [9]. Therefore, in order to detect the shape in the image, an algorithm must be introduced to detect the line in the image. When the line is detected in the image, an algorithm should find the relation between the lines and determine the geometric shape [10].

There are various line detection algorithms which are available for image processing. Radon transforms is one of them [11]. If there is a line in the image, there will be a peak in the Radon transform domain [11]. The ability of the Radon transform in transforming lines (line-trends) inside an image into a domain of possible line parameters even for a very noisy image has led to many line detection applications within image processing, computer vision etc. This is illustrated in Figure 2-4 block diagram:

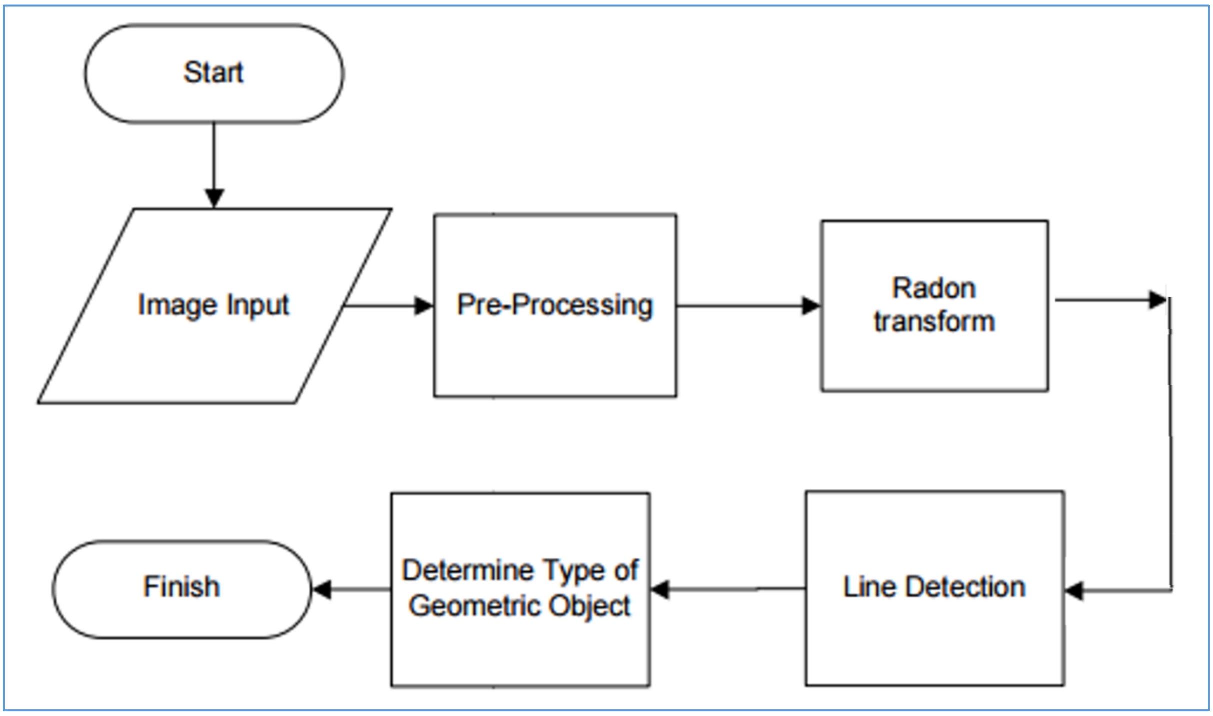


Figure 2-4: Block diagram of object detection algorithm research [12]

2.3.3. Region based object detection

Sometimes object detection tasks start by grouping pixels into segments as object contenders [13]. For example, some researchers for land planning projects [13] grouped contenders bridge pixels into possible bridge segments based on their connectivity and geometric values and then confirmed these segments according to their directions and connectivity with road segments [14].

The content mentioned in the above section is not directly needed by this research. But these are used in the data preparation step of this research. These are used for the identification of the moving object in the video. Trajectories will be created for these identified moving objects.

2.4. Projection

Projection is the process of changing the number of dependent variables. This section helps the author in chapter 3 while dealing with the manipulation of given input video stream environment. If there is a two-dimensional object and it is represented by a function, then such a function will need two variables to represent the object. Similarly, if there is a three-dimensional object, then its function will need three variables to represent the object. Projection plays an important role in video processing applications, especially for re-generation of moving object trajectories. Actually, two types of projections are possible which are explained below:

2.4.1. Three-dimensional to two-dimensional projection

Orthographic projection is the simplest way of projecting a three-dimensional object into two-dimensional objects [15]. In mathematical terms, the dropping of z component from a three-dimensional coordinate system is the projection of a three-dimensional object into a two-dimensional object. This can be written as below:

$$x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} y \tag{2.1}$$

The above equations show a three-dimensional point with the resultant two-dimensional projected point with ‘x’ and ‘y’. The Figure 2-5 is showing a graphical representation of the projection.

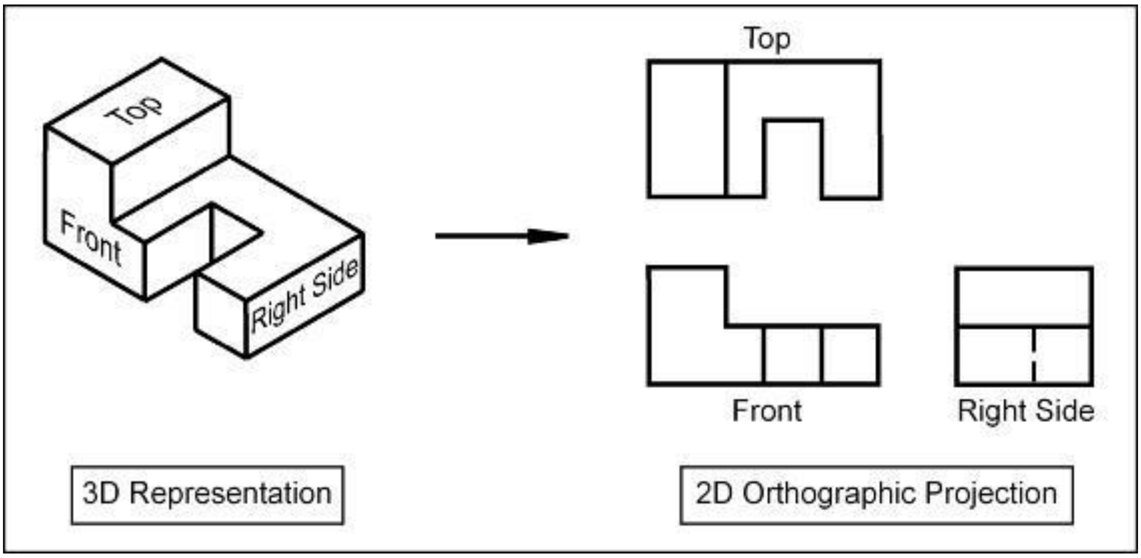


Figure 2-5: Three dimensional to two-dimensional projection [16]

In the Figure 2-5, the left-hand side of the figure has a three-dimensional object. The right-hand side of the figure has three results of orthographic projection. These projections are created with the help of top, front and side views of the three-dimensional object.

2.4.2. Two-dimensional to three-dimensional projection

Projection of two-dimensional objects to the three-dimensional objects is needed for a proper analysis within the environment that have limitations. Also, getting the three-dimensional object data is not possible in every case or it is too expensive. In the case of regeneration of moving object trajectories from a video data, this two-dimensional to three-dimensional protection is needed. Fourier synthesis, holography Monte Carlo and algebraic methods are used for this type of projection [17]. The Figure 2-6 is showing an example of usage of two-dimensional object projection to the three-dimensional objects:

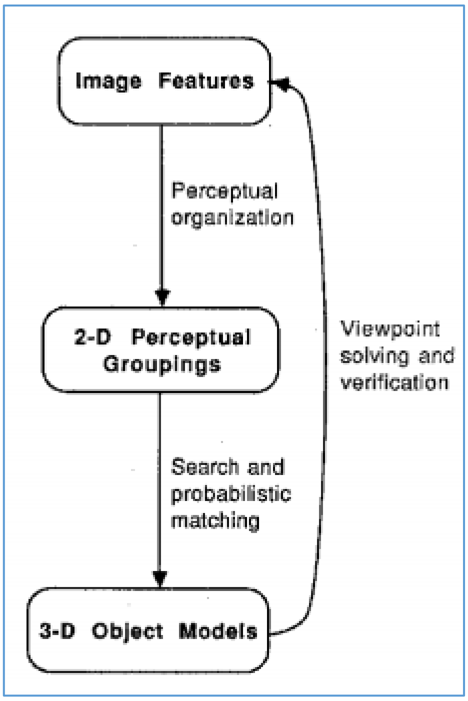


Figure 2-6: Three-Dimensional Object Recognition from Single Two-Dimensional Images [18]

The Figure 2-6 shows the model for visual recognition which makes use of prior knowledge of objects and accurate verification for a particular object, since it can make use of the spatial information in the image to make the use of full degree of available resolution. As it is mentioned above, the author used a three-dimensional model for object recognition after creating a projection from two-dimensional object and this type of projection is used and explained further in the next chapters.

2.5. Segmentation

Image segmentation can be normally viewed as subdividing an image into multiple segments. Information from this section is used in chapter 5 during the implementation of the framework. The segmentation method delivers a more simplified image depiction as these segments can be separately examined without the need of human to perform manual segmentation as a first step [19].

There is huge range of segmentation methods which are available; for example, a simple straight forward segmentation with just defining the foreground and background of the image. This basic segmentation is not enough for the current trend of image processing especially in object recognition applications [20]. A more trustworthy segmentation method is needed to counter more complex cases by applying some useful domains. Colour information is one of the popular domains used for image segmentation [21]. The importance of the colour information in the image is mentioned in the section above. The Figure 2-7 is with the example of two stages segmentation process:

Figure 2-7: Simple segmentation process [22]

The Figure 2-7 shows the basic segmentation output of an image. Image on the left side is the input image, whereas the image on the right side is the corresponding segmentation result. The output of this process can be improved further with the help of artificial intelligence techniques.

2.5.1. Contours

Image segmentation techniques are used to generate the contours of polygons with the help of jagged outlines and many redundant points. The Figure 2-8 is an example of a contour generated a by image segmentation technique.

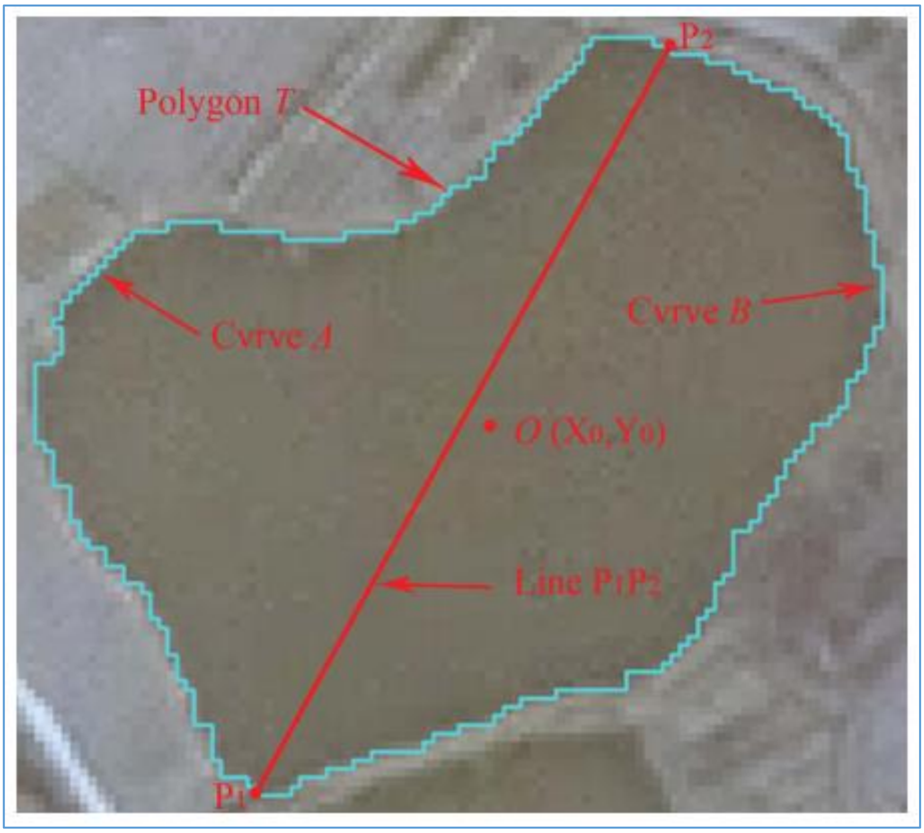


Figure 2-8: Image segmentation technique generated a contour polygon [23]

The Figure 2-8 is created with the help of geographic information system (GIS) data-producing standards. This figure shows the contour of pond. If the author can denote the Figure 2-8 polygon as 'T' then the centre point is as below.

$$O(X_0, Y_0) \tag{2.2}$$

While line passing from point P1 and P2 in the Figure 2-8 divides the contour in two curves namely A and B on both sides of line P_1P_2 respectively [23].

2.5.2. Fragmentation approaches

2.5.2.1. Divisive Clustering

Divisive clustering approach is also known as top-down approach. In this approach, the process starts with the help of complete image (or complete set of data). A usual method used in this approach is background subtraction [24]. The logic behind this is that of identifying the moving objects from the difference between the current frame and a reference frame (which is also known as "background model"). The background model must be a representation of the scene with no moving objects and must be kept regularly updated. The Figure 2-9 is an example of background subtraction process:

Figure 2-9: Background subtraction example [24]

In the Figure 2-9, the left-hand side part of the image is the real image, while the right-hand part is the result of background subtraction process. Moving objects are obvious in the right-hand part of the image.

2.5.2.2. Watershed

Watershed segmentation process is based on morphological features which is a classic and strong segmentation method [25]. This method forms a closed continuous region, a feature where methods are based on the spatial domain first order operator, second order differential operator [26].

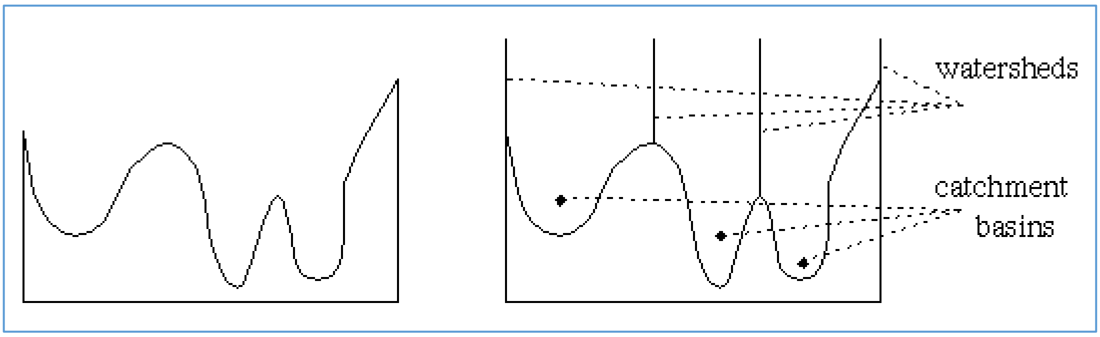


Figure 2-10: Watershed segmentation example values [27]

The left part of the Figure 2-10 shows the profile data of the image. While, the right part of the image shows the local minima of grey level altitude, the local maxima define the watershed lines.

2.5.2.3. Agglomerative clustering

Agglomerative clustering is an image segmentation approach usually used in high resolution image processing [28]. This approach offers some real advantages, such as more flexible clustering, and often produces higher quality segmentation trees. But this has been used less frequently in image processing because it is normally assumed to be prohibitively expensive [$O(N^2)$ or worse] [29]. The Figure 2-11 is an example of agglomerative clustering approach.

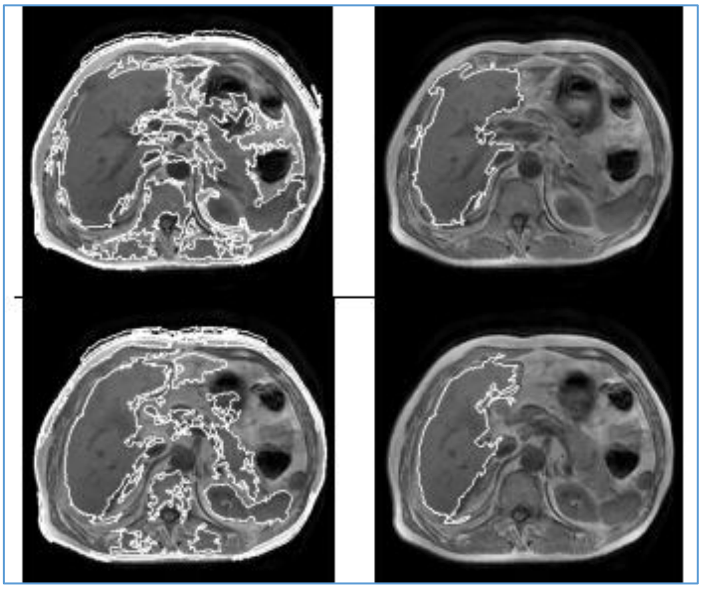


Figure 2-11: Agglomerative clustering approach on the liver MR image [30]

The left column of the Figure 2-11 contains the best clustered numbers, while the right column only contains the required segmented liver section.

2.5.2.4. Probabilistic aggregation

Sometimes this approach is also known as bottom-up approach. As its name indicates, in this approach pixels are gradually merged to produce larger and larger regions. In each step this approach considers pairs of adjacent regions and provides a probability measure to assess whether they should be included in the same segment.

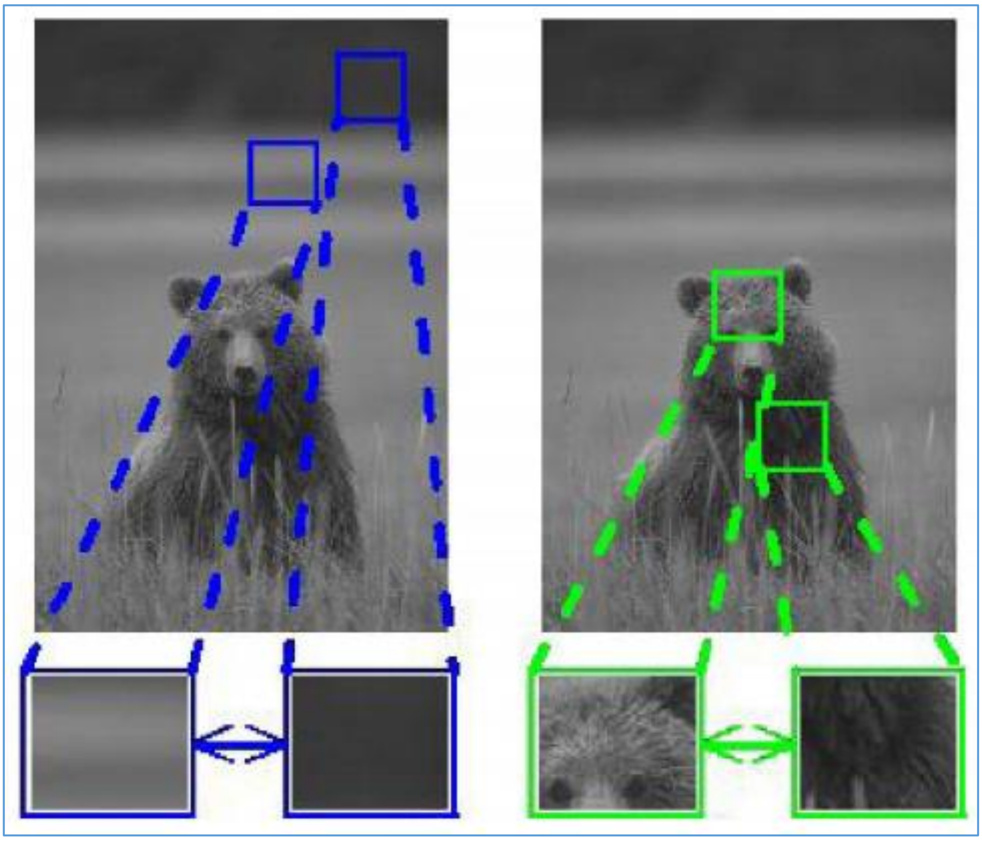


Figure 2-12: Probabilistic aggregation example [31]

The left two patches in the blue boxes in the Figure 2-12 have different intensity. These patches should be in different segments according to the probabilistic approach. The two patches in the green boxes have similar intensity, so these should be in the same segment.

Similarly, like the previous section, the content of this section is not directly needed by this research. But these are essential to segment out the moving object from the video in order to re-generate the trajectory.

2.6. Motion Structure

This section provides the basics for the re-construction of trajectories of moving human object. Most current structures from motion algorithms use a two-image method in their initial stages and, even though two images contain less information than a larger number, recent research has shown that structure and motion can be estimated from two images with surprising accuracy and robustness [32].

Usually, the structure from motion algorithms computes an initial estimate of the camera motion in first stage. In the second stage these algorithms improve the initial estimate by minimizing an error function with respect to the unknown motion and/or structure [33]. The Figure 2-13 is a figure with the plane and two cameras showing the structure from motion scenario.

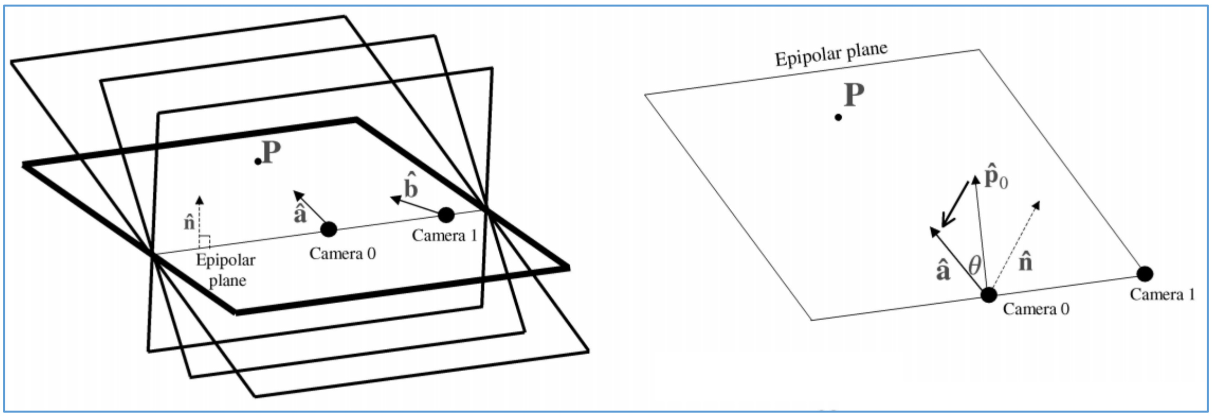


Figure 2-13: Structure from motion description [34]

The left section of the Figure 2-13Figure 2-13 shows the two stages of the process, choosing an epipolar plane passing through the camera positions and minimizing the result over all choices of the plane. The right section of the Figure 2-13 shows the optimal estimate after fixing the epipolar plane. The structure from the motion is described in further details in the following section.

2.6.1. Triangulation

The difficulty of finding a point's 3-dimensional position from a set of corresponding image locations and known camera positions is known as triangulation [35]. Conversion of image from 2 dimensions to 3 dimensions is also handled by this method. Delaunay triangulation is a popular method [36], which converts 2 dimensions image into 3 dimensions. The Figure 2-14 is an example of converting 2-dimensional image into 3 dimensions.

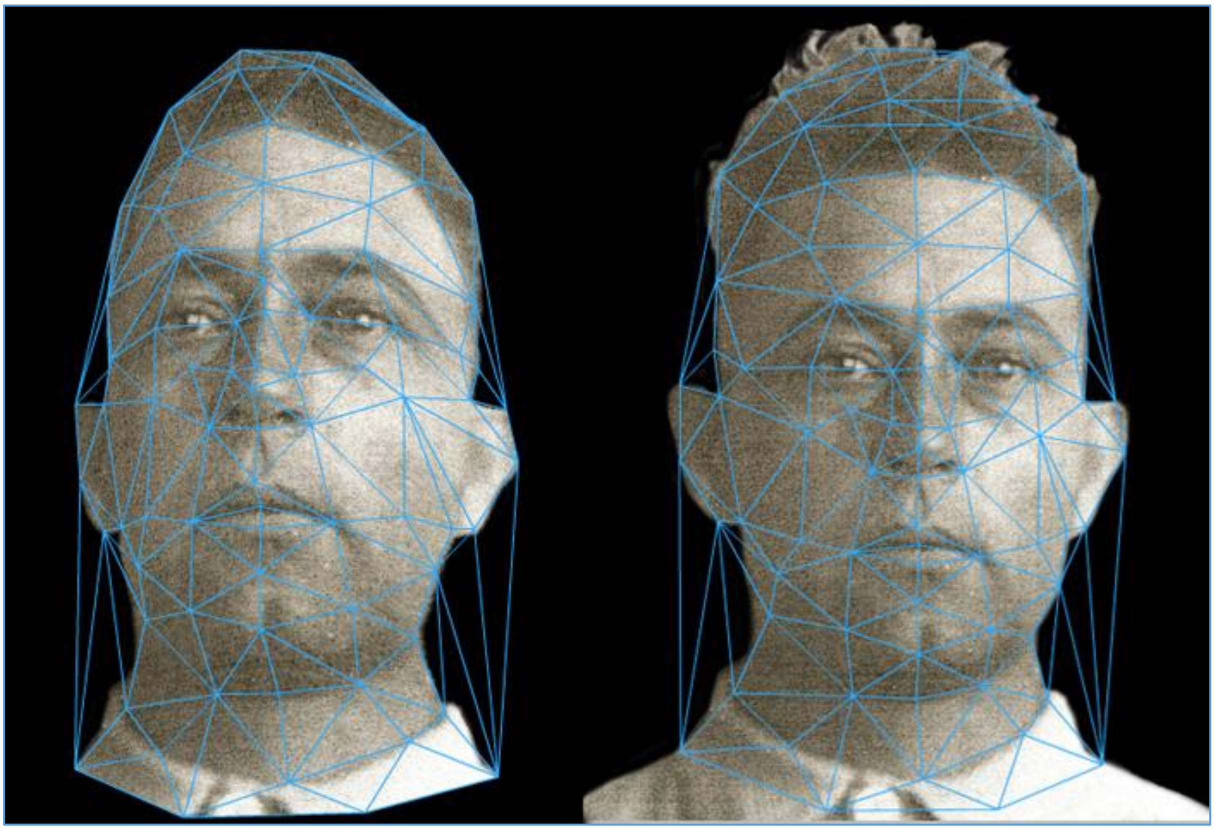


Figure 2-14: 2-Dimensional image conversion into 3 -Dimensional image [37]

2.6.2. Factorization

Factorization is a simple and effective way to reduce dimensions and extract features of interest in a video. This is used in several video processing applications such as background modelling in surveillance type videos, video content analysis. The conventional factorization method is not an online process. This factorization process is executed on the data matrix constructed by the observed examples [38]. The Figure 2-15 is an example of factorization of video frame.

Figure 2-15: Image factorization [39]

In the Figure 2-15, the left-hand part shows the real frame of a video. While the right-hand part of the figure shows the result generated after the process of factorization.

2.6.3. Bundle adjustment

Bundle adjustment is a combined name for techniques that refine an initial estimate of structure and camera motion, by minimising the re-projection errors over all images [40]. It has been shown that it is possible to use this bundle adjustment solvers even in a real time applications to improve a structure from motion estimation. Other works have studied techniques for improving the robustness of bundle adjustment [41]. Recent progress has been made in terms of stability and speed, especially for large scale problems where several thousand cameras and millions of points are refined and bundle adjustment can now be used to solve even city scale problems [42]. The Figure 2-16 is an example of an application of the bundle adjustment on an image:

Figure 2-16: Bundle adjustment on an image [43]

The left part of the Figure 2-16 is the real image, while the right part of the figure is a shaded representation of the bundle-adjustment triangulation.

2.6.4. Constrained structure and motion

This technique is based on a simple thought that even though severe appearance changes may occur to the target object in the tracking process, in most tracking tasks, the change of the structure of the target object in consecutive frames is relatively small [44]. Opposed to the simple motion model used in which the location of the tracker in frame t + 1 is assumed to be equally likely to appear within a radius r of the tracker location in frame t, the structure-constrained motion model measures the changes on the structure of the target object caused by displacing the tracker at different locations in frame t+ 1. And only those locations leading to small change on the structure of the target object counts for searching the tracker location in frame t + 1 [45]. The Figure 2-17 is the figure showing the object tracking with the help of constrained structure and motion.

Figure 2-17: Tracking an object [46]

First two parts of the Figure 2-17 are at time ‘t’ and ‘t + 1’. Both frames are consecutive frames. Featured points of the target object are tracked in both frames. The third part of the Figure 2-17 shows the estimated confidence in the root location of the target object in the frame t + 1, with the structure-constrained motion approach. The confidence increases with the intensity of the red colour. The circle in the third part of the Figure 2-17 denotes the simple motion model with the location of the tracker is being assumed to be equally likely to appear within a radius r (r = 30 pixels) of the tracker location in the previous frame.

Constrained structure and motion are an efficient method which is being used in this research for locating, tracking and regeneration of the moving object in the video.

2.7. Machine learning methods

Classification is the method of grouping data according to the properties that associate them with each other [47]. First, data are divided into two parts: training and testing data. Training data are the data used for training of machine learning algorithm. Testing data are the data to test the machine learning algorithm. Then the machine learning algorithms are used to classify the given data.

Machine learning methods information is used in chapter 4 and have more information related to this research. Machine learning can be supervised or unsupervised [48]. When an algorithm that classifies the data does not know which class the data belongs to, this is unsupervised learning. On the other hand, in supervised learning the labelling information of the data is given to the algorithm which will classify the data. This information is used to add experience in the machine, then it is expected to classify the data that the machine does not recognize.

2.7.1. Type of learning methods

Incremental re-construction of moving object trajectories is a requirement in many security and analytics’ related video processing applications. Few of these applications are available in the form of publications [49][50][51][52][53] and the results are productive but these results are not easily integrated into security systems and need a significant amount of resources to process the input video stream and the results are not accurate in all conditions. Therefore, an algorithm that can map the results and the requirements in an efficient and accurate way, is needed. Looking at the machine learning area for mapping is a useful tool to detect the moving object. Supervised and unsupervised are two learning methods used for machine learning algorithms. These are discussed in the section below.

2.7.1.1. Supervised learning methods

Supervised learning methods attempt to discover the relationships between input features and target attributes [54]. For each observation of the testing value, for example the output made by the [92][93] and [94] software.

The two main models used for linking the observations to the target outputs (such as moving human objects in the input video stream) by supervised learning methods are ‘Classification’ and ‘Regression’. Classification links the observed ‘features’ into pre-defined classes whereas regression models links the features on to a real-valued information. There are many classification and regression models used in different researches such as K Nearest Neighbour, decision trees, neural networks and support vector machines for representing classifiers. Some commonly used regression models include linear regression and logistic regression models while KNN can be used for either classification or regression.

2.7.1.1.1 Moving human object trajectory information

In this research, moving human object is identified for the trajectory information with the help of moving body parts in input video stream. These moving parts are identified and classified with the help of below classifiers:

1) Colour based classifiers.
2) Shape based classifiers.
3) Depth based classifiers.

2.7.1.2. Unsupervised learning methods

Unsupervised learning methods are methods for which for every observation produces a vector of measurement is produced but without labelled responses for the target [55]. In this research, the author focuses on supervised learning methods.

2.7.2. Application of machine learning methods

As mentioned above, this research uses supervised machine learning method as per the requirements and data availability. A supervised learning algorithm have a set of examples of pre-analysed data for which the required target properties are defined. This algorithm learns how to estimate the corresponding target properties of new given data, which are not known. The original set of examples are used as training data. Once the learning algorithm has been trained using all of the given data, if the same data is re-used to compute the error of the learning method, an excessively optimistic estimate of the error of the model should be obtained. This is because the training or learning process tries to ensure that the error of the model for the training data is as low as possible. Therefore, the learning method will be specifically suited to the training data. To get a more accurate estimate of how the model performs with new given data, some unseen data which is not used for the training process must be available along with its known properties for testing. It is therefore a common practice to set aside part of the original data for checking and not to use it in the training process. This dataset may be called the validation dataset [54]. After training the model using the remaining dataset, now known as the testing dataset, the performance may be tested or validated on the validation dataset [56].

The validation dataset can be used to fine-tune learning method. For example, one can use validation dataset with multiple sets of coefficients for a learning model and find the error produced by each set of coefficients. This will help in terms of finding out the most appropriate set of coefficients for a learning model. These cases also help in estimating the accuracy of a learning method with new given data. This is the case because we choose the final set of coefficients after seeing the accurate results produce with validation dataset.

Some researches [57] put another set of data aside from the original given data and it is neither used for training nor used in validation. This data set is usually called the test dataset in order for it to be kept different from the validation dataset. The error produced by the fine-tuned learning method applied to the test dataset gives a realistic estimate of the performance of the method on a completely new given data.

2.7.3. Researches using machine learning methods

There is an extensive range of published and commercialised frameworks for visual data processing which depend upon machine learning methods. Huge number of those are analysed for this research. Few of those are mentioned below which are investigated and analysed for this research.

2.7.3.1. Automated Detection of Dermatological Disorders through Image-Processing and Machine Learning

According to the authors of Dermatological diseases research, due to their high complexity, variety and scarce expertise is one of the most difficult terrains for quick, easy and accurate diagnosis especially in developing and under-developed countries with low healthcare budget [58]. The authors suggested three phases in this research to handle this difficult condition like any other machine learning method. These three phases are as below [58]:

1) Creation of dataset.
2) Data augmentation.
3) Data separation.

Dermatological diseases research uses the same approach as the one used in this thesis. This research creates a dataset of different types like Eczema, Herpes, Melanoma, and Psoriasis which uses augmentation techniques such as Synthetic Minority Over-Sampling Technique and rotation of images etc., to increase the amount of sample data. The Figure 2-18 illustrates the data flow.

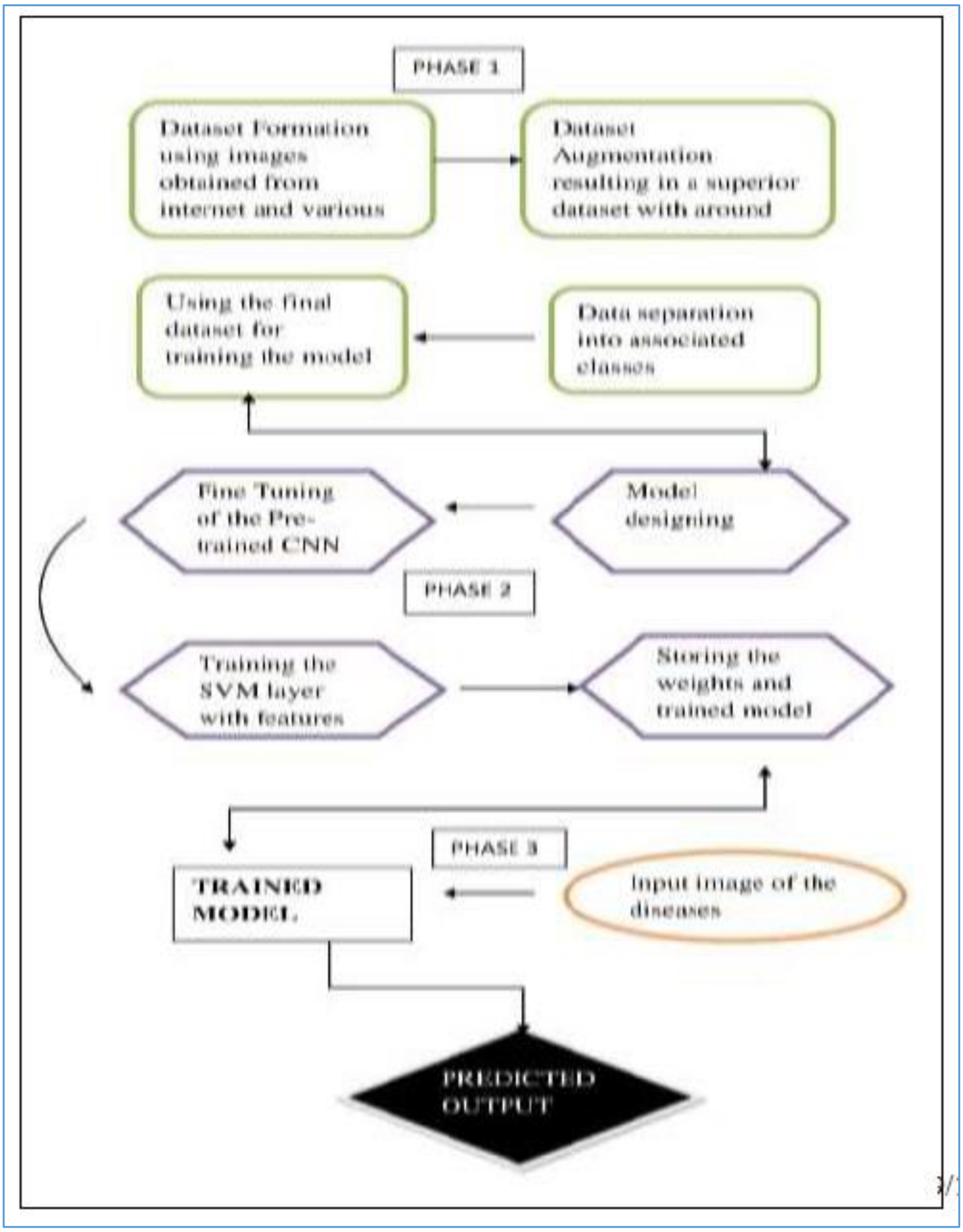


Figure 2-18: Data flow [58]

This research uses an amalgamation of Convolutional Neural Network (CNN) and Support Vector Machines (SVM) [58]. These methods are used to increase the accuracy of the entire Neural Network structure. The data obtained after pre-processing is processed using CNN. The features extracted by CNN are then passed through the SVM for further classification of the diseases.

The training process of this research method is divided in to two parts; first, the pre-trained model on their reduced dataset is trained and then they take the trained features from the final Convolutional layer and train the SVM classifier. Labels of trained data are shown in Figure 2-19:

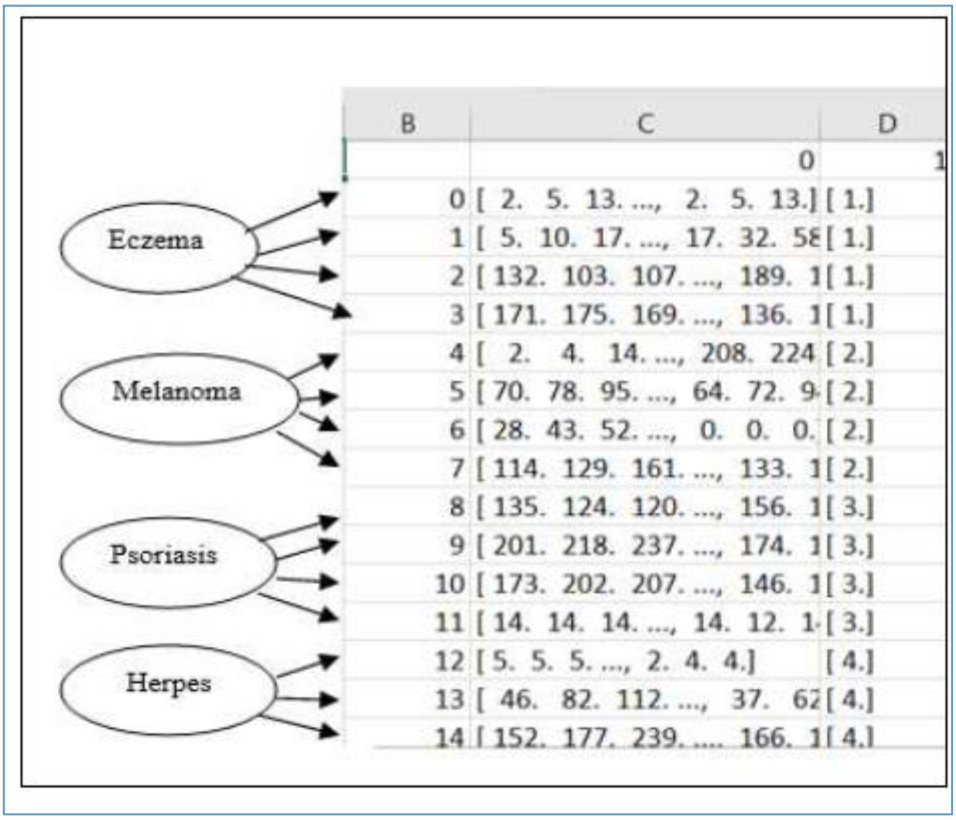


Figure 2-19: Labels are using training dataset [58]

The Figure 2-19 shows few of the images used for training and the array of labels generated as classifiers.

2.7.3.2. Marginal Space Deep Learning: Efficient Architecture for Volumetric Image Parsing

The authors of the "Efficient Architecture for Volumetric Image Parsing" research try to develop a system for anatomical object detection and segmentation support. The authors presented a solution to all the challenges, a novel feature-learning-based framework for parsing volumetric images split in a two-stage approach: anatomical object localization and non-rigid shape estimation [59]. For the first task, the authors present Marginal Space Deep Learning (MSDL), a solution exploiting the computational benefits of Marginal Space Learning (MSL) [60] and the automated, self-learned feature design of Deep Learning (DL). For the segmentation task the author proposes a learning based Active Shape Model (ASM) using a deep-learning-based system to guide the shape deformation. The Figure 2-20 shows the architecture of the research framework.

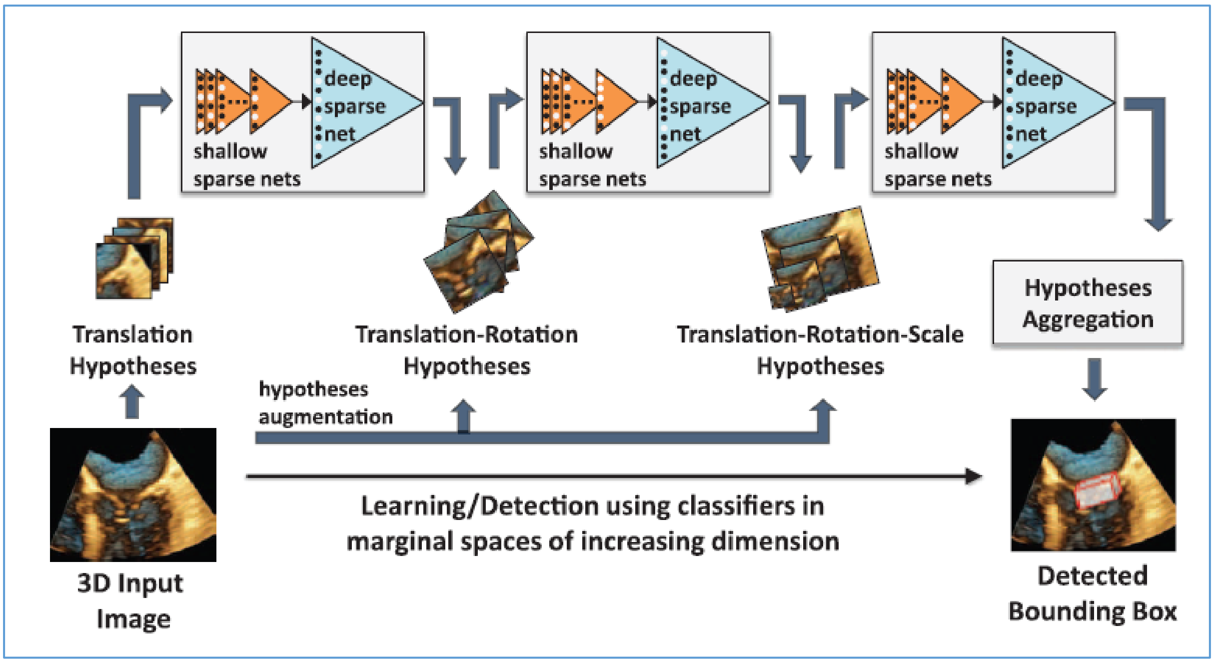


Figure 2-20: Schematic visualization of the marginal space deep learning framework [59]

2.7.3.3. Kernel based learning approach for satellite image classification using support vector machine

In this research, the procedural development for the analysis of multi spectral remote sensing data involves geo-referencing, image registration to a reference image, sub-setting of selected image region from the scene before doing the actual image classification using two methods (Maximum Likelihood and SVM) [61]. These geospatial processes were executed in sequence one after another in a workflow setup [62]. This workflow is shown in the Figure 2-21:

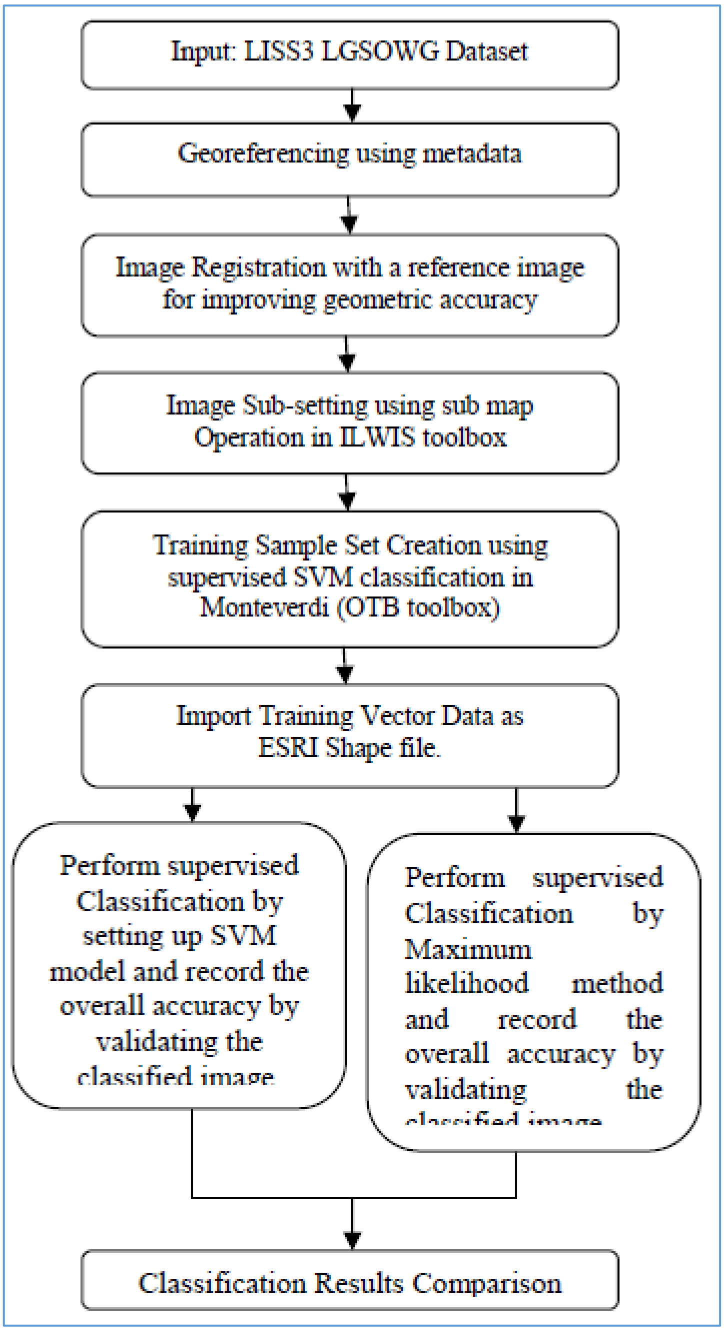


Figure 2-21: Supervised satellite image classification workflow [62]

The Figure 2-21 shows the procedural development for the analysis of multi-spectral remote sensing data involves georeferencing, image registration to a reference image, sub-setting of selected image region from the scene before doing the actual image classification using two methods (Maximum Likelihood and SVM) [62]. These geospatial processes were executed in sequence one after another in a workflow setup as shown in Figure 2-21. ILWIS [63] and Monteverdi toolbox have advanced process tree structure for geospatial processing which makes the analyst job comfortable [9]. In this workflow, few geospatial tasks were also carried out using utility software such as GDAL [64] to translate LGSOWG image data format into ILWIS native format which is nothing but general raster format with ASCII metadata files and project shape files for compatibility. LISS3 data set collected by National Remote Sensing Agency (NRSC) [65].

2.8. Moving object trajectory

The mission of robust tracking is very challenging because of the fast motion, occlusion, illumination variation, background clutters, real time restriction, structural deformation, etc. Massive studies are taking place in this area. Number of systems are using moving object trajectories as one of the main features in their systems. Few of the pertinent researches are presented in the following sections.

2.8.1. Heterogeneous Moving-Object Trajectory

A. Boulmakoul, L. Karim et al. [66] proposed an object-orientated data model of interoperability and information for traditional trajectories. The authors suggested the use of UML (Unified Modelling Language) to model the different real word objects [66]. They suggested a data model framework that combines the streaming technology and Service-Oriented Architecture (SOA) to improve the application performance and interoperability [66]. The Figure 2-22 shows the system's architecture:

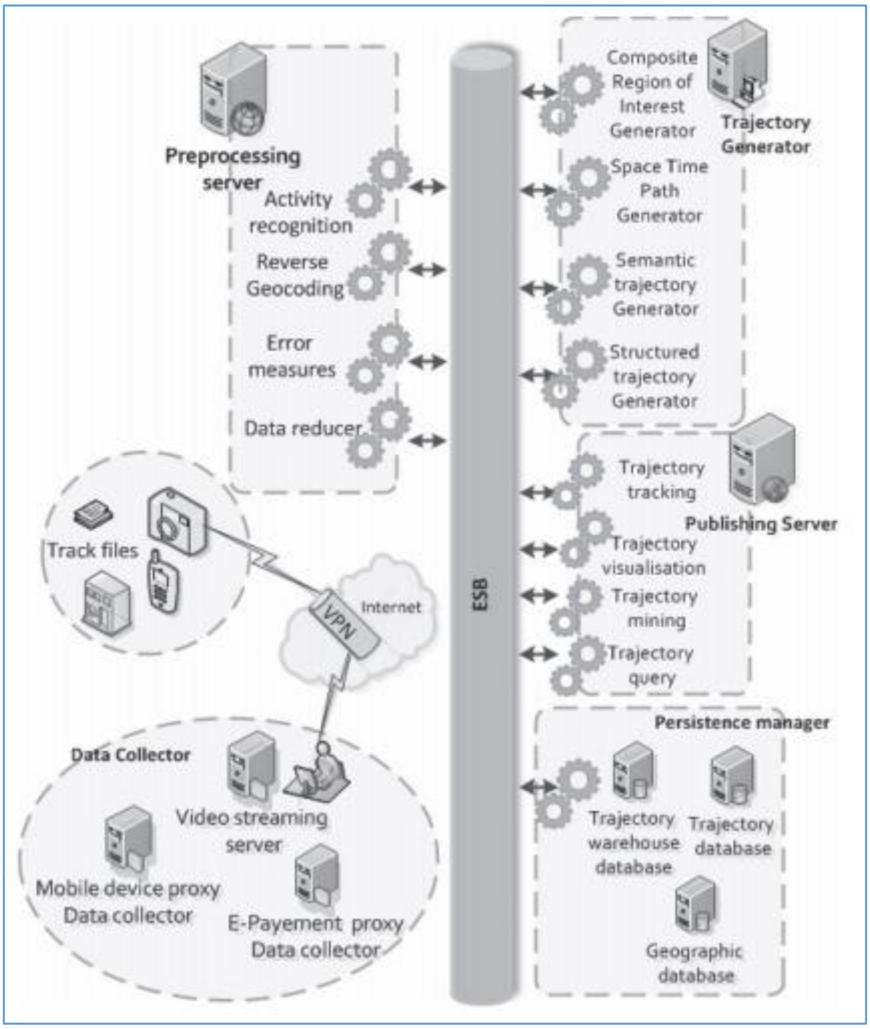


Figure 2-22: Heterogeneous moving object trajectory system architecture [66]

As shown in the Figure 2-22, the data collector layer contains mobile devices, e-payment, and video-streaming proxy servers, respectively, to collect data from mobile devices, self-service terminals, and cameras. Trajectory generator layer generates a set of spatial–temporal points which is transformed from a cleaned raw trajectory presentation to a structured, semantic trajectory.

2.8.2. Moving Objects Trajectories Detection algorithm

L. Zhang, X. Wen, W. Zheng, B. Wang [67] proposed an algorithm based on the relative location object, (according to the principle of relative motion of objects) in the video, which calculates the relative distance between various objects shown in the video, and then designs an object-location database, which is used to store the information of objects [67]. Based on the database of the trajectories of objects, the algorithm solves the problems which are caused by the camera movement [67]. The Figure 2-23 shows an example of object detection in the frames.

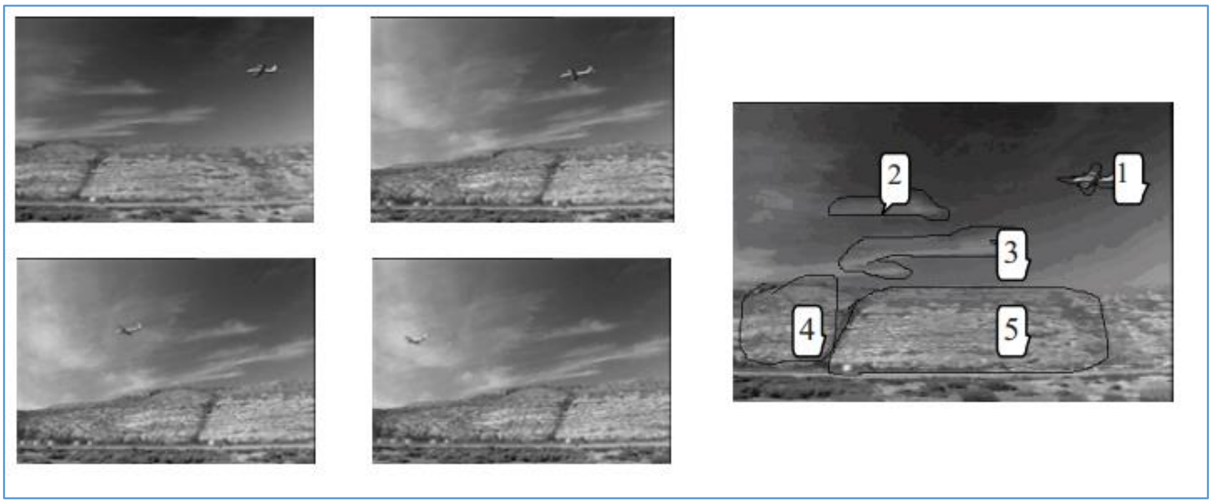


Figure 2-23: Four video frames and object tracking [67]

The left part of the Figure 2-23 shows the four frames of aeroplane video while the right-hand part shows the five detected objects in the frames with the labels 1 to 5. With the help of determining the displacement of moving object, they proposed the trajectories of moving objects as displayed in the table below.

frameid	iobjectid	jobjectid	xdistance	ydistance
1	1	4	-250	115
2	1	4	-66	116
3	1	4	39	96
4	1	4	116	85

Figure 2-24: Table showing the trajectory data of moving object [67]

As shown in the table, only object with the id ‘1’ has displacement, so, this is the only object moving in the frame and will have a moving object trajectory.

2.8.3. Content-based Retrieval using Trajectories

C. Shim and J. Chang [68] propose a new spatio-temporal representation scheme for multiple moving objects' trajectories in video data to efficiently retrieve the content of the data. They approximate the position of an object with the help of Minimum Bounding Rectangle (MBR). They define a moving object as one whose position is changed over a given time interval and the following as the centre point of an object in 2-dimensional plane:

$$(x_i, y_i) \tag{2.3}$$

So, the trajectory of a moving object is represented as below:

$$[(x_0, y_0, t_0), \quad (x_1, y_1, t_1), \dots, \quad (x_n, y_n, t_n)] \tag{2.4}$$

For time t0, t1, …, tn.

C. Shim and J. Chang [68] performed experiments on the football video and extracted the trajectories of football and these experiments results were 15% to 20% better than Li's and Shan's research results [68][69][70].

2.8.4. Surveillance video summarization and trajectories

Z. Ji, Y. Su, R. Qian, J. Ma [71] proposed an efficient framework for analysis and summarization of surveillance videos with the techniques of moving object detection and trajectory extraction [71]. According to them the video is first partitioned into segments based on moving object detection, then trajectory is extracted from each moving object, and then key frames are selected, together with the trajectories to represent the video segment [93]. They used hybrid motion detection algorithm which is proved to be fast and effective [72]. The Figure 2-25 shows few examples.

Figure 2-25: Moving object trajectory [93]

The figure displays three examples of moving object trajectories with the red colour trajectory.

2.9. Tracking moving objects in software applications

Usually, a moving object representation is a model of the tracked object that is used by a tracking algorithm. This model must contain information about the shape and the appearance of the object in question [73]. Mostly the model for a specific target object are computed in different ways as described below [74]:

- Defined with the help of previous research.
- Learned with the help of training samples.
- Image of the tracked object.

The above-mentioned ways are used in different applications. The selection of any of the above ways depends upon the requirements of the applications.

2.9.1. Medical applications

Moving object tracking in videos has been increasingly used by medical systems in order to support the diagnosis process and to speed up the operator's task [75]. Few examples are introduced below.

2.9.1.1. Ventricular motion tracking

Automated algorithms track the ventricular motion in ultrasound images. This is an example of tracking concerned object with the help of learning from training samples [76]. The Figure 2-26 is a figure which tracks the left ventricle from an ultrasound data.

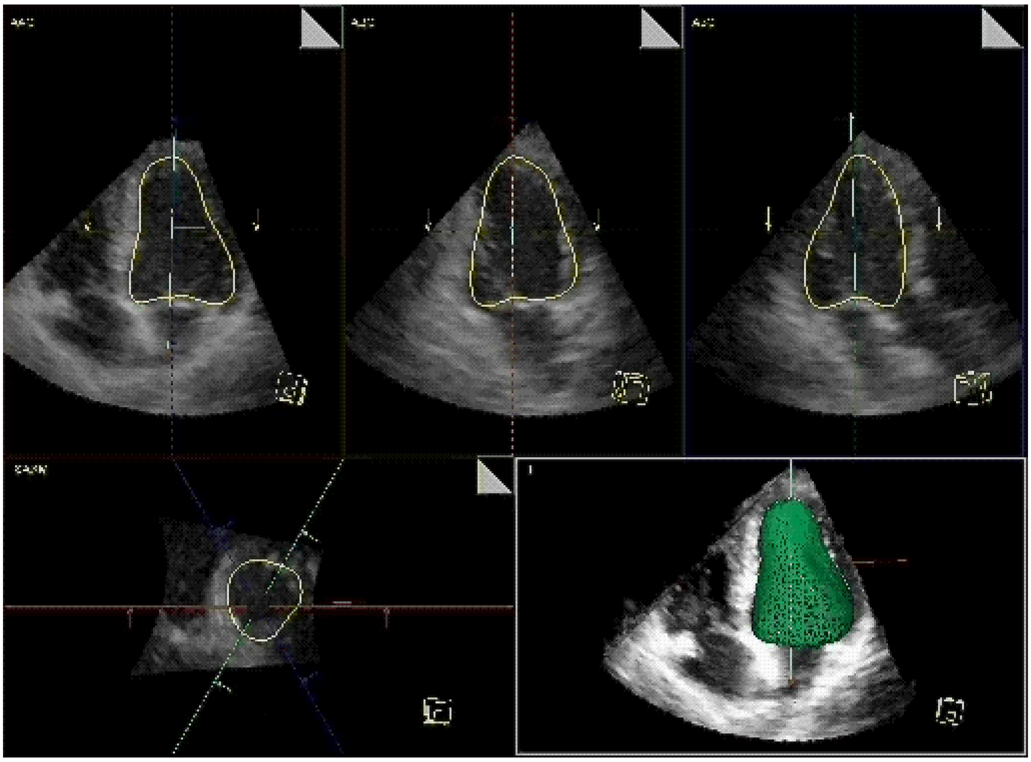


Figure 2-26: Tracking the left ventricle (LV) in 3D ultrasound data [77]

The Figure 2-26 shows the application using motion manifold learning, and introduction of two collaborative trackers to achieve both temporal consistency and failure recovery.

2.9.1.2. Soft tissues tracking

Needle surgery is one of the most complex fields of minimally invasive surgeries which has attracted several researchers [78][79]. When a surgeon inserts the needle into the tissue, he/she tries to use the needle body to steer it through a static or dynamic path to reach a specified target; the static and dynamic paths refer to offline and online path planning approaches respectively [79]. Mostly, the needle steering doesn't happen easily, since there is still no exact, real-time and comprehensive displacement-force feedback from the areas of interest, due to inherent properties of minimally invasive surgeries. On the other hand, tissue penetration causes some undesired events; e.g. target repositioning, obstacle reshaping and tissue overstretching [80].

Due to unpredictable events during a needle-based surgery, like uncertain bleeding of penetrated tissues, the human must be replaced by robots [81]. Robots must be able to dynamically re-plan and correct the pre-set trajectory and change it to a safer path. The Figure 2-27 is a block diagram of needle-based surgery system.

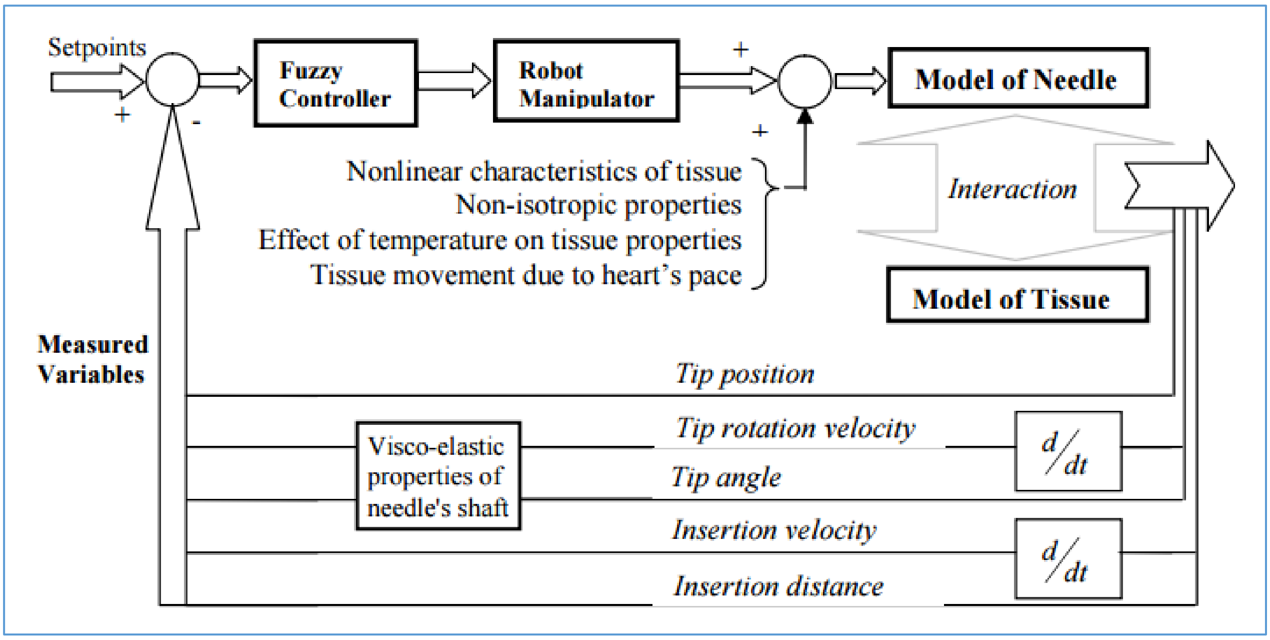


Figure 2-27: Block diagram for the robotic control and steering of surgical needles [82]

The Figure 2-27 shows closed loop system for fuzzy control of the needle in the tissue. This type of systems learns initially on the basis of different training samples.

2.9.2. Surveillance

In surveillance systems of dynamic objects, tracking can be used either as a forensic instrument or as a processing way of categorising activities. The Figure 2-28 shows an example of advance surveillance system.

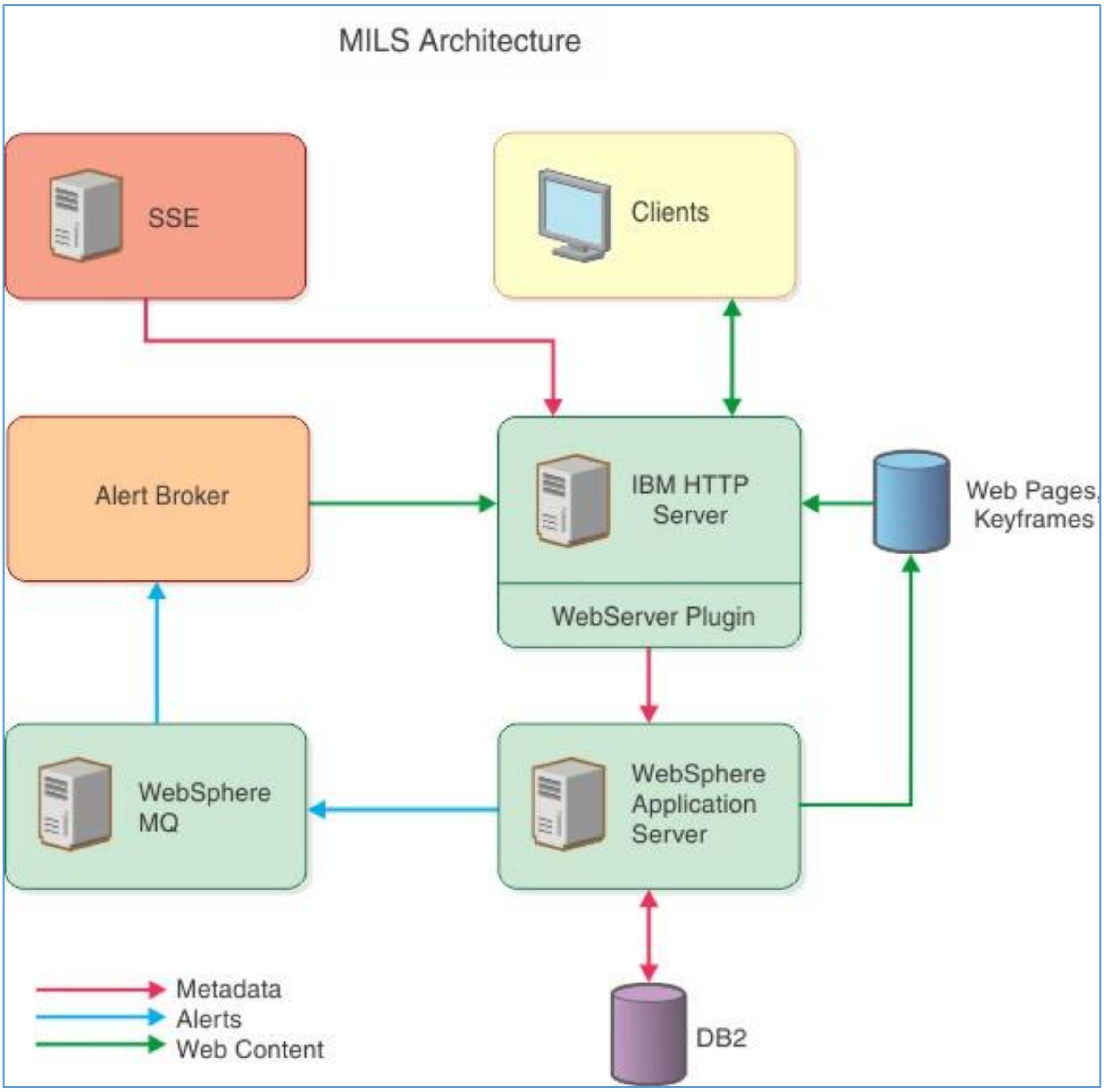


Figure 2-28: IBM system for Large Scale Surveillance [83]

The Figure 2-28 shows a system which index and digests the Meta data of video sequence. This data is full of events, which are necessary and needed for the forensic analysis. The data provided by this system are used with other surveillance solutions available in the market [84].

2.9.3. Business intelligence

Moving objects tracking in videos is also served as a reflection and measuring tool in retail environments [85]. For example, in superstores, the location of customers is tracked over time. Trajectory data combined with information from the points of sales (till) is used to build behavioural models of customers spending their time in the shop, how they interact with products depending on their location, and what items they buy [86]. By analysing this information, the marketing team can improve the product placement in the retail space. The Figure 2-29 illustrates the paths of customers in a superstore.

Figure 2-29: Moving paths of customers in a superstore [87]

The Figure 2-29 is used to show a trick to attract customers to spend more money. In this way they slow down the trolley of the customer with the help of floor.

Moreover, gaze tracking in front of billboards are used to automatically select the type of advertisement to show or to dynamically change its content based on the attention or the estimated marketing profile of a person, based for example on the estimated gender and age [88].

2.9.4. Unmanned vehicles

Usually, robot uses feedback information extracted from a vision sensor in order to control the motion. Sequential Monte Carlo methods (also known as Particle Filter - PF) is one of the best objects tracking method which overcomes most of the issue like background cluster, occlusion etc.[89]. This has been used for visual tracking because of non-Gaussian nonlinear assumption and multiple hypothesis property [90]. This estimates the target posterior based on Bayesian Framework. The basic idea was introduced by Hammersley and Morton [91]. CamShift Guided Particle Filter (CAMSGPF) has increased the efficiency of PF and solved the problem of sampling degeneracy and impoverishment by performing CamShift algorithm on a sample [92]. An example of CAMSGPF with the objects of same colour is displayed in Figure 2-30.

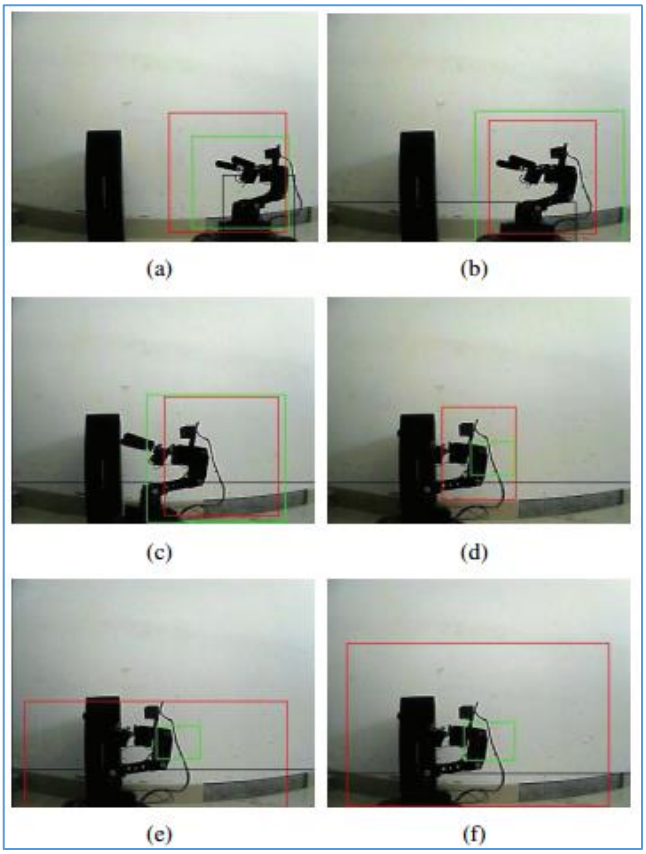


Figure 2-30: Tracking results of CAMSGPF and CamShift under colour clutter [93]

The Figure 2-30 shows a blacked arm robot. This robot is moving from right hand side of the image to the left-hand side. The authors [93] shows the result with the help of green rectangle. It is obvious that green rectangle tracks the object properly, while red rectangle is not able to track the object when two objects of the same colour are close to each other.

2.10. Mathematics primitives and transformations

Prior to efficiently analysing and handling videos, researchers need to find a vocabulary for defining the description of a video. In this research, one also need to recognize the image formation process that formed an image given a set of lighting circumstances, scene geometry, surface attributes, and camera optics. In this section, the investigator presents a simplified primitives and transformations which are required for the video processing and these primitives will be used for analytical description of moving objects in the video data. This information provides a basis for chapter 3.

2.10.1. Geometric primitives

The elementary concepts of planar geometry are known to anyone who has learned mathematics even at a fundamental level. In fact, they are so much of our everyday practice that we take them for granted. At a fundamental level, geometry is the study of points and lines and their relationships [94]. These parts play a basic role in image or video data processing.

2.10.1.1. Two dimensional Points and Two-dimensional Lines

As a normal understanding, a point in the plane may be represented by the pair of coordinates as follows:

$$(x, y) \ in\ plane\ R \ \mathrm{R} \tag{2.5}$$

Thus, the above is a common way to identify the plane with plane R. Considering IR2 as a vector space, the coordinate pair (x, y) is a vector – a point is identified as a vector [95]. In this section the regular notation for points and lines on a plane is introduced.

2.10.1.2. Two dimensional Conics

A conic is a curve defined by a second-degree equation in the plane [96]. In Euclidean geometry conics are of three main types: (apart from the so-called degenerate conics, which will be defined later) [97]:

- Hyperbola.
- Ellipse.
- Parabola.

Typically, the above three types of conic are generated as conic sections produced by planes of different orientation (the degenerate conics generated from planes which contain the cone vertex) [98]. However, a 2D projective geometry of all non-degenerate conics are equivalent under projective transformations. The conic equation below is written in inhomogeneous coordinates:

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{2.6}$$

This equation is a second-degree polynomial equation and with further substitutions:

$$x \rightarrow \frac{x_1}{x_3}\,, \qquad y \rightarrow \frac{x_2}{x_3} \tag{2.7}$$

Equation 2.2 becomes:

$$ax_1^2 + bx_1x_2 + c{x_2}^2 + d\,\frac{x_1}{x_3} + e\,\frac{x_2}{x_3} + fx_3^2 = 0 \tag{2.8}$$

And in the matrix form, this can be written as:

$$x^T C x = 0 \tag{2.9}$$

Where the conic coefficient matrix C is given by:

$$\begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix} \tag{2.10}$$

The above conic coefficient matrix is symmetric. In the situation of the homogeneous representation of points and lines, only the ratios of the matrix elements are important, since multiplying C by a non-zero scalar does not affect the above equations. Thus, C is a homogeneous representation of a conic. The conic has five degrees of freedom which can be thought of as the following ratios:

$$\{a : b : c : d : e : f\} \tag{2.11}$$

This is equivalent to the six elements of a symmetric matrix [99] and the Figure 2-31 shows the three-conic section,

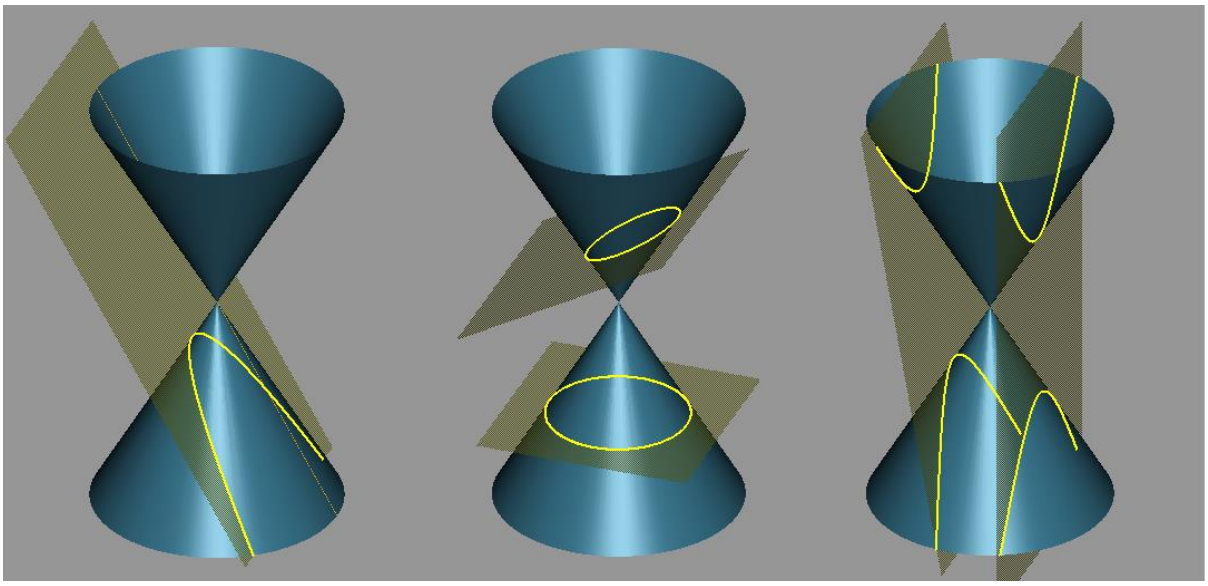


Figure 2-31: Conic sections [100]

The first part of the Figure 2-31 shows parabola, cutting plane parallel to the side of cone. The second part shows circle and eclipse. The third part shows hyperbola.

2.10.1.3. Three-dimensional Points and Three-dimensional Lines

The following three-dimensional coordinates are described either using inhomogeneous coordinates:

$$(x, y, z) \in R3 \tag{2.12}$$

or homogeneous coordinates:

$$(\tilde{X}, \tilde{Y}, \tilde{Z}, \widetilde{W}) \in P3 \tag{2.13}$$

2.10.1.4. Three-dimensional Planes

A plane in 3-dimensional space may be written as:

$$\pi_1 X + \pi_2 Y + \pi_3 Z + \pi_4 = 0 \tag{2.14}$$

Clearly this equation is not affected when multiplied by a non-zero scalar. Therefore, the three independent ratios take the following form:

$$\pi_1 : \pi_2 : \pi_3 : \pi_4 \tag{2.15}$$

The plane coefficients are significant in equation (2.10). This means that any plane has 3 degrees of freedom in 3-dimensional space. The homogeneous description of the plane is the 4-dimensional vector as shown below:

$$\pi = (\pi_1, \pi_2, \pi_3, \pi_4)T \tag{2.16}$$

One can homogenize equation (2.10) by the replacements described below:

$$X \rightarrow \frac{x_1}{x_4} \tag{2.17}$$

$$Y \rightarrow \frac{x_2}{x_4} \tag{2.18}$$

$$Z \rightarrow \frac{x_3}{x_4} \tag{2.19}$$

Which results in the following:

$$\pi_1 x_1 + \pi_2 x_2 + \pi_3 x_3 + \pi_4 x_4 = 0 \qquad (2.20)$$

or more concisely, one can write the following:

$$\pi^T X = 0 \qquad (2.21)$$

Which states that the point X is on the plane π. The first 3 parts of π correspond to the plane normal of Euclidean geometry. Therefore, inhomogeneous equation (2.17) becomes the known plane equation written in 3-dimensional vector notation as,

$$n.\tilde{X} + d = 0 \qquad (2.22)$$

Where,

$$n = (\pi_1, \pi_2, \pi_3)^T \qquad (2.23)$$

$$\tilde{\mathrm{X}} = (\mathrm{X}, \mathrm{Y}, \mathrm{Z})^{\mathrm{T}} \qquad (2.24)$$

$$x_4 = 1 \qquad (2.25)$$

And

$$d = \pi_4 \qquad (2.26)$$

Given this, the distance of the plane from the origin is as follows:

$$\frac{d}{||n||} \tag{2.27}$$

2.10.2. Transformations

2.10.2.1. Two-dimensional Transformations

2.10.2.1.1 Translations

All the researchers state that moving object trajectories consist of translation of points in the plane. Translation is the motion of a body from one point to another in the plane. Simply use a point as a normal point at position ‘P1’. This is a simple operation that is easy to be formulated mathematically. Let’s move the point from position ‘P1’ to a new position ‘P2’.

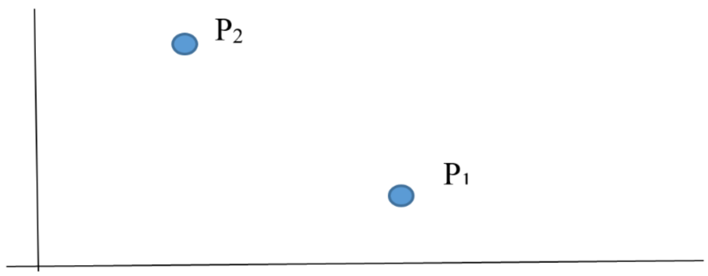


Figure 2-32: Translation of two points

This translation can either be written as,

$$x' = x + t \tag{2.28}$$

Or as,

$$x' = [I \quad t]\ \tilde{x} \tag{2.29}$$

Where I is as identity matrix and then the above equation can be written as below:

$$x' = \begin{bmatrix} I & t \\ 0^T & 1 \end{bmatrix} \tilde{x} \tag{2.30}$$

2.10.2.1.2 Rotations

When trajectories of moving objects are created, the rotation motion of the moving objects is considered. Rotation matrix is a matrix that is used to show a rotation in Euclidean space [101]. The rotation of object can be written as below:

$$x' = [R \quad t]\, x \tag{2.31}$$

Where,

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{2.32}$$

We can show the rotation of two points in the Figure 2-33:

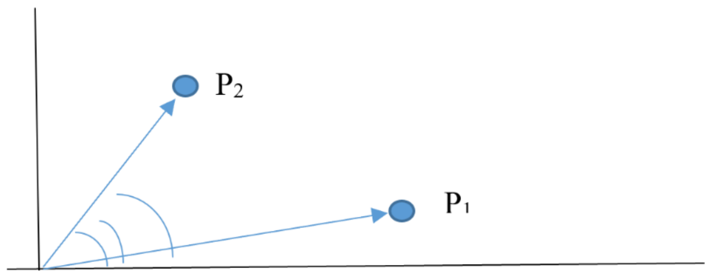


Figure 2-33: Two points rotation

2.10.2.2. Hierarchy of Two-dimensional Transformations

During the image processing, sometimes detected moving object images are subject to geometric distortion introduced by viewpoint irregularities. These irregularities can be improved with the help of different transformations. The Figure 2-34 shows the list of transformations and names of transformations.

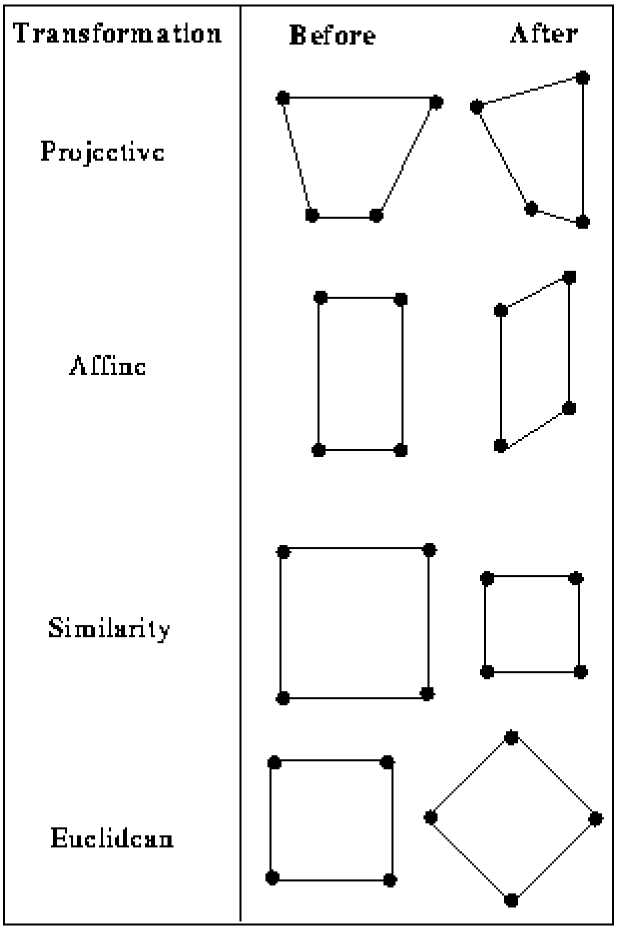


Figure 2-34: Image transformation hierarchy [102]

2.10.2.3. Three-dimensional Transformations

2.10.2.3.1 Translations

As mentioned in the previous section, translation is the displacement of object from one point to another point in the plane.

This translation can be written as below:

$$x' = x + t \tag{2.33}$$

Or this can also be written as,

$$x' = [I \quad t] \; \tilde{x} \tag{2.34}$$

Where I is as identity matrix with 3 x 3.

2.10.2.3.2 Rotations

Axis is an important component to describe the rotation in the 3-dimensional plane. This information is needed to describe the rotation of moving object in the video. The Figure 2-35 shows an example of rotation scenarios about x-axis:

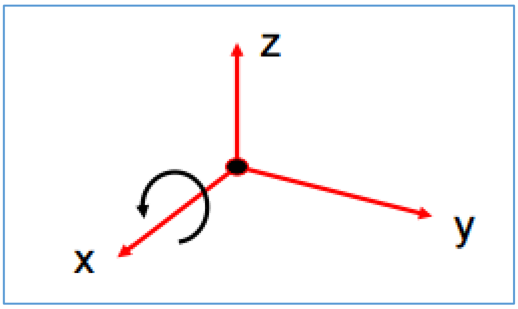


Figure 2-35: Rotation about x-axis [103]

Let's assume that there is a point 'X' which rotates about x-axis, this can be written as below:

$$X' = R_X\,(\theta)\,X \tag{2.35}$$

This can also be written as below if we replace all the components:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.36}$$

2.10.2.3.3 Exponential twist

Exponential twist is used to characterize the rotation motion vector of the body in the plane. This is also known as axis-angle representation. This can be used to describe the rotation of a moving body in an image. The Figure 2-36 is an example to show exponential twist of a vector.

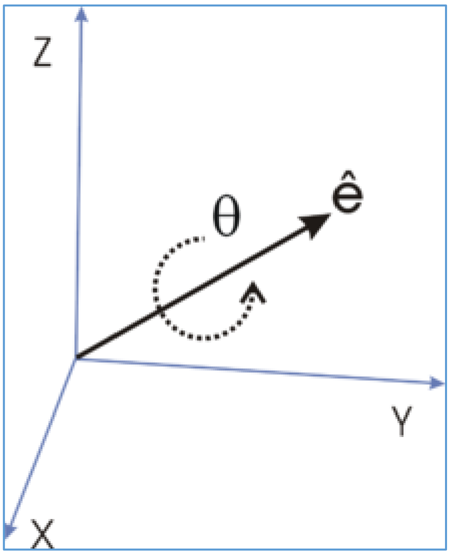


Figure 2-36: Exponential twist [104]

This is represented with the help of unit vector and the angle with the vector, so the Figure 2-36 can be written as below,

$$e = \theta\,\hat{e} \tag{2.37}$$

Where $\hat{e}$ is the direction of rotation of vector and θ is the angle of rotation.

2.10.2.3.4 Unit Quaternions

The above sections of this thesis have shown that a rotation in three-dimensional planes about an axis through the origin can be represented by a 3×3 orthogonal matrix with a determinant of 1. However, the matrix representation seems redundant because only four of its nine elements are independent [105]. Also, the geometric understanding of such a matrix is not clear until we carry out several steps of calculation to extract the rotation axis and angle. Furthermore, to compose two rotations, we need to compute the product of the two corresponding matrices, which requires twenty-seven multiplications and eighteen additions.

Quaternions are very efficient for analysing circumstances where rotations in three-dimensional planes are involved. A quaternion is a 4-tuple, which is a more concise representation than a rotation matrix. Its geometric meaning is also more obvious as the rotation axis and angle can be trivially recovered. The quaternion algebra to be introduced will also allow us to easily compose rotations. This is because quaternion composition takes merely sixteen multiplications and twelve additions.

2.10.3. Point Operators

The most basic kinds of image processing transform are point operators, where each output pixel's value depends on only the corresponding input pixel value (plus, potentially, some globally collected information or parameters) [106]. Examples of such operators include brightness and contrast adjustments as well as colour correction and transformations. Usually, in the image processing literature, such operations are also known as point processes [107]. The Figure 2-37 shows a digital representation of an image.

101	100	103	105	107	105	103	110
110	140	120	122	130	130	121	120
134	134	135	131	137	138	120	121
132	132	132	133	133	150	160	155
134	140	140	135	140	156	160	174
130	138	139	150	169	175	170	165
126	133	138	149	163	169	180	185
130	140	150	169	178	185	190	200

Figure 2-37: Digital image [108]

2.10.3.1. Pixel Transformation

A general image processing operation is a function that takes one or more input images and produces an output image. In the continuous domain, this can be written as below [109]:

$$g(x) = h(f(x)) \tag{2.38}$$

Or, this can also be written as below if we replace input image:

$$g(x) = h(f_0(x), \quad f_1(x), \dots, \quad f_n(x)) \tag{2.39}$$

Where, h, is the transformation function and $f_n(x)$ is the input image.

2.10.3.2. Histogram Equalization

Histogram equalization uses a monotonic, non-linear mapping which re-assigns the intensity values of pixels in the input image such that the output image contains a uniform distribution of intensities (i.e. a flat histogram). This technique is used in image comparison processes (because it is an effective enhancement) and in the improvement of non-linear effects introduced by, say, a digitizer or display system [110]. This image compression function played a vital role in moving objects detection.

2.11. Conclusion

Despite the research that is highlighted above which has, no doubt, made a good contribution in this area of interest, there are still unresolved problems which need to be explored. This chapter discusses different aspects needed for this research for video analysis. First, the author discusses the mathematical requirements for the main aim of reconstruction of incremental trajectories. This include the survey of different available methods and finding of methods best suited for this research requirements. Similarly, to ease the calculation and make the trajectory re-construction process fast, building a framework needs projection and manipulation of the visual scene under observation. These aspects are researched, and different approaches are examined to select the appropriate method. These approaches are mostly used during implementation of the framework.

The reviewed literature mostly concerns the methods for extraction and classification based on direct processing of the image frames obtained from the input visual stream. Very rarely if ever they consider the logical relationships between static and dynamic entities of the environment, which is of principal importance for any real-time application. It would be difficult to recognise, for instance, a pattern of a "man walking in a corridor" in a building based on the methods analysed in this chapter, because such a scenario would involve processing of too many information that needs to be processed in real-time: identification of the background in the scene, recognition of the walking activity, estimation of the distances between different entities involved, tracking of moving parts of moving human object, establishing the direction of movement for a limited period of time, etc. This might be possible by visual data processing methods in some identical conditions, but this would require huge computational power because of the volume of information which needs to be processed in real time and the complexity of the images which contain many layers of information. The original problem can be simplified by splitting the analysis limited to the required information only – first a researcher must detect the moving human object with essential features which needs to be analysed into the input visual stream. Then we have tracked the moving parts of the moving object. These are the required information from visual scene and need processing from input stream data. Afterwards, we will have to do data post processing in order to find the direction of motion, view direction etc. These data post processing tasks need less computation power and are possible in real time. These are necessary for re-construction of moving object trajectories and enriching it by organising it into efficient model. The model-driven analysis and information holding simplify the analysis and reconstruction of moving object trajectories

The research conducted in this chapter in several areas related to video analysis and re-construction of moving object trajectories from visual data input has raised the awareness not only of the available methods with their advantages, but also of their limitations; amongst them are:

- An efficient reconstruction of moving object trajectory from the video in real time.
- An intelligent algorithm is needed to improve the moving object trajectory processing time in order to enhance the performance of the overall system.

This is extremely important for the success of this research like any other research; its requirements must be well-informed, and options are selected carefully. The aim of the extraction methods is to successfully identify static and moving objects in the input visual scene taken by a physical camera, which can be monitoring different environments. The classification methods try to classify the moving objects and movements in order to establish their trajectories.

This research aims to investigate which features are most likely to help in the detection of moving object in visual scene and which algorithms are best suited for the detection task from visual input. It may be possible to improve on existing algorithms by tailoring them or by combining with some other techniques. There are several machine learning techniques available for training classifiers and regression-based techniques. Methods of training, validating and testing these are recommended in the machine learning literature. The next chapter introduces the conceptualization of framework and describing the model needed for the re-construction of moving object trajectories. The literature contains some examples of combination of the different methods - to make the analysis of visual scene fast and accurate. However, none of the reviewed frameworks considers incorporating of a full data model driven by real world data that would be used for reconstruction of trajectories in real time. These frameworks are compared comprehensively in chapter 6. Chapter 3 documents the various improvements made to the model and chapter 4 describes the required classifiers and their usage~~.~~

3. Conceptualization of moving objects trajectories framework

Humans can recognize and reconstruct the moving object trajectory based on visual observation and individual level of knowledge. Now, in this age of advanced technology, it is necessary to delegate the process of extracting information to machines.

The benefits of this would:

- Allow anyone to automate the tracking and reconstruction of human body movements by the machines.
- Improve security and safety management.
- Gain better business customer insight.
- Further advance analysis of human movements in different fields (like in sport or in educational institutes), etc.

Performing such information extraction in real time is a challenging task, due to the necessity of combining complex visual as well as analytical data processing which requires a significant amount of computing resources.

Whereas the processing of visual data is simpler due to current technological developments, but this requires model based information extraction which makes the task much more difficult.

There are several ways to model a moving object, which helps in the reconstruction of its trajectories. This chapter will, therefore, discuss a few different models in order to make the trajectories reconstruction process efficient and easy.

The demonstration of the moving objects in graphical models takes different shapes at different stages of this research process.

3.1. Types of the trajectories of moving object

Moving human object trajectories can be classified into four different types depending upon the view of extracting information which are provided below.

3.1.1. Dynamic trajectories

These trajectories are perceived directly from visual observation of physical movements and body parts of individuals moving object in an input video stream. An example of this type of trajectory can be a person walking in a corridor.

3.1.2. Cognitive trajectories

Trajectories are directed by goals of a moving object, intentions and actions in pursuit of reaching them. An example of a cognitive trajectory is, when a customer enters a grocery shop and picks items from the shelves.

3.1.3. Psychological trajectories

Moving human object trajectories are driven by the emotional state of moving objects. For instance, when a person is crying and going to a bedroom.

3.1.4. Social trajectories

These trajectories depend upon the interrelations that exist between moving human objects and their actions towards each other, such as a kid walking towards their mother or two friends meeting with each other.

The desire of this research is on extracting information about moving objects for all the above type of trajectories from input video stream. There is another way of extracting information of moving object by analysing the above-named trajectories. The scrutinization analysis of the various types of trajectories will improve the accuracy of the moving object trajectory's outcome and even add the possibility of trajectories prediction. The main aim of this research is to gather information on moving human object(s) and their moving parts from an input video stream in order to re-construct the moving object trajectory. As stated above, a moving object is tracked based on the use of models, which help in the reconstruction of its trajectories. This will be discussed below in further details.

3.2. Models of the moving object for trajectories reconstruction

The research initially started with a dot-shape based model. However, after exploring the possible benefits and limitations, a decision was taken to switch to a triangle-shape based model. These models are discussed below.

3.2.1. Trajectories of dot-based model

In this model, a human moving object is represented with a dot. Initially, the first step of this research was to model the whole moving object in a dot-based model as illustrated in Figure 3-1 :

Figure 3-1: Trajectory with the help of dot-based model [111]

The moving object is walking in a forward direction in the above screenshot of the video where the red dots represent the different positions of the moving human object through different video frames. Such a continuous stream of these dots helps in the reconstruction of the trajectory of the object. These dots are approximate positions in each frame.

3.2.1.1. Benefits of usage of dot-based model

The moving object is tracked in the video and is represented by a dot shape. Mathematical representation of the dots is simple and needs co-ordinates for the reconstruction of trajectories. Modelling the object with the dot-based model is easy and needs relatively less processing resources when compared to a complex trajectory model. In this case, only a minimal amount of information is extracted for the purpose of trajectory reconstruction.

3.2.1.2. Theory about usage of dot-based model

The displacement in the space leads to changes in the spatial-temporal state since object movement from one place to another is time-bound and is expressed with the help of a dot coordinates at a time instance 1 as shown below.

$$D_{t1} = (x_{t1}, y_{t1}) \tag{3.1}$$

Similarly, this is shown at time instance ‘n’,

$$D_{tn} = (x_{tn}, y_{tn}) \tag{3.2}$$

As such, for the purposes of reconstructing the trajectory of the object, this model requires less information from the video.

3.2.1.3. Input and output of dot-based model

The input of this dot-based model is a video input stream and it processes one video frame at a time. The model detects the moving object(s) in the video frame and locates its centroid. The result of the graphical output of this process is shown in the picture above entitled ‘Trajectory with the help of dot-based model’. The textual output generated by this model from the console is shown in the Figure 3-2:

```
Processing frame 1
Find the moving object(s)
0 object(s) detected
Processing frame 1
Find the moving object(s)
1 object(s) detected
Dot location => X = 152, Y = 155
Processing frame 2
Find the moving object(s)
1 object(s) detected
Dot location => X = 172, Y = 160
Processing frame 3
Find the moving object(s)
1 object(s) detected
```

Figure 3-2: Textual output of dot shape-based model in the console

The console output processes each video frame at a time during which it finds the moving object and subsequently reports its x and y co-ordinates.

3.2.1.4. Limitation of dot-based model

This dot shape-based model is limited in its illustration of different sizes as well as actions of human moving objects. For example, if an object bends or is seated the dot-based model has no way of representing that information.

3.2.2. Updating from dot-based model to prismatic model

After considering the above limitations, the dot-based model was discarded in favour of the prismatic model. The aim of developing the prismatic shape-based model is to introduce more information about the moving object's trajectory.

3.2.3. Trajectories of prismatic model

The representation of the moving object in this model is done with a prismatic shape instead of single sphere-based model, which allows for a comprehensive coverage of the object. See the Figure 3-3 for an illustration.

Figure 3-3: Trajectory with the help of prismatic model [111]

As shown in Figure 3-3, the prismatic shape-based model draws the prismatic shape in each frame, which covers the moving object and the information of these prismatic changes formulates the trajectory of the moving object while maintaining the same correct position of the moving object as it was in the dot-shape based model.

3.2.3.1. Benefits of using a prismatic model

The prismatic shape-based model helps in estimating the size of the object and provides the potential of using prismatic shapes to represent different parts of the object. i.e. one prismatic shape for each of the 4 limbs etc. This may also allow for the creation of a new shape, modelled after a part of an object in motion. For instance, when you wave your arm (which is represented by one prismatic shape) in a full circle motion, it creates a sphere.

3.2.3.2. Theory of the prismatic model usage

This model is represented with three coordinates and the value of these coordinates change for each video frame each time the moving object(s) change(s) the position. This can be represented as shown below for an initial time instance,

$$P1_{t1} = (x_{p1t1}, y_{p1t1}) \tag{3.3}$$

$$P2_{t1} = (x_{p2t1}, y_{p2t1}) \tag{3.4}$$

$$P3_{t1} = (x_{p3t1}, y_{p3t1}) \tag{3.5}$$

Similarly, for time instance ‘n’, these co-ordinates can be shown as below:

$$P1_{tn} = (x_{p1tn}, y_{p1tn}) \tag{3.6}$$

$$P2_{tn} = (x_{p2tn}, y_{p2tn}) \tag{3.7}$$

$$P3_{tn} = (x_{p3tn}, y_{p3tn}) \tag{3.8}$$

The continuous stream of values of the equations above, reconstruct the trajectories of moving objects.

3.2.3.3. Input and output of the prismatic model

Input of this prismatic shape-based model is the video which consists of multiple frames and it processes each frame at a time. Graphical output is shown in the Figure 3-3 and the Figure 3-4 is a screenshot of the console output of the detected points for a moving object.

```
Processing frame 1
Find the moving object(s)
0 object(s) detected
Processing frame 1
Find the moving object(s)
1 object(s) detected
Point P1 => X = 161, Y = 122
Point P2 => X = 134, Y = 188
Point P3 => X = 175, Y = 188
Processing frame 2
Find the moving object(s)
1 object(s) detected
Point P1 => X = 171, Y = 133
Point P2 => X = 145, Y = 193
Point P3 => X = 186, Y = 193
Processing frame 3
```

Figure 3-4: Textual Console output of the prismatic model

As shown in Figure 3-4, each frame is processed one at a time and after that, it tries to detect the moving object. If the object is detected, it then tries to find the three co-ordinates to draw a prismatic shape and this shape covers the whole body of the moving object.

3.2.3.4. The limitations of the prismatic model

Although the prismatic model provides better coverage and therefore a more accurate size of the object when compared to the dot-shape based model, it is nonetheless, not an exact size. Furthermore, this model neither covers any bending motion of the moving object nor the thickness (or width) of the moving object.

3.2.4. Transformation of the prismatic model to a sphere-based model

To overcome the above-mentioned weaknesses, the prismatic model was transformed into a sphere-based model which is capable of representing more information.

3.2.5. Trajectories of sphere-based model

This sphere-based model creates continuous spheres covering the moving object's body at different frames and the continuous stream of these spheres reconstructs the moving object's trajectory as shown in Figure 3-5.

Figure 3-5: Trajectories of the sphere-based model [111]

3.2.5.1. Benefits of usage of sphere-based model

This sphere-based model provides a more accurate size of the moving object compared to the prismatic model and it needs less information to draw the sphere around the object. This model also estimates the width of the moving object.

3.2.5.2. Theory about usage of sphere-based model

To draw a sphere around a moving object, the centre point of the object within a scene must be located first. For example, a centre of an object can be the abdominal area. Once the centre is located, the furthest point from it is then calculated and the sphere is drawn around the object/individual. For instance, when a centre is the abdominal area, the furthest other point from it is the head and/or feet.

This furthest distance acts as the sphere radius of the moving object while the centre of the body acts as the centre point of the sphere. At time instance ‘t_1’, the centre point can be expressed as:

$$S_{t1} = (x_{t1}, y_{t1}) \tag{3.9}$$

And the radius as,

$$R_{t1} = l_{t1} \tag{3.10}$$

Similarly, for time instance ‘t_n’, the above two equations can be written as,

$$S_{tn} = (x_{tn}, y_{tn}) \tag{3.11}$$

$$R_{tn} = l_{tn} \tag{3.12}$$

3.2.5.3. Input and output of sphere-based model

The input of the sphere-based model is a video, whereby, each frame is processed one at a time. Next, the moving object in each frame is detected and the sphere’s centre point along with its radius are determined. To illustrate, the graphical screenshot is shown in the Figure 3-5 and the textual console values are shown in Figure 3-6.

```
Processing frame 1
Find the moving object(s)
0 object(s) detected
Processing frame 1
Find the moving object(s)
1 object(s) detected
Centre point => X = 153, Y = 152
Radius => R = 43
Processing frame 2
Find the moving object(s)
1 object(s) detected
Centre point => X = 173, Y = 163
Radius => R = 41
Processing frame 3
```

Figure 3-6: Textual values of sphere-based model

As shown in Figure 3-6, each frame contains the centre of a sphere and the radius data and this information is only available if there is a moving object in the frame. The first frame does not have any moving object; hence no value is associated with that.

3.2.5.4. Limitation of sphere-based model

A limitation of this model lies in the assumption that the width as well as the height of the object in motion is the same in all directions. This is due to the fact that, the sphere-based model does not allow for the proper detection of object rotation.

3.2.6. Updating from spheres-based model to seven spheres-based model

After considering the above limitations, the model was further improved and developed to be seven spheres-based model. The goal of this update is to capture more information on the trajectories of the moving object and their respective moving parts.

3.2.7. Trajectories of seven spheres-based model

The seven spheres-based model creates a continuous sphere covering the body of the moving object and the resultant stream of values reconstruct the trajectory; the capture of such information is depicted in the Figure 3-7.

Figure 3-7: Trajectories of seven spheres based-model [111]

The Figure 3-7 above shows a moving object, heading in a forward direction and its associated seven spheres-based model draws the seven sphere shapes each time for each frame around the individual and their related parts.

3.2.7.1. Benefits of usage of seven spheres-based model

The seven sphere-based model gives a more accurate size of the moving object compared to the single sphere model; this model also helps in estimating the width of the moving object. The model not only shows the information of a moving object, but also tracks related moving parts.

3.2.7.2. Theory about usage of seven spheres-based model

In a seven sphere-based model, a sphere is drawn for each part of the body. To do this, first, the centre point, which can be an elbow in the case of an arm and is identified once. The next task is to locate the furthest point from it, which can be the shoulder and/or the fingers and this will make up the radius of the sphere. This sphere is then representative of an arm. The same applies to the other arm as well as all other limbs. At time instance 't_1', the centre point can be expressed as:

$$S_{t1} = (x_{t1}, y_{t1}) \tag{3.13}$$

And the radius as,

$$R_{t1} = l_{t1} \tag{3.14}$$

Similarly, for the time instance 't_n', the above two equations can be written as below:

$$S_{tn} = (x_{tn}, y_{tn}) \tag{3.15}$$

$$R_{tn} = l_{tn} \tag{3.16}$$

The above equation is for the sphere representing one arm, similarly, there's an equation for the other arm and the rest of the other limbs.

3.2.7.3. Input and output of seven spheres-based model

The input of this seven sphere-based model is identical to that of the single sphere-based model with the exception of the output where the output in this model is a stream of seven sphere instead of one. To illustrate this, the graphical screenshot is shown in the Figure 3-7 with the title 'Trajectories of the seven sphere-based model' and the textual console values are shown in Figure 3-8.

```
Processing frame 1
Find the moving object(s)
0 object(s) detected
Processing frame 2
Find the moving object(s)
1 object(s) detected
Head sphere centre point => X = 153, Y = 172
Head sphere radius => R = 8
Left hand sphere centre point => X = 141, Y = 164
Left sphere radius => R = 8
Right sphere centre point => X = 168, Y = 166
Right sphere radius => R = 8
Torus sphere centre point => X = 156, Y = 162
```

Figure 3-8: Textual values of seven spheres-based model

As shown in Figure 3-8, each frame contains the centre of a sphere and the radius data and this information is only available if there is a moving object in the frame. The first frame does not have any moving object; hence no such data is available of the analysis.

3.2.8. Brief characterization of models

As mentioned previously, the dot-based model is simple and needs little information to be extracted from the video, but it does not hold the information about the size of the moving object. Similarly, the prismatic model has few details about the size of the moving object, but those details do not cover each part of the moving object. The prismatic model is complex and needs more data extraction from the input video, while the sphere-based model is simple and a step closer to capturing the proper dimensions of the moving object. Based on the above information, the choice was made to use the seven spheres-based model, which does not only show the moving object but also each of its moving parts.

3.3. Moving object depth calculation

One challenging factor in this research is to find the distance between a camera and the moving object which is also known as depth calculation. There are a few known methods to perform depth calculation, but such methods have their own limitations. For instance, some of them need an exact camera focal length and other values for calibration while others suggest using two cameras.

This research uses a simple and easy way to solve this challenge because all the models described in the above sections have co-ordinates values about the moving object and its moving parts. These values can be used to find the depth distance when considering the environment as shown in the Figure 3-9.

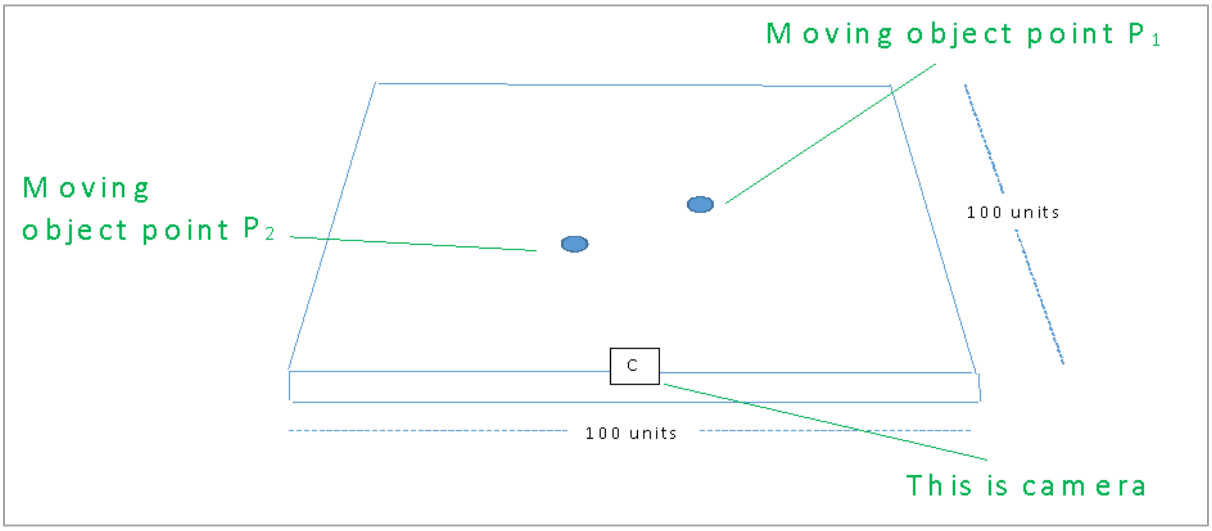


Figure 3-9: Moving object at point P1 and P2 at two different time instances

The Figure 3-9 shows that there is one moving object in the environment and initially, at time 't_1' this moving object is at point 'P_1', while later at time instance 't_2', the object moves to point 'P_2'.

Let's assume that the above is an explanation of the two video frames of the same video at two different time instances; the figure shows that the video frame has 100 pixels in x-axis direction and 100 pixels in y-axis direction with a camera at the bottom centre position of the frame.

Let's further assume that the point 'P_1' at time instance 't_1' can be shown as below:

$$P_1 = (65, 40) \tag{3.17}$$

Similarly, point 'P_2' at time instance 't_2' can be shown as below:

$$P_2 = (30, 60) \tag{3.18}$$

So, the distance between the camera and the point 'P_1' can be calculated as below:

$$D_1 = 100 - 40 \tag{3.19}$$

$$D_1 = 60 \; units \tag{3.20}$$

Similarly, the distance between the camera and the point ‘P_2’ can be calculated as below:

$$D_2 = 100 - 60 \tag{3.21}$$

$$D_2 = 40 \tag{3.22}$$

The above explanation is illustrated in the Figure 3-10 detailing the calculation of the distance between a moving object and the camera.

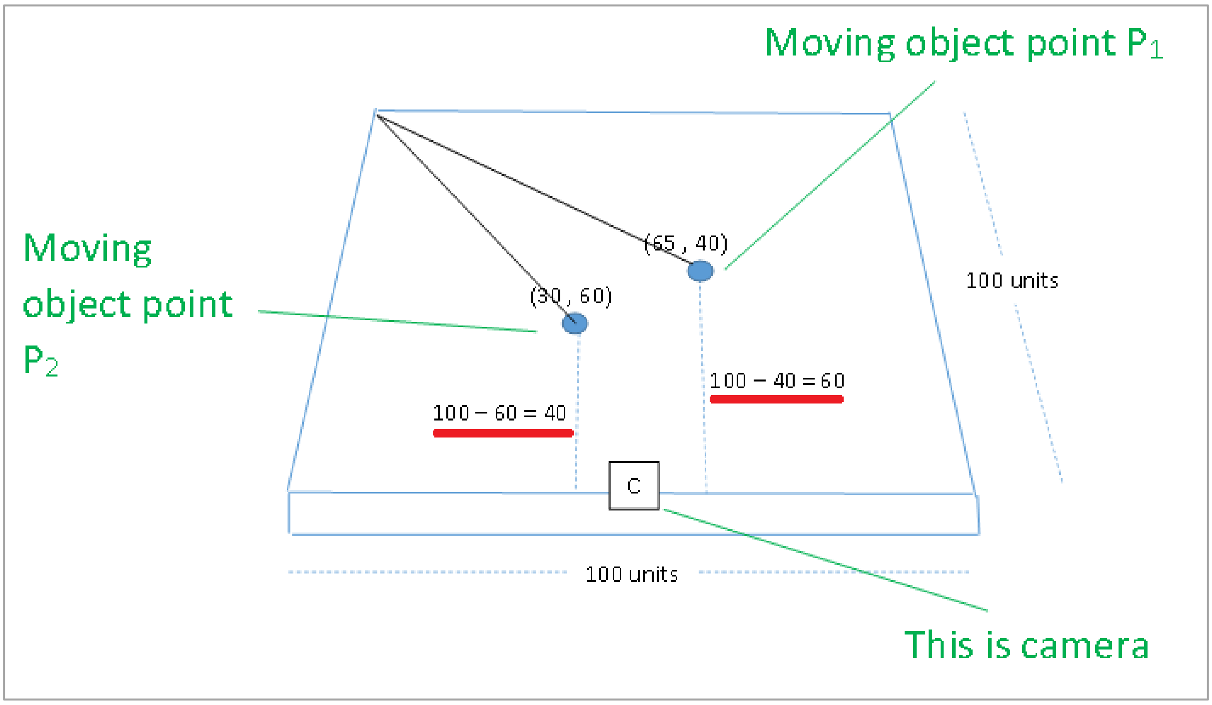


Figure 3-10: Moving object distance from camera

3.4. Origin manipulation

3.4.1. Normal Origin

In this research, the normal two-dimensional space origin is considered to be at the top left (or in three-dimensional space, at the top left bottom) corner as shown in Figure.3-11:

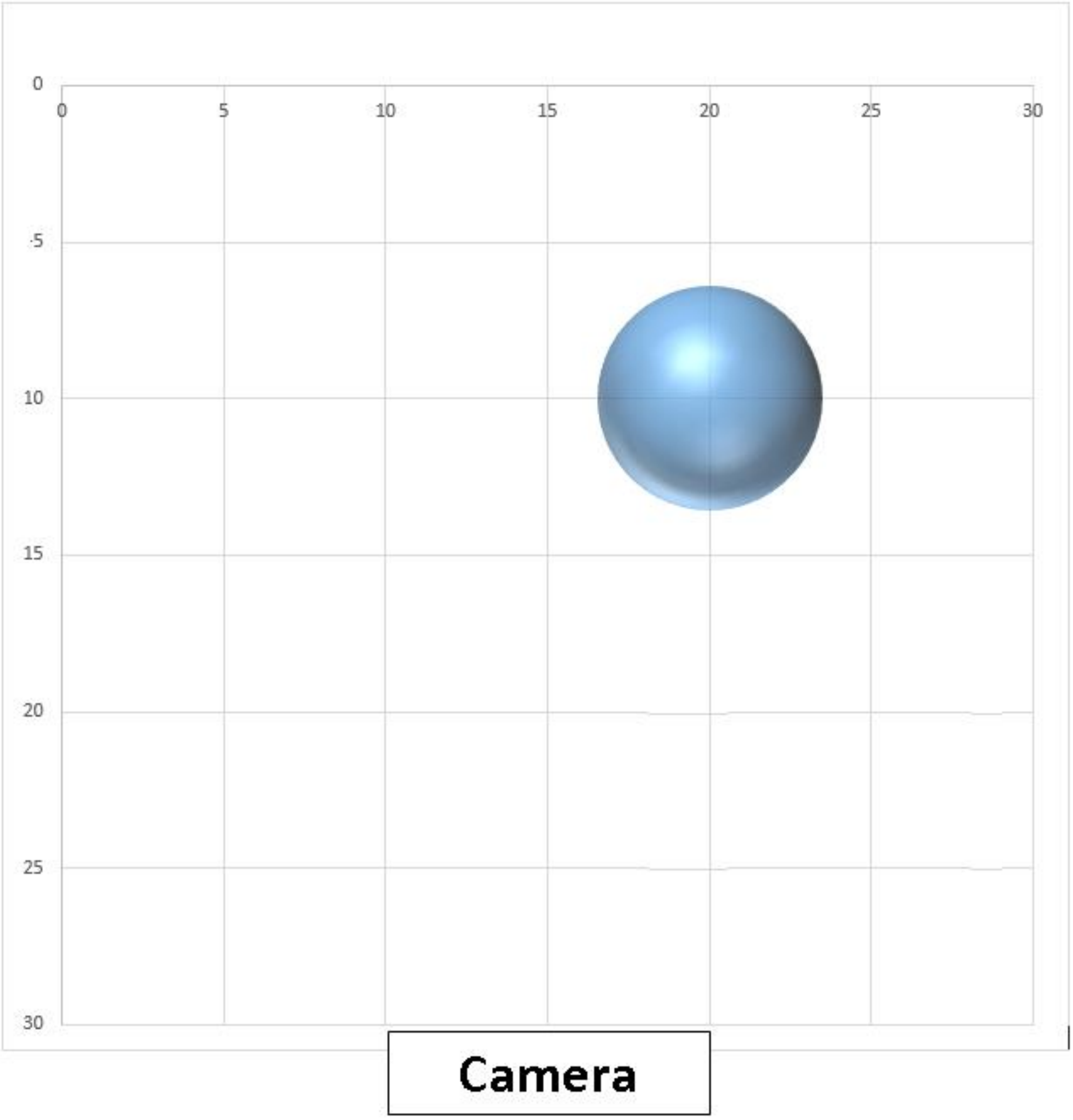


Figure.3-11: Normal space with origin at top left corner

All figures and descriptions in this document use 2D space representation so as to characterise the details clearly and easily. The camera is assumed to be at the bottom centre of the space and this place is standardised with the help of origin.

This calibration is not easy to understand since the origin of the scene and the camera are in two different places and it depends upon the space rather than the position of the camera.

From the Figure.3-11, the equation of the origin can be written as shown below, where origin is at 'x_{o0}' on x-axis and similarly at 'y_{o0}' on y-axis:

$$\text{O} = (x_{O0}, y_{O0}) \tag{3.23}$$

And the position of the camera is represented below which is at a different place when compared to the origin position in the scene:

$$C = \ (x_{C0}, \ \ y_{C0}) \tag{3.24}$$

While the position of the sphere is denoted as:

$$\text{S} = (x_{S0}, y_{SO}) \tag{3.25}$$

The vector from a camera to the sphere can be written as:

$$\overrightarrow{CS} = \vec{C} - \vec{S} \tag{3.26}$$

$$\overrightarrow{CS} = \ (x_{SO} - x_{c0}, y_{s0} - y_{c0}) \tag{3.27}$$

The length of the vector can be written as:

$$|\ \overrightarrow{CS}| = \ \sqrt{(x_{SO} - x_{c0})^2 + (y_{s0} - y_{c0})^2} \tag{3.28}$$

3.4.2. Origin at camera location

To make the calculation of the moving object's displacement easy, the origin needs to be moved to the camera position and the red rectangles are used in Figure 3-12 to denote the updated calibrations.

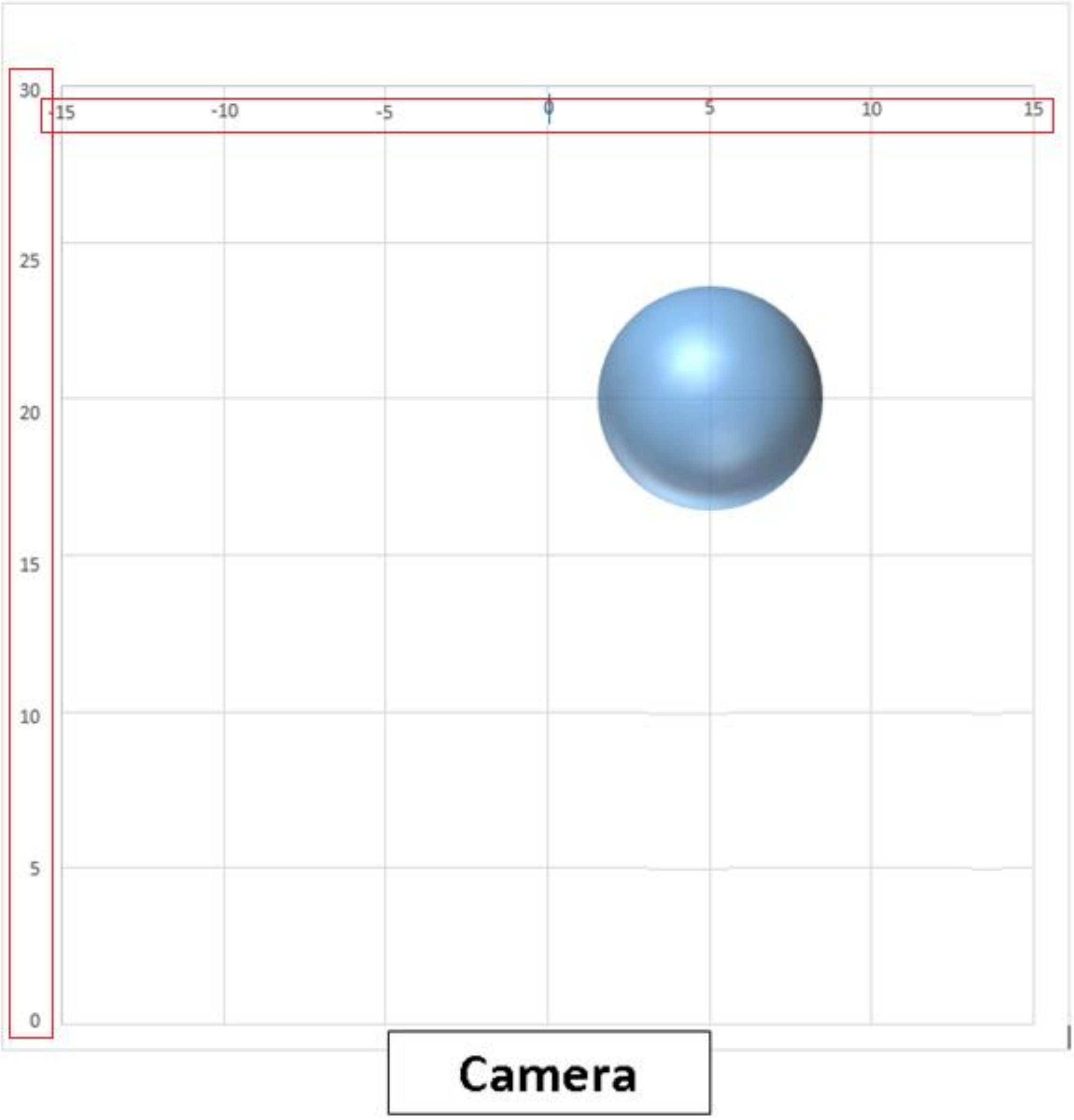


Figure 3-12: Modified space with the origin at camera location

Let's assume that new origin position can be written as below:

$$O' = (x_{o0} + a, y_{o0} + b) \tag{3.29}$$

Where,

'a' is the displacement value of origin along X-Axis,

While,

'b' is the displacement value of origin along Y-Axis.

Due to the above origin displacement, the camera position will be the same as origin:

$$C' = (x_{o0} + a, y_{o0} + b) \tag{3.30}$$

The position of a sphere with the help of new origin can be written as:

$$S' = (x_{s0} - a, y_{s0} - b) \tag{3.31}$$

By default, value 'a' and 'b' should be subtracted from the original value of the sphere and if, origin displacement involves a crossing of the sphere position then the final value should be multiplied by -1.

Now, the new vector between camera and sphere can be written as:

$$\overrightarrow{C'S'} = \overrightarrow{C'} - \overrightarrow{S'} \tag{3.32}$$

$$\overrightarrow{C'S'} = (\ (x_{s0} - a) - (x_{o0} + a),\ (y_{s0} - b) - (y_{o0} + b)\ ) \tag{3.33}$$

Then, the length of the vector can be given as:

$$\overrightarrow{C'S'} = \sqrt{((x_{s0} - a) - (x_{o0} + a))^2 + ((y_{s0} - b) - (y_{o0} + b)\ )^2} \tag{3.34}$$

3.4.3. Origin at camera location and the use of a circle to estimate distance.

The circle is used to calculate the distance between the sphere and the camera in the space. The circle radius will be the same for the sphere anywhere in the space, as illustrated in Figure 3-13.

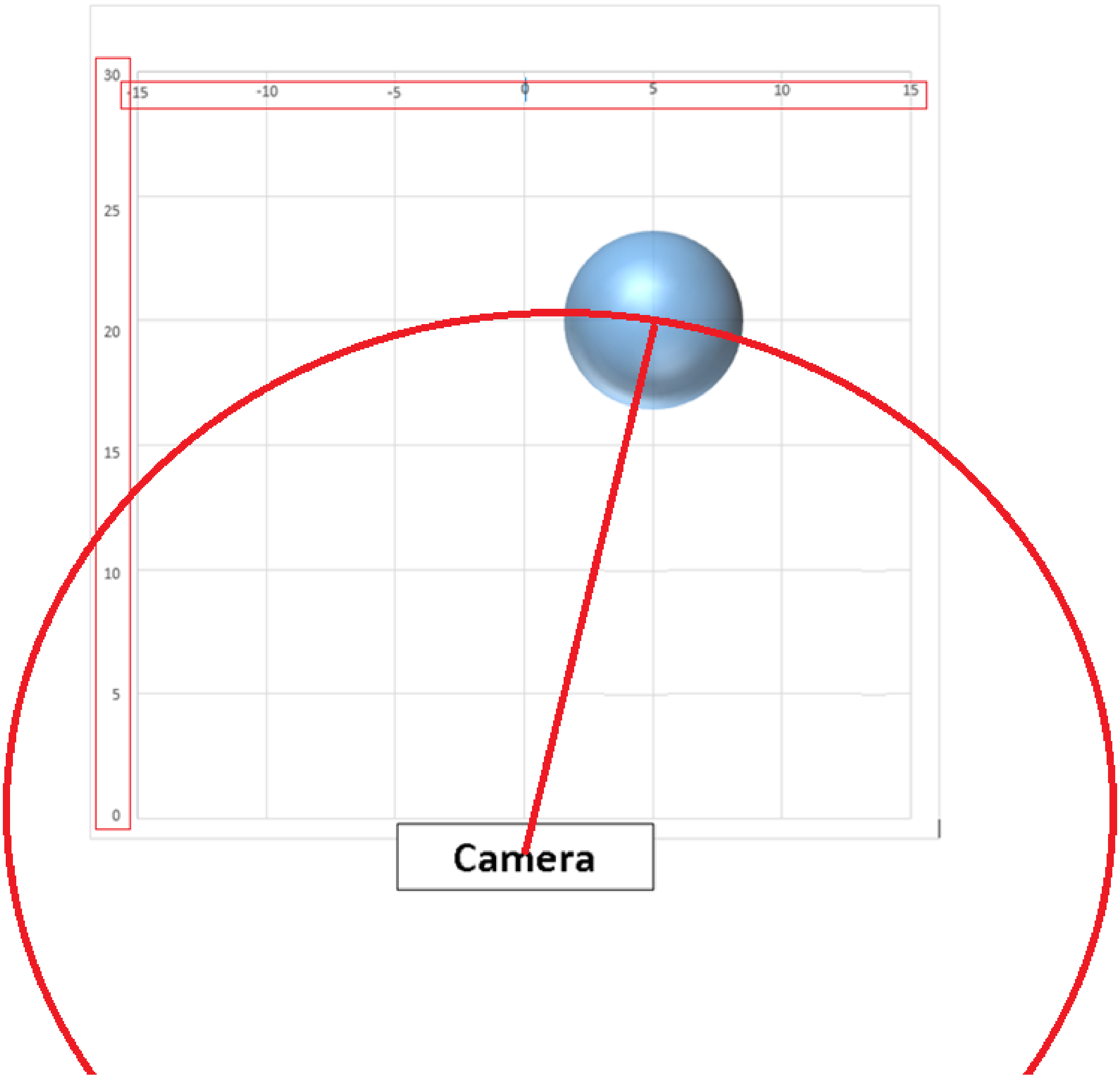


Figure 3-13: Modified space with the origin at camera location and use circle for distance calculation

As stated in the previous section there are two equations 2.5 and 2.6 which are sufficient in order to calculate the angles between vector and axis.

If $\overrightarrow{C'S'}$ is the vector between a camera and a sphere then the angle from X-axis can be written as below,

$$\cos\theta = \frac{\overrightarrow{C'S'}_x}{\overrightarrow{C'S'}} \tag{3.35}$$

and if the values are replaced then,

$$\cos\theta = \frac{(x_{s0} - a) - (x_{o0} + a)}{\sqrt{((x_{s0} - a) - (x_{o0} + a))^2 + ((y_{s0} - b) - (y_{o0} + b))^2}} \tag{3.36}$$

Similarly, for Y- axis,

$$\sin\theta = \frac{\overrightarrow{C'S'}_y}{\overrightarrow{C'S'}} \tag{3.37}$$

and if the values are replaced, then,

$$\sin\theta = \frac{(y_{s0} - b) - (y_{o0} + b)}{\sqrt{((x_{s0} - a) - (x_{o0} + a))^2 + ((y_{s0} - b) - (y_{o0} + b))^2}} \tag{3.38}$$

3.5. Sample space with objects

Space is considered to be as a frame of a video and there can be multiple instances of space with respect to time (i.e. the number of frames of a video at different time instances). Positions of objects within the space can be different at different time instances and few objects can have continuous motions and finally disappear from the space, while some other objects may move at a later/different time instance. There is also the possibility of one or more objects to be completely stationery or not moving at all in the space.

3.5.1. Space at initial time instance

To understand clearly what is stated above, 3 objects in the space are considered in Figure 3-14.

Consider a grey sphere, which can be shown as:

$$G = (x_{g0} - a, y_{g0} - b) \tag{3.39}$$

Similarly, a blue sphere as:

$$B = (x_{b0} - a, y_{bo} - b) \tag{3.40}$$

And finally, a purple sphere as:

$$P = (x_{p0} - a, y_{p0} - b) \tag{3.41}$$

The above descriptions of the spheres' position within the space are presented in the Figure 3-14.

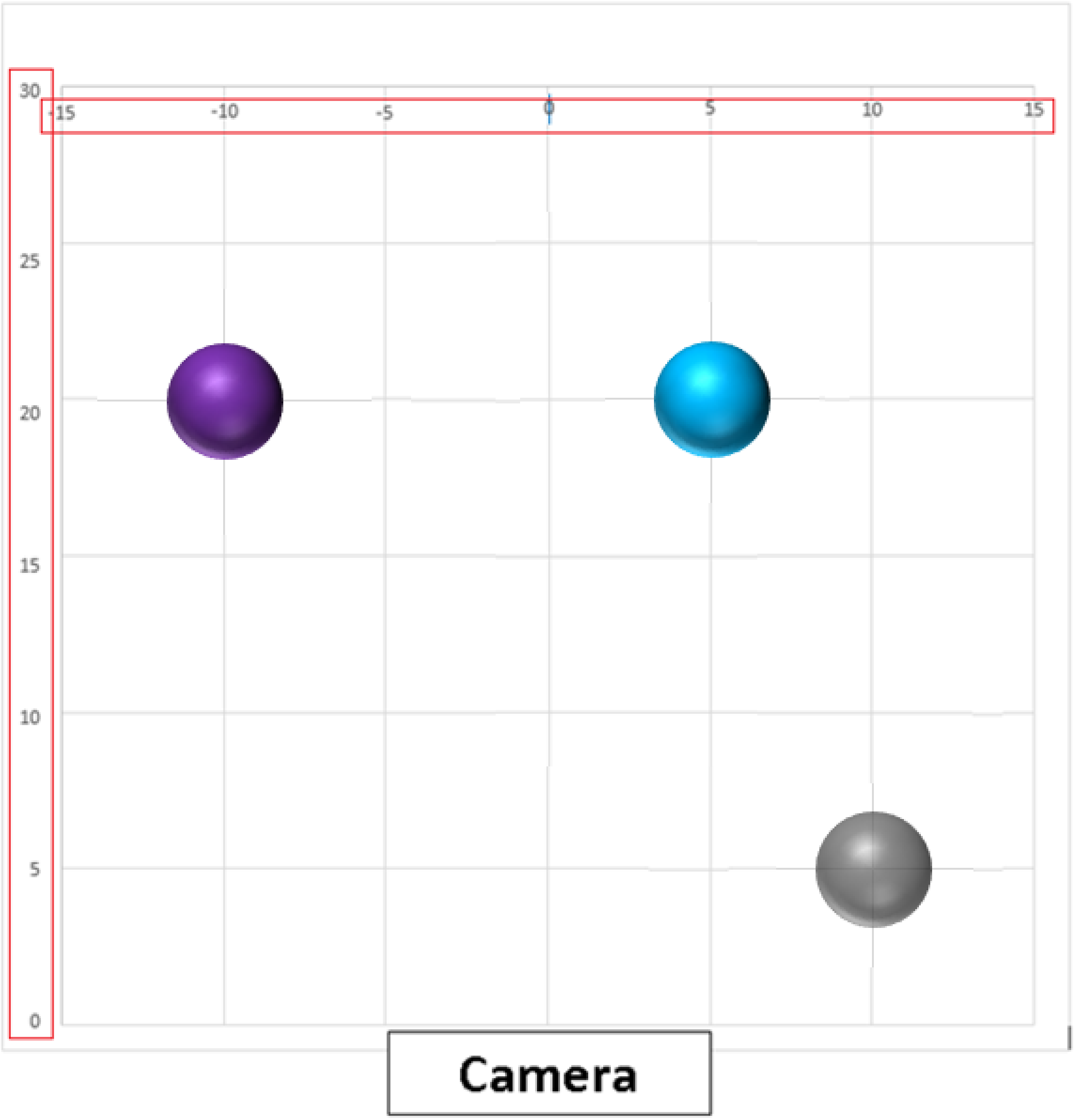


Figure 3-14: Sample space with three different objects

The Figure 3-15 is illustrated from a real-life video example (shown below), which details the 3 spheres depicted above being mapped based on the 3 objects detected below.

Figure 3-15: Sample space example in a video [112]

3.5.2. Space at final time instance

The Grey sphere (on the right-hand side of figure 3-14) was in a continuous motion and eventually disappears from the space entirely. The final position of the sphere can be specified as,

$$G_f = (\infty, \infty) \tag{3.42}$$

While, in the mid-time instances, the position of the grey sphere is specified as,

$$G_m = (x_{gm} - a, y_{gm} - b) \tag{3.43}$$

The Blue sphere starts motion after several time instances and this sphere is still in the space and can be specified as,

$$B_f = (x_{bf} - a, y_{bf} - b) \tag{3.44}$$

The blue sphere's initial position in time (before motion) is specified as,

$$B_m = (x_{b0} - a, y_{b0} - b) \tag{3.45}$$

The Purple sphere has similar behaviour as the blue sphere, which is specified in the following two different equations:

$$P_f = (x_{pf} - a, y_{pf} - b) \tag{3.46}$$

$$P_m = (x_{p0} - a, y_{p0} - b) \tag{3.47}$$

The Figure 3-16 illustrates the two spheres being in the space with the exception of the grey sphere which has left the space.

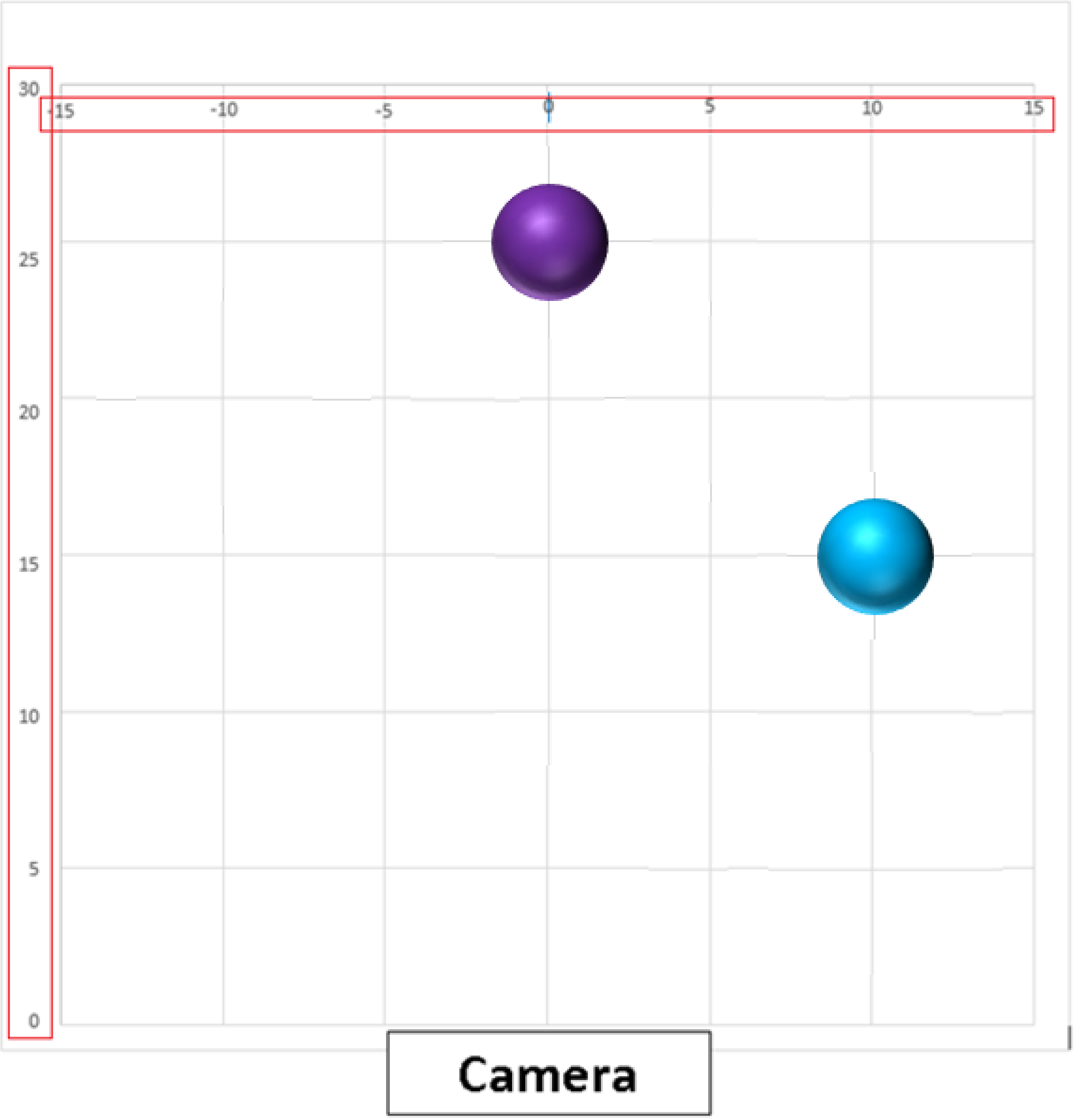


Figure 3-16: Space sample at final time instance

The object-based representation in the Figure 3-16, corresponds to the video example shown below which illustrates the 3rd (grey sphere denoted) object having left the scene at a final time instance.

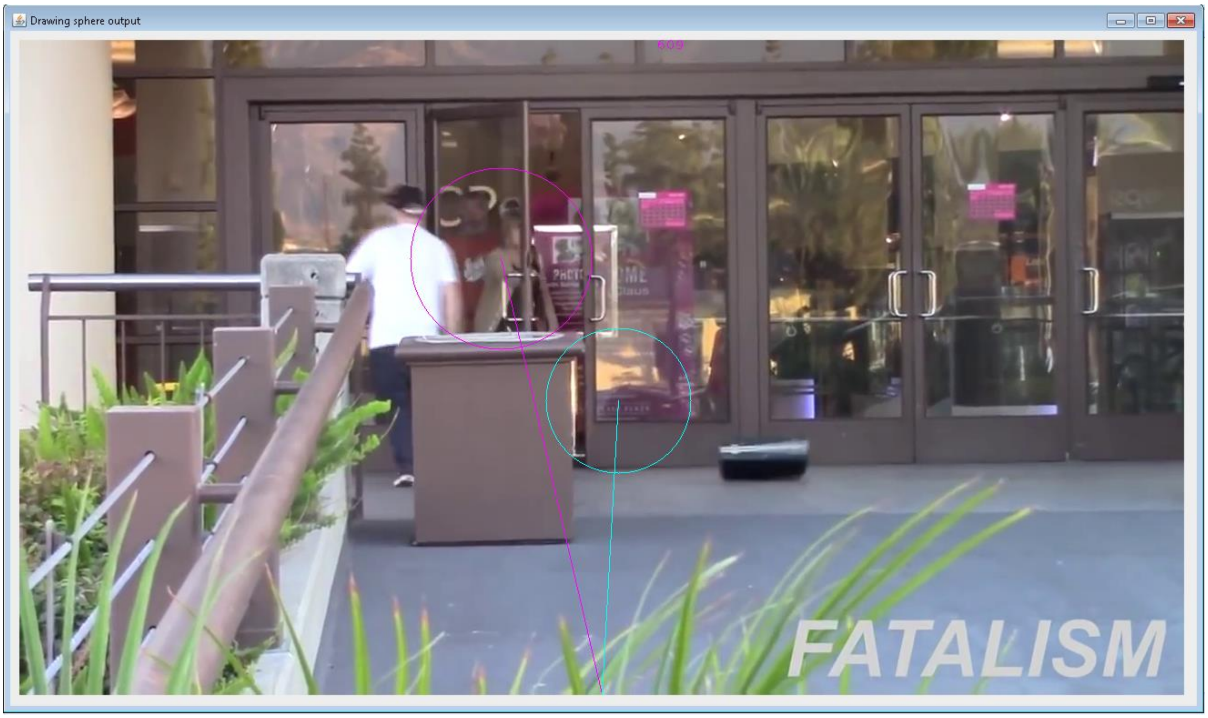


Figure 3-17: Video space sample at final time instance [112]

The Figure 3-18, is the middle-time instance of the video showing the grey denoted object still in the scene (during continuous motion).

Figure 3-18: Video space sample at middle time instance [112]

3.6. Movements in space

All the movements of objects in the space must be tracked in order to provide effective trajectories which consist of several types of motions. A few important types of motions are discussed below.

3.6.1. Continuous linear motion moving object

The author considers the linear motion when sphere is moving in a straight direction without doing any rotation action and generates the simple trajectory.

Continuous linear motion is the movement of an object in a straight direction, free from any rotation. This results in a simple trajectory, which is discussed further below.

The grey-sphere denoted object’s position discussed previously, moved in a direct, linear motion and its continuous motion vector is shown below.

$$G_C = (x_{gc} - a, y_{gc} - b) \tag{3.48}$$

The equations below specify the continuous motion vector at different time instances, 1 and 2 respectively.

$$G_{c1} = (x_{gc1} - a, y_{gc1} - b) \tag{3.49}$$

And

$$G_{c2} = (x_{gc2} - a, y_{gc2} - b) \tag{3.50}$$

So, the angle from X-axis for the first vector can be calculated as,

$$\cos\theta_{c1} = \frac{\overrightarrow{G_{xc1}}}{\overrightarrow{G_{c1}}} \tag{3.51}$$

And as shown below when replaced with the actual values.

$$\cos\theta_{c1} = \frac{x_{gc1} - a}{\sqrt{(x_{gc1} - a)^2 + (y_{gc1} - b)^2}} \tag{3.52}$$

Similarly, the second vector's angle from X-axis, can be calculated as,

$$\cos\theta_{c2} = \frac{\overrightarrow{G_{xc2}}}{\overrightarrow{G_{c2}}} \tag{3.53}$$

This become as shown below when replaced with the actual values.

$$\cos\theta_{c2} = \frac{x_{gc2} - a}{\sqrt{(x_{gc2} - a)^2 + (y_{gc2} - b)^2}} \tag{3.54}$$

As the motion is linear, the angle between x-axis and vectors is the same at any and all time instances.

$$\cos\theta_{c1} = \cos\theta_{c2} \tag{3.55}$$

Similarly, the first vector's angle from Y-axis, can be calculated as,

$$\sin \emptyset_{c1} = \frac{\overrightarrow{G_{yc1}}}{\overrightarrow{G_{c1}}} \tag{3.56}$$

And as shown below when it is replaced with the actual values.

$$\sin \emptyset_{c1} = \frac{y_{gc1} - a}{\sqrt{\left(x_{gc1} - a\right)^2 + \left(y_{gc1} - b\right)^2}} \tag{3.57}$$

Then, the second vector's angle from Y-axis, can be calculated as below:

$$\sin \emptyset_{c2} = \frac{\overrightarrow{G_{yc2}}}{\overrightarrow{G_{c2}}} \tag{3.58}$$

And as shown below when it is replaced with the actual values.

$$\sin \emptyset_{c2} = \frac{y_{gc2} - a}{\sqrt{\left(x_{gc2} - a\right)^2 + \left(y_{gc2} - b\right)^2}} \tag{3.59}$$

As the motion is linear, the angle between y-axis and the vectors is the same at any and all-time instances.

$$\sin \emptyset_{c1} = \sin \emptyset_{c2} \tag{3.60}$$

The same equivalence applies when considering the Z-axis's angle with the vector, which is shown in the Figure 3-19 by the person walking from frame number 160 to frame number 202.

Figure 3-19: Continuous linear motion [112]

Red rectangle in the Figure 3-19 is drawn manually to show the object is in a phone conversation.

3.6.2. Static object start motion

The motion is shown by the blue sphere in the above space and at initial time instance the sphere was not moving but after few time instances, the sphere starts doing so. Initially, there is no trajectory for this type of motion, however, it will start appearing after few time instances. In this case, the blue sphere can have any type of motion (linear or rotational). The general continuous tracking of the sphere will be shown as below:

$$B_C = (x_{bc} - a, y_{bc} - b) \tag{3.61}$$

As this sphere was not moving initially and after few time instances, it starts its motion, so vectors at two different time instances can be given as below:

$$B_{c1} = (0, 0) \tag{3.62}$$

And

$$B_{c2} = (x_{bc2} - a, y_{bc2} - b) \tag{3.63}$$

In the sample video, the object shown in Figure 3-20 starts moving after frame number 77.

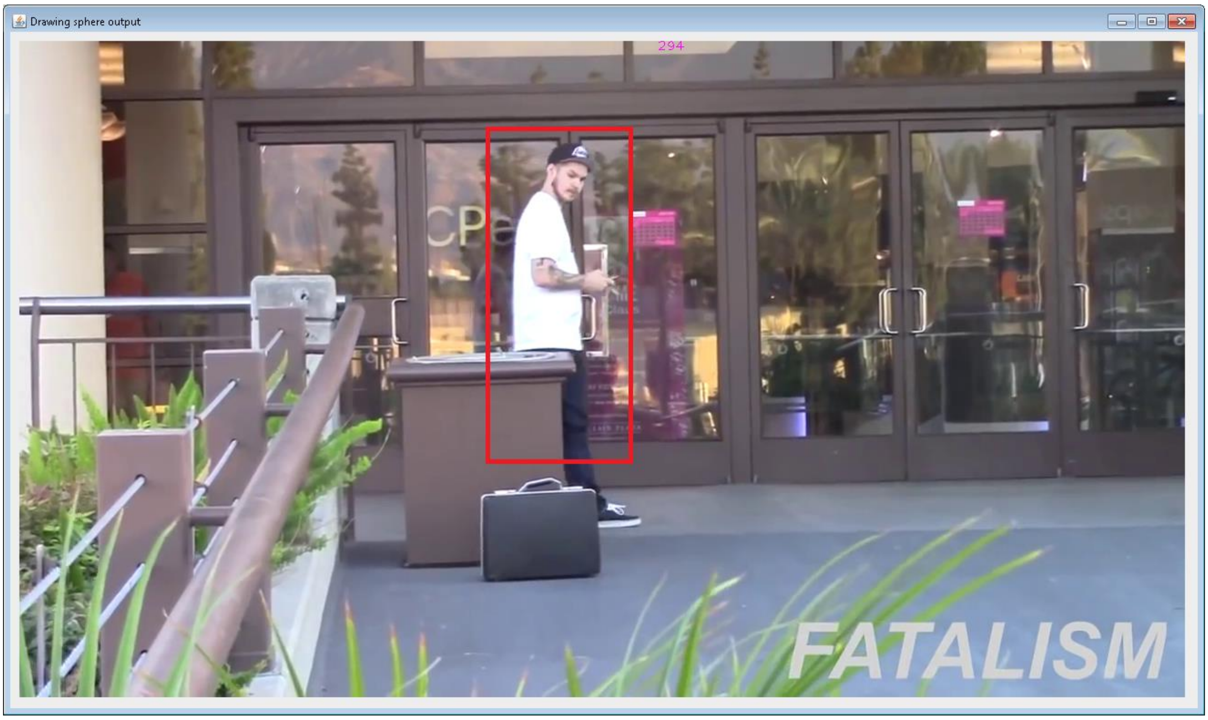


Figure 3-20: Object start moving after few time instances [112]

3.6.3. Static object picked up by another moving object

In the sample space shown in the Figure 3-14, the purple sphere is a static object and this sphere starts moving with the help of another object. The motion of a purple sphere can be shown as below:

$$P_C = (x_{pc} - a, y_{pc} - b) \tag{3.64}$$

As this sphere was not moving initially then it starts its motion, so vectors at two different time instances can be given as below:

$$P_{c1} = (0, 0) \tag{3.65}$$

And

$$P_{c2} = (x_{pc2} - a, y_{pc2} - b) \tag{3.66}$$

In the sample video, this static object starts moving with the help of another object from frame number 478. This is shown in the Figure 3-21.

Figure 3-21: Static object start moving with the help of other object [112]

3.6.4. Moving object disappearing from the scene

In the sample space, the grey sphere disappears from the space; this is shown in the Figure 3-16 with the title of final time instance and that changes the moving object trajectory. The continuous generation of vector for this type of object can be expressed as below,

$$G_C = (x_{gc} - a, y_{gc} - b) \tag{3.67}$$

Let’s assume that the following two vectors show the position at two different time instances as stated below,

$$G_{c1} = (x_{gc1} - a, y_{gc1} - b) \tag{3.68}$$

And

$$G_{c2} = (x_{gc2} - a, y_{gc2} - b) \tag{3.69}$$

But for this case of disappeared sphere, there should be another final explanation to show this disappearance as expressed below:

$$G_{c3} = (\infty, \infty) \tag{3.70}$$

In the sample video, the grey sphere (the person) disappears from the video and this happens from the frame number close to 252.

Figure 3-22: Sphere disappear from the space

3.6.5. Moving object rotation motion

In the example explained above, the blue sphere is having the rotation motion. Tracking the rotation with the help of a single sphere will not be easy and there should be some reference points on the sphere in order to explain the rotation. At a later stage of this research this single sphere is replaced with multiple spheres and each sphere shows the different part of body and makes the rotation motion tracking easy and possible.

For the explanation at this instance, the tracking of these reference points in a single sphere can be expressed as below,

$$B_{cn} = (x_{bcn} - a, y_{bcn} - b) \tag{3.71}$$

Where,

'n' shows the number of reference point in the sphere.

The post calculation on vectors of different reference points helps in showing the rotation motion of a sphere.

At the example of the video, the blue sphere has a rotation motion in frame number near 404 and this is shown in the picture below.

Figure 3-23: Rotation action of sphere

The red rectangle in the Figure 3-23 is used to show that the object is active. This red rectangle is drawn manually.

3.7. Conclusion

The extraction of information for re-construction of moving object trajectory directly from an input video stream is a difficult task due to the complex nature of visual data that needs to be processed. Format, resolution, frame rate of the input stream, physical capabilities and limitations of hardware, or the ever-changing environmental conditions such as illumination variations or dynamic cluttering which all have an impact on the visual output that must be tracked in real-time.

The main goal of this research is to construct an efficient framework for real-time re-construction of moving object trajectories and tracking the moving parts of moving object. This is based on dynamic movements of individuals or groups of individuals moving in enclosed public spaces at walking speeds using information extracted from live input video stream.

The key to the success of this task is the implementation of efficient algorithms for recognition of the moving human objects and their moving parts using data from input stream about the objects in the scene, their location within the space of the scene with estimation of the dimensions of their constituent parts and information about the direction of possible movement.

The approach, which is adopted in this research, is based on replacing the necessity for moving object of the entire video stream with seven sphere based model and approximate information generated by the algorithm. This approach was originally presented in articles [113] [114] and is elaborated in two subsequent articles prepared for publication (in 2016 and 2018).

Different moving objects models are discussed in this section. The author explored the benefits and limitations of different proposed models and uses these models in the framework of reconstruction of moving object trajectories. The manipulation of the centre of origin of the environment is also discussed in this section.

Since the algorithm is processing data in matrix data format programming techniques, it is anticipated that the data will be output in a standard vector notation to describe the location, velocity, dimensions and orientation of a moving object and parts identified in the input video stream. In order to represent more complex motions, the input data can be extended to include twisting, bending, rotation, and so on. Such data can be also represented in XML format. The theoretical foundations for such an extension is discussed in detail in the next chapter.

4. Classifiers for moving object trajectories reconstruction

In this research, the algorithm's task is to detect a moving human object and their body parts (hands, torso, legs etc.) through analysing data from video input streams with the help of trained data sets of classifiers. This is discussed in detail later in this chapter.

Unlike existing research methods and approaches, which make use of RGB or infrared sensors to detect a human moving as a whole or just some body parts or a combination of both, this research explores the benefits of using classifiers for human object trajectory reconstruction. These classifiers are discussed and outlined in detail in the sections below.

4.1. Motivation

Motion tracking is a very active research area in various fields such as surveillance applications, gaming industry, sports etc.

Full body tracking is also used in the rehabilitation of patients and other related activities. A research titled 'Full Body Interaction for Serious Games in Motor Rehabilitation' conducted by C. Schönauer, T. Pintaric and H. Kaufmann [115], captures the motion with the help of passive markers, based on infrared option-flow motion tracking system. This research aids in the motion tracking of patients suffering from pain in the lower back and neck.

Another related study of N. Ukita et al [113] focuses on the usefulness of human body part tracking. The work focuses on particle, filtering-based tracking using prior models, which have several advantages for social interactions research. This research also includes work on body parts' trajectory smoothing in videos by using existing models.

4.2. Classifiers essentials

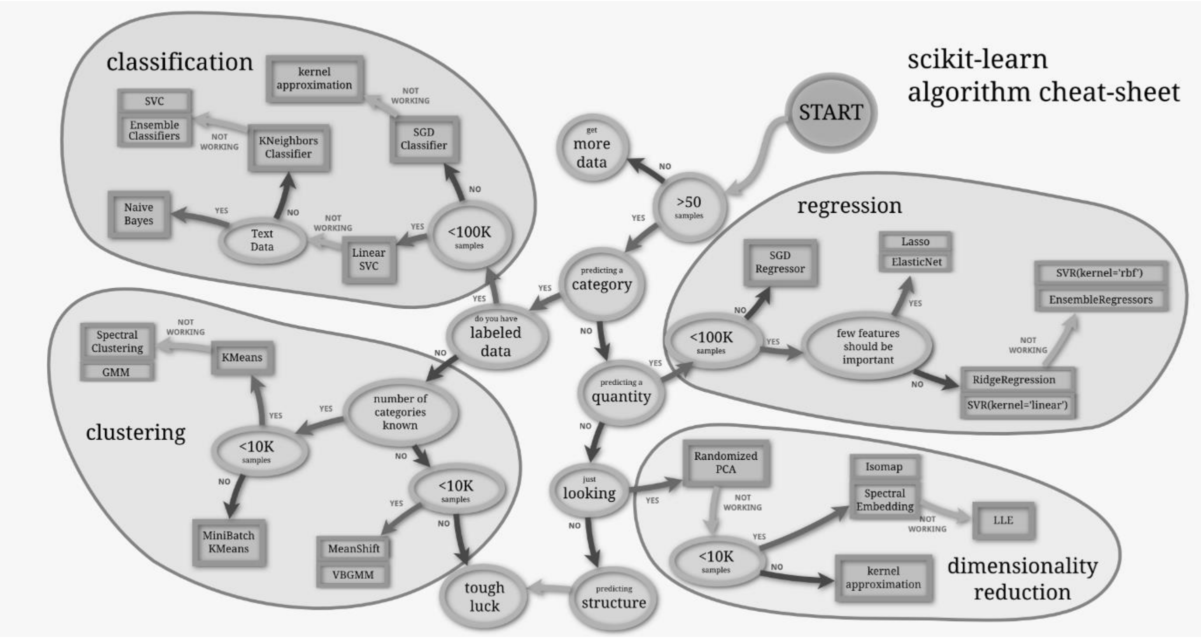


Figure 4-1: Selection of classifiers [116]

4.3. Classifier types & data processing

The classifiers used in this research are based on colour, shape and depth of the moving object in a video stream. To gather the necessary information on different features, the features are blended with trained data for concluding the classification map. In this framework approach, moving object tracking is based on the object-centric representation of the position which forms a tube-like model of the spatial navigation and allows isolated manipulation of the video objects within the focus. This is achieved through an incremental algorithm for processing and classification of the information flow, as illustrated in the Figure 4-2.

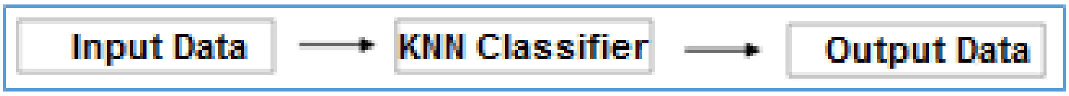


Figure 4-2: Classifiers execution and extracting information steps

The detailed description of different feature generations is given below.

4.3.1. Shaped-based features extraction

Shaped based features are extracted with the help of differential morphological profiles (DMPs).

4.3.1.1. Morphological profile

Morphological profiling is a method widely used and reviewed in the medical industry for the detection, characterization, and identification of particles [117][118]. This method depends upon the characteristics of target item and is used in this research to identify the human body part shapes. The morphological profiling technique needs a proper human body part isolation as well as the visibility of the body part, under standardized conditions, in order to be valid and suitable for correct identifications.

4.3.1.2. Principal component analysis

The most important characteristic of the principal component analysis is its ability to measure the similarity of two or more sets of data. This section will discuss how this will meet the requirements of this research.

4.3.1.2.1 Profile Segment

Consider the relationship between the eigenvectors and eigenvalues of the data sets, and for simplicity purposes, we consider two data sets. Each of the two data sets may have n number of dimensions, this then, will help us to identify similar characteristics in multi-dimensional data sets. The Figure 4-3, shows hyperbole values of the first data set, denoted as S1 and will serve as a template.

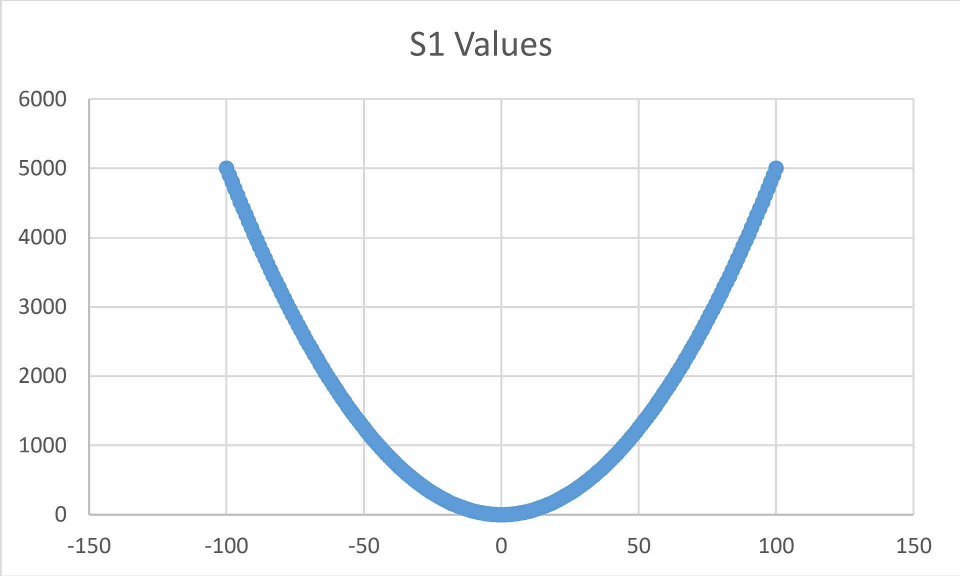


Figure 4-3: SI Data set values

The Figure 4-4 shows the hyperbole of the second data set (S2) and has unknown characteristics which the template above will identify.

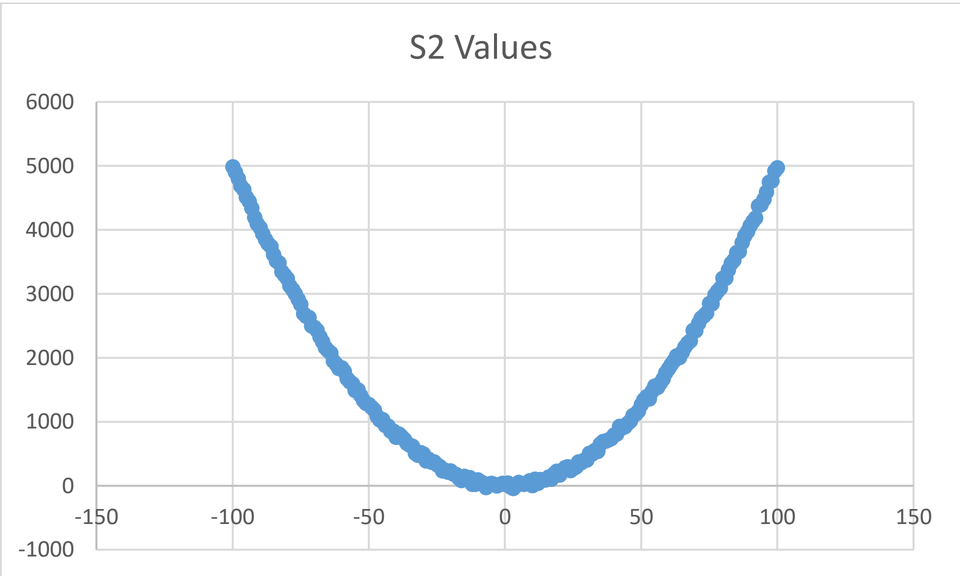


Figure 4-4: S2 Data set values

S2 data set values are calculated by adding small noise in the S1 data set values. The difference of values in S1 and S2 data sets is shown in the Figure 4-5.

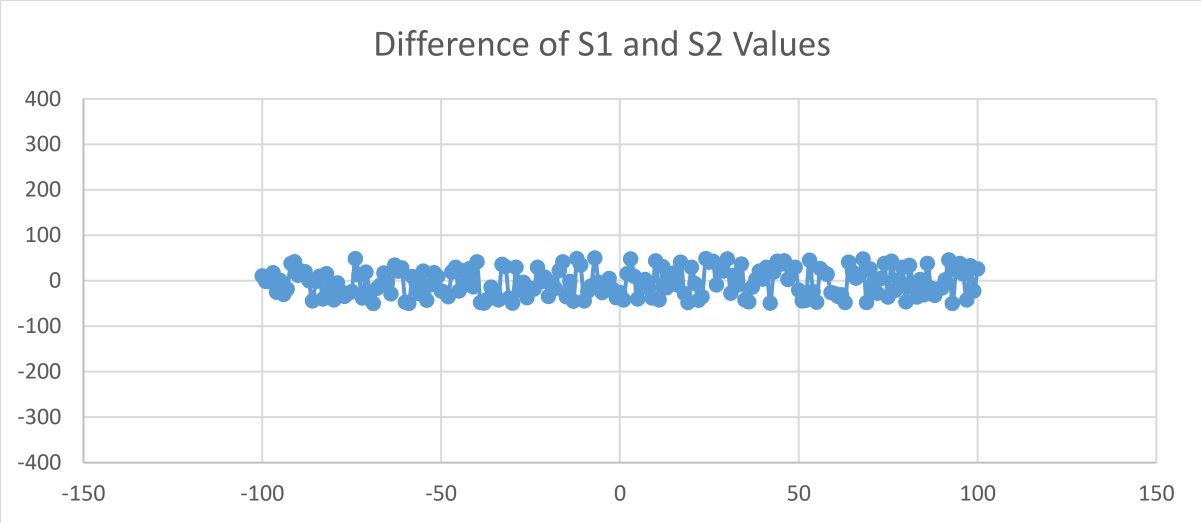


Figure 4-5: Difference of values in S1 and S2 data sets

S1 data set values are profile values of a shape in an image. The main task of the principal component analysis in this research is to find the vectors which best account for the distribution of the required features within the entire image space and these vectors define the subspace of image features. Vectors of length N^2; represent an image with a matrix of NxN image and is a linear combination of the original images. These vectors are also the eigenvectors of the covariance matrix, corresponding to the original images and are referred to as the eigenvalue. Each feature of an image in the original image space can be represented entirely in terms of a linear combination of the eigenvalue. The steps for feature identification can be summarised as follow.

- Determine the image space by acquiring an initial set of images and their respective features which form the training set.
- Apply the principal component analysis to the training set, to find the eigenvalue and keep only M number of images, which have the highest eigenvalues. These M images define the feature space.
- When a new image is presented to the system, calculate a set of weights based on the input image and the M eigenvalue by projecting the input image onto each of the eigenvalues.
- Determine if the input image belongs to the image space by checking the values of the projected weights from last step.

- If the input image belongs to the image space, it will be further classified by comparing it with other images and their features in order to determine whether it's a human body.

Results produced by this classification method are accurate and the eigenvalue provides a better solution for the feature extraction process. The principal component analysis depends on the eigen-decomposition of positive and semidefinite matrices formed mathematically by the input data set. The following are the steps needed to calculate the principal component analysis required value.

- **Data Set**: Consider a matrix 'D' of dimensions m x n, where 'm' is the number of observations and 'n' is the number of dimensions
- **Mean Subtraction:** This step is performed for each dimension and the resultant matrix has zero mean value. This is explained below in more details.
    - Calculate the mean value for each dimension of matrix 'D' and save the results in matrix 'R' as shown below,

$$R(n) = \frac{1}{M} \sum_{m=1}^{M} D(m\,X\,n) \tag{4.1}$$

    - Subtract the mean matrix 'R' for each column of data matrix 'D' and save the result in matrix Q as shown below with the dimension m X n,

$$Q = D - IR \tag{4.2}$$

Where, 'I' is identity matrix

- **Covariance Matrix**: The covariance matrix for above matrix 'Q' can be written as below for 'n' dimensions,

$$C = \begin{bmatrix} cov(Q_1, Q_1) & \cdots & cov(Q_1, Q_n) \\ \vdots & \ddots & \vdots \\ cov(Q_n, Q_1) & \cdots & cov(Q_n, Q_n) \end{bmatrix} \tag{4.3}$$

The above matrix 'C', has all possible covariance values between all dimensions and each value in the matrix is the covariance between any two dimensions, which can be calculated as shown below if there's a matrix with only two dimensions A and B,

$$con(A, B) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)} \quad (4.4)$$

Where, 'X' is mean value of data X and 'Y' is the mean value of data Y. Covariance is always calculated only between two dimensions and is used to calculate the relation between dimensions (of features).

- **Eigen Vectors and Eigen Values:** Eigen vectors and Eigen values are calculated for the covariance matrix with,
  - 'n' Eigen vectors for matrix 'C' along their Eigen values.
  - Eigen vectors normalized to unit length.
- **Arranging Eigen Vectors and Eigen Values:** Arrange the Eigen values in a decreasing order with the Eigen vector and keep the Eigen value and Eigen vector pairing,

$$O = (eigen_1, eigen_2, eigen_3, \dots, eigen_n,) \quad (4.5)$$

Where,
'$eigen_n$' is the i[th] eigen value. The highest value is at the start of the sequence while the lowest value is at the end of the sequence. Principal component of the data set is the Eigen vector with the highest Eigen value.

- **Data Set Projection:** Let's project the data set matrix D as shown below,

$$P = O^T Q^T \quad (4.6)$$

Where, O is the Eigen vector from previous step and Q is the mean subtracted data from the step before the 2 steps above. P is the data set with the observation in the columns and dimensions in the rows of the matrix. This data is now in the vector axes instead of the normal ‘x’ and ‘y’ axes and this new vector axes transform the data into Eigen vectors.

The steps above are used to calculate the values for both data sets, S1 and S2 and these are discussed below.

- Determine the mean subtracted data for each data set:

$$\overline{S1} = S1 - mean(S1) \tag{4.7}$$

$$\overline{S2} = S2 - mean(S2) \tag{4.8}$$

- Covariance matrix is calculated for each of the data set.
- Eigen value and Eigen Vector for each data set is determined as below using covariance matrix:

$$O = \begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} ; Eigen\ Vector \tag{4.9}$$

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} ; Eigen\ Value \tag{4.10}$$

Diagonal elements of matrix A are Eigen values which are A_{11} and A_{22}. A_{11} is the minimum value and $A_{22}$is the maximum Eigen value. The Eigen value shows the relationship between the two data sets. However, their value depends upon a number of factors for instance the number of data points involved. These Eigen values are arranged in decreasing value order so that the highest value appear first, in this way $O_{21}$and O_{22} is the first Eigen vector associated with maximum A_{22} Eigen value while $O_{11}$and O_{12} is the second Eigen vector associated with maximum A_{22} Eigen value. The angle between first Eigen vector and x-axis is measured as shown below,

$$\theta_1 = \tan^{-1}\frac{O_{22}}{O_{12}} \tag{4.11}$$

And for the second Eigen vector is given as below:

$$\theta_2 = \tan^{-1}\frac{O_{21}}{O_{11}} \tag{4.12}$$

When applying principal component analysis between two completely different data sets, the results as shown below,

$$\theta \neq 45° \tag{4.13}$$

And the Eigen values as below,

$$\frac{A_{11}}{A_{22}} \neq 0 \tag{4.14}$$

As both data set are not similar which is visible from the last two equations and in this case, different template data set is needed until the result of $\theta = 45°$ is obtained.

4.3.1.2.2 Line Segment

Line segment allows the shape feature to be determined in the image as it cannot be calculated with the help of profile segmentation since the line segment is two dimensional. For the application of the principal component analysis on the image, the image is converted into a column vector. A column vector is constructed by concatenating each row with the previous in sequence. If there is an image matrix with m x n dimensions (as shown in the equation below) then the column vector matrix would be of mn x 1 dimensions and it can be described as below,

$$image = \begin{bmatrix} e_{11} & e12 & \dots & e_{1n} \\ e_{21} & e_{22} & & e_{2n} \\ \dots & & & .. \\ e_{m1} & & & e_{mn} \end{bmatrix}; \ m\ x\ n\ dimension\ matrix \tag{4.15}$$

$$image\ vector = \begin{bmatrix} e_{11} \\ e_{12} \\ \cdots \\ e_{1n} \\ e_{12} \\ e_{22} \\ \cdots \\ e_{2n} \\ e_{m1} \\ \cdots \\ e_{mn} \end{bmatrix};\ \ mn\ x\ 1\ dimension\ matrix \tag{4.16}$$

The previous section showed that if $\theta = 45°$ then both data sets are identical. Keeping this into consideration, there are two other possible cases, C1 (when $\theta > 45°$) and C2 (when $\theta < 45°$). To reach a point to select the correct data set template, there are multiple logical flows possible in algorithm, which are discussed in the following section. For example, consider that there is a data set for image E1 as a template dataset and E2 image with unknown characteristics. The shape of any object is defined by its width and height and these characteristics are extracted from images. W1 and L1 are the width and height of image E1 and W2 and L2 for image E2.

C1: Case $\boldsymbol{\theta}$ > 45°

There are four possible situations in this case which are written below,

a) C1-1: (W1 = W2) and (L1 < L2)
b) C1-2: (W1 < W2) and (L1 = L2)
c) C1-3: (W1 < W2) and (L1 < L2)
d) C1-4: (W1 < W2) and (L1 > L2)

The logical flow for the case above (and situation) is shown in Figure 4-6.

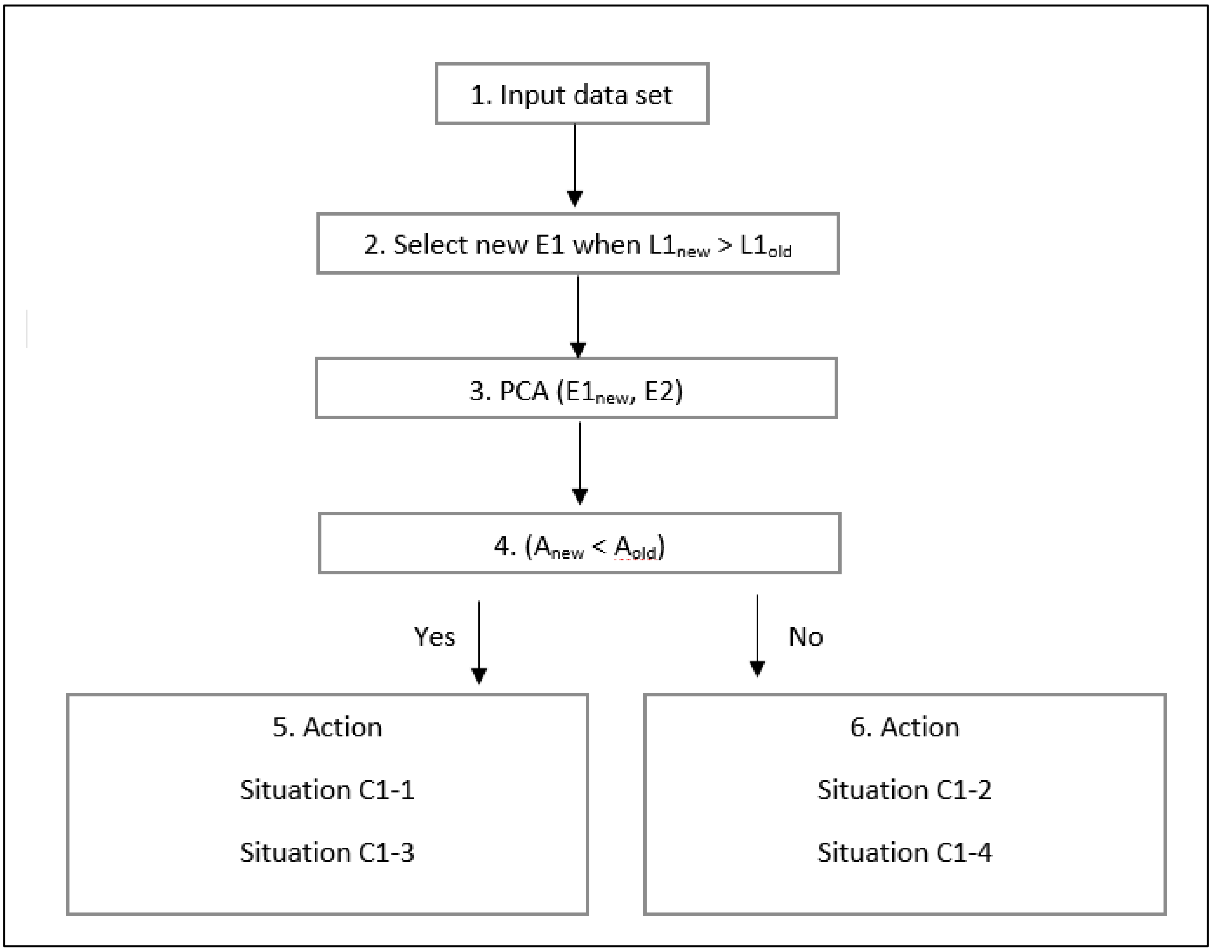


Figure 4-6: Case 'C1' for template selection

The algorithm logical flow above comes in the execution path when the condition of $\theta > 45°$ is met. This condition is tested after the initial calculation of the principal component analysis, keeping in mind the equations mentioned above. In the following stage, a different dataset template is selected to obtain the longer length and Eigen value ratio is checked until it equals 45°, in otherwords, the condition $\theta = 45°$ is met. The inner flow of step 5 is shown in Figure 4-7.

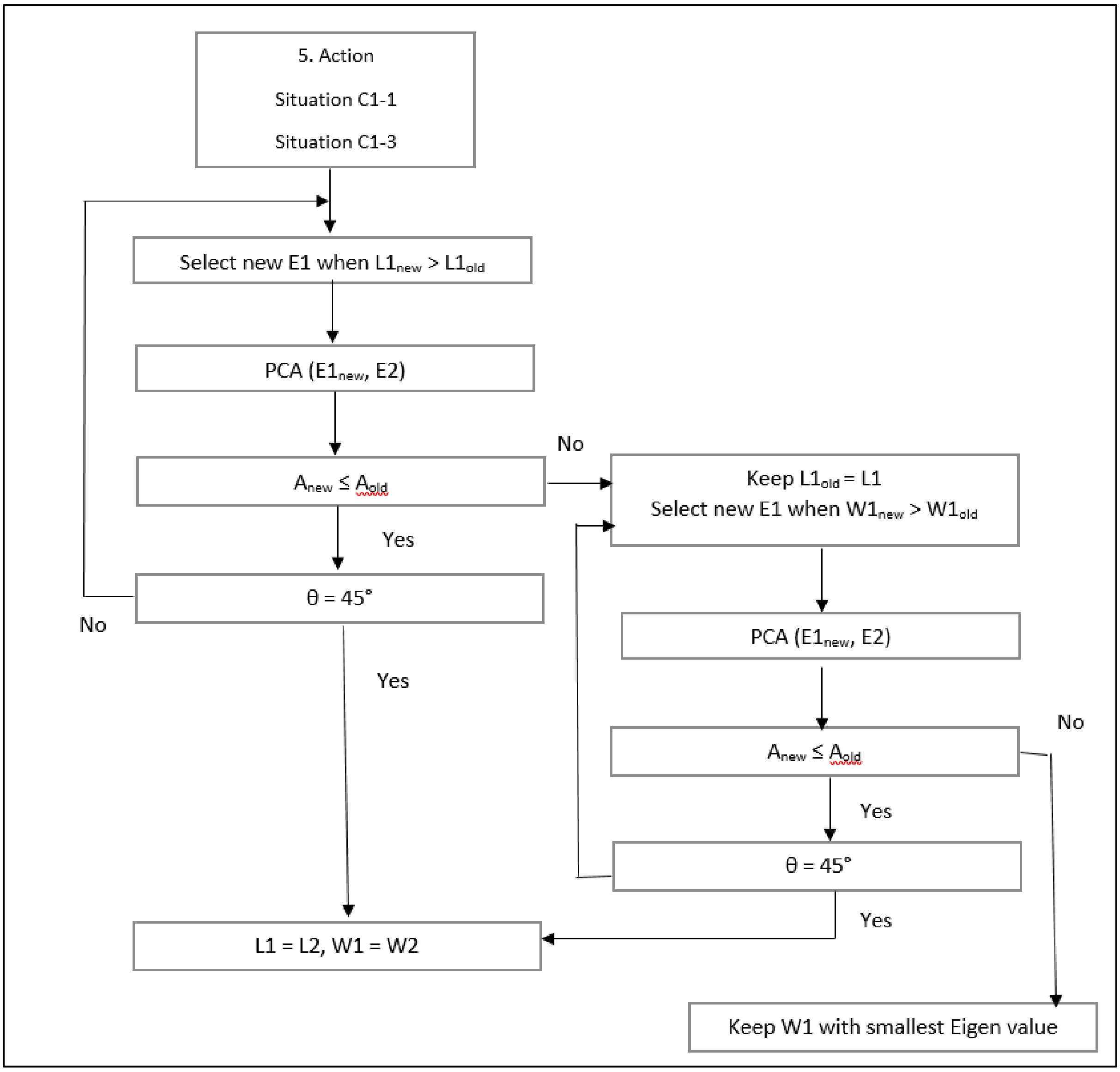


Figure 4-7: Detailed algorithm logical flow for step 5

L1 of E1 is longer E1 of L2 and the algorithm retains E1 as shown in the logical flow diagram in Figure 4-7. If L1 is smaller than L2 then the algorithm selects a new template image. The same logic applies for the width of both images E1 and E2 as shown above. Both images are similar when $\theta = 45°$ with L1 = L2 and W1 = W2. On the other hand, the Figure 4-7 shows two situations, C2 and C4 as well as the length L1 and L2, which if L1 is shorter than L2, then the algorithm retains the old length.

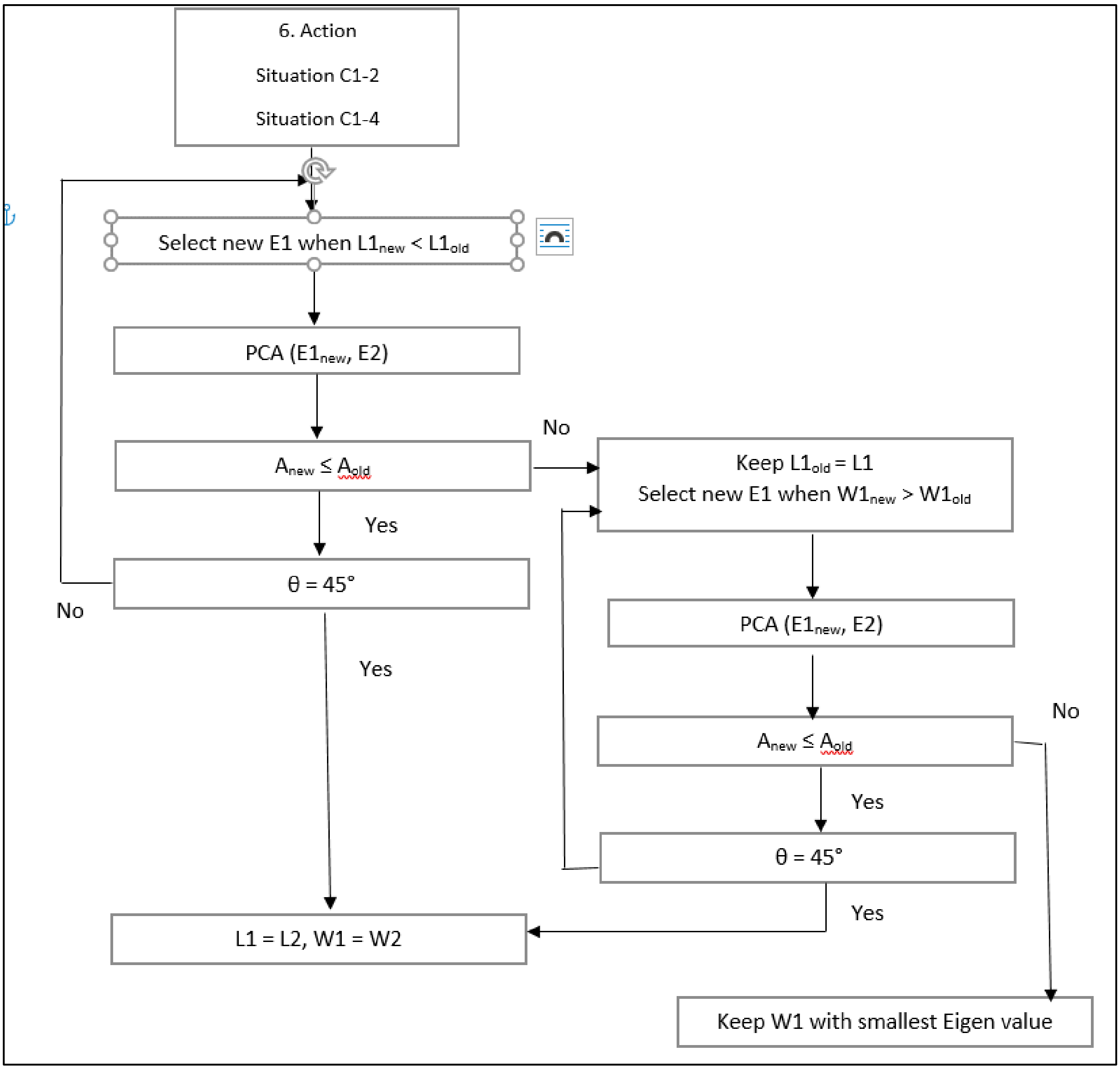


Figure 4-8: Detailed algorithm logical flow of step 6

C2: Case $\theta < 45°$

There are four possible situations in this case which are,

a) C2-1: (W1 = W2) and (L1 > L2)
b) C2-2: (W1 > W2) and (L1 = L2)
c) C2-3: (W1 > W2) and (L1 > L2)
d) C2-4: (W1 > W2) and (L1 < L2)

The logic flow diagram in Figure 4-9 illustrates the logic flow of the algorithm for the situations discussed above.

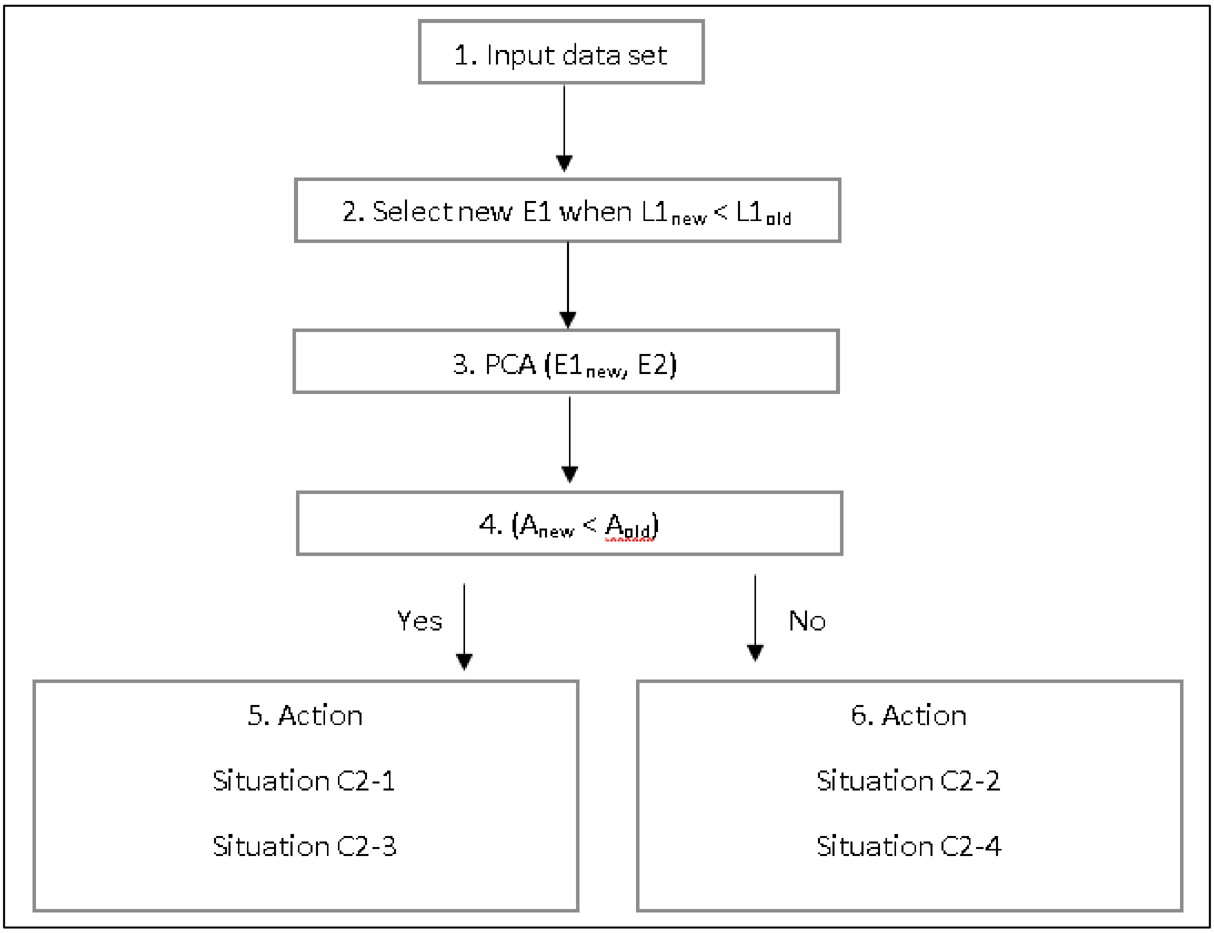


Figure 4-9: Case 'C2' for template selection

This particular path (illustrated in Figure 4-9) executes when the algorithm's logic meets the condition of $\theta < 45°$. This condition is tested after the calculation of initial principal component analysis in the same way it is done in the previous section. In the following stages, the algorithm tries to select the template's dataset with the shorter length and keeps checking the Eigen value ratio. The inner flow of step 5 can be shown in Figure 4-10:

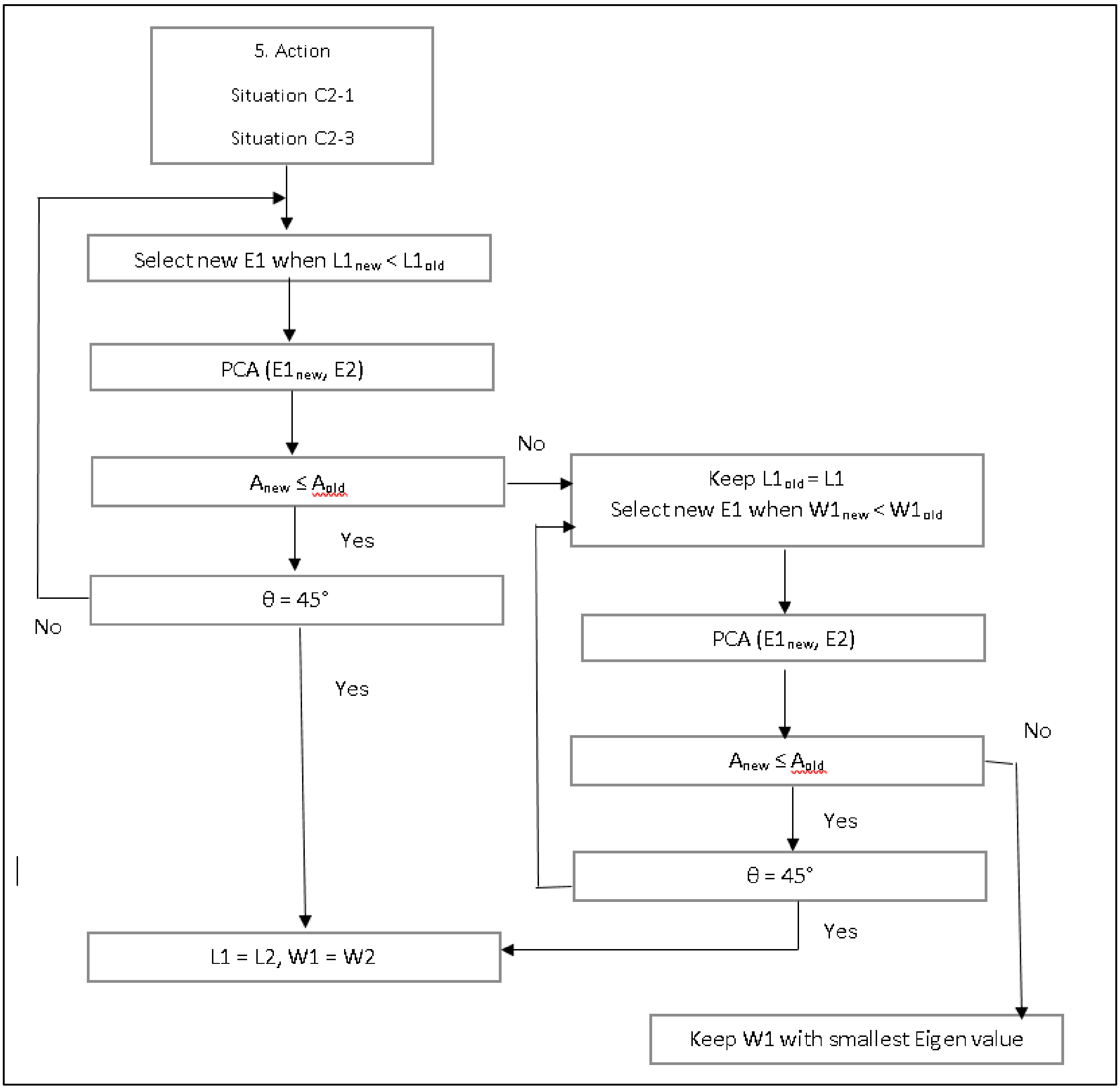


Figure 4-10: Detailed algorithm logical flow of step 5

Length/L1 of Image/E1 is shorter and therefore the algorithm keeps E1. If L1 was greater than L2 then the algorithm selects a new template image. The same logic applies for the widths of both images E1 and E2. Both images will be similar when $\theta = 45°$ with L1 = L2 and W1 = W2. On the other hand, the Figure 4-11 shows two situations, C2 and C4 as well as L1 and L2, and since L1 is shorter than L2, the older length is saved.

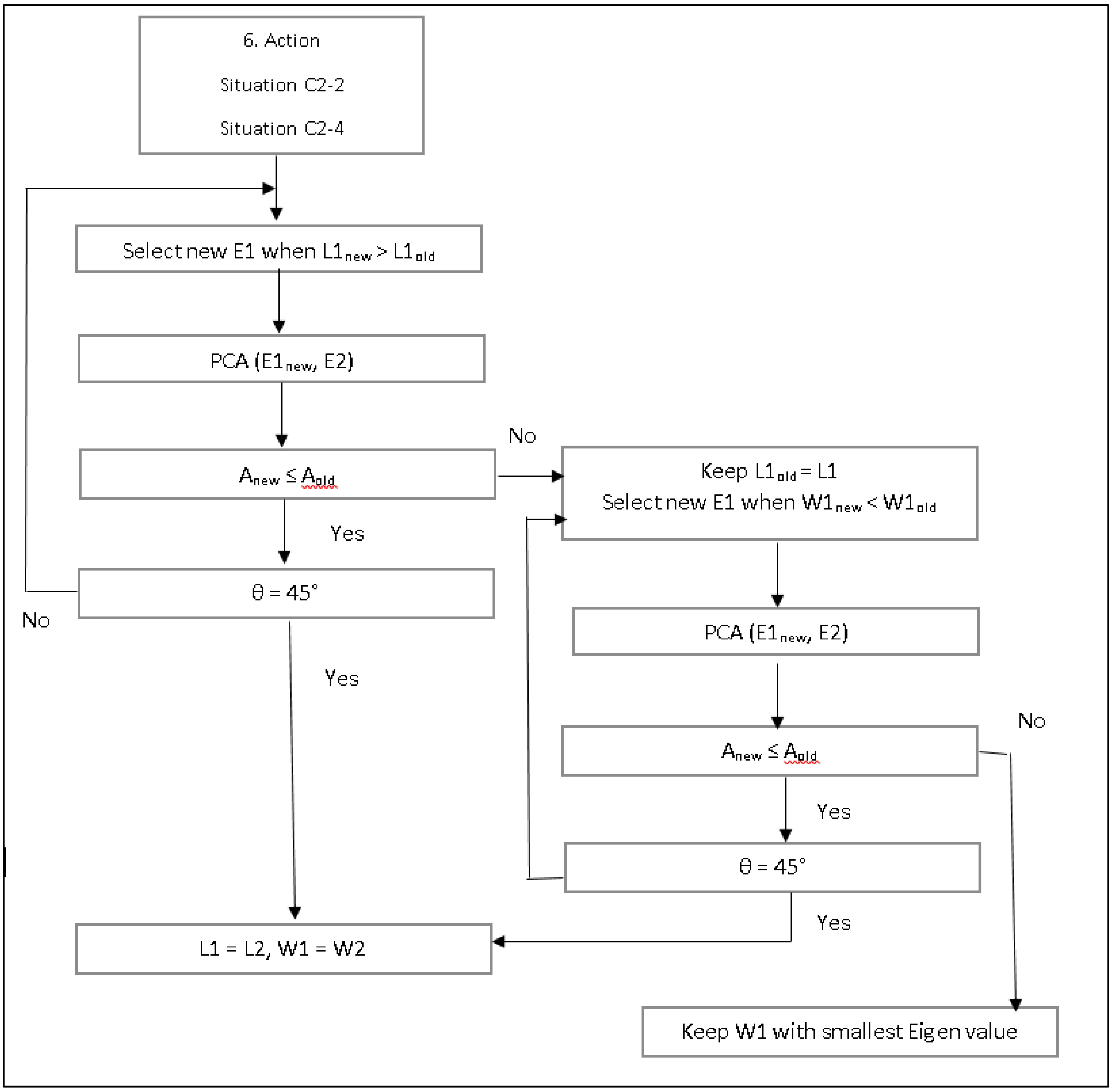


Figure 4-11: A detailed logical flow of step 6

In the principal component analysis, if the calculated value is $\theta \neq 45°$ and $\frac{A_{11}}{A_{22}} \neq 0$, then the two datasets are not perfectly matched.

The steps described above are taken by the algorithm to change the template's dataset in order to find similarities between the two sets and if there aren't any similarities, the algorithm then re-calculates the Eigen value and vector till both datasets match.

4.3.2. Classifiers training

This section describes the classifiers training steps and the feature vectors created at the learning stage. These vectors are used for segmentation of moving objects based on the features extracted from the input video stream and the order of execution of steps can be shown in Figure 4-12:

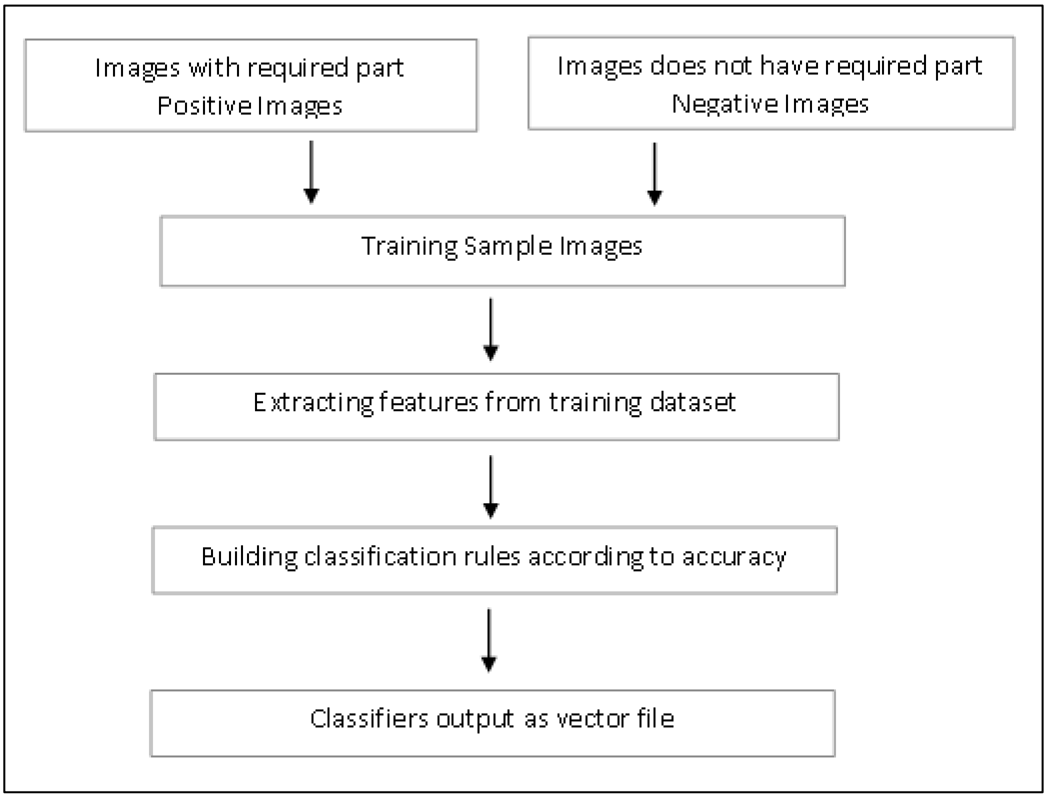


Figure 4-12: Classifiers training steps

The process can be explained using a series of equations, calculated at each step. They lead to the formation of the feature matrix used by the classifiers. Let's assume that input video stream containing all features and data can be described as follows:

$$A = [\, a_1; a_2;\; a_3; \dots a_n] \in \Re^{n*m} \tag{4.17}$$

The above equation describes the input data as a multi-dimensional matrix with m as the number of features and n as the number of samples. The jth sample is:

$$a_j \in \Re^{1*m} \tag{4.18}$$

While, the jst feature vector is,

$$f_j(j = 1, \dots m) \tag{4.19}$$

In accordance with this, the multi-dimensional matrix of combined features and samples takes the form,

$$A = f_1; f_2;\ f_3; \dots f_m \tag{4.20}$$

For a matrix C, the Frobenius norm can be calculated for samples and features as:

$$||C||_{Fi} = \sqrt{\sum_{i=i}^{n} ||c^i||^2{}_2} \tag{4.21}$$

$$||C||_{Fj} = \sqrt{\sum_{J=i}^{m} ||c_j||^2{}_2} \tag{4.22}$$

Using this measure, the features can be shown as:

$$||C||_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=i}^{m} c_{ij}^2} \tag{4.23}$$

Where, ci and cj denote a row and a column of the original multi-dimensional matrix, respectively. This matrix contains all information for the features used by the classifier. To estimate a single featuref_j, the following linear regression model is used:

$$f_j \approx \sum_{i=1}^{m} f_i \, s_{i,j} = A_{sj}, \in j = 1,2, \dots , m \tag{4.24}$$

Where, $s_{i,j}$ represents the i^th feature vector to the j^th sample. In this case the co-efficient vector of the featuref_j, can be formulated as:

$$s_j = \left[\, s_{1,j}; \dots ; s_{ij}; \dots \; s_{m,j} \right] \in \Re^{m+1} \tag{4.25}$$

As a result, the multi-dimensional matrix can be written as:

$$A \approx \; AS \tag{4.26}$$

Where A is the linear combination of all features and,

$$S = \left[s^1; \dots , s^j; \dots ; s^m \; \right] \in \Re^{m+m} \tag{4.27}$$

The value of S can be calculated as follows;

$$min \; \|A - AS\|_F^2 \tag{4.28}$$

To reduce the redundancy and keep the features unique the co-efficient matrix of $| < s^i$, $s^j < \; |$ is used, where, si and sj denote i^th row and j^th row vector of S, respectively. To use all vectors, the following formulas hold:

$$\Omega(S) = \sum_{i=1}^{m} \sum_{j=1, j \neq 1}^{m} | < s^i \, , \qquad s^j > | \tag{4.29}$$

$$\Omega(S) = \sum_{i=1}^{m} \sum_{j=1}^{m} \left| < s^i \, , \qquad s^j > \right| - \sum_{i=1}^{m} | < s^i \, , \qquad s^j > | \tag{4.30}$$

$$\Omega(S) = \sum_{i=1}^{m}\sum_{j=1}^{m}|< s^i , s^j >| - \sum_{i=1}^{m}\left\|s^i\right\|_2^2 \tag{4.31}$$

The values calculated in the above equation are required to identify the features in the input video stream and to track the moving objects and their parts. The framework to reconstruct the trajectories of moving objects uses these features values.

Videos are split into frames to create the data in order to train the algorithm. These frames contain different moving objects in relation to their respective environments or scene. The raw data used for training purposes consists of only images which represent moving body parts. During the training of the classifiers, the first stage is to detect the shapes of the moving parts and save them as subjects in the training samples and the second stage involved the detection of colours and consequent creation of subjects.

4.4. Supervised machine learning

The framework classification is based on different categories of data which are different classes or types such as arms and legs. Each category is assigned a label and the regression model is used for classification purposes.

4.4.1. Multiple Linear Regression

Multiple linear regression approach is a supervised learning approach. The equation below describes the linear regression model:

$$A = \beta_0 + \beta_1 B_1 + \beta_2 B_2 + \cdots + \beta_k B_k + \varepsilon \tag{4.32}$$

A in the above equation is dependent upon the k features of B. If there are ‘n’ number of subjects, then it can be written as below:

$$A_1 = \beta_0 + \beta_1 B_{11} + \beta_2 B_{12} + \cdots + \beta_k B_{1k} + \varepsilon_1 \tag{4.33}$$

$$A_2 = \beta_0 + \beta_1 B_{21} + \beta_2 B_{22} + \cdots + \beta_k B_{2k} + \varepsilon_2 \tag{4.34}$$

$$A_n = \beta_0 + \beta_1 B_{n1} + \beta_2 B_{n2} + \cdots + \beta_k B_{nk} + \varepsilon_n \tag{4.35}$$

The accuracy of the model is determined with the help of $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ and this can be given as below;

$$accuracy = (\varepsilon_1)^2 + (\varepsilon_2)^2 + \cdots (\varepsilon_n)^2 \tag{4.36}$$

The results of multi regression model can be influenced due the training data which is possible in any other machine learning algorithm.

4.4.1. K Nearest Neighbour (KNN)

K Nearest neighbour is the simplest machine learning algorithm and in this particular framework, it is used for the classification of different moving parts. Its learning process is only dependent on the training data provided and it does not depend on the relationship between features and labels, which can be linear or non-linear. For classification purposes, the distance between the template's data subject and the unknown characteristic's data subject, is calculated and the distance is Euclidean distance. This distance measurement is an important integral part of the KNN algorithm and provides positive results. If there are two points (two data sets) as shown below:

$$a = a_1, a_2, a_3, \dots, a_n \tag{4.37}$$

$$b = b_1, b_2, b_3, \dots, b_n \tag{4.38}$$

Then the Euclidean distance can be determined as,

$$euclidean\ distance\ (a, b) = \sqrt{\sum_{i=1}^{n}(b_i - a_i)^2} \tag{4.39}$$

For this framework, since point ‘a’ and ‘b’ are multi-dimensional, the result can be called as Euclidean space. The classification category selected is the one with the nearest distance and it is illustrated in table below where different distance values are shown between the new data set and the training data:

Table 4-1: K Nearest neighbour distances and labels

Subject	Distance	Label
12	0.1	0
25	0.8	2
27	0.2	0

Label ‘0’ appears twice with the smallest distance, so the category labelled as ‘0’ will be selected for the unknown characteristic subject. This process is executed in a sequence of steps, as shown in the Figure 4-13:

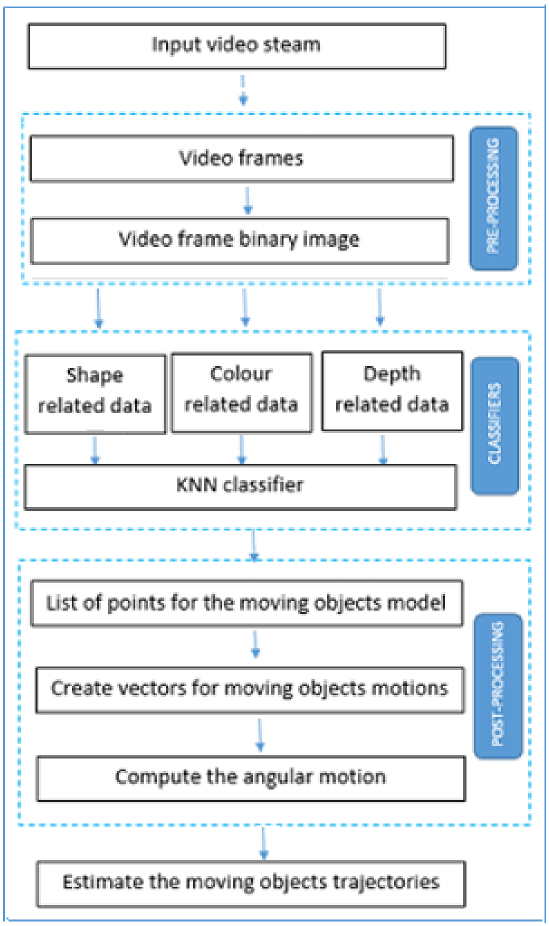


Figure 4-13: Flow of video stream analysed using KNN classifiers

Features information generated with the help of the equations presented above and the KNN classifier decide if the moving object is a human being or not. Similarly, classifiers decide about different moving parts of a moving object.

4.5. Data Post processing

In order to provide informative reconstruction of the trajectories, it is essential to perform some post processing of the data generated after the classifier completes its task. The most important processing steps are as follows:

4.5.1. Estimating the viewing direction

The viewing direction is calculated with the help of the head sphere of the moving object model and with the position of the eyes in the head sphere. If the eyes direction and moving object direction are the same, then the object is viewing in the direction of movement.

4.5.2. Orientation of the moving parts

This information is calculated with the help of position of the face and head hairs. This step is necessary in order to distinguish between left and right hand. The same is applied on the legs of moving object.

4.5.3. Completing the invisible body parts

The missing body parts of moving object of seven sphere-based model are estimated in order to generate meaningful trajectory data.

4.5.4. Estimating the depth of 2D projection

The depth of moving object in the video stream is calculated with the help of geometric calculations.

4.5.5. Detection of the moving objects

The moving objects can be detected with the help of some historical information. All static objects do not change the position in a sequence of frames, while the dynamic object does, and this can be a criteria for identifying new objects in the scene.

4.5.6. Origin adjustment

The logical centre of the scene can be adjusted in order to make the displacement and movement calculations easier.

4.5.7. Camera position adjustment

The camera position can be adjusted to coincide with the origin of the visual scene. This step is needed to ease the run-time data processing.

The above tasks are executed after the trajectory data is calculated using the information obtained during the trajectory reconstruction to facilitate the further analysis by the behaviour analyser of the video analytics framework. These steps are described into more details in the next chapter.

4.6. Conclusion

This chapter presents the logical flow of the algorithm, classifiers as well as data pre and post processing for the proposed framework capable of reconstructing trajectory from input video streams. The framework uses a limited amount of data, extracted from real-time video streams, to track activities as well as events occurring within surveillance monitored zones. There are several advantages of this approach.

Firstly, the framework’s dependence on classifiers reduces the complexity of data processing by eliminating the need of having precise data to derive patterns of features.

Secondly, the control over the accuracy of the results of the framework can be maintained through fine tuning and the utilization of training data with a large number of parameters, such as different lightening conditions of the environment, availability of multiple colours in the training data samples, as well as brightness variations that dictate how items are recognised and logged in the event.

Lastly, the training of classifiers performed in the background and in advance, can be utilised during the run-time so that one can immediately validate the functionality and estimate the efficiency of the analysis through simple observations.

The framework has been created as an independent, self-contained, stand-alone framework which up to this stage, has only been utilized for the purposes of trajectory re-construction of live video streams. However, the framework has other potential uses in direct computer applications where there is a need for more sophisticated engines, capable of processing input video streams in real time for analysis.

5. Implementation of moving objects trajectories framework

5.1. Introduction

This chapter will discuss the proposed model for the regeneration of the moving object’s trajectory in the video and provide high-level description of its internal properties. In order to validate the framework objectives, the system must be successfully built upon the previously proposed model.

5.2. Language

This framework is based on Java programming language, which is an outstanding language for data processing purposes, its components are interrelated, and it also supports heterogeneous environments.

5.3. Libraries

OpenCV library is used for the purposes of developing the framework. OpenCV library is an open source library, which is widely used in image processing applications and is written in C language. It is fast and easy to use and supports elementary image and video processing tasks, which are based on logical and arithmetic operations.

At the same time OpenCV also supports complex image and video processing operations like object detection and moving object trajectories.

As such, the library satisfies all the requirements mentioned above, in chapter 3. The modules used by the framework are listed below.

5.3.1. Image processing (imgproc)

This module enables the processing of real-time data and it includes:

- Smooth (Smooth the image with the help of different available methods).

- Sobel (Derivatives calculation using Sobel operator).
- Scharr (Determine image derivatives using Scharr operator).

5.3.2. High-level GUI and Media I/O (highgui)

This module is used for rich UI framework. It allows for mapping and marking of the output values, in addition to reading different data type inputs.

5.3.3. Geometric Image Transformations

This module is mainly used for the geometrical transformation of moving objects. The methods in this module do not change the video frame content but deform the pixel grid and map this deformed grid to the output frame. This module mainly solves two issues:

- Extrapolation of non-existing pixels
- Interpolation of pixel values

5.3.4. Structural Analysis and Shape Descriptors

This module is mainly used for extracting information from the moving object which can help to identify contours and arc length.

5.4. Tools

This framework primarily needs two tools to perform operations properly. One is the source of input, which can be a video camera or a computer system with a recorded video. The second, is a processing system with the required software mentioned in the above section. This processing system takes the input and regenerates the trajectories of moving objects which constitutes the input source.

5.5. Environment

For the purposes of this research, the framework is only used on a Windows operating system but as it is based on Java, it can run on any other major OS.

5.6. Video Processing framework

The video processing framework does the actual processing on the frames within the video sequences. Most of the classes in the framework deal with the regeneration of moving object trajectories. The framework block diagram is shown in Figure 5-1:

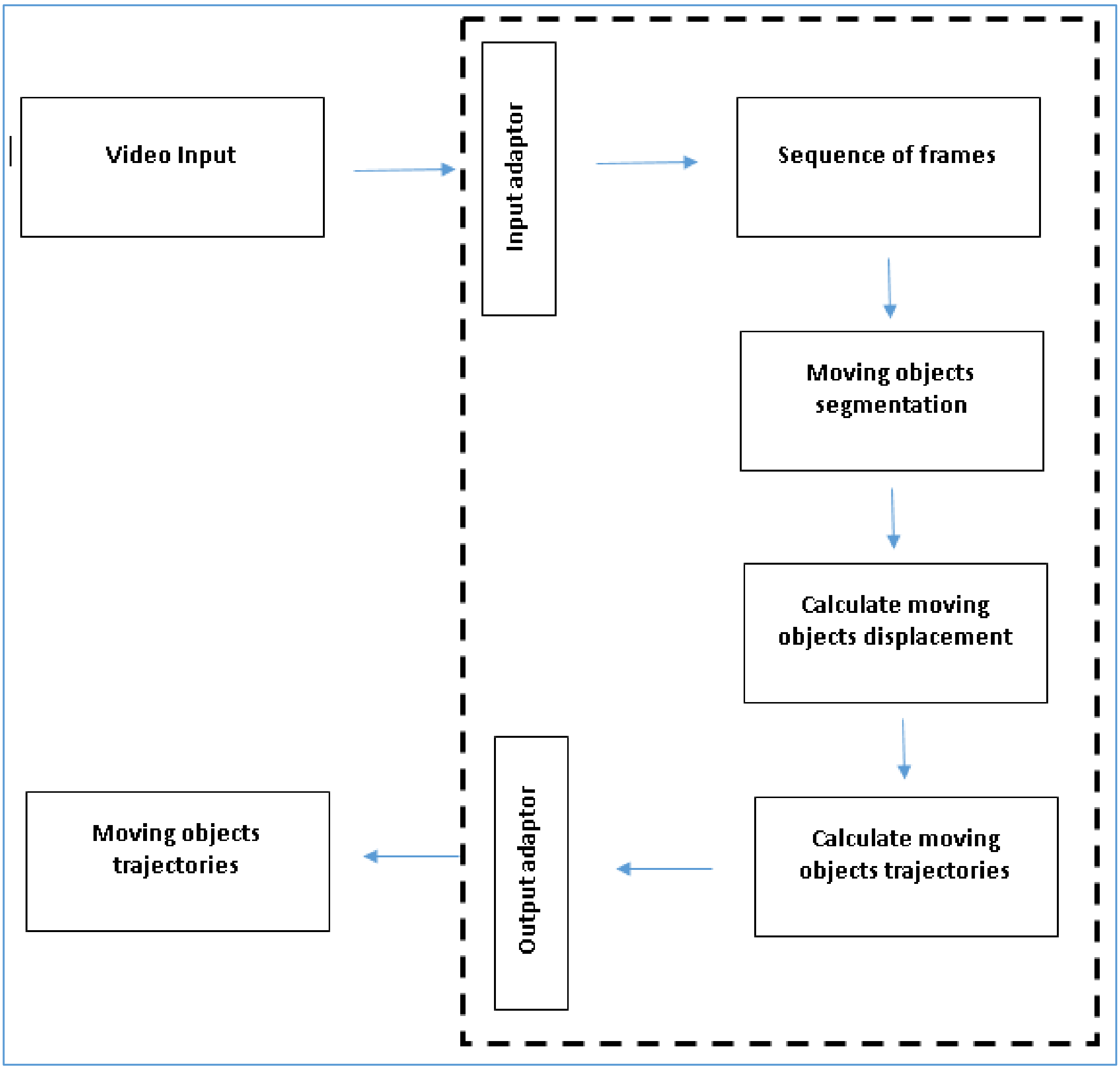


Figure 5-1: Block diagram of framework

Figure 5-1 shows the flow of data in different sections of framework and needed for re-construction of trajectories.

5.6.1. Video frame transformation module

Video data consists of video frames which can be thought of as pictures. These frames are then combined in a time sequence to form one whole video. Generally, videos consist of thirty frames per second, but this can be different, i.e., 25 or 60 FPS.

The Figure 5-2 shows an example of video frames.

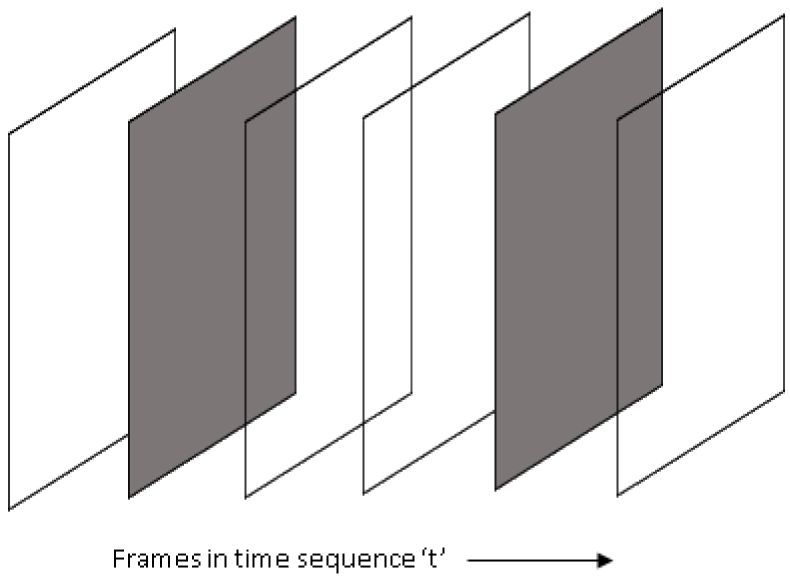


Figure 5-2: Sequence of frames

The two types of frames denoted above as Grey and White make up a video but for the purposes of balancing the load, the framework only processes frames coloured in grey. This means that in a 30FPS video, not all frames are processed.

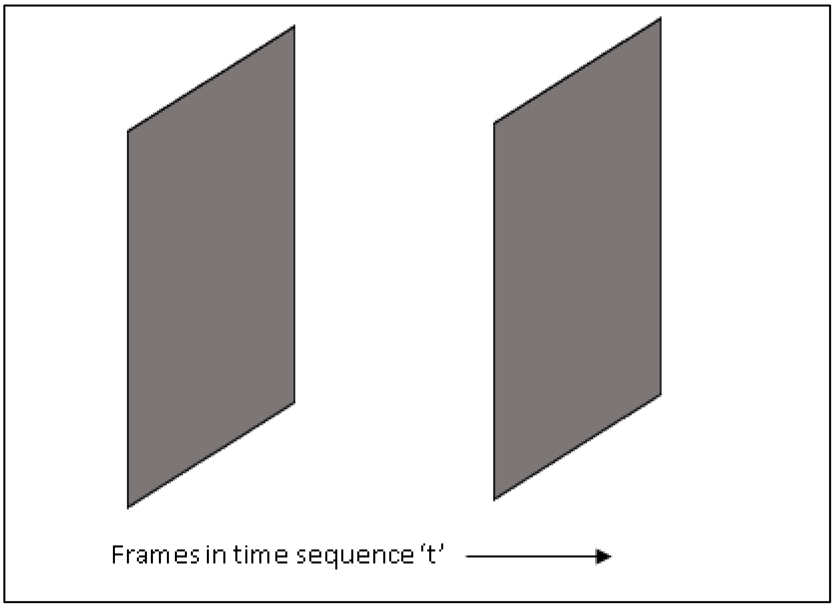


Figure 5-3: Selected frame to process

The Figure 5-3 shows only the video frames selected for processing by the framework and the video frame splitting is done by the video frame transformation module.

5.6.1. Moving objects segmentation component.

This component performs operations on all of the selected frames in a video sequence and it lists the points that represent the moving object in the frame. This component first converts the information of the frame into binary format which is easier to process and then proceeds to detect the moving object.

The Figure 5-4 shows an example of a moving object which is captured and denoted with dashed lines.

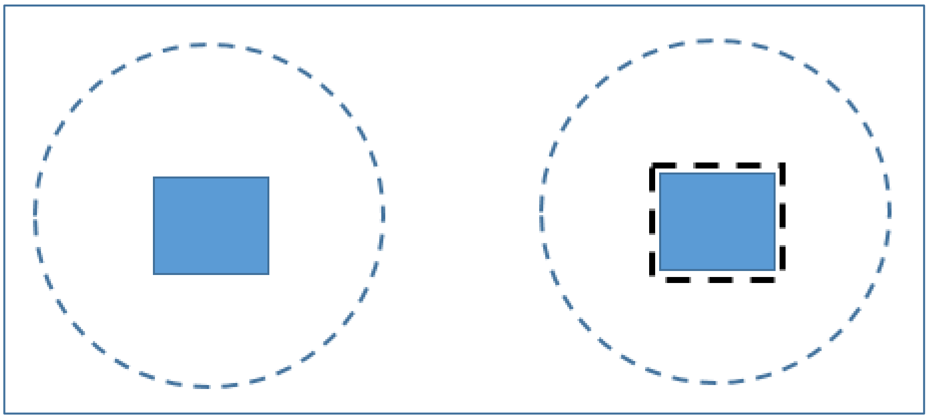


Figure 5-4: Moving object and sequences of points around the moving object

The blue square on the left is the moving object in the Figure 5-4 whereas the right-hand side square is surrounded with the black dashed line which highlight the moving object's boundaries.

5.6.2. Computation of the moving object's displacement component

After identifying the moving object from the video, this component computes the trajectory of the moving object and thereby monitoring the moving object.

When a moving object is represented as a triangle, its position as well as angular velocity is taken and computed separately for each frame. The position of the triangle is denoted by the matrix x_k and transition with by 'F' matrix as shown below.

$$x_k = Fx_{k-1} \tag{5.1}$$

The equation calculates the position of the triangle with the help of previous time at k-1 and the transition F. This transition F will be a 2D matrix (with 2 rows and two columns) trajectory changes of the triangle. The equation above is generated using Kalman filter.

Similarly, the centroid of the moving triangle can be given as:

$$x = \frac{1}{n_h}\sum_{i=0}^{n_h} x_i \qquad (5.2)$$

And

$$y = \frac{1}{n_v}\sum_{i=0}^{n_v} y_i \qquad (5.3)$$

Where n_h are the number of horizontal pixels of the moving triangle while n_v are the number of vertical pixels of the moving triangle.

5.6.1. Estimation of the moving object's trajectories

The calculation of the moving triangle's centroid provides temporal data on the moving objects centroid at each frame which is done with the help of last two equations:

$$t_i = (x_i, y_i) \qquad (5.4)$$

The equation above provides the centroid co-ordinates of the i^{th} frame and the angular velocity of the moving triangle in the video mentioned previously can be shown as below if the position of triangle is Φ and angular velocity is ω.

$$\Phi(t_{i)} = \omega t_i \qquad (5.5)$$

5.7. Trajectories generation components details

Having discussed the overall component structure of the video processing framework, there is still one important component to discuss, therefore, this section will discuss the components used to generate the trajectories.

To understand the functionality of generating a moving object's trajectory, one needs to understand the mathematics involved. The following sections provide mathematical equations to elaborate on the process of generating trajectories from moving objects.

5.7.1. Assumptions

Prior to further development some assumptions have to be made which are as follows:

1) The model framework is linear. It means that, if the system can be modelled at time 't' then it can be modelled at 't+1' after some calculations.
2) Basic noise model is used in order to mimic the effect of many random processes that occur in nature.
3) All disturbing factors (like noise) are Gaussian in nature

5.7.2. Theoretical explanation

This section is based on the details presented above and will discuss the way the framework is implemented, and it is considered as the starting point of the development. The high-level pseudo code is shown in Figure 5-5.

```
Initialize the frame as null
Initialize the normalImage as null
Initialize the binaryImage as null
List of movingObjects as empty
While the frame is available from the videoDataSource
        normalImage = clone frame
        binaryImage = conversion from normalImage
        movingObjects = get list of moving object with the help of normalImage and binaryImage
        Get and display the list of boundary points of the moving objects
        Compute the moving objects displacements
        Compute the moving objects vectors
        Computer the angular motion of the moving objects
        Re-generate the moving object trajectories
        Display the trajectories visual and in textual forms
```

Figure 5-5: Pseudo code screenshot

The Figure 5-6 shows the pseudo logic in the graphical format.

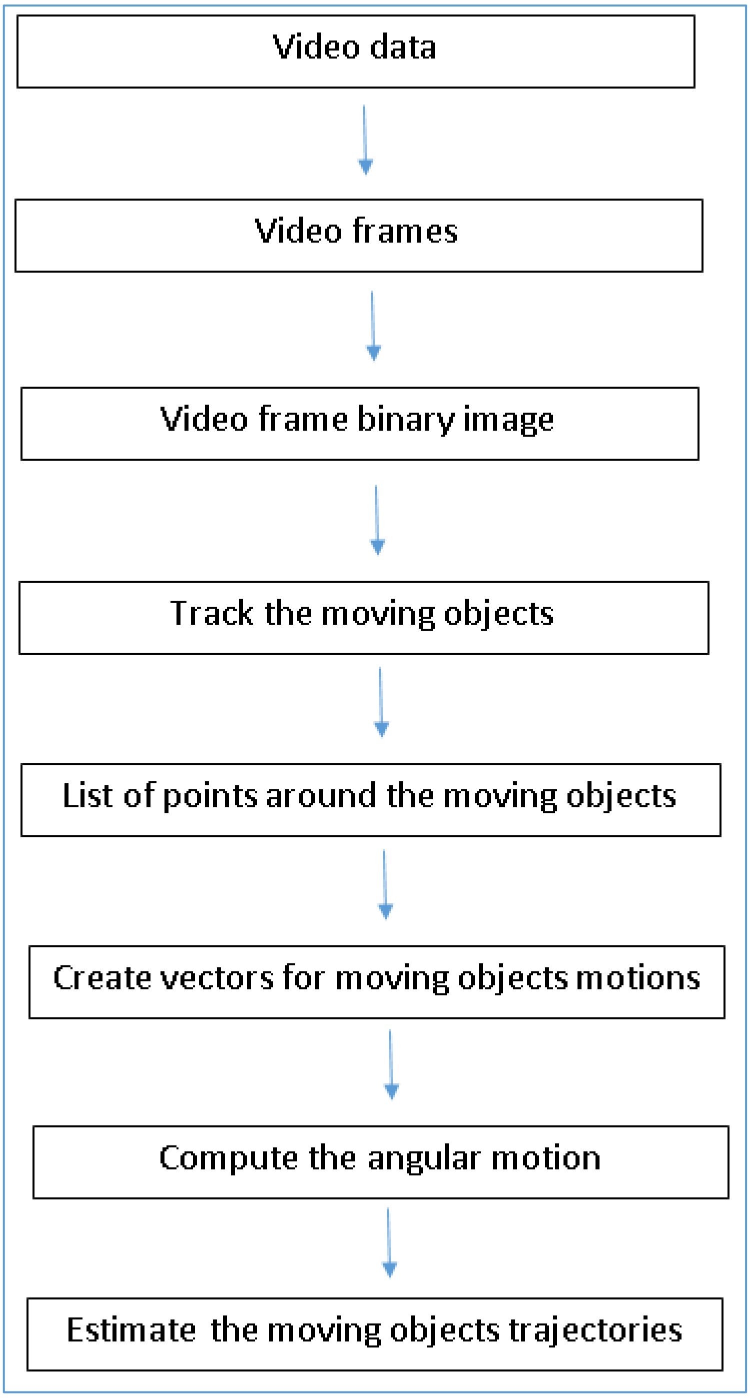


Figure 5-6: Pseudo logic in graphical format

The Figure 5-6 shows the basic steps involved in the re-generation of the trajectories of the moving objects. These regenerated trajectories will be in the format of visual as well as textual output.

5.8. Framework Namespaces

This framework code consists of mainly six namespaces where each namespace is responsible for different elements of the framework. Some few important namespaces are discussed below.

5.8.1. VideoDecomposition namespace

The VideoDecomposition namespace contains the classes that control the operations related to decomposing the video stream into frames and other namespaces work on these frames. This namespace is also responsible for reading the video data input as shown in Figure 5-7.

```
/*
 * Class to handle the video frame operation
 *
 *
 */
public class FramesManagement {

    public String source;
    public VideoCapture video;

    /*
     * Create the new DetectObject object
     *
     * @param   Mat frame       Frame object to detect the moving object
     *
     *
     */
    public FramesManagement(String source){

        this.video = new VideoCapture();
        this.video.open(source);

    }

    /*
     * Get the frame of a video
     *
     *
     *
     */
    public Mat getFrame(){

        Mat frame = new Mat();

        this.video.read(frame);

        return frame;

    }
```

Figure 5-7: Frame management class

5.8.2. ObjectDetection namespace

Classes in this namespace perform operations to detect the moving objects. Canny edge detector and few other algorithms are used in this namespace. This namespace is one of the most important namespaces in this framework. Classes in this namespace get inputs from the VideoDecomposition and AnalyseObject namespaces.

The Figure 5-8 is a screenshot of a class responsible for object detection.

```
/*
 * Class to detect the object
 *
 *
 */
public class DetectObject {

    public Mat frame;

    /*
     * Create the new DetectObject object
     *
     * @param   Mat frame       Frame object to detect the moving object
     *
     *
     */
    public DetectObject(Mat frame){

        this.frame = frame;

    }

    /*
     * Return the edges of the objects
     *
     */
    public Mat findEdges(){

        Mat frameGray = new Mat();
        Mat temp = new Mat();
        int thresh = 120;

        // Convert image to gray and blur it
        Imgproc.cvtColor(frame, frameGray, Imgproc.COLOR_BGR2GRAY);
        frameGray.convertTo(temp, 0);
        Imgproc.blur(frameGray, frameGray, new Size(5,5));

        // Detect edges using canny
        Mat edges = new Mat();
        Imgproc.Canny(frameGray, edges, thresh, thresh * 3);

        return edges;

    }
```

Figure 5-8: Class detect the objects in the video frame

5.8.3. ObjectDisplacement namespace

This namespace gets inputs from the ObjectDetection namespace. Classes here use the values of two different consecutive time instances in order to calculate the displacements. Displacement values are calculated with the help of the Euclidian distance. The screenshot in Figure 5-9 shows the class performing vector calculation operations.

```
/*
 * Class to perform arithmetic option for vectors
 *
 *
 */
public class Calculation {

    public Point startPoint;
    public Point endPoint;


    /*
     * Create the new calculation object
     *
     * @param   Point startPoint        Start Point
     * @param   Point endPoint          End Point
     *
     *
     */
    public Calculation(Point startPoint, Point endPoint){

        this.startPoint = startPoint;
        this.endPoint   = endPoint;

    }


    /*
     * Calculate the distance between points
     *
     *
     */
    public double distance(){

        return Math.sqrt(Math.pow((this.endPoint.x - this.startPoint.x),2)  + Math.pow((this.endPoint.y - this.startPoint.y), 2));

    }

}
```

Figure 5-9: Vector distance calculation

5.8.4. AnalysisObject namespace

Classes in this namespace use CascadeClassifier and analyses the available objects in the frames. The output of this namespace plays an important role for the ObjectDectection namespace. The Figure 5-10 is screenshot with an example of a class that extracts object information from a frame.

```
/*
 * Class to find contours of the object
 *
 *
 */
public class Contours {

    public Mat frame;

    /*
     * Create the new Contours object
     *
     * @param   Mat frame       Frame object to detect the moving object
     *
     *
     */
    public Contours(Mat frame){

        this.frame = frame;

    }

    /*
     * Return the contours of the objects
     *
     */
    public List<MatOfPoint> findContours(){

        DetectObject detectObject = new DetectObject(this.frame);
        Mat edges = detectObject.findEdges();

        List<MatOfPoint> contours = new ArrayList<MatOfPoint>();
        Mat hierarchy = new Mat();

        Imgproc.findContours(edges, contours, hierarchy, Imgproc.RETR_EXTERNAL, Imgproc.CHAIN_APPROX_SIMPLE, new Point(0,0));

        return contours;

    }
```

Figure 5-10: Class to find the object contours

5.8.5. CalculateTrajectory namespace

This namespace generates the output which is the trajectory of a moving object. This namespace mainly depends on the input from ObjectDetection and ObjectDisplacement namespaces. The main operation here is the vector manipulations and the screenshot in Figure 5-11 shows the Trajectory class.

```
/*
 * Class to handle trajectory operations
 *
 *
 */
public class Trajectory {

    List<SimpleSpehereData> listSimpleSpehereData;

    /*
     * Create the new Trajectory object
     *
     *
     */
    public Trajectory(){

        this.listSimpleSpehereData = new ArrayList<SimpleSpehereData>();

    }

    /*
     * Add the value in the trajectory data
     *
     */
    public void addValue(SimpleSpehereData simpleSpehereData){

        this.listSimpleSpehereData.add(simpleSpehereData);

    }
```

Figure 5-11: Trajectory operation class

5.8.6. DataManagement namespace

Classes in this namespace hold the data needed for different namespace communications. Class ‘SimpleSpehereData’ contains the information which is generated from ObjectDetection namespace and passes it to the CalculateTrajectory namesace. This class is mainly used for the sphere-based model of moving objects and the class holding the sphere data is shown in the Figure 5-12 with screenshot.

```java
/*
 * Class to hold the data to show in the video
 *
 *
 */
class SimpleSpehereData {

    public Point point;
    public int radius;


    /*
     * Create the new object
     *
     * @param   int x           Point x co-ordinate
     * @param   int y           Point y co-ordinate
     * @param   int radius      Radius of the circle
     *
     */
    public SimpleSpehereData(int x, int y, int radius){

        this.point = new Point(x, y);
        this.radius = radius;

    }

}
```

Figure 5-12: Code screenshot holding data about sphere

5.9. Framework Interaction

The main components of the framework have already been discussed and this section will go through the ways these components interact with each other within the boundaries of the framework. The Figure 5-13 shows the flow of the video data in through each component of the framework.

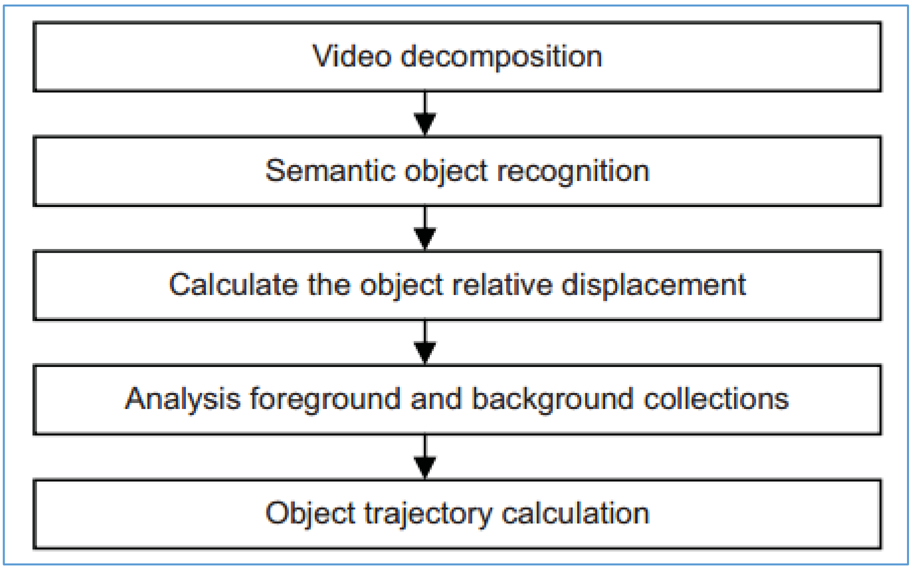


Figure 5-13: Interaction between the Framework's components.

As such, the program shown in Figure 5-14 and Figure 5-15, has been developed to handle basic object movements.

```
package openvideo;

import java.util.ArrayList;
import java.util.List;

import org.opencv.core.Core;
import org.opencv.core.Mat;
import org.opencv.core.MatOfPoint2f;
import org.opencv.core.Point;
import org.opencv.core.Scalar;
import org.opencv.core.Size;
import org.opencv.highgui.Highgui;
import org.opencv.highgui.VideoCapture;
import org.opencv.imgproc.Imgproc;
import org.opencv.imgproc.Moments;
import org.opencv.core.MatOfPoint;

public class TrajectoryReGeneation {

    public static void main(String[] args) {

        int thresh = 100;
        int max_thresh = 255;

        System.loadLibrary(Core.NATIVE_LIBRARY_NAME);

        System.out.println(System.getProperty("sun.arch.data.model"));

        VideoCapture cap = new VideoCapture();
        boolean $result = cap.open("video.avi");
        System.out.println("\nOpen");
        System.out.println($result);

        if(!cap.isOpened()) {
            System.out.println("No, video data input");
        } else {
            System.out.println("Yes, video data input");
        }
```

Figure 5-14: Basic code screenshot part 1

```java
int loop = 1;
Mat frame = new Mat();
cap.read(frame);

Mat frameGray = new Mat();
Mat temp = new Mat();
Mat prevImg = new Mat();

// Convert image to gray and blur it
Imgproc.cvtColor(frame, frameGray, Imgproc.COLOR_BGR2GRAY);
frameGray.convertTo(temp, 0);
Imgproc.blur(frameGray, frameGray, new Size(5,5));
Imgproc.bilateralFilter(temp, prevImg, 5, 20, 20);

System.out.println("Processing in frame loop start");
while(true) {

    System.out.println("Processing in frame: " + loop);

    if(frame.empty()) {
        break;
    }

    // Detect edges using canny
    Mat edges = new Mat();
    Imgproc.Canny(frameGray, edges, thresh, thresh * 2);

    // Find contours
    Mat hierarchy = new Mat();
    List<MatOfPoint> contours = new ArrayList<MatOfPoint>();
    Imgproc.findContours(edges, contours, hierarchy, Imgproc.RETR_TREE, Imgproc.CHAIN_APPROX_SIMPLE, new Point(0,0));

    // Get the moments
    List<Moments> mu = new ArrayList<Moments>();
    for( int i = 0; i < contours.size(); i++ ) {
        mu.add(i, Imgproc.moments(contours.get(i), false ));
    }

    //  Get the mass centers:
    List<Point>  mc = new ArrayList<Point>();
    for( int i = 0; i < contours.size(); i++ ) {
        Moments p = mu.get(i);
        int x = (int) (p.get_m10() / p.get_m00());
        int y = (int) (p.get_m01() / p.get_m00());
        mc.add(i, new Point(x, y));
    }

    // Draw contours
    Mat drawing = Mat.zeros(edges.rows(), edges.cols(), 3);
    for( int i = 0; i< contours.size(); i++ ) {
        Scalar color = new Scalar(0,255,255);
```

Figure 5-15: Basic code screenshot part 2

5.9.1. Linear motion of moving object

A single sphere covers the whole moving object. Once the sphere shape has been scaled to the size of a model, it assigned as a DataManagement class object. From this point onwards, every calculation made on this object has been applied to properties of the model associated with the object. As this is a linear motion of the moving object, simple calculations of CalculateTrajectory namespace reveals this type of motion. This type of motion is discussed in the previous chapter about sphere-based model with the title ‘Continuous linear motion moving object’. All mathematical explanation is coverd about this type of motion.

5.9.2. Static object start motion

This type of motion and sphere-based model is discussed in the previous chapter with the title 'Static object start motion'. As mentioned before, the static object is covered by a sphere and data object instance is passed to the class of CalculateTrajectory namespace. In initial time instance, there will be no displacement but after few time instances ObjectDisplacement namespace and CalculateTrajectory namespace generate the values associated with this type of object motion.

5.9.3. Static object picked up by another moving object

In the environment, it is possible that a static object starts motion due to another moving object. This type of motion has similar features like the previous type of motion. But here, the motion direction and displacement depend upon another moving object. Title 'Static object picked up by another moving object' is used in previous chapter to discuss about sphere-based model for this type of motion.

5.9.4. Moving object disappearing from the environment

This type of motion is described with the title 'Moving object disappearing from scene' in the previous chapter. This happens when the moving object disappeared from the scene or in other words go out from the focus of the camera. Before this moving object disappears, ObjectDisplacement namespace and CalculateTrajectory namespace generate the values associated with this type of object motion. But the classes in these namespaces do not generate any value after disappearing.

5.9.5. Moving object rotation motion

This type of motion is not covered by sphere-based model properly alone and this needed post-processing of data. This type of motioned is described in previous chapter with the title 'Moving object rotation motion' and uses reference points in the sphere to identify these motions. The proposed model with data post-processing handle these motions easily and efficiently.

5.10. Conclusion

The content presented in this chapter is about the framework (for the reconstruction of trajectories of moving object) implementation and is needed to develop the appropriate framework for re-generation of trajectories and their related rational details.

6. Comparative analysis of classifier algorithms

The aim of this chapter is to investigate and compare the accuracy of different frameworks involved in the identification of human body part in motion.

6.1. Researches comparison

Cross validation technique is used to compare the statistical results generated by the framework against those of existing research frameworks. The research aims to estimate the accuracy of the framework results and carries out the model comparison used in this research with other existing research models.

6.1.1. Objective

The main objective is to compare the results of this framework with the results found in two existing researches entitled 'A Method for Body Part Detection, Tracking and Pose Classification using RGB-D Images' (M5AIE) [119] as well as 'Scale and Rotation Invariant Approach to Tracking Human Body Part Regions in Videos' [120].

6.1.2. Challenges

- Creating a prototype in order to analyse the output of the previously mentioned research undertakings
- Transforming the outputs to a format such that all 3 results can be compared together
- Comparing the outputs of the three frameworks

6.1.3. Opportunities

There are multiple existing researches focused on the identification of different body parts; each one of which uses different techniques dissimilar to the proposed ones. These solutions do not track all body parts, as it is the case in this research.

6.1.4. Existing Research Review

Below is an overview of the previously mentioned research frameworks, which will later be compared.

6.1.4.1. A Method for Body Part Detection, Tracking and Pose Classification using RGB-D Images

The author of this research uses a modified Accumulative Geodesic Extrema approach for detecting body part candidates [119]. It also uses Medial Axis transformation, Adapted AGEX, ASIFT, Aligned Images, and Estimation for human body part detection and tracking [119]. Also, this research classifies the human poses which is out of the scope of this research.

6.1.4.1.1 Strategy

Depth sensors are used to address an extensive range of problems, including skeleton tracking [121], gesture recognition [122], activity monitoring, collision detection [123], 3D reconstruction [124] and robotics. Depth information can also facilitate background subtraction [119]. Early publications subtract background using stereoscopic cameras [125] [126] and other more recent publications use Kinect sensors with the help of other techniques [127] [128]; however, there are only minor contributions that address the topic directly [119] with M5AIE research going further by describing how to fill this gap, and provides a starting point for further research.

Figure 6-1: RGB and depth images [1]

For body parts detections and tracking, M5AIE research uses a combination of the AGEX and ASIFT methods. These methods then align RGB and depth images to identify five major body parts (hands, feet and head) and track each of the body parts using an adapted ASIFT using matching algorithm.

6.1.4.1.2 Model

The Accumulative Geodesic Extrema (AGEX) points are detected through a graph construction that uses the human body's pixels [119]. Dijkstra algorithm is then applied [129] to find the extreme points of the generated graph. These extremes are considered to be the main body parts: head, hands and feet. The authors of AGEX use the results as an input for a motion capture system in a later study [130]. Baak et al. also uses the geodesic extrema points as one of the stages for full body pose reconstruction [131].

Once the human body part is detected and labelled, it is then possible to track each of the parts. The author also uses the ASIFT algorithm, which takes advantage of the SIFT method. The SIFT method was proposed by Lowe, and it finds similarities between different images by extracting features that are invariant to rotation, illumination and isotropic scale [132].

6.1.4.1.3 Strength

This uses less pre-processing or training for detecting and tracking human body parts. Affine-SIFT is used for feature extraction and body part location estimation both of which are then combined and used for tracking objectives of movements without self-occlusions [119].

6.1.4.1.4 Weakness

As discovered in later experiments; this research is limited to indoor environment, static background, static position and orientation of the sensor and to single-user segmentation [1]. Investigations carried out as part of analysing the results, show that to be correctly tracked, sequences must not have body part occlusions [119].

6.1.4.2. Scale and Rotation Invariant Approach to Tracking Human Body Part Regions in Videos

The author here proposes a method to track human body part, which is independent of scale and rotation variations. This method depends upon the assembly of human body parts with the spatial and temporal constraints of a human body plan [120]. It uses Dynamic Programming method to make the human body part assembly scalable and efficient.

6.1.4.2.1 Strategy

The tracking of the body part region is formulated into an optimization problem and assigns a body part region to each of the graph nodes so that the assignment cost is minimized [120]. However, this research assumes that body part must be in a fixed position (i.e. the location of the head and hands in are pre-defined) and not dependent on an input [133]. These body part locations are randomly merged with super pixels whose overall shape has a high object-ness score [15]. Different from homogeneous super pixels, the body part location arrangements have a high chance of finding the correct body parts even if the target subject wears a mono colour clothing [15]. The proposed method assembles the chosen body part locations for a valid human shape configuration [120].

6.1.4.2.2 Model

This research uses a human body plan consisting of five body parts: a torso, two arms and two legs. The graph in Figure 6-2 shows the connections between the body parts in each video frame and between successive video frames.

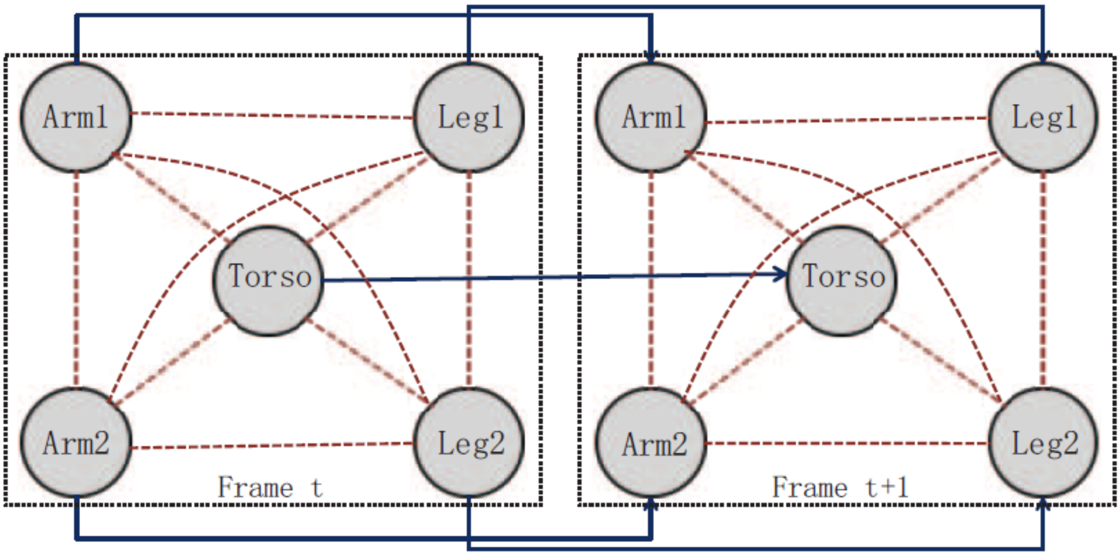


Figure 6-2: Interaction between body parts in two adjacent video frames [2]

Each node in the above graph indicates a body part and the edges indicate the interactions among these body parts [120]. However, only the arm-torso, leg-torso, arm-arm and leg-leg constraints are considered as well as the enforcement of the coupling between arms and legs [120]. The graph edges also connect the corresponding body parts to successive video frames to enforce smooth transitioning of each part through time [120].

6.1.4.2.3 Strength

The rotation and scale independent method used in this research for detecting different human body parts is efficient since it can detect the part regardless of any executed motion.

Using Dynamic Programming, an effective method was developed and used to find ‘n’ best body configuration where ‘n’ is a relatively small number.

6.1.4.2.4 Weakness

When it comes to the limitations of this research, there are three main points. Firstly, the research produces good results only in outdoor environment. Secondly, further investigation of the results show that the proposed methods do not work properly with fast moving objects and lastly, in order to get the best results, the clothing colour of each body part must be different otherwise the results will be inadequate. This is hardly the case in the real world where people are likely to wear clothes of the same colour (i.e. blue t-shirt and blue jeans).

6.1.5. Strength of PhD thesis

In this research, our focus is mainly on designing and computing efficient and accurate model to reconstruct the trajectories of moving human object without imposing restrictions on the environment and other constraints.

6.1.5.1. Strategy

The goal is to estimate the location of body parts in the input video with the help of trained classifiers. The Figure 6-3 shows an example of a seven sphere-based model being used in this research.

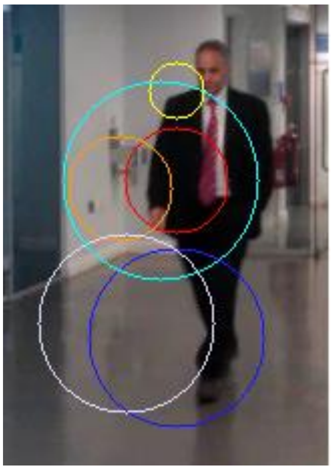


Figure 6-3: Seven sphere-based model with a single moving object

First, k-nearest neighbour algorithm is adopted for detection of body parts of the moving object. The output of this is then used to determine if a moving object is a human being with the help of Logistic regression algorithm.

6.1.5.2. The Model

The proposed model is a seven-sphere based model used to model the moving body parts in input video stream. Each sphere receives the information about the body part with the help of trained k-nearest neighbour's algorithm. The 3D reconstruction of the object movements using their 2D projections on the frames of a digital video signal is the choice of a suitable composite model. Different approximations are possible depending on whether the precision needed and the complexity of the recognition algorithms can be afforded. It is an optimal in the sense that it combines the simplicity of the spherical shape with the sophistication of the composite capsule as illustrated in the Figure 6-4.

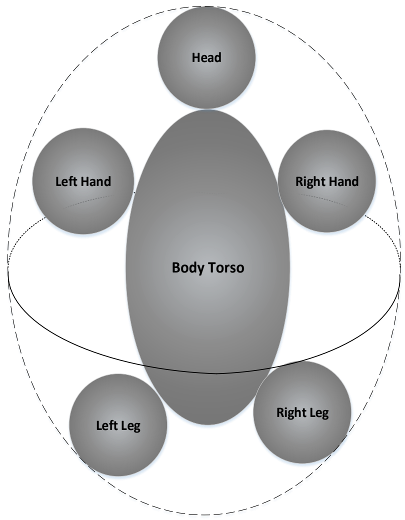


Figure 6-4: Seven sphere-based model

6.1.5.3. Strength

Different models are described and initially used in this research and each of them have different benefits and drawbacks from outcome point of view. The following table summarizes the comparison of moving object models with the seven-sphere based model with better performance for different characteristics.

Table 6-1: Moving object comparison model

Moving object models	Model characteristics			
	Algorithmic Complexity	Accuracy	Representing rotation	Amount of information
Point-based model	Low	Low	No	Low
Spherical model	Low	Medium	No	Medium
Prismatic shape model	Medium	Medium	Yes	Medium
Seven spheres model	Medium	High	Yes	High

6.1.5.4. Weakness

The usage of classifiers evidently improves upon the existing trajectory re-construction method but also raises new questions when dealing with existing problem of training of classifiers. Especially, the challenging topic is the gathering of training data and the construction of ground-truth information needed for classifiers' training. It is also noticed that the presence of the shadow of the moving object in the environment affects the accuracy of the re-constructed trajectories. Seven sphere-based model produces good results when the whole moving object is in the active scene due the requirement of tracking each part of the moving object according to the model.

6.1.6. Competition analysis

This is essential to compare using different test cases as appearance of the body parts in different videos has different depth and complexities. Different research endeavours track different body parts as outlined in the table below.

Table 6-2: Body parts considered in different research

	Existing Research [121]	Existing Research [122]	This thesis
Head	Yes	No	Yes
Hands	Yes	Yes	Yes
Legs	Yes	Yes	Yes
Torso	No	Yes	Yes
Overall body	No	No	Yes

Background and lighting arrangements in the scene under investigation can make the tracking and identification process difficult. To evaluate different research studies, the following six cases are examined.

6.1.6.1. Case 1: Fast hand movement in outdoor sunlight environment

In this input video stream, the human object is moving hands in sunlight and is in an outdoor environment to test the framework. The hands are detected with the help of colour of skin classifier. The Figure 6-5 is a screenshot of one frame of the input video stream.

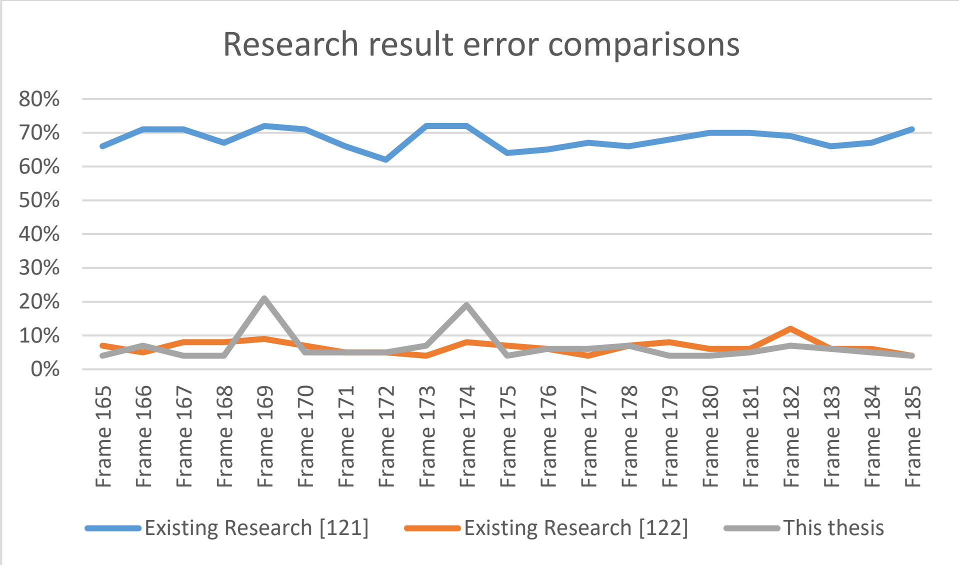
Figure 6-5: Frame of input video stream when human object moving hand in outdoor environment

The following are the details about the input video stream with interesting section of input stream according to the variation in results:

Frames per second = 24

Total frames = 233

Interesting frames = from 165 to 18



The graph in the

Figure 6-6 shows the results of the frames labelled as “interesting” and the different error values.

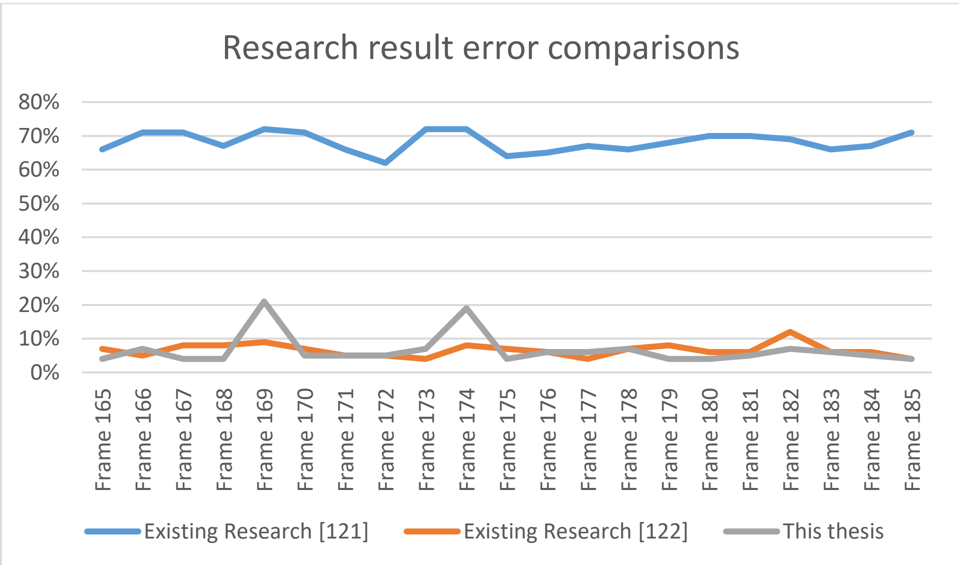


Figure 6-6: Graphical view of output accuracy in outdoor environment

Below is the tabular result of the interesting frames and the resulting percentage of error among the two existing research frameworks as well as the proposed framework.

Table 6-3: Error percentage in results when human object is in outdoor environment

	Existing Research [121]	Existing Research [122]	This thesis
Frame 165	66%	7%	4%
Frame 166	71%	5%	7%
Frame 167	71%	8%	4%
Frame 168	67%	8%	4%
Frame 169	72%	9%	21%
Frame 170	71%	7%	5%
Frame 171	66%	5%	5%
Frame 172	62%	5%	5%
Frame 173	72%	4%	7%
Frame 174	72%	8%	19%
Frame 175	64%	7%	4%
Frame 176	65%	6%	6%
Frame 177	67%	4%	6%
Frame 178	66%	7%	7%
Frame 179	68%	8%	4%
Frame 180	70%	6%	4%
Frame 181	70%	6%	5%
Frame 182	69%	12%	7%
Frame 183	66%	6%	6%
Frame 184	67%	6%	5%
Frame 185	71%	4%	4%

As evident from the above results, it is obvious that the ‘Research [1]’ is not performing well due to the sunlight but it would perform better in indoor conditions. On the other hand, ‘Existing Research [122]’ produces accurate results as it is meant to work in outdoor environments and the condition of light was good. This proposed framework was only inaccurate in frame number 169 and 172 dues to the presence of shadows. However, overall, ‘Research [2]’ as well as this research produce good results.

6.1.6.2. Case 2: Playing music in indoor environment

In this input video stream, human object is moving hands while playing the guitar in an indoor environment to test the framework. The Figure 6-7 is a screenshot of one frame from the input stream.

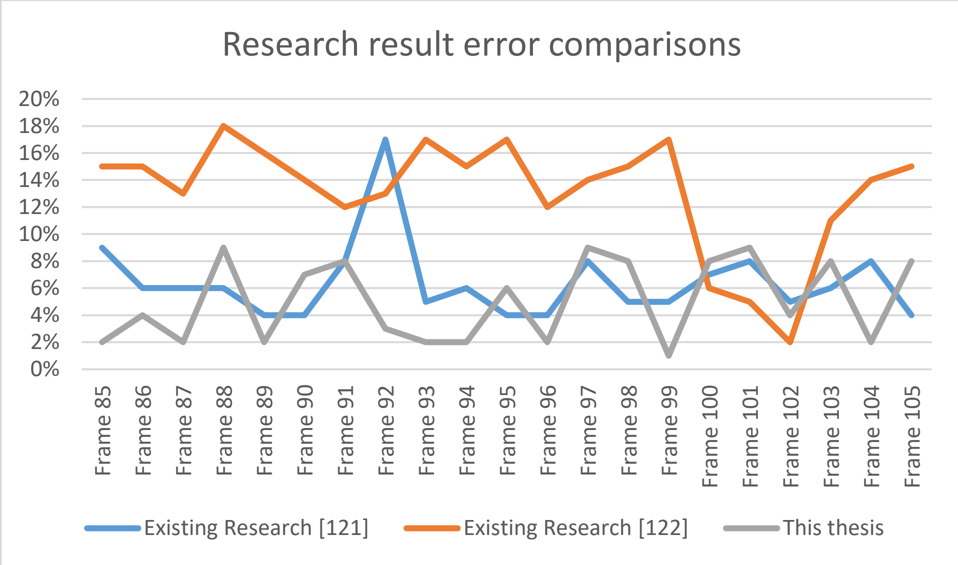
Figure 6-7: Frame of input video stream when human object moving hand in indoor environment

The following details are taken from the video input stream and frame 85 to 105 are mainly the range of frames considered and are referred to as Interesting Frames.

Frames per second = 24

Total frames = 722

Interesting frames = from 85 to 105



The graph in the

Figure 6-8 shows the results of the frames labelled as "interesting" and the different error values.

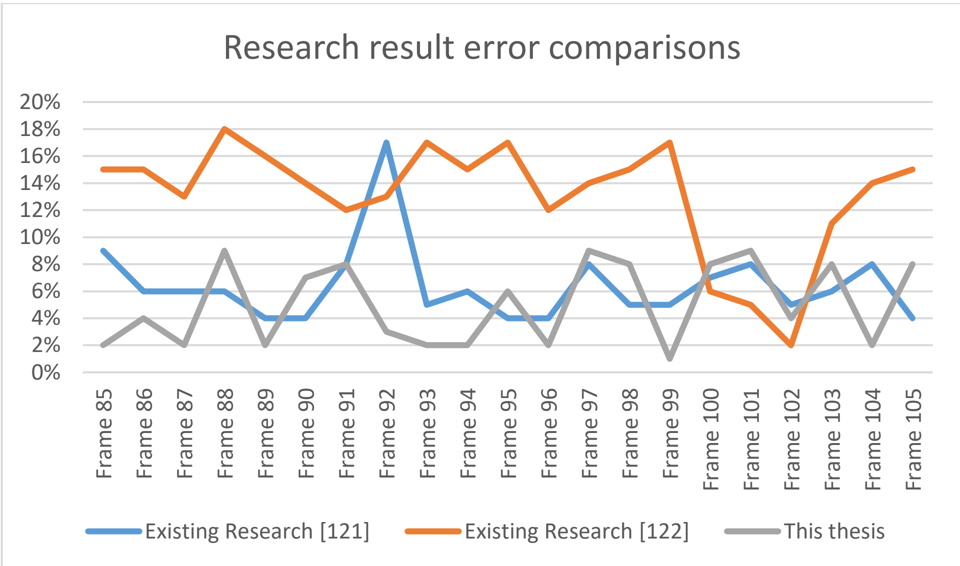


Figure 6-8: Graphical view of output accuracy when human object moves hands indoor environment

Below is the tabular result of the interesting frames and the resulting percentage of error among the two existing research frameworks as well as the proposed framework.

Table 6-4: Error percentage in results when human object is in indoor environment

	Existing Research [121]	Existing Research [122]	This thesis
Frame 85	9%	15%	2%
Frame 86	6%	15%	4%

Frame 87	6%	13%	2%
Frame 88	6%	18%	9%
Frame 89	4%	16%	2%
Frame 90	4%	14%	7%
Frame 91	8%	12%	8%
Frame 92	17%	13%	3%
Frame 93	5%	17%	2%
Frame 94	6%	15%	2%
Frame 95	4%	17%	6%
Frame 96	4%	12%	2%
Frame 97	8%	14%	9%
Frame 98	5%	15%	8%
Frame 99	5%	17%	1%
Frame 100	7%	6%	8%
Frame 101	8%	5%	9%
Frame 102	5%	2%	4%
Frame 103	6%	11%	8%
Frame 104	8%	14%	2%
Frame 105	4%	15%	8%

As part of the comparison ‘Existing Research [121]’ and the research of this thesis both produce good results. ‘Existing Research [121]’ generates good results in indoor environment. The current research does not depend upon light while it is more dependent on the clearance of body parts in the input video stream. ‘Existing Research [122]’ did not produce good results in this input video stream. However, once lighting was sufficient, it did produce good results due the fact that it is light dependent.

6.1.6.3. Case 3: Throwing item in outdoor environment

In this input video stream, human object is moving hands in outdoor environment and throwing items. Below is the Figure 6-9 with the screenshot of a frame of the input video stream.

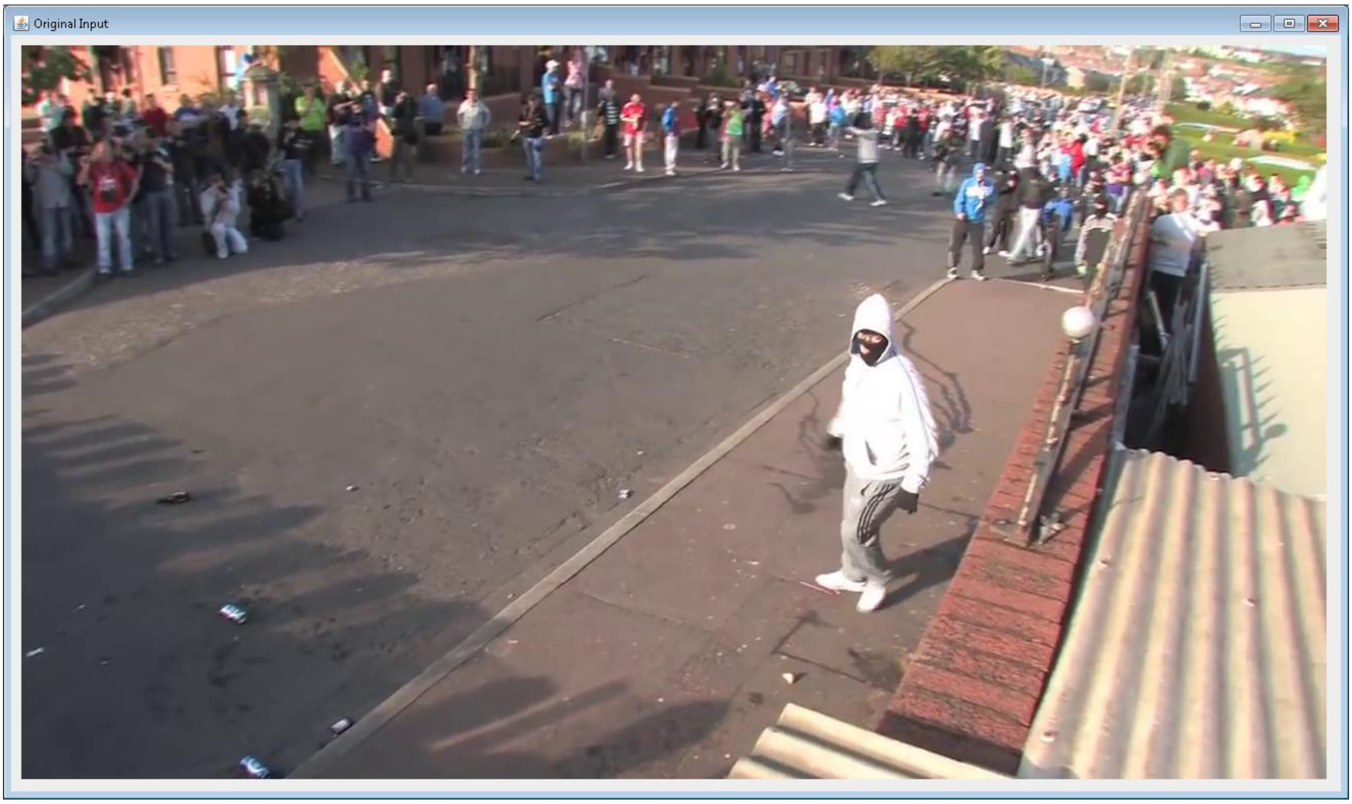


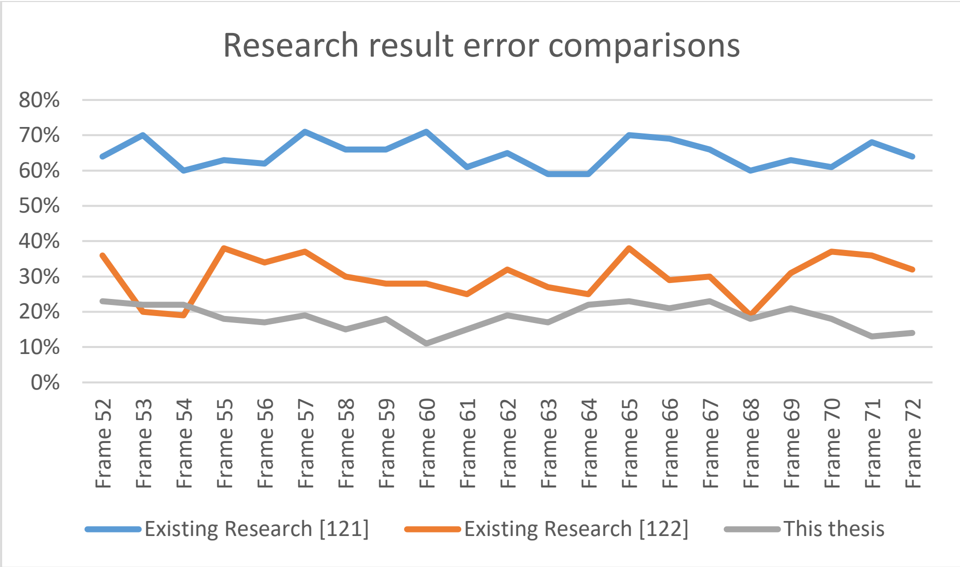
Figure 6-9: Frame of input video stream when human object throwing items

The result of interesting frames ranging from 52 to 72 is shown below.

Frames per second = 25

Total frames = 201

Interesting frames = from 52 to 72



The graph in

Figure 6-10 shows the results of the frames labelled as “interesting” and the different error values:

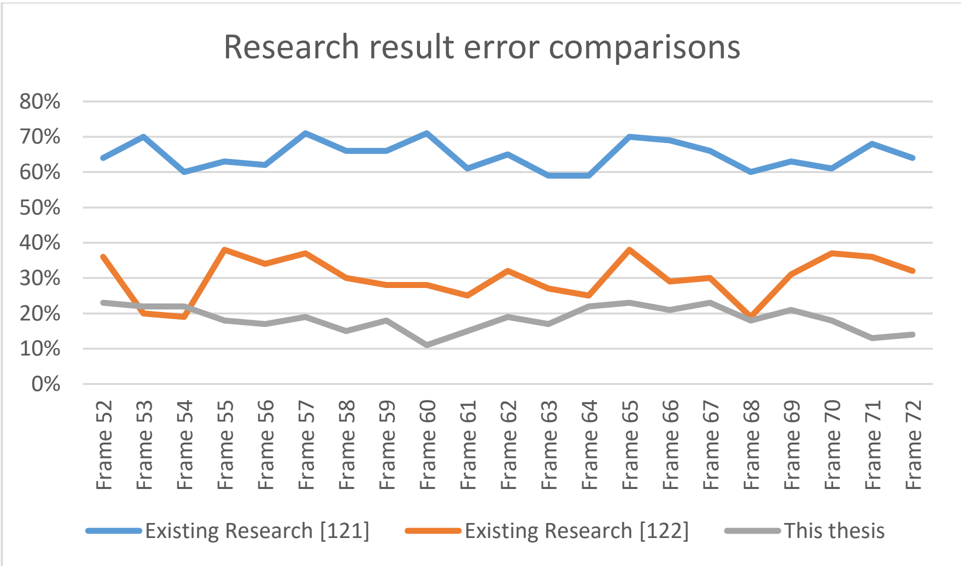

Research result error comparisons
80%
70%
60%
50%
40%
30%
20%
10%
0%
Frame 52
Frame 53
Frame 54
Frame 55
Frame 56
Frame 57
Frame 58
Frame 59
Frame 60
Frame 61
Frame 62
Frame 63
Frame 64
Frame 65
Frame 66
Frame 67
Frame 68
Frame 69
Frame 70
Frame 71
Frame 72
Existing Research [121]
Existing Research [122]
This thesis

Figure 6-10: Graphical view of output accuracy when human object throwing item

The tabular result of the interesting frames and the resulting percentage of error among the two existing research frameworks as well as the proposed framework is shown below.

Table 6-5: Error percentage in results when human object throwing items

	Existing Research [121]	Existing Research [122]	This thesis

Frame 52	64%	36%	23%
Frame 53	70%	20%	22%
Frame 54	60%	19%	22%
Frame 55	63%	38%	18%
Frame 56	62%	34%	17%
Frame 57	71%	37%	19%
Frame 58	66%	30%	15%
Frame 59	66%	28%	18%
Frame 60	71%	28%	11%
Frame 61	61%	25%	15%
Frame 62	65%	32%	19%
Frame 63	59%	27%	17%
Frame 64	59%	25%	22%
Frame 65	70%	38%	23%
Frame 66	69%	29%	21%
Frame 67	66%	30%	23%
Frame 68	60%	19%	18%
Frame 69	63%	31%	21%
Frame 70	61%	37%	18%
Frame 71	68%	36%	13%
Frame 72	64%	32%	14%

In this particular case, ‘Existing Research [121]’ and ‘Existing Research [122]’ did not produce good results. The results of the current investigation obviously produce better results as the shape of the human body parts is clearly visible. The research framework does not produce 100% accurate results due to the person wearing the same colour, making it difficult for the framework to determine the different moving parts.

6.1.6.4. Case 4: Person walking in indoor environment

In this input video stream (see the screenshot in Figure 6-11), the person is moving indoors, walking in a normal speed with the hands moving as well.

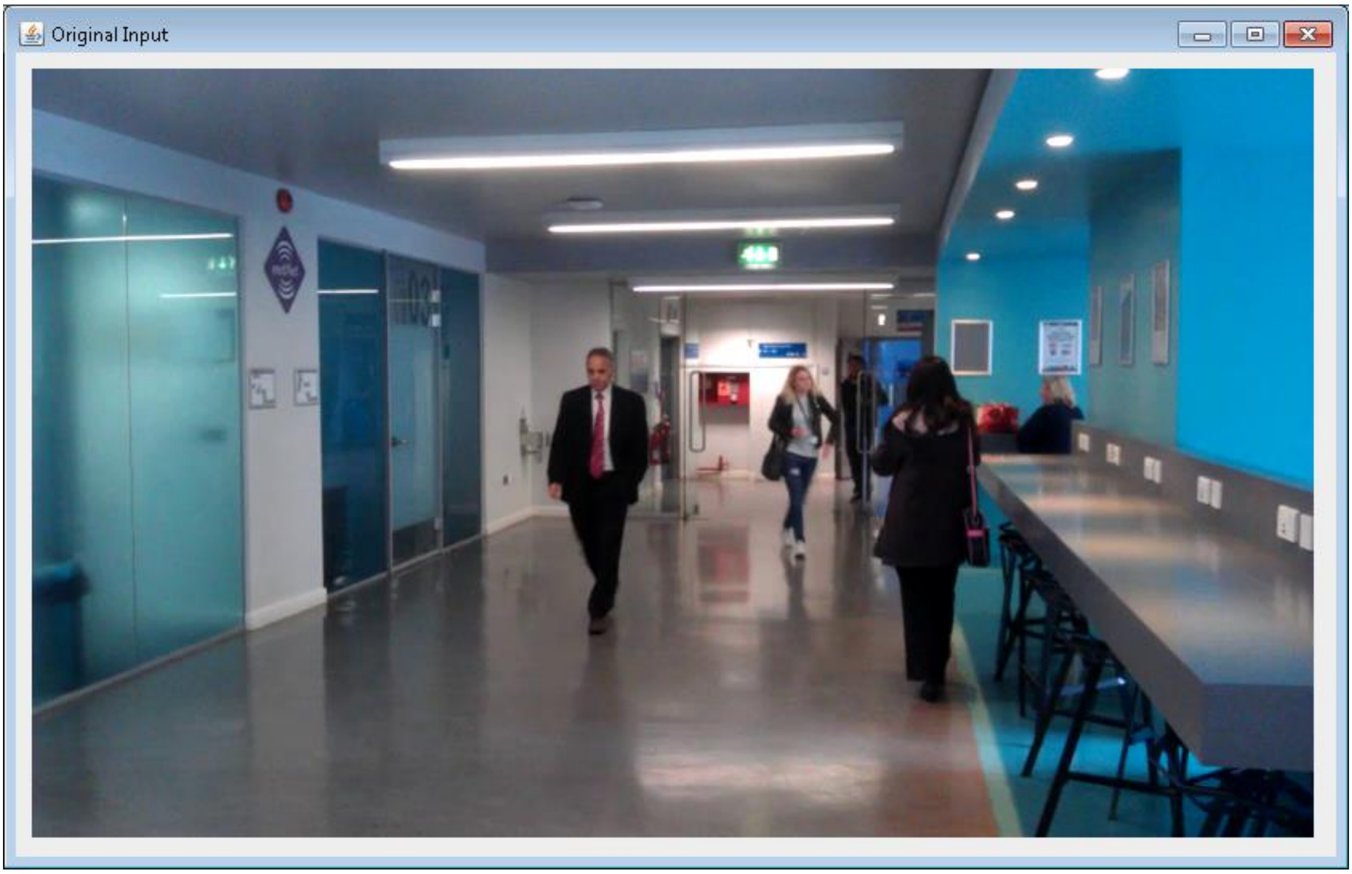


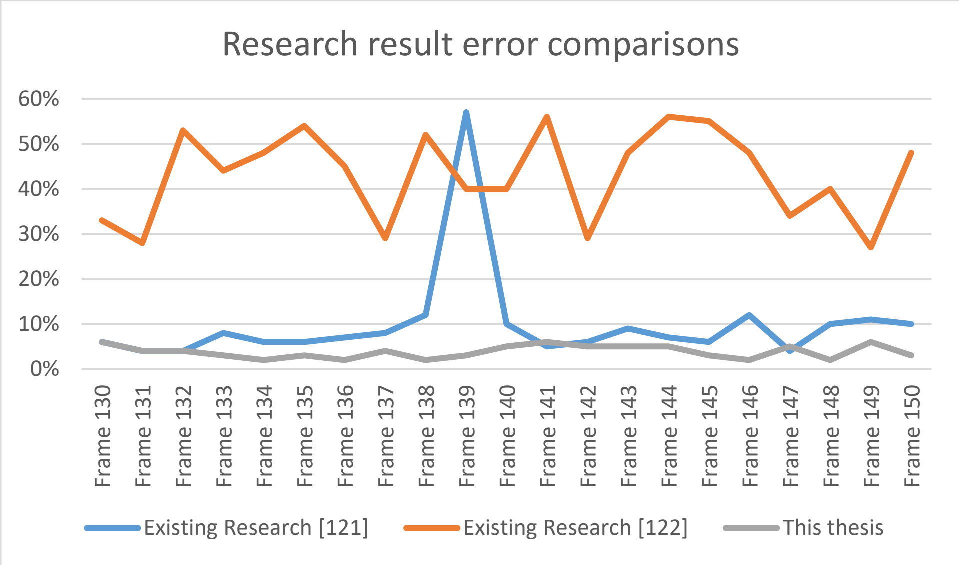
Figure 6-11: Frame of input video stream when human object is walking in indoor environment

Here is the result of interesting frames ranging from 130 to 150.

Frames per second = 19

Total frames = 170

Interesting frames = from 130 to 150



The graph in

Figure 6-12 shows the results of the frames labelled as "interesting" and the different error values.

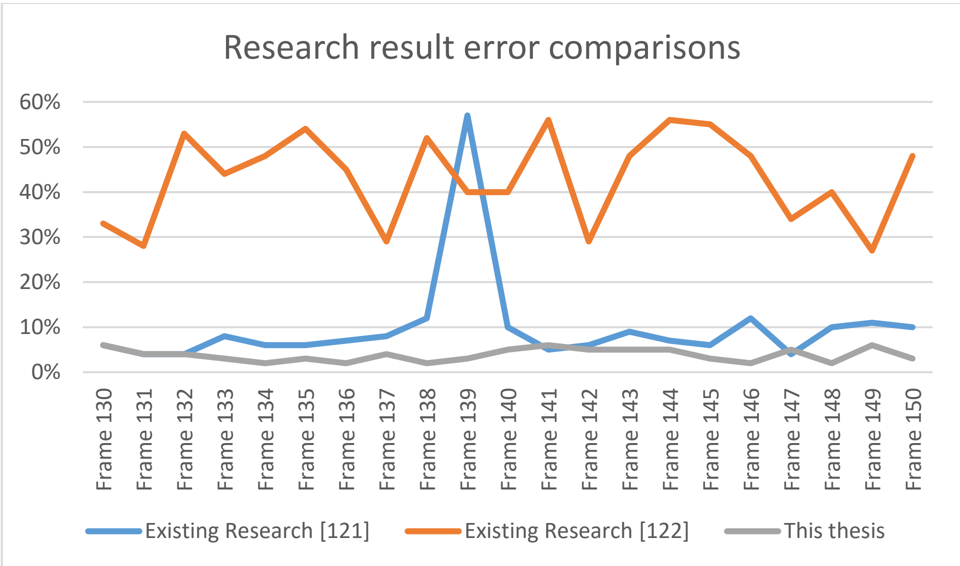


Figure 6-12: Graphical view of output accuracy when person walking in indoor environment

The tabular result of the interesting frames and the resulting percentage of error among the two existing research frameworks as well as the proposed framework is shown below.

Table 6-6: Error percentage in results when human object walking in indoor environment

	Existing Research [121]	Existing Research [122]	This thesis

Frame 130	6%	33%	6%
Frame 131	4%	28%	4%
Frame 132	4%	53%	4%
Frame 133	8%	44%	3%
Frame 134	6%	48%	2%
Frame 135	6%	54%	3%
Frame 136	7%	45%	2%
Frame 137	8%	29%	4%
Frame 138	12%	52%	2%
Frame 139	57%	40%	3%
Frame 140	10%	40%	5%
Frame 141	5%	56%	6%
Frame 142	6%	29%	5%
Frame 143	9%	48%	5%
Frame 144	7%	56%	5%
Frame 145	6%	55%	3%
Frame 146	12%	48%	2%
Frame 147	4%	34%	5%
Frame 148	10%	40%	2%
Frame 149	11%	27%	6%
Frame 150	10%	48%	3%

For Research [1], the results show that when light intensity increases, error values become significant. As a result, the framework in this research cannot properly track the object. However, when the amount of light is stable, the results are more accurate as the error values are lower. The framework proposed in this thesis is light independent, whether light intensity is high, balanced or low it does not affect the output. Framework [2] produces bad results (higher error values) when the amount of light is low and good results when it is balanced.

6.1.6.5. Case 5: Two persons are walking in outdoor environment

To test the framework as part of process, this video stream (see screenshot in Figure 6-13) captures two individuals walking outdoors, in a lit up area.

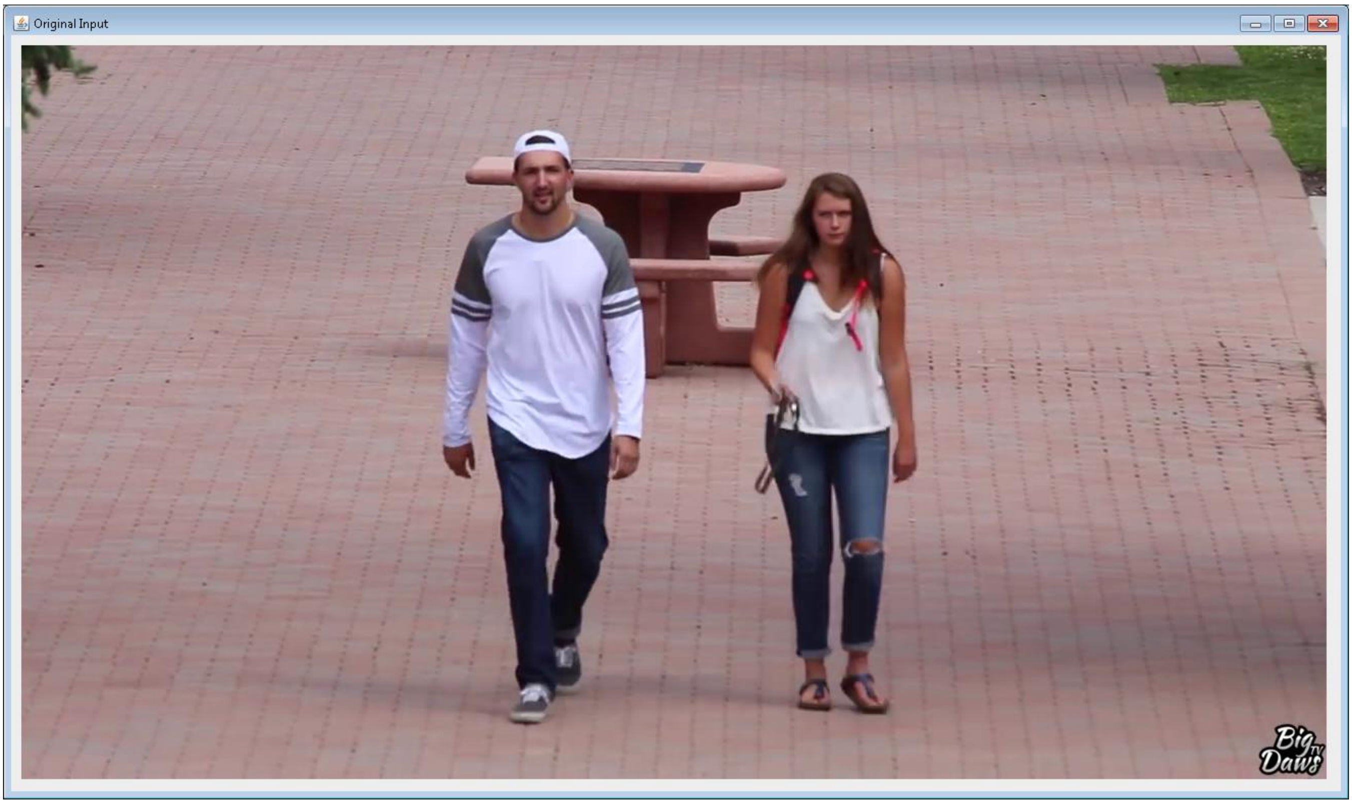


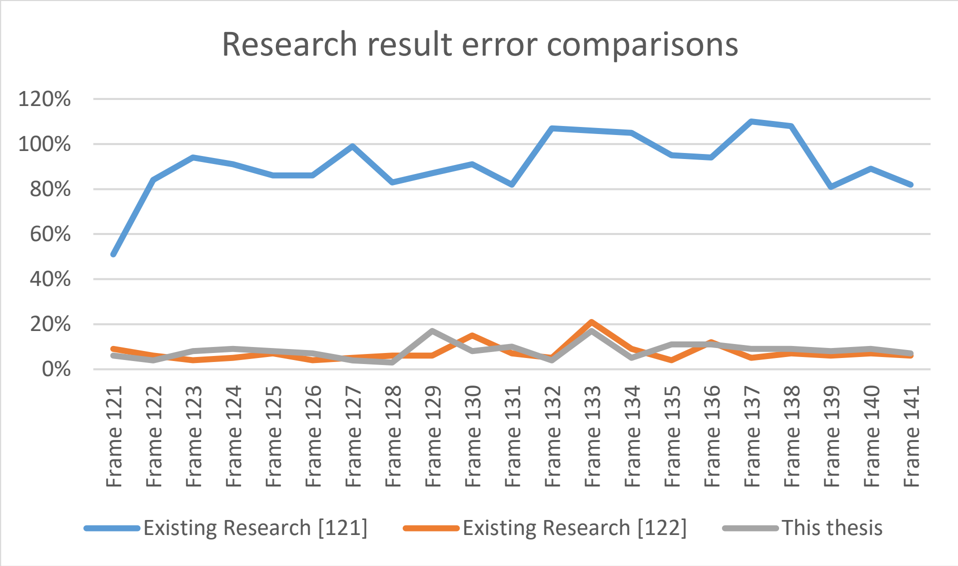
Figure 6-13: Frame of input video stream when two human object is walking in outdoor environment

The result of interesting frames ranging from 121 to 141 is as follows:

Frames per second = 23

Total frames = 383

Interesting frames = from 121 to 141



The graph in

Figure 6-14 displayed below shows the results of the frames labelled as "interesting" and the different error values.

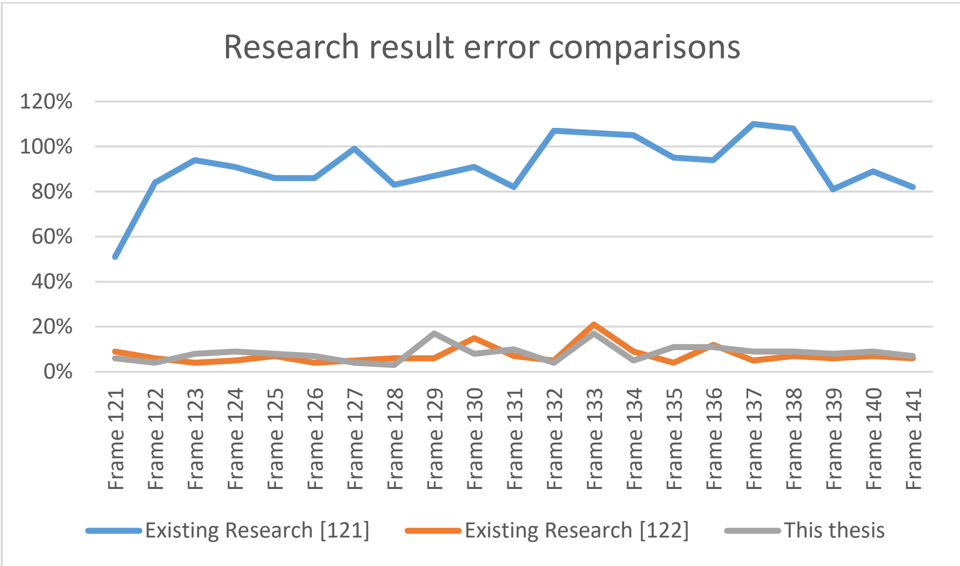


Figure 6-14: Graphical view of output accuracy in outdoor environment when two persons walking

The table below presents the tabular result of the interesting frames and the resulting percentage of error among the two existing research frameworks as well as the proposed framework.

Table 6-7: Error percentage in results when human objects are walking in outdoor environment

	Existing Research [121]	Existing Research [122]	This thesis
Frame 121	51%	9%	6%
Frame 122	84%	6%	4%

Frame 123	94%	4%	8%
Frame 124	91%	5%	9%
Frame 125	86%	7%	8%
Frame 126	86%	4%	7%
Frame 127	99%	5%	4%
Frame 128	83%	6%	3%
Frame 129	87%	6%	17%
Frame 130	91%	15%	8%
Frame 131	82%	7%	10%
Frame 132	107%	5%	4%
Frame 133	106%	21%	17%
Frame 134	105%	9%	5%
Frame 135	95%	4%	11%
Frame 136	94%	12%	11%
Frame 137	110%	5%	9%
Frame 138	108%	7%	9%
Frame 139	81%	6%	8%
Frame 140	89%	7%	9%
Frame 141	82%	6%	7%

As evidenced by the results, the framework presented in the second framework performs better as it was meant to be in an outdoor environment. From frame 131 to 136, the error values increase with the speed of the object's movements. In other words, the faster the object moves, the higher the error values. In the case of the framework presented by the first research, the performance is inadequate due to the presence of bright light in the environment.

6.1.6.6. Case 6: Person picking stuff up in an outdoor environment

In this input stream (see screenshot in Figure 6-15), a person is picking up an item outdoors and in a sunny environment.

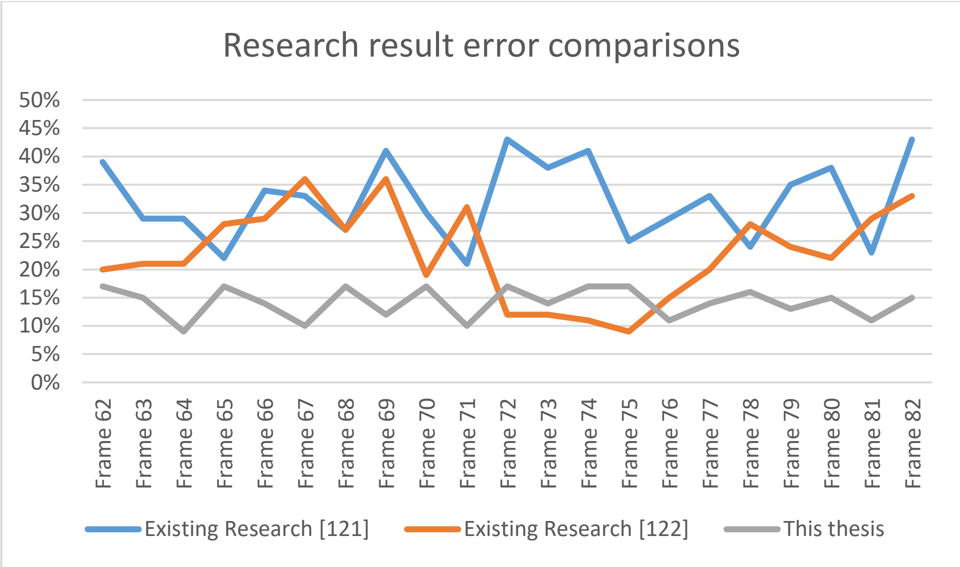
Figure 6-15: Frame of input video stream when one human object is picking up item in outdoor environment

Presented below is the result of interesting frames ranging from 62 to 82.

Frames per second = 25

Total frames = 279

Interesting frames = from 62 to 82



The graph in

Figure 6-16 shows the results of the frames labelled as "interesting" and the different error values.

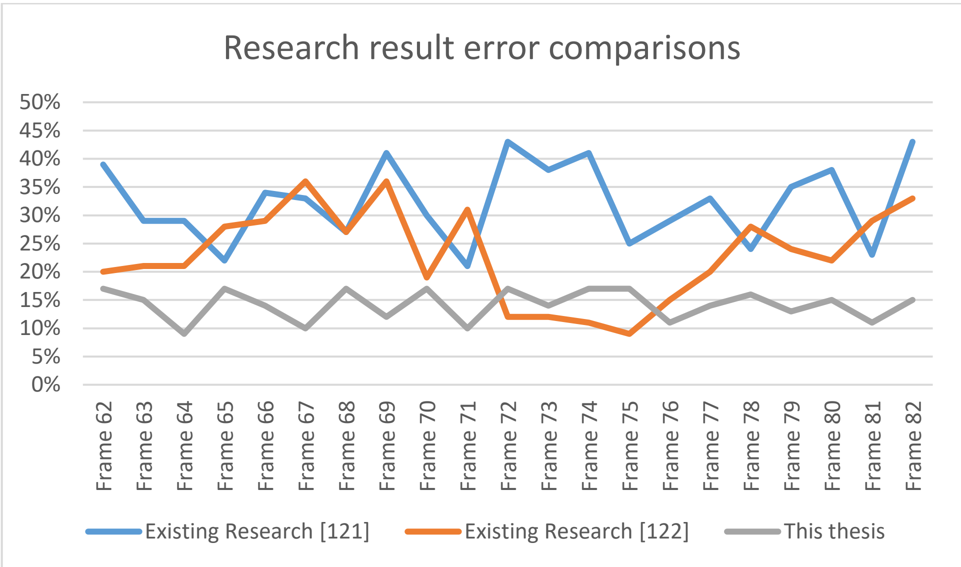


Figure 6-16: Graphical view of output accuracy when person picking up item in outdoor environment

Shown below is the tabular result of interesting frames and the percentage of error among three researches in the results.

Table 6-8: Error percentage in results when human object is picking up item in outdoor environment

	Existing Research [121]	Existing Research [122]	This thesis

Frame 62	39%	20%	17%
Frame 63	29%	21%	15%
Frame 64	29%	21%	9%
Frame 65	22%	28%	17%
Frame 66	34%	29%	14%
Frame 67	33%	36%	10%
Frame 68	27%	27%	17%
Frame 69	41%	36%	12%
Frame 70	30%	19%	17%
Frame 71	21%	31%	10%
Frame 72	43%	12%	17%
Frame 73	38%	12%	14%
Frame 74	41%	11%	17%
Frame 75	25%	9%	17%
Frame 76	29%	15%	11%
Frame 77	33%	20%	14%
Frame 78	24%	28%	16%
Frame 79	35%	24%	13%
Frame 80	38%	22%	15%
Frame 81	23%	29%	11%
Frame 82	43%	33%	15%

From frame 72 to 75, the person picks up the item at a slow speed so that the framework presented by the second research can detect the hand movement of the body. However, in the case of the first research, the framework is not able to detect the movement properly due to the bright sunlight.

6.1.7. Competition discovery

The variations of output of the framework depends on different elements of the environment under observation. These elements sometimes help in producing good results. However, these parameters can produce bad results in some situations such as in a sunny environment.

The results analysed above reveal that the following elements are the important elements to consider in the environment.

6.1.7.1. Environment lighting

Environment Lighting has a significant impact on the overall results or output of both, research 1 framework as well as that of the second. However, it is minimal in this research framework. Framework [1] works and produces good results in indoor environments with limited light in. On the other hand, framework [2] works and produces good results in outdoor environments under bright lights.

In both cases, the environment lighting factor has an impact on the results with varying rates.

6.1.7.2. Background

The environment background plays an important role in video analytics framework and the impact it can have has been documented in the tabular results. Results from the first framework are heavily impacted by the environment's background whereas the impact is limited in the case of the second framework. In terms of this research, the proposed framework is less dependent on the background of the scene. However, there is still a need for visible and clear body parts (i.e. skin colour is visible as well as the body parts).

6.1.7.3. Speed of moving object

Speed of a moving object is important for all three cases. Research [2] framework is not able to produce good results for fast moving objects but in the case of both, the first as well as this proposed framework, the results produced were accurate with slightly faster speed of movement. An ideal speed for both is an average walking speed of a human being.

6.2. Classifiers contribution

This research uses mainly three types of classifiers to identify the moving human body parts. Each classifier plays an important role in producing the overall results. In some instances (depending on the video stream), different classifiers play different important roles. For example, one classifier may play a more important role than the second or third and in other cases, it is a combination of classifiers that play an important role.

The following table shows the importance of each classifier:

Table 6-9: Framework classifiers contribution

	Sphere based model	Colour Classifier	Shape Classifier	Depth Classifier	Colour and Shape Classifiers	Shape and Depth Classifiers	Colour and Depth Classifiers	Three classifiers
Case 1	Moving object	Medium	Medium	Low	Close to high	Medium	Medium	High
Case 2	Moving object	Low	Close to high	Low	Close to high	Close to high	Low	High
Case 3	Moving object	Low	Close to high	Low	Close to high	Close to high	Low	Medium
Case 4	Moving object	Medium	Close to high	Low	Close to high	Close to high	Close to high	High
Case 5	Moving object	Medium	Close to high	Low	Close to high	Close to high	Close to high	High
Case 6	Moving object	Low	Low	Low	Medium	Medium	Medium	Medium

6.3. Parameters comparison

In this section, few parameters of the input video stream are being varied and the results are analysed.

6.3.1. Speed of moving object

The speed of the input video stream was increased by skipping frames in order to increase the speed of the moving object. This speed factor heavily influenced the trajectories' reconstruction results. These tests have been conducted based on the six types of input video stream cases mentioned above.

6.3.1.1. Case 1: Fast hand movement in outdoor sunlight environment

In this particular case, the person moves the body parts (the hands) with fast speed. When the frames are skipped, the speed of the movement of the hands increases. This is illustrated in the graphical view in the Figure 6-17 which demonstrates that when 4 frames are skipped, the framework begins to produce poor results.

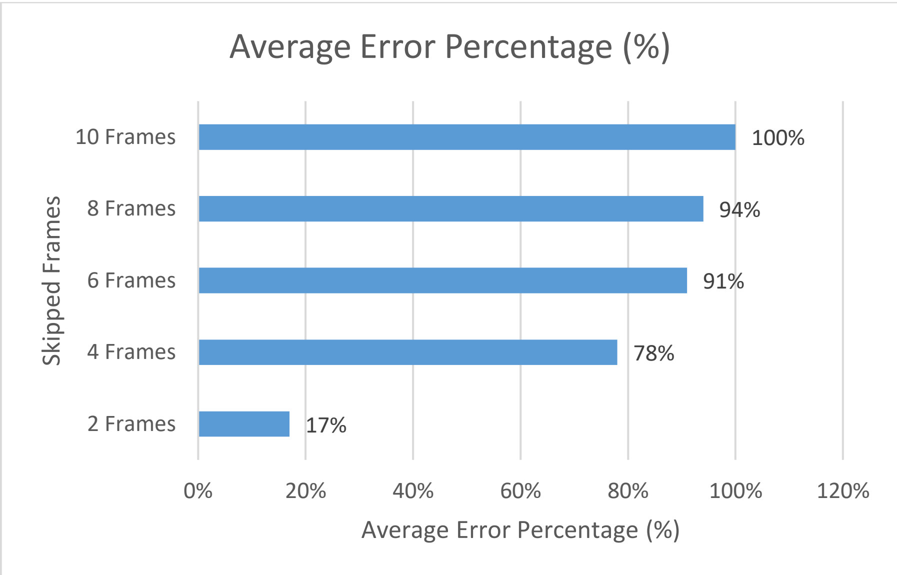


Figure 6-17: Graphical view of error percentage in results for outdoor environment when skipping frames

The tabular results below show error values in percentage caused by skipped frames during the reconstruction of trajectories.

Table 6-10: Error percentage due to skipped frames in results when human object is in outdoor environment

Skipped Frames	Average Error Percentage (%)
2 Frames	17%
4 Frames	78%

6 Frames	91%
8 Frames	94%
10 Frames	100%

6.3.1.2. Case 2: Playing music in indoor environment

In the video input stream (see Figure 6-7) the person's position is static and even though the body part, which in this case is the hand playing the guitar, moves at a slow speed the framework still detects it easily. This means, when skipping frames under such circumstances, the effect is lessened so long as the person does not change his position. In fact, this is the case even when 10 frames are skipped since there is no big movement involved. The only requirement in this particular case is that the video stream should be long enough to give the framework the chance to analyse the input data.

This is illustrated in the graphical view in Figure 6-18 which shows error values corresponding to the number of skipped frames.

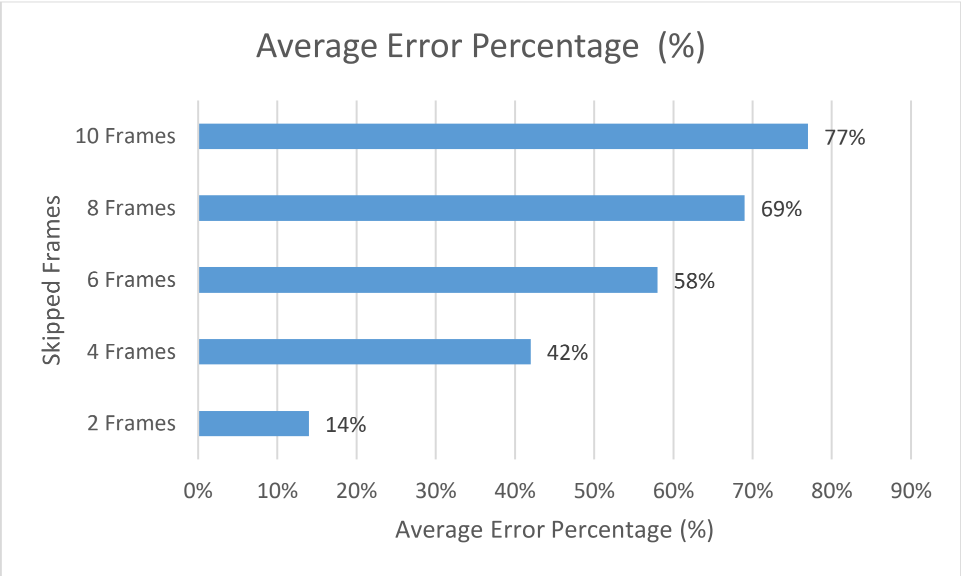


Figure 6-18: Graphical view of error percentage in results when human object moves hands indoor environment with skipping frames

The tabular results below show error values in percentage caused by skipped frames during the reconstruction of trajectories.

Table 6-11: Error percentage due to skipped frames in results when human object is in indoor environment

Skipped Frames	Average Error Percentage (%)
2 Frames	14%
4 Frames	42%
6 Frames	58%
8 Frames	69%
10 Frames	77%

6.3.1.3. Case 3: Throwing item in outdoor environment

In the input video stream shown in the Figure 6-9, the person's movement is fast and fluid (throwing object, going back and forth etc.) and if a few frames are skipped, the impact on the error values is significant. When 10 frames are skipped, the trajectory reconstruction is not possible. The chart in the Figure 6-20 shows the error values corresponding to the number of skipped frames.

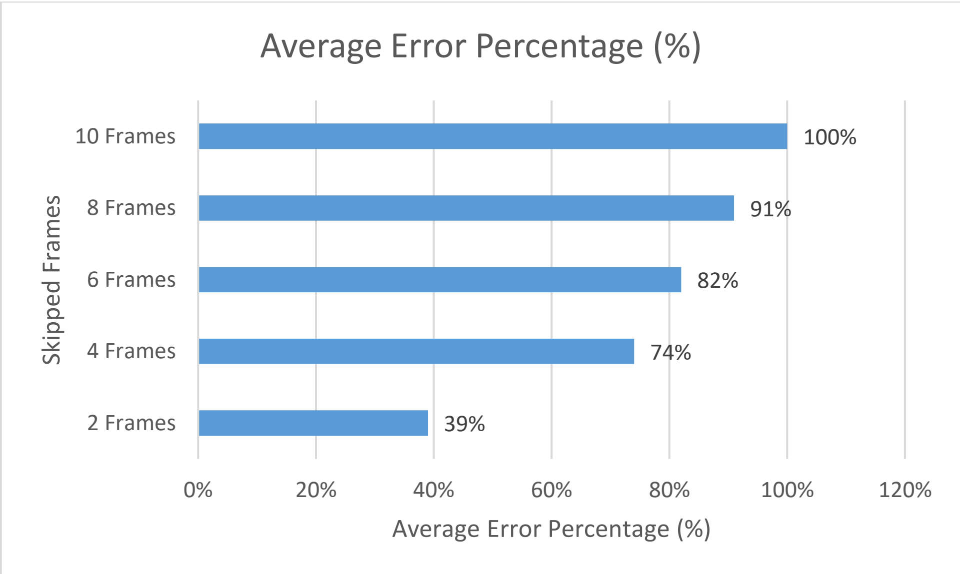


Figure 6-19: Graphical view of error percentage in results when human object throwing item and frames skipped

The tabular results below show error values in percentage caused by skipped frames during the reconstruction of trajectories of the person throwing the object.

Table 6-12: Error percentage due to skipped frames in results when human object throwing items

Skipped Frames	Average Error Percentage (%)
2 Frames	39%
4 Frames	74%
6 Frames	82%
8 Frames	91%
10 Frames	100%

6.3.1.4. Case 4: Person walking in indoor environment

The object in this video input stream is moving against static background and if the frames are skipped, the framework cannot clearly detect or track the position of the object at each frame.

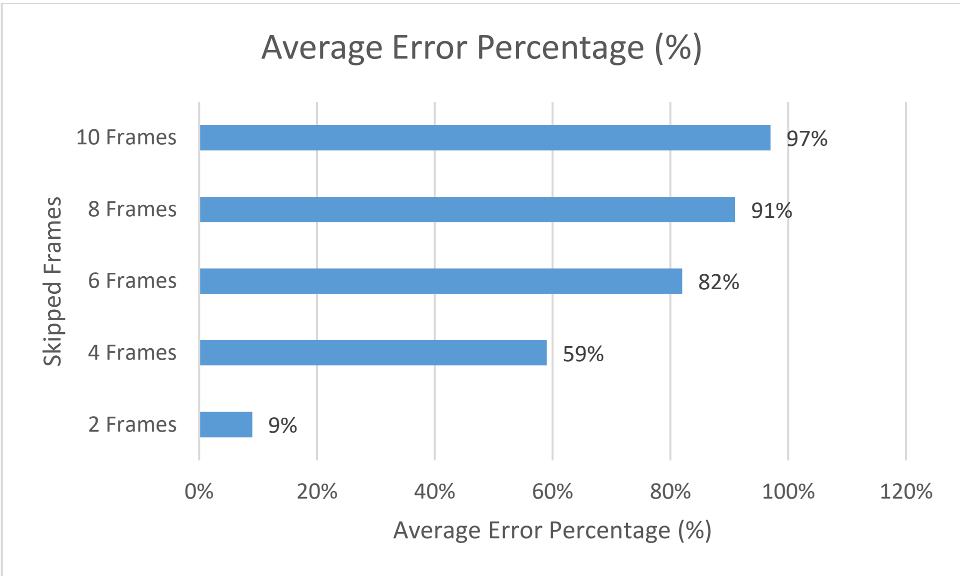


Figure 6-20: Graphical view of error percentage in results when person walking in indoor environment frames skipped

Shown below is the tabular result of skipped frames and the percentage of error values caused by these skipped frames in the results during the re-construction of trajectories of the moving human and their body parts.

Table 6-13: Error percentage due to skipped frames in results when human object walking in indoor environment

Skipped Frames	Average Error Percentage (%)

2 Frames	9%
4 Frames	59%
6 Frames	82%
8 Frames	91%
10 Frames	97%

6.3.1.5. Case 5: Two persons are walking in outdoor environment

In this case, the input video stream has two objects/individuals. Both are moving and the background is changing with respect to their positions. In the normal speed of frames and processing, the framework can handle and track the moving objects and their body parts.

During an average speed of movement, the framework can handle and track the two individuals and their body parts (hands, legs, torso etc.) whereas, when the frames are skipped, the framework is not able to track the movements. This is illustrated in the graphical view in the Figure 6-21 which shows when the frames are skipped, the framework begins to produce poor results.

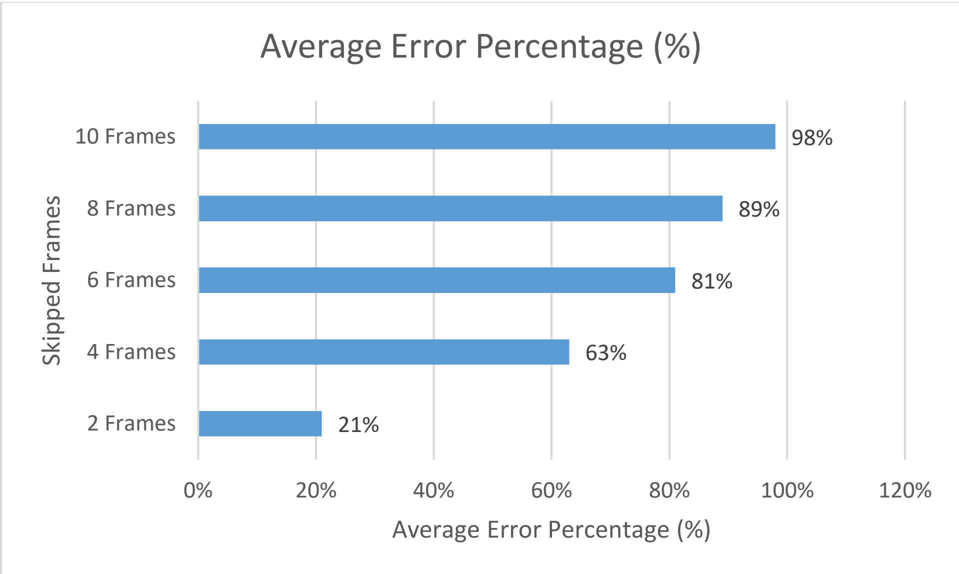


Figure 6-21: Graphical view of error percentage in results in outdoor environment when two persons walking with skipped frames

Shown below is the tabular result of skipped frames and the percentage of error in values caused by these skipped frames in the results of re-construction of trajectories of moving human object(s) and their parts.

Table 6-14: Error percentage due to skipped frames in results when human objects are walking in outdoor environment

Skipped Frames	Average Error Percentage (%)
2 Frames	21%
4 Frames	63%
6 Frames	81%
8 Frames	89%
10 Frames	98%

6.3.1.6. Case 6: Person picking up stuff in outdoor environment

This scenario is difficult for the framework since the face of the person is at some point hidden and non-detectable. This means eyes, nose, mouth and hair that the framework uses to build up and detect a person's head, only hair is visible and as such, detectable. When four or more frames are skipped, the framework is not able to create good results.

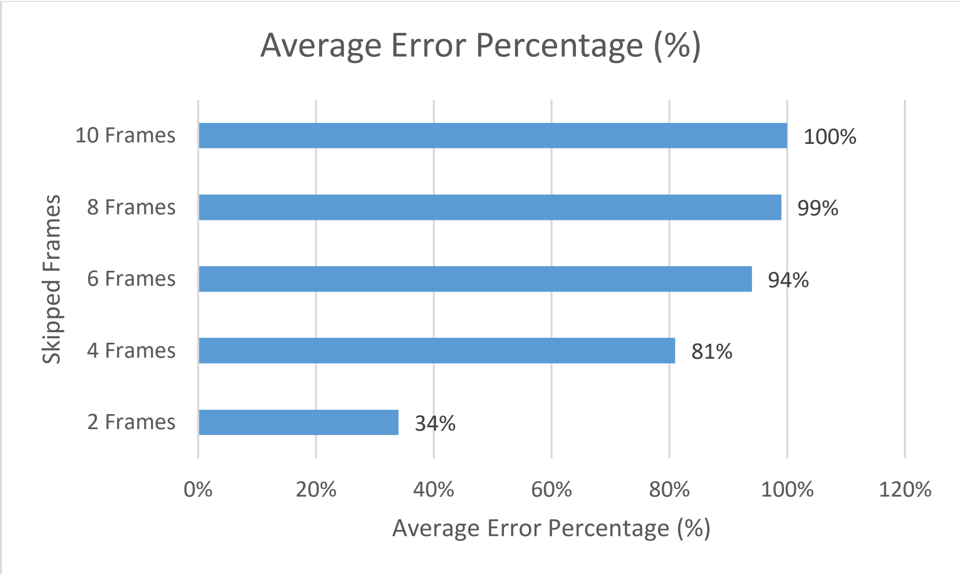


Figure 6-22: Graphical view of error percentage in results when person picking up item in outdoor environment and frames skipping

Illustrated below is the tabular results of skipped frames and the percentage of error in values caused by these skipped frames in the results of re-construction of trajectories of moving human object(s) and their parts.

Table 6-15: Error percentage due to skipped frames in results when human object is picking up item in outdoor environment

Skipped Frames	Average Error Percentage (%)
2 Frames	34%
4 Frames	81%
6 Frames	94%
8 Frames	99%
10 Frames	100%

6.3.2. Light condition in the environment / Environment Lighting Condition

Below is a list of factors that can affect the results of the framework's trajectory reconstruction.

- Speed of moving object (relative to speed to meter)
- Light condition in the environment
- Brightness (wavelength of the spectrum)
- Black/White video
- Colour of moving object and background

6.4. Dataset comparison

In this section, a comparison of the investigated framework with other existing frameworks is performed. Simulated experiments is carried out to demonstrate the advantage of the proposed classifier-based framework of reconstruction of trajectories in comparison with other three approaches namely CEMMT [134], DCOMT [135] and KSP [136]. To evaluate the performance of different approaches, two most commonly used datasets PETS S2L1 and PETS S3MF1 are selected and used. These datasets have different challenges, such as occlusion, people with same colour of clothing, pose changes and exit and entry of a scene.

To compare the multi object tracking algorithms, the CLEAR metrics [137] was adopted which is the most widely used protocol for quantitative evaluation. The different measures for comparison in this benchmark are as follows:

- **Groundtruth (GT)**: The number of trajectories in the groundtruth.

- **Mostly tracked trajectories (MT)**: The percentage of trajectories that are successfully tracked for more than 80 percent divided by ground truth.
- **Mostly lost trajectories (ML)**: The ratio of mostly lost trajectories, which are successfully tracked for less than 20 percent.
- **Partially tracked trajectories (PT)**: The ratio of partially tracked trajectories.
- **ID switches (IDS)**: The total number of times that a tracked trajectory changes its matched groundtruth identity.
- **Recall (Rec.)**: The number of correctly matched detections divided by the total number of detections in groundtruth.
- **Precision (Prec.)**: The number of correctly matched detections divided by the number of output detections.
- **Multi-Object Tracking Accuracy (MOTA)**: A measure of tracking accuracy that takes into consideration, false positive, false negatives and ID switches.
- **Multi-Object Tracking Precision (MOTP)**: This measures the position of objects in experimental results with the actual dataset.

6.4.1. Quantitative evaluation

6.4.1.1. PETS S2L1 dataset

The table below shows the experiment comparison values of PETS S2L1 dataset.

Table 6-16: Comparison values of PETS S2L1

Methods	Comparison Values							
	Rec.	Prec.	GT	MT	ML	IDs	MOTA	MOTP
CEMMT [134]	94.2	98.4	23	21	1	11	90.6	80.2
DCOMT [135]	90	98.7	23	19	0	18	88.3	79.6
Our research	85.9	97.6	23	6	0	2	82.6	90.1

This is a difficult dataset as it has 794 frames. Moving objects (mainly people) in the dataset are wearing the same colour cloths. Dataset has three different backgrounds house, grass and street. As shown in the above table, this dataset is used with different object tracking algorithms:

- CEMMT [134] generates multiple few hypothesis for each detection and selecting those which have minimized energy, in this way moving object tracking is the minimization of continuous energy.
- DCOMT [135] a simple closed form solution is used as continuous fitting problem for trajectory estimation.

The approach adopted in this research outperforms the other methods in terms of ID switches and MOTP. CEMMT [134] obtained the best results in terms of Recall (94.2), MT (21) and MOTA (90.6) but has more ID switches than the method used for this research. The best precision (98.7) value is obtained by DCOMT [135]. The Figure 6-23 shows the comparison of the values obtained by using different methods during the experiments:

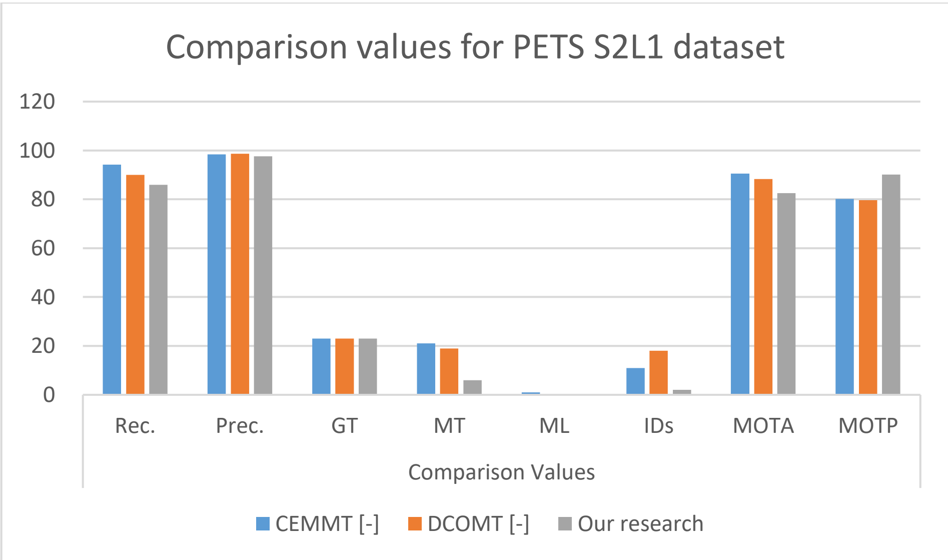


Figure 6-23: Comparison of the values for PETS S2L1 dataset (Y-Axis is showing the percentage)

It is clear from the above graphs that DCOMT [135] has high number of ID switching while the approach used in this research has low ID switching. This approach also outperforms MOTP.

6.4.1.2. PETS S3MF1 dataset

The table below shows a comparison of the experimental values of PETS S3MF1 dataset.

The table shows the resulting values from the experimental comparison of PETS S3MF1 dataset.

Table 6-17: Comparison values of PETS S3MF1

Methods	Comparison Values							
	Rec.	Prec.	GT	MT	ML	IDs	MOTA	MOTP
CEMMT [134]	97.7	99.4	7	7	0	0	97.1	83.4
KSP [136]	87.9	95.4	7	6	1	0	83.7	77.8
Our research	96.8	98.7	7	7	0	0	95.6	94.7

This dataset has 107 frames and three different backgrounds namely a house, grass and a street which is similar to the previous dataset. Initially, the objects move in a uniform direction and then in random directions. As shown in the table above, this dataset is used with different object tracking algorithms in the same way as the previous table. The approach used in this research obtains the best results in terms of the accuracy of multi object tracking with a difference of 11.4 percent.

In comparison and as illustrated in the Figure 6-24, CEMMT [134] fairs better in recall (0.9 percent) and precision (0.7 percent).

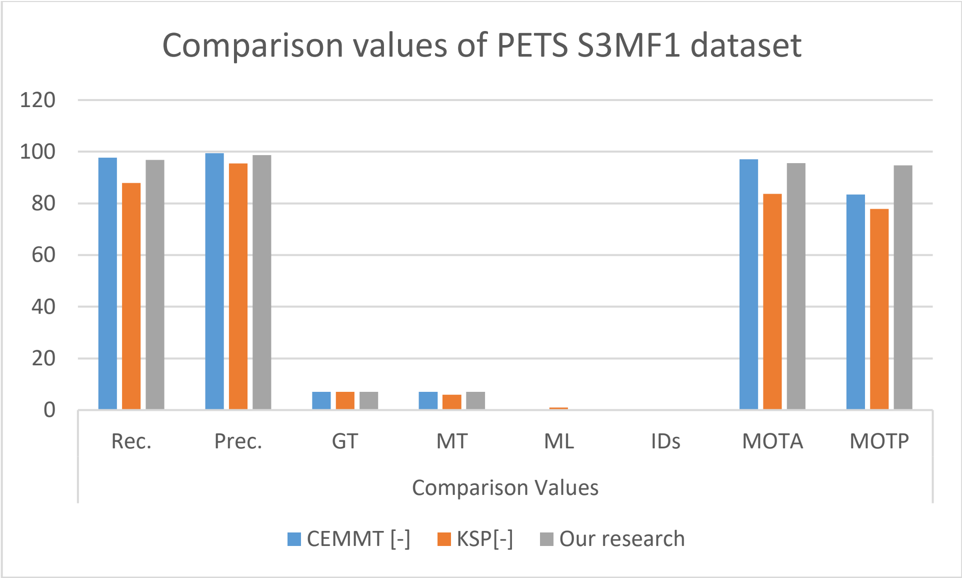


Figure 6-24: Comparison of the values for PETS S3MF1 dataset (Y-Axis is showing the percentage)

6.4.2. Qualitative evaluation

6.4.2.1. PETS S2L1 dataset

In this research, the framework developed is tested using PETS S2L1 dataset. The Figure 6-25 shows how the changing frames track several moving objects.

The Figure 6-25 shows three frames during the process of tracking two moving objects as shown in frame 290. At frame 295, the two individuals within the red frame are overlapping each other and the framework processes it as one object. The second object then walks away and is clearly separate from the first one and in this case, the framework detects and tracks both successfully.

Figure 6-25: PETS S2L1 dataset (frame number 290, 295 and 319)

In the Figure 6-25, the trajectory of objects with ID=9 and ID=1 occupy two different positions in frame 290. After five frames in frame 295 object with ID=9 covers object with ID=1. However, object with ID=1 does not lose its trajectory and there is no ID switch. Finally, in frame 319, object with ID=1 does not cover object with ID=9 anymore, its direction of movement has changed, and the trajectories split. This shows fewer ID switches even though the moving objects overlapped.

6.4.2.2. PETS S3MF1 dataset

Dataset PETS S3MF1 is applied to this research approach and the Figure 6-26 shows the tracking of new moving objects entering the scene.

Figure 6-26: PETS S3MF1 dataset (frame number 38 and 68)

The Figure 6-26 shows that ID=6 and ID=7 are entering in view in frame number 38.

Looking at frame 68, the framework is able to track both individuals in the highlighted box which shows that the method for this research allows for the tracking of not only existing objects in a frame but also handles and detects the trajectories of new objects entering the scene.

6.5. Conclusion

The methodology of experimental analysis is used in this chapter to compare and contrast the three result sets of the trajectory reconstruction. The comparison of results is of four types.

The first type of comparison is based on a comparison of the expected results against the achieved results of the developed framework as well as those of the previous two mentioned earlier. These experiments are performed on six different cases in order to cover all possible aspects and demonstrate the efficiency and accuracy of the three frameworks.

The second type of comparison is based on analysing the contribution of different classifiers for re-constructed trajectories' output, which shows the importance of difference classifiers in different cases. This type of comparison uses the same cases used in previous comparison and the results show that colour and shape classifiers are the most important for this framework.

In the third type of comparison, experiments are performed under different conditions by changing different values of the input stream. This has been in such a way that the relation between the different video spatial values could be established.

The frequency of the Input video stream frames, is changed by skipping a certain percentage of frames in order to examine the effect of computational power on different machines and the possibility of partial data loss before reaching as input video stream.

Finally, the fourth series of experiments focused on using standard datasets and comparing the results of the developed framework with those of other existing research which is carried out in order to investigate the framework on standard datasets.

7. Conclusion and recommendation for future work

This research presents an efficient model-driven approach for moving object trajectory reconstruction using classifiers which can be used for real-time video analytics. This approach has a number of advantages compared to other existing approaches e.g. [119][120].

Firstly, the use of classifiers makes the extraction of trajectory data easier and possible in real-time video stream. Secondly, trajectory data can be reconstructed using less information because of the simpler geometry which lowers the requirements for preliminary visual image processing. Thirdly, the reconstruction of the trajectory is more efficient because of the simpler approximation, which makes this approach preferable for real-time systems.

Finally, all object trajectory reconstruction algorithms are far simpler than the other algorithms (e.g. [119] and [120]) reviewed in the literature and as a result, the software which implements them becomes more compact, which allows for easy integration in other software for visual analytics.

7.1. Reflection on the research

Incremental re-construction of trajectories from input video stream is an area of extensive research in both, the research community and industry due to its wide applicability to various areas such as video surveillance and security, accidents and safety management, business customer insight and video analytics.

The literature provides countless examples of issues and ways of tackling the complexity of this challenging task but the problem still remains difficult. There are multiple factors which make the video analytics aspect difficult. Two major factors influence the real-time performance of the video analysis are the enormous volume of visual data, which has to be processed in real-time, and the need to combine video data processing with complex analytic as well as symbolic data processing. Researchers in the field usually concentrate on direct visual processing of full image data frames recorded by surveillance cameras and are typically concerned with identification of single actions of humans normally executed in controlled environments on the basis of statistical data. These approaches in combination with the latest technological advancement address a few of the issues of extracting critical visual features from image frames, but when it comes to more complex activities, it quickly becomes an overwhelming task. This research hypothesis combines partial information extracted from videos with some data-post processing to make this analysis swift, feasible and easy.

In the initial stages of this research, there were certain anticipations about the potential outcomes but were not certain and did not formulate an exact achievable goal. The formulated goal was to create a framework, which must be efficient and intelligent to create effective and accurate trajectories to the actual trajectories of moving human objects and their body parts.

However, taking into account the fact that the input data is a continuous video stream and the present concerns regarding the load and processing speed, there was uncertainty about which methods to choose and what the potential for success. This approach is still based on the data extracted from the provided input video stream information is minimal and therefore the framework reduces the need of performing complex data processing on the input video stream.

As this research advanced and development progressed, it became more and more apparent and certain that the initial assumptions were correct. The framework managed to detect the human object along with their body parts whilst in motion and extracted the positions near to real values in real-time.

This has a direct impact on this research and results showed that it achieved the required goal. Through extensive analyses, investigations and experimentations, in the end, the framework proved that it is possible to re-construct trajectories from different cases with different issues and variations like environments, colour, speed etc.

7.2. *Originality and contribution to the knowledge*

As a result of this detailed research, we have successfully implemented a framework of model-driven classifiers, based on incremental trajectory reconstruction of moving object and obtained very promising experimental results. The results prove the applicability of the framework to areas where video analytics is limited due to various reasons. The results are shown and presented in detail in the previous chapter. The resulting contributions can be summarized as follows:

- The simpler geometry lowers the requirements for preliminary visual image processing and makes the reconstruction of trajectories possible by using less information. During the analytical phase of this research and as a necessary step towards formulating a correct and efficient model of moving object, a basic model of the moving object was developed. This was used as a starting point of this research but as this research progressed, the model was further polished and became more information rich. This model defines key important aspects, relations and visual appearances as well as an event-driven model of a constraint dynamic world, which can be easily adopted in the conceptualisation and construction of various systems for analysing visual scenes.
- Particularly, this model is helpful for analysing multiple moving objects within a given scene allowing for the tracking and analysis automation of a group of people's movements which is the next, more advanced stage in video analytics. This model-based trajectory reconstruction approach was presented for the first time during "The First International Conference on Applications and Systems of Visual Paradigms, VISUAL 2016, Barcelona, Spain" [113]
- A self-contained, standalone framework was developed for the purposes of analysing as well as reconstructing trajectories of moving objects from input video stream which has the potential of applicability in other areas such as video analysis, video surveillance and security, accidents and safety management, business customer insight, and user experience enhancement. Using these classifiers made the extraction of the required data from video stream input easy and straightforward and it also has a number of advantages over other image processing techniques including but not limited to:
    - It reduces the requirements for processing large volume of video data in real time.

  - It creates the trajectory details with diverse level of granularity and is able to do post-processing on the available data to make the data more meaningful.

  The approach to using classifiers was introduced for the first time during "The Third International Conference on Applications and Systems of Visual Paradigms, VISUAL 2018, Venice, Italy" [114]

- As a by-product of this research, a sub-module was developed which can be used independently for video analytics and is useful for simply extracting information about moving objects and their moving parts from a video stream. This can be utilised as a separate software module which is pluggable into various systems, enhancing their ability to analyse the input stream data, such as intelligent cameras, robot planning systems, etc.
- This whole framework can be used as a Java library. The potential applicability of the framework is supported by its modular structure, which allows processing the information flow in a standard XML format, as well as a separate configuration file allowing for easy configuration. This reconstruction of trajectories feature can be used in conjunction with other features capable of producing the minimal information required for further analysis of the visual scene. Examples include, object identification, description and classification, location, directions and trajectories of movements. The framework was presented first during the World Congress in Computer Science in June 2016 in Las Vegas, USA - see Gasiorowski, Vassilev and Ouazzane (2016). A further extension of this whole framework is currently under development by other members of the Video Analytics research group at the Cyber Security Research Centre of London Metropolitan University and will be published in the future.
- During the development of this framework, the useful method was implemented for depth estimation of moving object in input video stream for ease of calculation. The usage of Pythagorean Theorem made it possible to establish meaningful relationships with other dimensions in the scene and as a result, to avoid complex calculations and difficult variations. Instead of analysing the moving object with complex values, framework is capable of processing input video stream and reconstruct trajectories with quicker speed thanks to the depth estimation method. This approach has provided a higher degree of precision in trajectory re-construction. This work was first introduced

in the publication "The First International Conference on Applications and Systems of Visual Paradigms, VISUAL 2016, Barcelona, Spain" [113]

- During this research, different logical solutions were formulated to handle the difficult and complex conditions present in video stream inputs. These solutions can be useful in other researches and can make the data processing easier. The following are examples of the mentioned solutions:
  - To reconstruct a continuous trajectory, a timeline history of the frames is used and saved in a simple maths matrix format.
  - To reduce the complexity of moving object positions in the scene, project the origin of the scene to the camera location and this makes all the calculations and positions relative to camera.
  - Usually, the methods and techniques used to estimate a location of an object or individual in a scene, the starting point is the top left of the scene. However, in order to reduce the complexity of estimating the location of an individual in the post processing stage, the framework estimates the location of the individual starting from the position of the camera (bottom centre) and works forward towards the individual’s location.

7.3. Framework Limitations

The framework produced in light of this research has a few limitations. At the same time there are some edge cases (cases of rare occurrences), which cause issues as well. All limitations are listed below.

7.3.1. Shadow of a moving object

The framework identifies the moving object and its parts with the help of shapes and colours. In some videos, an object can cast a shadow and when a shadow is present, it makes the scene complex. If the shadow is obvious (easily seen) then it affects the accuracy of trajectories.

7.3.2. Human object going from a static state to a moving state

Movements of a human object(s) and/or their body parts make the framework aware of their presence. However, when a human is initially in a fixed or static state (not in motion), the framework cannot generate any trajectory information but will start as soon as movement begins. This means that when an object first starts moving, the framework starts generating trajectories continuously until the object is either out of camera sight or stops moving. If movement is stopped, trajectory values continue generating but remain the same. However, when the object resumes movements within the scene, trajectories will be generated accordingly.

7.3.3. Classifiers require training

Like all other classifier-based frameworks, this framework uses classifiers and classifiers need training. This training is an extra step and requires additional resources.

7.3.4. Improved accuracy when human skin colour is present/detected

The presence of human body colour in the scene helps the framework in identifying the body parts. When the human body colour as well as all body parts are present, the framework produces very good results and as was proven in the previous chapter. Environments in which no human body colour is detectable, or present can sometimes produce less accurate results.

7.4. Recommendations for future development

This framework can be enhanced in several ways that can increase not only the accuracy of the human vs non-human object distinction but also the usability of the software in other projects. This section indicates the identified room for improvement and presents a few suggestions about the potential future work.

7.4.1. Multiple and potentially moving cameras

As future work, one improvement is to consider using multiple cameras distributed within the boundaries of the physical space being under surveillance.

It is an understandable improvement for extending the framework but would require changing of some of the underlying mathematical theory behind the positioning of moving objects to accommodate the relativity of the different viewing points. It would also require additional synchronisation of the analysis performed during the different stages of post processing and in the scene under observation.

7.4.2. Spatial details of the environment

At the moment, the framework is focused on only moving objects and ignores the static ones. It does not account for designated boundaries of the space under observation, such as walls and ceilings. This prototype implementation considers everything to be on one and the same floor. Considering the ceilings and the walls would allow for the recognition of more fine-grained details about moving objects, which involves traversing of partitioned spaces, such as a continuous walk along a corridor after leaving a room, or changing the floor of a building etc.

7.4.3. Enhancement of moving object model

Currently, framework is based on the seven-sphere shape-based model, which helps in performing fast and efficient calculations of locations, motion and directions. Furthermore, it artificially adds approximation of object shapes to allow capturing the parts of moving objects. In some cases, the framework gets the data by way of assumption rather than from real data. These limitations could be lifted with the introduction of more realistic physical shapes of the objects to account for their real dimensions with higher precision.

7.4.4. Implementation of an intelligent, human object tagging system

The current implementation identifies the moving human object with the help of movements and with the detection of presence of a few body parts. In order to make this process more efficient and robust, as such, it is essential to perform the updates and streamline this process by adding tags in the input video stream for which a separate adapter can be developed.

8. References

[1] D. Walther, U. Rutishauser, C. Koch, and P. Perona, On the usefulness of attention for object recognition. In Paletta, L., Tsotsos, J. K., Rome, E., and Humphreys, G. W., editors, Proc. of the 2nd international workshop on attention and performance in computational vision (WAPCV ’04), Conference: Prague, Czech Republic, 2004, pages 96–103.

[2] Pawel Gasiorowski, Individual and group dynamic behaviour patterns in bound spaces, School of Computing and Digital Media, London Metropolitan University, Dec. 2017.

[3] C. Stauffer and W. E. L. Grimson, “Learning patterns of activity using real-time tracking,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 8, pp. 747–757, Aug. 2000

[4] R. Fergus, P. Perona and A. Zisserman, “A sparse object category model for efficient learning and exhaustive recognition,” CVPR 2005.

[5] T. Thummanuntawat, W. Kumwilaisak, J. Chinrungrueng, "Unsupervised Object of Interest Discovery in Multi-view Video Sequence", International Conference on Advanced Communication Technology, IEEE, 2009, pp. 1622-1627

[6] M.V.Naquest, S.Ullman. "Object Recognition with Informative Features and Linear Classification," Proceedings of 9th International Conference on Computer Vision, 2003, pp. 281-288.

[7] H. Zhang, W. Gao, D. Zhao, “Object detection using spatial histogram features”, Image and Vision Computing, ACM Digital Library, April 2006, vol 24 no 4.

[8] Featured based object detection research, https://www.youtube.com/watch?v=BQ1W021iYEE [Last accessed: 20-04-2019]

[9] Shape, space and measures, http://www.counton.org/resources/ks3framework/pdfs/geometrical.pdf [Last accessed: 25-04-2019]

[10] Qiang Zhu et al, “Fast human detection using a cascade of histogram of oriented gradients,” CVPR 2006.

[11] K. Jafari-Khouzani and H.Soltanianzadeh, "Radon transform orientation estimation for rotation invariant texture Analysis," IEEE Trans. on Pattern Analysis and Machine Intelligence, Jun 2005, vol. 27 (6), pp.1004-1008.

[12] M. Ilke Sari, A. B. Suksmono, I. Iwut T, "Detection and reconstruction of geometric object based on line detection using Radon Transform", International conference on Electrical Engineering and Informatics, IEEE, 17 – 19 July, 2011,

[13] S.Z. Li, et al. "Statistical Learning of Multi-View Face Detection," Proc. of the 7th European Conf on Computer Vision, Springer Link, 2002, vol. 2353.

[14] H. Zhang, W. Gao, X. Chen, D. Zhao, "Learning Informative Features for Spatial Histogram-Based Object Detection”, International Joint Conference on Neural Networks, Montreal, Canada, 2005, pp 1806-1811

[15] M. Watanabe and S. K. Nayar. Telecentric optics for computational vision. In ECCV, 1996.

[16] Orthographic Projection, http://www.engineeringessentials.com/ege/ortho/ortho_page2.htm [Last accessed: 19-04-2019]

[17] L. A. Shepp and B. F. Logan, “The Fourier reconstruction of a head section,” IEEE Trans. Nucl. Sci., vol. NS-21, pp. 21-43, 1974

[18] D. G. Lowe, Three-Dimensional Object Recognition from Single Two-Dimensional Images in Artificial Intelligence, ACM Digital Library, 1987, Artificial Intelligence, pp 335-395

[19] M. Martínez, P. Mittrapiyanuruk, and A. C. Kak, “On combining graph-partitioning with non-parametric clustering for image segmentation,” Computer Vision and Image Understanding, vol. 95, July 2004, pp. 72-85

[20] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and Texture Analysis for Image Segmentation," International Journal of Computer Vision, vol. 43, June 2001, pp. 7-27

[21] M.Y. Choong, C.F. Liau, J. Mountstephens, M.S. Arifianto, and K.T.K. Teo, “Multistage Image Clustering and Segmentation with Normalised Cuts,” 3rd International Conference Intelligent Systems, Modelling and Simulation (ISMS), February 2012, pp. 362-367.

[22] M. Yeen Choong, Y. Kwong Chin, L. Angeline, K. Tze Kin Teo, "Image Segmentation via Normalised Cuts and Clustering Algorithm", IEEE International Conference on Control System, Computing and Engineering, 2012, pp 430-435

[23] J. Liu, F. Xu, Z. Huang, and Y. Li, "Adaptive Algorithm for Automated Polygonal Approximation of High Spatial Resolution Remote Sensing Imagery Segmentation Contours", IEEE transactions on geoscience and remote sensing, 2014, pp 1099-1106.

[24] Telecommunications and Imaging Laboratory - http://www2.ulg.ac.be/telecom/research/vibe/ [Last accessed: 26-04-2019]

[25] L. Jian-ping, Ji Bao Gold, Multi-resolution analysis based on morphological watershed algorithm for image segmentation, Research Gate, Computer, 2010.

[26] L. Xiying, W. Shuquan, W. Xiangmin, “Research of Medical Image Segmentation Based on Improved Watershed Algorithm”, Microelectronic Technology, 2003, 8(4): 39-42,.

[27] Chapter 5, Part III Segmentation: Region growing segmentation - http://user.engineering.uiowa.edu/~dip/lecture/segmentation3.html [Last accessed: 20-04-2019]

[28] L. Rongjie, S. Fengjing, L. Guanfeng, "An Agglomerative Hierarchical Clustering based High-Resolution Remote Sensing Image Segmentation Algorithm”, International Conference on Computer Science and Software Engineering, 2008, pp 403 - 406

[29] M. Kulkarni, K. Pingali, G. Ramanarayanan, B. Walter, P. Carribault, P. Chew, and K. Bala. Scheduling strategies for optimistic parallel execution of irregular programs. ACM Symposium on Parallelism in Algorithms and Architectures, June 2008

[30] D. Chi Y. Zhao, M, Li, Automatic Liver MR Image Segmentation with SelfOrganizing Map and Hierarchical Agglomerative Clustering Method, International Congress on Image and Signal Processing, 2010, pp 1333-1336

[31] S. Alpert, M. Galun, A. Brandt, Image Segmentation by Probabilistic Bottom-Up Aggregation and Cue Integration, IEEE transactions on pattern analysis and machine intelligence, 2012, pp 315 - 319

[32] Z. Zhang, "On the Optimization Criteria for Two-Frame Structure from Motion," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 7, pp. 717-729, July 1998.

[33] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision. Cambridge, 2000.

[34] J. Oliensis, Exact Two-Image Structure from Motion, IEEE transactions on pattern analysis and machine intelligence, 2002, pp 1618-1626

[35] L. Rila, A. G. Constantidies Image coding using data-dependent triangulation, IEEE, pp 531-541

[36] Z. Guan, B. Zhang, Y. Zhang, S. Zhang, F. Wang, "Delaunay triangulation-based localization scheme", 29th Chinese Control And Decision Conference (CCDC) IEEE, 1948-9447, Chongqing, China, 2017

[37] 2D to 3D Photo with Delaunay triangulation, http://www.neuroproductions.be/experiments/fle_delaunay_triangulation/ [Last accessed: 26-04-2019]

[38] Y. Li, J. Xu, R. Jacobs, "An integrated algorithm of incremental and Robust PCA," Proc. of IEEE International conference on image processing, Barcelona, Spain, 2003

[39] Bucak, S.S. ; Gunsel, B., "Video Content Representation by Incremental Non-Negative Matrix Factorization", Proc. of IEEE International Conference on Image Processing, 2007

[40] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In Vision Algorithms: Theory and Practice, pages 298–375, 2000.

[41] G. Klein and D. Murray. Parallel tracking and mapping on a camera phone. In ISMAR'09, October 2009

[42] J.-M. Frahm, P. Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building rome on a cloudless day. In ECCV'10, 2010

[43] P. Fua, Using Model-Driven Bundle-Adjustment to Model Heads from Raw Video Sequences. Proc. of IEEE International Conference on Computer Vision, 1999.

[44] H. Grabner, M. Grabner, and H. Bischof, “Real-time tracking via online boosting,” BMVC, 2006.

[45] K. Zhang, L. Zhang, and M. Yang, “Real-time compressive tracking,” ECCV, 2012

[46] H. Zhang, L. Quan, Real-Time Object Tracking with Generalized Part-Based Appearance Model and Structure-Constrained Motion Model, proc. of IEEE 22nd International Conference on Pattern Recognition (ICPR), 2014, pp 1224-1229

[47] Q. Su, “A Cancer Gene Selection Algorithm Based on the K-S Test and CFS”, Biomed Research International, 2017, pp. 1-6

[48] M. Morovvat and A. Osareh, “An Ensemble of Filters and Wrappers for Microarray Data Classification” Machine Learning and Applications: An International Journal (MLAIJ) Vol.3, No.2, June 2016.

[49] Y. Azdoud, A. Amine, N. Alioua, M. Rziza, "Pre collision detection system for pedestrian safety based on HOL", IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), 2015,

[50] P. Vasuki, S. Veluchamy, "Pedestrian detection for driver assistance systems", IEEE Fifth International Conference on Recent Trends in Information Technology, 2016

[51] J. Balcerek et al., "Video processing approach for supporting pedestrians in vehicle detection", Signal Processing; Algorithms, Architectures, Arrangements, and Applications (SPA), 2014, pp. 100-103

[52] C. Conaire, N. Connor, E. Cooke, A. Smeaton, Multispectral Object Segmentation and Retrieval in Surveillance Video, IEEE International Conference on Image Processing, 2006, PP. 2381-2384

[53] E. MuraliKrishnan, E. Gangadharan, "Enhanced Performance of H.264 Using FPGA Coprocessors in Video Surveillance", IEEE International Conference on Signal Acquisition and Processing, 2010, pp. 157-161.

[54] S. Kotsiantis, I. Zaharakis, P. Pintelas, "Supervised machine learning: A review of classification techniques", Emerging Artificial Intelligence Applications in Computer Engineering, 2007.

[55] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, Unsupervised learning. Springer, 2009.

[56] V. Naumovich, V. Vapnik. Statistical learning theory, volume 1. Wiley New York, 1998.
[57] B. Archambeault, J. Diepenbrock, "Quantifying the quality of agreement between simulation and validation data for multiple data sets", Asia-Pacific International Symposium on Electromagnetic Compatibility, IEEE, 2162-7673, Beijing, China, 2010
[58] Y. Hasija, N. Garg, S. Sourav, "Automated detection of dermatological disorders through image-processing and machine learning", IEEE International Conference on Intelligent Sustainable Systems, 2017, pp. 1047-1051
[59] F. Ghesu, E. Krubasik, B. Georgescu, "Marginal Space Deep Learning: Efficient Architecture for Volumetric Image Parsing", IEEE Transactions on Medical Imaging, 2016, pp. 1217-1228

[60] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu, "Four-chamber heart modeling and automatic segmentation for 3-D cardiac CT volumes using marginal space learning and steerable features", IEEE Trans. Med. Imag., vol. 27, no. 11, November, 2008, pp. 1668–1681
[61] S. Moorthi, N. Dube, D. Dhar, B.Kartikeyan and R.Ramakrishnan, "RESOURCESAT-1 (IRS-P6) value added data products," Journal of the Indian Society of Remote Sensing, 2005, 33 (2), pp. 277–283
[62] S. Moorthi, I. Misra, R. Kaur, N, Darji, R. Ramakrishnan, "Kernel based learning approach for satellite image classification using support vector machine", IEEE Recent Advances in Intelligent Computational Systems, 2011, pp. 107-110
[63] "ILWIS 3.0 User's Guide," Unit Geo Software Development Sector Remote Sensing & GIS IT Department International Institute for Aerospace Survey and Earth Sciences (ITC) Enschede, The Netherlands, May 2001.
[64] S. Steiniger and E. Bocher, "An Overview on Current Free and Open Source Desktop GIS Developments," International Journal of Geographical Information Science 2009, pp.1345-1370.
[65] LISS3 data set
https://www.usgs.gov/centers/eros/science/usgs-eros-archive-isro-resourcesat-1-and-resourcesat-2-liss-3?qt-science_center_objects=0#qt-science_center_objects [Last accessed: 26-04-2019]

[66] A. Boulmakoul, L. Karim, A. Elbouziri, and A. Lbath, A System Architecture for Heterogeneous Moving-Object Trajectory Metamodel Using Generic Sensors: Tracking Airport Security Case Study, proc of IEEE SYSTEMS JOURNAL, VOL. 9, NO. 1, MARCH 2015

[67] L. Zhang, X. Wen, W. Zheng, B. Wang , "An Algorithm for Moving Semantic Objects Trajectories Detection in Video". Proc. of IEEE International Conference on Information Theory and Information Security (ICITIS), 2010

[68] C. Shim and J. Chang, "Content-based Retrieval using Trajectories of Moving Objects in Video Databases", Proc. of the Seventh International Conference on Database Systems for Advanced Applications (DASFAA í01)

[69] J. Z. Li, M. T. Ozsu, D. Szafron, "Modeling of Moving Objects in a Video Database", In Proceedings of the International Conference on Multimedia Computing and Systems, pp. 336-343 (1997).

[70] M. Shan and S. Lee, "Content-based Video Retrieval via Motion Trajectories", In Proceedings of the International Conference on SPIE, Vol. 3561, pp. 52-61 (1998).

[71] Z. Ji, Y. Su, R. Qian, J. Ma, "Surveillance video summarization based on moving object detection and trajectory extraction", proc of IEEE 2nd International Conference on Signal Processing Systems (ICSPS), 2010

[72] Robert T. Collins, Alan J.Lipton, Takeo Kanade, “A System for Video Surveillance and Monitoring”, Pittsburgh, cmu, 2000

[73] S. Vahora, N. Chauhan, N. Prajapati, A Robust Method for Moving Object Detection Using Modified Statistical Mean Method, proc. of International Journal of Advanced Information Technology (IJAIT) Vol. 2, February 2012

[74] In Su Kim, Hong Seok Choi, Kwang Moo Yi, Jin Young Choi, and Seong G. Kong. : Intelligent Visual Surveillance - A Survey. International Journal of Control, Automation, and Systems (2010) 8(5):926-939

[75] Tanaka, Keiji, “Mechanisms of visual object recognition: monkey and human studies,” Current Opinion in Neurobiology, 7 (1997), pp. 523–529.

[76] 5. S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA, 1998: 232- 237

[77] Centre for Biomedical imagine and informatics - http://pleiad.umdnj.edu/CBII/rad.html [Last accessed: 19-04-2019]

[78] A. M. Okamura, C. Simone, and M. D. O'Leary, "Force modeling for needle insertion into soft tissue," IEEE Trans. Biomed. Eng., vol. 51, no. 10, pp. 1707-1716, Oct. 2004.

[79] R. Alterovitz and K. Goldberg, "Motion Planning in Medicine: Optimization and Simulation Algorithms for Image-Guided Procedures," Springer-Verlag, Berlin Heidelberg, 2008.

[80] S. P. DiMaio and S. E. Salcudean, "Needle steering and motion planning in soft tissues," IEEE Trans. Biomed. Eng., vol. 52, no. 7, pp. 965-974, Jul. 2005.

[81] K. B. Reed, V. Kallem, R. Alterovitz, K. Goldberg, A. M. Okamura, and N. J. Cowan. "Integrated planning and image-guided control for planar needle steering. In Proc. IEEE/RAS-EMBS Intl. Conf. on Biomed. Robotics and Biomechatronics, Arizona, USA, pp. 819-824, 2008.

[82] N. Sadati, M. Torabi, R. Vaziri, R. Dehestani-Ardekani, Soft-Tissue Modeling and Image-Guided Control of Steerable Needles, proc of IEEE Annual International Conference, 2009, pp 5121 - 5125

[83] Middleware for Large Scale Surveillance, http://www-01.ibm.com/support/knowledgecenter/SS88XH_1.6.0/iva/ov_mils.dita [Last accessed: 20-04-2019]

[84] Features of Intelligent Video Analytics, http://www-01.ibm.com/support/knowledgecenter/SS88XH_1.6.0/iva/ov_features.dita?lang=en [Last accessed: 20-04-2019]

[85] E. K. Frentzos, "Trajectory data management in moving object databases," Ph.D. dissertation, Univ. Piraeus, Dept. Informat., Piraeus, Greece, 2008.

[86] L. Chen, L. Mingqi, and G. Chen, "A system for destination and future route prediction based on trajectory mining," Pervasive Mobile Comput., vol. 6, no. 6, pp. 657–676, Dec. 2010

[87] Daily mail news - http://www.dailymail.co.uk/news/article-2619148/From-floor-tiles-slow-trolley-beer-nappies-We-supermarket-mind-games-make-spend-more.html [Last accessed: 22-04-2019]

[88] Minority Report-style advertising! Billboards could soon track Google Glass users' gaze to play 3D videos and movie trailers - http://www.dailymail.co.uk/sciencetech/article-

2568379/Minority-Report-style-advertising-Magazines-billboards-soon-track-Google-Glass-users-gaze-play-3D-videos.html [Last accessed: 22-04-2019]

[89] J. Vermaak and P. Perez. Monte carlo filtering for multi-target tracking and data association. IEEE Transaction on Aerospace and Electronic Systems, 41(1):309–332, 2005.

[90] Peter Galison , Image and Logic: A Material Culture of Microphysics, 1997, University of Chicago Press.

[91] John Michael Hammersley - http://www.statslab.cam.ac.uk/~grg/papers/jmh_biom.pdf [Last accessed: 20-04-2019]

[92] S.Birchfield, “Elliptical head tracking using intensity gradients and color histograms,” in Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1998, pp. 232–237.

[93] Z. Zhang, R. Sa, Y. Wang, "A Real Time Object Tracking Approach for Mobile Robot Visual Servo Control" pp 500 - 504

[94] Tony Barnard, Department of Mathematics, King’s College London, http://www.mth.kcl.ac.uk/staff/ad_barnard/S&E.pdf [Last accessed: 20-04-2019]

[95] Dennis G. Zill, Patrick D. Shanahan, A First Course in Complex Analysis with Applications, eights edition, 2004, McGraw Hill Higher Education

[96] Michiel Hazewinkel , Encyclopaedia of Mathematics, Volume 6, 2001, Kluwer Academic Publishers

[97] Vagin, Shadows Of The Circle: Conic Sections, Optimal Figures And Non-Euclidean Geometry, 1998, Wold Scientific Publishing cooperation.

[98] Wikipedia about Hyperbola - http://en.wikipedia.org/wiki/Hyperbola [Last accessed: 20-04-2019]

[99] Davis (Emeritus) Donald C. Benson Professor of Mathematics University of California, A Smoother Pebble: Mathematical Explorations, 2003, Oxford University Press.

[100] Source of the image - http://en.wikipedia.org/wiki/Hyperbola

[101] Rotation matrix - http://en.wikipedia.org/wiki/Rotation_matrix

[102] Affine Transformation - http://homepages.inf.ed.ac.uk/rbf/HIPR2/affine.htm [Last accessed: 21-04-2019]

[103] 3D Geometrical Transformations, http://www.cs.brandeis.edu/~cs155/Lecture_07_6.pdf [Last accessed: 20-04-2019]

[104] Axis-angle representation,

http://en.wikipedia.org/wiki/Axis%E2%80%93angle_representation [Last accessed: 20-04-2019]
[105] Quaternions and Rotations - https://www.cs.iastate.edu/~cs577/handouts/quaternion.pdf [Last accessed: 20-04-2019]
[106] R. Szeliski, “Computer Vision“ - Algorithms and Applications, Springer e-ISBN 978-1-84882-935-0 2010.
[107] Point operations - http://pippin.gimp.org/image_processing/chap_point.html
[108] Fundamental of digital image processing, http://homepages.inf.ed.ac.uk/rbf/BOOKS/VERNON/Chap004.pdf [Last accessed: 20-04-2019]
[109] Digital image processing transform, http://www.tutorialspoint.com/dip/Image_Transformations.htm
[110] Histogram equalization - http://homepages.inf.ed.ac.uk/rbf/HIPR2/histeq.htm
[111] Person walks https://www.youtube.com/watch?v=kc5ZwzXkTiI [Last accessed: 20-04-2019]
[112] Persons and briefcase - https://www.youtube.com/watch?v=IDPCxNV92zw [Last accessed: 20-04-2019]

[113] M. Afzal, K. Ouazzane, V. Vassilev and Y. Patel, "Incremental Reconstruction of Moving Object Trajectory", The First International Conference on Applications and Systems of Visual Paradigms, VISUAL 2016, Barcelona, Spain
[114] M. Afzal, K. Ouazzane and V. Vassilev, "K-Nearest Neighbours Based Classifiers for Moving Object Trajectories Reconstruction", The Third International Conference on Applications and Systems of Visual Paradigms, VISUAL 2018, Venice, Italy
[115] C. Schönauer, T. Pintaric and H. Kaufmann, "Full Body Interaction for Serious Games in Motor Rehabilitation", Proceedings of the 2nd Augmented Human International Conference, March 2011
[116] http://scikit-learn.org/stable/tutorial/machine_learning_map/ [Last accessed: 28-09-2017]
[117] P. S.W., "Phylogenetic analysis of Aspergillus species using DNA sequences from four loci." Mycologia 100 (2):205-226

[118] P. J. I. and R. and Samson, "Nomenclatural considerations in naming species of Aspergillus and its teleomorphs" Stud Mycol 59 (1):67-70

[119] A. Brand, L. A. F. Fernandes and E. Clua. A Method for Body Part Detection, Tracking and Pose Classification using RGB-D Images. International Conference on Computer Vision Theory and Applications, 367 - 377, 2014

[120] Y. Bo, H. Jiang. Scale and Rotation Invariant Approach to Tracking Human Body Part Regions in Videos. IEEE Conference on Computer Vision and Pattern Recognition, 1041 - 1047, 2013.

[121] A. Kar, "Skeletal tracking using Microsoft Kinect," Methodology, vol. 1, pp. 1–11, 2010, department of Computer Science and Engineering, IIT Kanpur.

[122] M. Tang, "Recognizing hand gestures with Microsoft's Kinect," Stanford University, Tech. Rep., 2011, department of Electrical Engineering, Stanford University.

[123] J. Pan, S. Chitta, and D. Manocha, "Probabilistic collision detection between noisy point clouds using robust classification," in International Symposium on Robotics Research (ISRR), 2011.

[124] Y. Cui and D. Stricker, "3d shape scanning with a kinect," in ACM SIGGRAPH 2011 posters, ser. SIGGRAPH '11. New York, NY, USA: ACM, 2011.

[125] G. Gordon, T. Darrell, M. Harville, and J. Woodfill, "Background estimation and removal based on range and color," in Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference, vol. 2, 1999.

[126] Y. Ivanov, A. Bobick, and J. Liu. Fast lighting independent background subtraction. International Journal of Computer Vision, vol. 37, no. 2, pp. 199–207, June 2000.

[127] L. Xia, C. C. Chen, and J. K. Aggarwal, "Human detection using depth information by Kinect," in International Workshop on Human Activity Understanding from 3D Data in conjunction with CVPR (HAU3D), Colorado Springs, CO, Jun. 2011.

[128] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun, "Real-time identification and localization of body parts from depth images," in Proceedings of the IEEE International Conference on Robotics & Automation (ICRA), Anchorage, Alaska, USA, 2010, pp. 3108–3113.

[129] E. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik, vol. 1, no. 1, pp. 269–271, 1959.

[130] V. Ganapathi, C. Plagemann, S. Thrun, and D. Koller, “Real time motion capture using a single time-of-flight camera,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 2010, pp. 755–762.

[131] A. Baak, M. M¨uller, G. Bharaj, H.-P. Seidel, and C. Theobalt, “A data-driven approach for real-time full body pose reconstruction from a depth camera,” in IEEE 13th International Conference on Computer Vision, Bacelona, Spain, Nov 2011, pp. 1092–1099.

[132] D. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, January 2004.

[133] I. Endres and D. Hoiem, “Category independent object proposals”, ECCV 2010.

[134] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multi-target tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 58-72 2014.

[135] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking", In Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'12), Providence, RI, USA, pp. 1926-1933, June 2012.

[136] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization", IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1806-1819, 2011.

[137] R. Stiefelhagen, K. Bernardin, R. Bowers, J.S. Garofolo, D. Mostefa, and P. Soundararajan, "The CLEAR 2006 Evaluation," In Proc. Int. Conf. Classification of Events, Activities and Relationships, 2006.