TRACKING POINTER AT ENDOSCOPIC IMAGES FOR TELEPOINTER REMOTE GUIDED

MUHAMAD HAMIZI ZAIDI BIN MOHD JONHANIS

B.ENG (HONS.) ELECTRICAL ENGINEERING (ELECTRONICS)

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : MUHAMAD HAMIZI ZAIDI BIN MOHD JONHANIS

Date of Birth : 5 APRIL 1997

Title : TRACKING POINTER AT ENDOSCOPIC IMAGES FOR TELEPOINTER REMOTE GUIDED

Academic Session : 2021/2022

I declare that this thesis is classified as:

- ☐ CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*
- ☐ RESTRICTED (Contains restricted information as specified by the organization where research was done)*
- ☑ OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature)	b/p: (Supervisor's Signature)
970405-23-5033	Ts. Dr. Rohana Binti Abdul Karim
New IC/Passport Number Date: 27 June 2022	Name of Supervisor Date: 27 June 2022

NOTE: * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name
Thesis Title

Reasons (i)

(ii)

(iii)

Thank you.

Yours faithfully,

b/p:

____
(Supervisor's Signature)

Date: 27 June 2022

Stamp:

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Electrical Engineering (Electronics).

b/p:

____

(Supervisor's Signature)

Full Name : Ts. Dr. Rohana Binti Abdul Karim

Position : Supervisor

Date : 27 June 2022

____

(Co-supervisor's Signature)

Full Name :

Position :

Date :

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

(Student's Signature)

Full Name : MUHAMAD HAMIZI ZAIDI BIN MOHD JONHANIS

ID Number : EA18008

Date : 27 JUNE 2022

TRACKING POINTER AT ENDOSCOPIC IMAGES FOR TELEPOINTER REMOTE GUIDED

MUHAMAD HAMIZI ZAIDI BIN MOHD JONHANIS

Thesis submitted in fulfillment of the requirements
for the award of the
B.Eng (Hons.) Electrical Engineering (Electronics)

College of Engineering
UNIVERSITI MALAYSIA PAHANG

JUNE 2022

ACKNOWLEDGEMENTS

First and foremost, thanks to Allah S.W.T, the Most Merciful and Beneficent, and peace be upon His Messenger in the name of Allah. Above all, I want to thank Him for blessing me with patience and tenacity of mind so that I could finish this research project thesis titled "Tracking Pointer at Endoscopic Images for Teleporter Remote Guided" and meet the requirements of Universiti Malaysia Pahang (UMP) and the College of Engineering. I am grateful for the opportunity to convey my gratitude to all who made this research project thesis possible.

Secondly. I really want like to thank and compliment my supervisor, Ts. Dr. Rohana Binti Abdul Karim who was always there whenever I need help. Despite the fact that she is always busy with her own commitments, she is still prepared to assist me with guidance and support to help me resolve my problems and doubts once this assignment is over. She is a wonderful supervisor who is prepared to give me her full attention. Without her continual care and wit, I would not have made any progress or comprehended the purpose of our research work. Thank you so much for your encouragement, criticism, and invaluable teachings.

Finally, I really would like to express my gratitude to my parents and family members for their spiritual and financial support throughout this research project. They always encourage me when I'm having difficulties and urge me to keep going till I finish this project successfully. Not to mention, I much appreciate all of the contributions whose articles and publications appear in my working papers. I owe my gratitude to all of the authors who have been of significant assistance to me. Thank you so much for everything.

ABSTRAK

Tanda tempat telepointer atau Telepointer Landmark (TL) telah menjadi alat penting untuk membimbing pakar bedah semasa pembedahan pada masa kini. Untuk terus mengesan dan menjejaki Telepointer Landmark (TL) pada tisu, pendekatan semasa seperti kaedah pemadanan templat gambar telah digunakan. Malangnya, disebabkan perubahan dalam ciri-ciri tisu seperti pergerakan berterusan, tekstur yang tidak tegar, homogen, dan pencahayaan yang berbeza-beza, ketepatan templat gambar yang dipadankan adalah tidak tepat. Beberapa pilihan kaedah lain diperlukan untuk mengekalkan aktiviti menjejaki objek sasaran berdasarkan aspek tisu tersebut. Masalah pertama ialah pengesanan kursor. Pakar bedah mungkin akan terlepas pandang lokasi kursor semasa pembedahan. Dalam kaedah tradisional, kursor sukar untuk dikesan dengan menggunakan mata kasar. Ini kerana warna kursor biasanya berwarna putih dan putih mempunyai kontras yang tidak baik dengan warna darah dan tisu jadi ia akan mengelirukan pakar dalam mengesan penunjuk atau kursor. Masalah kedua adalah pakar perlu menunjukkan kursor atau penunjuk ke lokasi tanda dan mengikuti pergerakan secara manual. Manipulasi kamera dan koordinasi antara pakar bedah dan pembantu boleh menjadi kompleks dan memerlukan kemahiran yang tinggi dari doktor dan pakar bedah dalam ruang bedah. Kajian ini bertujuan untuk mengesan dan menjejaki kursor sistem menggunakan teknologi penglihatan komputer. Oleh itu, matlamat pertama projek ini adalah untuk mengesan kursor hijau kerana warna kursor yang digunakan adalah warna hijau. Matlamat kedua adalah untuk mengesan lokasi tanda yang telah ditandakan oleh penunjuk atau kursor. Dalam projek ini, untuk pengesanan kursor, pengesanan warna yang dicadangkan untuk pengesanan kursor adalah ruang warna RGB, ruang warna HSV dan kluster k-means. Untuk penjejakan kursor, kaedah penjejakan kursor yang dicadangkan adalah pengesanan gumpalan (ciri warna) dan pengesanan titik ciri. Hasil eksperimen untuk sistem yang dicadangkan untuk pengesanan kursor menunjukkan bahawa HSV diserlahkan dengan aspek yang lebih positif daripada kaedah yang lain dan oleh itu lebih sesuai untuk pengesanan kursor warna hijau dan akan digunakan dalam kaedah pengesanan gumpalan (ciri warna). Untuk kaedah penjejakan kursor, hasil eksperimen menunjukkan bahawa sistem menggunakan pengesanan gumpalan (ciri warna) dan ciri -ciri pemadanan titik adalah satu kejayaan di mana sistem dapat mengesan dan menjejaki lokasi kursor atau tanda. Kedua -dua kaedah ini mempunyai ketepatan 100% untuk mengesan dan menjejaki kursor. Sistem masa depan yang dicadangkan untuk pengesanan dan pengesanan kursor harus mampu menggunakan algoritma yang dapat berfungsi dalam persekitaran yang lebih kompleks dan pelbagai jenis data.

ABSTRACT

Nowadays, the telepointer landmark (TL) has become a crucial tool for guiding surgeons during surgery. To keep detect and track of the telepointer landmark (TL) at tissues, the current approach such as Template Matching, is applied. Unfortunately, due to changes in tissue characteristics such as continuous movement, non-rigid, homogeneous texture, and varied illumination, the precision of the matched point is always bound in relying on a single hard decision. Several options are required to maintain track of actual activity based on the aspect of the tissues. The first problem is the detection of the cursor. Experts and surgeons are likely to overlook the cursor's location during surgery. In the traditional method, the pointer is difficult to detect with the naked eye. This is because the cursor's colour is usually white and white has bad contrast to blood and tissue colour so it will confuse the expert in detecting the pointer or cursor. The second problem is that an expert must point the cursor or pointer to the mark location and manually follow the movement. The manipulation of the camera and the coordination between surgeon and assistant can be complex and require excellent skills from the doctor and the surgeon on site. This study aims to detect and track the cursor a system using computer vision technology. Therefore, the first aim is to detect the green cursor or pointer since the colour of cursor used is green colour. The second aim is to track the mark location that the pointer or cursor has marked. In this project, for cursor detection, the proposed colour tracking for cursor detection is RGB Colour Space, HSV, Colour Space and K-Means Clustering. For cursor tracking, the proposed cursor tracking is Blob Detection (Colour Feature) and Feature Point Tracking. The experimental results for the proposed system cursor detection system show that HSV is highlighted with more positive aspects than other methods. Therefore, more suitable for green colour cursor detection and will be used in Blob Detection (Colour Feature) method. For the cursor tracking methods, the experimental results show that the system using Blob Detection (Colour Feature) and Features Point Matching is a success where the systems are able to detect and track the cursor or mark location. Both methods have 100% accuracy in detecting and tracking the cursor. The proposed future cursor detection and tracking system should be capable of applying the proposed algorithm in a more complex environment and various input data.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AR	Augmented Reality
BLOB	A binary large object
FAM	Fuzzy Associative Memory
FPM	Feature Point Matching
FYP 1	Final Year Project 1
FYP 2	Final Year Project 2
HCI	Human-Computer Interaction
HSV	Hue Saturation Value
KLT	Karhunen-Loève Transform
LBP	Local Binary Pattern
OR	Operating Room
PC	Personal Computer
PNCC	Partial Normalized Cross Correlation
RANSAC	Random Sample Consensus
RGB	Red Green Blue
SIFT	Scale-Invariant Feature Transform
TL	Telepointer landmark
TOB	Tissue of the body
VR	Virtual Reality
dHash	Difference Value Hash

CHAPTER 1

INTRODUCTION

1.1 Project Background

Long-distance quick and accurate communication has proven critical to human progress. Telemedicine is one of the disciplines that has benefited from the recent network and communications advancements, which have proven to be crucial in saving many lives.[1] With recent advancements in telemedicine technology, it is now possible to do closed-abdomen surgery on a remote patient's moving organ with a robotic instrument, reducing patient trauma and negative effects.[1] One of the leading Telemedicine technology is Telepointer landmark (TL). Telepointer landmark (TL) is an important medical tool used for guided inexperienced surgery regardless of the distance limitations. For a limited period, the landmark locations may be detected rapidly and precisely. Telepointer landmark (TL) is also known as medical endoscopy.

In recent decades, medical endoscopy has become a critical tool for minimally invasive diagnostics and minimally invasive surgery in the abdomen, joints, and other body parts in a range of body areas. Endoscopy is the tool that is used during endoscopic surgery. Endoscopy is a technique in which a long, thin tube is inserted into the body and used to inspect an internal organ or tissue in great detail. It can be used for various other things, including imaging and minor surgery. [2][3]

To keep track of the telepointer landmark (TL) at tissues, the current approach such as Template Matching is applied. Unfortunately, due to changes in tissues characteristics such as continuous movement, non-rigid, uniform texture, and varied illumination, the precision of the matched point is always locked in relying on a single hard decision. [1]

This study aims to detect and track the cursor during the endoscopic surgery using computer vision technology other than Template Matching. The involvement of computer vision technology can improve the conventional medical imaging method. Computer vision has been used in various healthcare applications to assist medical professionals in making better decisions regarding the treatment of patients. Medical imaging or medical image analysis is one method that creates a visualization of particular organs and tissues to enable a more accurate diagnosis especially in the tracking the tissues and internal organs (TDOD) movement, which considerably reduces the risk for the patients in surgery.

The method to be used and observed to detect the cursor in this paper is colour detection. There is 3 types of colour detection methods used to detect the cursor. It is RGB Colour Space, HSV Colour Space and K-Means Clustering. All the data obtained from these methods will be observed to measure and validate performance for any movement from the camera or organ itself. Next, the method to be used and observed to track the cursor in this paper is by using Blob Detection (Colour Feature) and Feature Point Tracking.

1.2 Problem Statement

Although this endoscopy technology has the potential to improve health care quality, concerns such as clinician attention, limited gesture information, and real-world applications must be addressed. The following are the problem statement of this research project:

a) The first problem is related to the cursor. Localization of object is one of the most important requirements for performing computer-assisted surgery in the operating room (OR). The typical shape of the cursor is a white pointing arrow. However, when it is used during the surgery, expert and surgeon tend to miss the location of the cursor. It is hard to detect the pointer by using bare of eyes. The movement of camera or the organ itself will confused the expert in detecting the pointer or cursor.[4] This is because the cursor's colour is usually white in colour and white has bad contrast to blood and tissue colour so it will confuse the expert in detecting the pointer or cursor.[5]

b) The second problem is that endoscopic surgery is more complex to perform for the surgeon and required experience. The manipulation of the camera and the coordination between the surgeon and the assistant can be complex and the images from the endoscopic camera images are not stable. The expert needs to manually point the cursor or pointer to the mark location and follow the movement manually. This is happening due to the movement of the camera or the movement of the organ itself.[6]

1.3 Objective

This research project has three goals to achieve. These goals must be specified to determine whether the guidelines have been followed and the goals have been fulfilled. This project focused on tracking the pointer at endoscopic images for telepointer remote guided with TDOD movement. The following are the objectives of this research project:

a) To detect the green cursor or pointer because it might not be in the right positon.

b) To track the mark location that has been marked by the pointer or cursor.

c) To measure and validate the performance of any movement from the camera or from the organ itself.

1.4 Scope of Project

This project covers a few different scopes of objects, which can be classified into primary scopes, which are the limits that this project must adhere to achieve its goal. The scope is divided into several categories collecting the data set in video, the type of dataset used to obtain the result for this project and the type of software used for this project. The following scopes are listed below:

a) This research project thesis will be carried out using MATLAB R2021b software.

b) All the dataset is taken from Hamlyn of Imperial Collage London website as requested by the supervisor.

c) There are three types of dataset used is rotation, zoom in zoom out and liver deforming.

d) Dataset type rotation is used to do initial analysis for detection and tracking.

1.5 Thesis Organization

There are five chapters and appendices in this thesis. This Chapter 1 briefly discusses the introduction and background of this project in general, the problem statement, the project's objectives, the project's scope and constraints, and the thesis organisation.

The next chapter, Literature Review, will analyse the article that was considered to be linked to the project's purpose and problem statement. There are two major concepts will be discussed: green colour pointer detection and cursor tracking. A review of the literature on a few algorithms utilised in the optimization research topic is also included.

In Chapter 3, the emphasis will be on project development. This chapter will go into detecting and tracking the cursors using MATLAB R2021b software. This chapter will also discuss the technique of detecting the green colour cursor and the process of tracking the cursor will also be discussed in this chapter. A flowchart has also been created for each cursor detection method and tracking method to describe the entire testing process.

In Chapter 4, the researcher presents the study's findings based on the information acquired as a result of the technique used. This chapter will also discuss and show the analysis of any information that has been collected, observed, generated, or created to validate original research findings for each method for green colour cursor detection and cursor tracking.

Chapter 5 summarises the research effort. It will present the project's findings as well as specific recommendations for future study to increase the efficiency and reliability of colour cursor identification and tracking using a better algorithm.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter briefly reviews the current endoscopic image detection, colour detection, object detection, and object tracking technology and research. The problem statement in Chapter 1 mentions limited research on tracking the pointer at endoscopic pictures for telepointer remote guided with TDOD movements. As a result, this chapter will provide a quick overview of cursor recognition, template matching, and data extraction based on a recent study. There are two parts to the summary for this research project thesis. The first is for cursor detection. The second method is template matching for endoscopic image detection.

2.2 Summary of Literature review for cursor detection

For cursor detection, many methods can be used to detect cursor. One of the method is by using the 3D cursor. David B. et al. introduced three method used in the medical field. It is by using user's hands or hand-controllers, a 2D cursor that can travel in 3D space, and a volume-subtending 3D cursor have all been referred to be 3D cursors in medicine. [7] They a thoroughly analyse the medical literature on the term "3D cursor" and explain its uses in diagnostic radiology and surgery in this paper. They went over the various applications of the 3D cursor in medicine, including using of a 3D cursor in conjunction with virtual reality (VR) and augmented reality (AR).[7] However, for this research project thesis, combining virtual reality (VR) and augmented reality (AR) has a high level of complexity to execute in real time. This is because the input for this project comes from the camera at the need of the equipment. Therefore, the method in this paper is not suitable to be used in this project because in this paper mainly uses hand movement, whereas the objective is only to detect the cursor.

The second paper that has been observed and reviewed is cursor movement by object detection based on image processing. Cursor movement based on image processing is the second paper examined and reviewed. The project uses an object detection technique to control mouse movement. In this research paper, this project eliminates of the use of a mouse or any other physical device to manage cursor movement.[8] The suggested work was implemented successfully with object detection and PyAutoGUI, and the mouse movement was achieved with high precision accuracy. This resulted in improved Human-Computer Interaction (HCI). It can also be used in Augmented Reality, modern gaming, and computer graphics. This technology has been utilised in modern gaming consoles to develop interactive games that detect and interpret a person's movements as orders in the context of computer graphics and gameplay.[8] The major goal is to eliminate the use of any physical device, such as a mouse, and instead rely on a web cam, which is widely available with a laptop, to manage the computer. Although this project offers numerous benefits, it also has significant disadvantages. If the background image clashes with the given image, it may provide an inaccurate result and not perform correctly. As a result, it's best to use this technique when the background light and the object’s colour don't combine. The system may run slower on PCs with poor resolution and computational capability. If the camera has a high resolution, the system may be slow, but this problem can be rectified by lowering the image resolution.[8]

The third paper that has been observed and reviewed is an approach to control mouse movement using a real-time camera. This system works with all mouse tasks and is based on computer vision techniques. Hajoon Park developed a method using vision technology and natural hand gestures to manipulate the mouse. The goal is to reduce the amount of work space necessary. In this paper, they propose a new technique to control the mouse system using a video device in this paper. This mouse system can do all mouse functions, including right and left clicks, double clicks, and scrolling. To do so, he uses a variety of image processing methods. However, because of the wide range in lighting and skin tones among the human race, it is difficult to obtain consistent results. [9] The majority of vision algorithms have difficulty with illumination. Following that, it is unable to obtain a clear image of the hand from the video streaming. It it also difficult to pinpoint the exact location of the centre of the hand. [9]

The fourth paper that has been observed and reviewed is the approach to control mouse movement using colour detection. The main focus of this research is on improving the human-computer connection. In this experiment, researchers is using web-cam colour detection to control cursor movement with all click events. This technology provides a simple and cost-effective approach to handle human-computer interaction. Ashvini Shastrakar et al. introduced a method by using a web camera with a colour recognition approach to control cursor movement with all click events in this experiment. To give commands to the system, the user needs to wear coloured tapes or capes. For execution, individual frames from the camera's real-time video must be processed separately. The processing methods require an image background subtraction algorithm for colour detection. The system uses detected colours to execute various functions, including tracking the mouse and performing all cursor commands. [10] At the end of this experiment, researchers list out few drawbacks of this method. The first one, in the presence of other colourful objects in the background, the system may respond incorrectly. Next, the system may run slowly on computers with limited processing capabilities. The system may run slowly if the camera's resolution is too high. However, this drawback can be counter to overcome the disadvantages. For the first problem, we can counter the drawback by using a filter. Using filter, command can be used to get the object perfectly. For the second problem, nowadays, most computer have advanced technologies within one computer, so we can counter this problem by using a high processing capabilities. For the last problem, if the resolution is too high, we can reduce the resolution, and it is possible to do that.[10]

From all the paper that have been observed and reviewed, the final conclusion that can be made is cursor detection by using colour detection tends to have high accuracy and good result, and it is less complex to execute in real time.

Table 2.1 Comparison journal for cursor detection

Title	Source / Author / Year Published	Method / Technology used	Result / Comment
"A systematic review of the 3D cursor in the medical literature" [7]	Source: Google Scholar (2018) - D. B. Douglas, R. E. Douglas, C. Wilke, D. Gibson, J. Boone, and M. Wintermark	To detect the cursor, few method is used. First using an individual's hands or a hand-controller are used to enter commands into the computer. Second method, utilising a non-volume-subtending digital 3D cursor. Third method, using a volume-subtending 3D cursor can be used.[7]	Combining virtual reality (VR) with augmented reality (AR) in the system is extremely difficult to implement in real time.[7]
"Cursor Movement by Object Detection Based on Image Processing"[8]	Source: IEEEXplore (2021) - N. Rajendra, R. N. Shree K, B. Rao, and A. Professor	Image Acquisition, Image Processing, Image Analysis, Morphologic Processing, Image Segmentation, and Gesture Recognition are the methods used to detect the cursor.[8]	The result in this paper is excellent but if the background image clashes with the specified image, it may generate an inaccurate result and may not perform correctly.[8]
"Cursor Movement Control Using Colour Detection" [10]	Source: IEEEXplore (2018) - A. Shastrakar, J. Raman, M. Paul,	Colour detection using a webcam is the method used to detect the cursor.[10]	Although the results shown in this work are outstanding, if the camera resolution is set too high, the system may

	N. Ramteke, and P. V. Sathwane		function slowly on PCs with limited computing power.[10]
"A Method for Controlling Mouse Movement using a Real Time Camera" [9]	Source: ResearchGate (2018) - Hojoon Park	The method used is by using a camera and computer vision technology, such as image segmentation and gesture recognition, which are employed to identify the cursor. [9]	The result in this paper is excellent but sometimes it is unstable due to the wide range in lighting and skin tones among human races, and it's difficult to pinpoint the exact location of the centre.[9]

2.3 Summary of Literature Review for Cursor Tracking

For the Cursor Tracking, there are many ways for this methods can be used to track the cursor. One of the methods is by using Template Matching for detect the similarity of the tire pattern. This paper detects the tire pattern's similarity based on template matching and Local Binary Pattern (LBP). Cai et al. introduced an approach optimized using Hue, Saturation, Value colour Model (HSV) and Sovel operator, so that the acquired target image has high accuracy. Instead of using the usual target extraction approach, this study proposes using HSV and Sobel edge detection to improve the template matching algorithm. [11] The test results reveal that the suggested technique is more robust and reliable than the previous one when using rotation-invariant LBP paired with difference value hash (dHash) algorithm for image texture feature extraction and similarity detection.[11]

The second paper that has been observed and reviewed is old and young face detection using the template matching method with Fuzzy Associative Memory (FAM) developed by Sembiring et al. Face detection with the aid of a computer is a difficult task since the human face has a significant level of intra-personal and extra-personal variability. The results of this study reveal that systems using template matching methods integrated with FAM can detect variations in human faces with an accuracy of 80%, which is 10% better than standard template matching. [12] With the template matching method and real-time system sensitivity when planting training samples on the system, it is emphasised that the sample must be in good light and that the training object clearly does not use attributes or objects that can block face attributes, because the results of the training sample will test many other samples. A clear test sample pattern of his face will significantly affect the success when testing.[12]

The third paper that has been observed and reviewed uses an efficient and Fast Partial Template Matching Technique which is Enhancement in Normalized Cross Correlation which is developed by M. Noorjahan and A. Punitha. In this research paper, efficiency limitations in any processing encompass a variety of elements, including the accuracy of the results and speed of image processing, which plays a critical role in computer vision algorithms. Thus any method that reduces the execution time is always welcome. In this paper, they are proposed a new method. It uses the partial template matching approach. The proposed method is put to the test. The experiment results, as expected, show a reduction in processing time or an improvement in operation speed. It also works with the highest level of precision. Since speed is always a consideration in image processing, the suggested technique is several times faster than the existing method for various image sizes and formats.[13] The experiment's findings demonstrate that processing time can be cut in half while operational speed can be increased. It also operates with high accuracy.[13]

The fourth paper that has been observed and reviewed is called Human Face and Facial Parts Detection using Template Matching Technique which is introduced by o Payal Bose et al. Face recognition has become relevant in recent years because of its potential uses, In this research paper, and the goal of this paper is to develop relevant techniques that provide both higher accuracy and efficient speed. A pattern recognition technique was used in this experiment to detect human faces and their unique facial characteristics. Normalized Cross-Correlation was utilized to recognise the faces and their facial parts in this case. [14] It is one of the most effective methods for detecting several faces in a group of photos of individuals. Although the correlation template matching technique is highly accurate, detecting it takes a long time.[14]

The fifth paper that has been seen and reviewed is Muhammad Dede Yusuf, RD Kusumanto, Yurni Oktarina, Tresna Dewi, and Pola Risma's proposal of a technique for fruit recognition and detection using Blob Analysis. This paper proposes using BLOB analysis to detect 5 fruits of various shapes and colours. BLOB extraction's aims to isolate the BLOB or targeted object within its binary image. [15] A BLOB is made up of a set of connected pixels, and it is up to the user to figure out which ones are neighbours and which are not. [15] In this research, BLOB analysis is performed using SCILAB, an open-source software that comparable to more expensive software like MATLAB. The simulation results show that BLOB analysis can be applied to various of fruits. This technology can eliminate noises like motion blur and lighting effects in agriculture. [15] As a result, a harvesting robot can use the proposed method.[15]

The sixth paper has been observed and reviewed is a project to create an automated vehicle counting system based on Blob Analysis for traffic surveillance. G. Salvi creates this system. The researcher offers a traffic surveillance system for vehicle counts in this paper. There are five steps in the suggested method. Background subtraction, blob detection, blob analysis, blob tracking, and vehicle counting are among the techniques used. Blob analysis is used to classify a vehicle modelled as a rectangular patch. [16] To detect and count moving cars, the virtual loop-based approach is utilized. The important features are derived by analysing the blob of vehicles. Moving targets are tracked by comparing extracted features and calculating the shortest distance between successive frames. [16] The results of the experiments suggest that the proposed system can offer real-time and valuable traffic surveillance information. [16]

The seventh research paper that has been observed and reviewed is about the face tracking using Feature Point Tracking, proposed by Guo-Shiang Lin and Tung-Sheng Tsai. Face detection and face tracking are elements of the proposed approach. A face detector using Haar-like features trained by the Adaboost algorithm is used in the proposed method. [17] The suggested method uses KLT feature tracking to monitor faces in adjacent frames. The proposed method, uses a re-tracking procedure to reduce feature tracking error. Several videos featuring various types of face movements are captured using a low-cost webcam to evaluate the proposed scheme. Experiments show that the proposed method can track facial regions in videos. Furthermore, the suggested face tracking approach has a detection rate of above 91%. [17] These results show that the suggested technique can track faces in noisy films in real life. [17]

The eighth research paper observed and reviewed is called Feature Point Matching with Matching Distribution. This paper presented by San Ratanasanya, Jumpol Polvichai, and Booncharoen Sirinaovakul. This research proposes a new method using Feature Point Matching for obtaining reliable and suitable matches. When the matches are dispersed across the complete point set, a high matching level can be attained. [18] As a result, the researcher proposed the quality of the matching verified by looking at the distribution of matches. [18] In terms of precision, the suggested method outperforms SIFT combined with RANSAC, according to the results. SIFT combined with RANSAC has high false positive matches than the suggested algorithm, based on the matching precision value. [18] The greater the number of false positives, the less effective the performance of feature point matching applications. [18]

The ninth research paper that has been observed and reviewed is research on improvement of feature point matching algorithm using depth prediction. This proposed method is designed by Yongbin Chen, Guitang Wang and Liming Wu. Researchers describe a unique strategy for improving feature matching accuracy by integrating depth estimate and SIFT in this study. The SIFT technique is utilized to extract and match features in this work, while the MonoDepth algorithm is employed to predict depth. [19] The feature matching is limited to the pixel region of adjacent images with identical depth values, reducing the number of mismatching points. [19] The feature matching with depth estimate has a higher accuracy than the old method, according to the findings of the experiments. [19]

From all the paper that has been observed and reviewed, the final conclusion that can be made is the Blob Detection (Colour Feature), and Feature Point Tracking technique is the most suitable to be used to track the mark location because the result for this method has high accuracy which is needed in medical purposes, especially during surgery.

Table 2.2 Comparison journal for Cursor Tracking

Title	Source / Author / Year Published	Method / Technology used	Result / Comment
"Tire pattern similarity detection based on template matching and LBP" [11]	Source: IEEEXplore (2019) - Cai, Chengtao, Lin, Mengqun	The method used in this paper is Target Extraction Methods and Similarity Detection Methods.[11]	There is no cursor or point to identify the target so it is a must to search all throughout the frame, adding to the system's complexity.[11]
"Improving the accuracy of old and young face detection in the template matching method with Fuzzy Associative Memory (FAM)" [12]	Source: Google Scholar (2020) - Sembiring, Rafika Sari Efendi, Syahril Suwilo, Saib	The method used in this paper is Template Matching methods combined with Fuzzy Associative Memory (FAM).[12]	This method used in this paper can only be used with a zero-orientation background and no patterned background. [12]
"An Efficient and Fast Partial Template Matching Technique – Enhancement in Normalized Cross	Source: ResearchGate (2019) - M. Noorjahan, A. Punitha	The method used in this paper is Partial Template Matching Algorithm or Partial Normalized Cross Correlation (PNCC) and the Block window technique. [13]	The experiment results show that processing time can be reduced or operating speed can be increased. It also works with the highest level of precision.[13]

Correlation" [13]			
"Human Face and Facial Parts Detection using Template Matching Technique" [14]	Source: Google Scholar (2020) - Payal Bose, Samir Kumar Bandyopadhyay	The method used in this paper is The Normalized Cross-Correlation technique. [14]	Although the correlation template matching technique is very accurate, it takes a long time to detect. [14]
"BLOB Analysis for Fruit Recognition and Detection" [15]	Source: IEEEXplore (2018) - Muhammad Dede Yusuf, RD Kusumanto, Yurni Oktarina, Tresna Dewi, Pola Risma	The method used in this paper is BLOB analysis. [15]	This approach can eliminate noises like motion blur and lighting effects in agriculture. The BLOB analysis was detected all five fruits in various sizes and colours.[15]
"An Automated Vehicle Counting System Based on Blob Analysis for Traffic Surveillance" [16]	Source: Semantics Scholar (2018) - G. Salvi	The BLOB analysis technique was implemented in this paper. Background subtraction, blob detection, blob analysis, blob tracking, and vehicle counting are the five steps of the proposed algorithm. [16]	This proposed method can give real-time and valuable traffic surveillance information. This approach has an accuracy of over 98 percent. [16]

"A Face Tracking Method Using Feature Point Tracking" [17]	Source: IEEEXplore (2017) - Guo-Shiang Lin and Tung-Sheng Tsai	The method used in this paper is Feature Point Tracking. [17]	Face tracking has a detection rate greater than 91 percent. The results show that the suggested technique can track faces in noisy films in real life. [17]
"Feature Point Matching with Matching Distribution" [18]	Source: Research Gate (2018) - San Ratanasanya, Jumpol Polvichai, Booncharoen Sirinaovakul	Feature Point Tracking is the method utilized in this article, which involves visualizing a specific zone in a point set. [18]	In terms of precision, the proposed algorithm outperforms SIFT combined with RANSAC, according to the results. [18]
"Research on feature point matching algorithm improvement using depth prediction" [19]	Source: Semantics Scholar (2018) - San Yongbin Chen, Guitang Wang, Liming Wu	Feature point matching using depth estimation and SIFT is the method used in this paper. [19]	The feature matching with depth estimate has higher accuracy than the old Feature Point Matching method, according to the findings of the experiments. [19]

2.4 Summary for Chapter 2

In this chapter, the past, previous, and present work on this project has gathered and studied. The articles and journals in this chapter guide the project's methodology and flow, and they were all about cursor detection and tracking. Each article or journal's concept, idea, and flow is utilized as a guide to achieve the project's objective detecting the cursor or pointer, tracking the mark locations that the pointer or cursor has marked and measuring and validating the performance.

CHAPTER 3

METHODOLOGY

3.1 Introduction

"Methodology" refers to a systematic, analytical examination of the research methods. This includes a detailed analysis of the methodologies and concepts associated with the information group. The Methodology provides a theoretical foundation for describing which process, methodologies, or best practices should be used for cursor detection and tracking. It goes over the objective, the scope of the project, and how to achieve the desired outcomes in great detail. This chapter also explains the Block Diagram, Flowchart, and mathematical equation involved.

This chapter will go into detecting and tracking cursors with the MATLAB R2021b software. This chapter will also discuss the technique of detecting the green colour cursor and the process of tracking the cursor will also be discussed in this chapter. Before beginning any task, the first thing to consider is planning, since a step-by-step planning process will generate a positive result. A planning Gantt Chart and Block Diagram have been constructed to provide a high-level overview of the project's implementation and procedures. A flowchart has also been created for each cursor detection method and tracking method to describe the entire testing process. In order to meet the objectives indicated in the previous chapter, the approach presented here reflects the implementation of codes relevant to the cursor detection and tracking process.

3.2 Project Management

For this project, a Gantt Chart was utilized as the framework. The Gantt chart below shows the timetables for Final Year Project 1 (FYP 1) and Final Year Project 2 (FYP 2), which may be used to schedule work around deadlines and properly allocate resources to accomplish the project's goal. The researcher devised a schedule for each action, includes reading research papers, writing a thesis, building the overall system's code, data collection and analysis, and presentation preparation.

3.2.1 Gantt Chart Final Year Project 1

Refer to Appendix C.

3.2.2 Gantt Chart Final Year Project 2

Refer to Appendix D.

3.3 Block Diagram

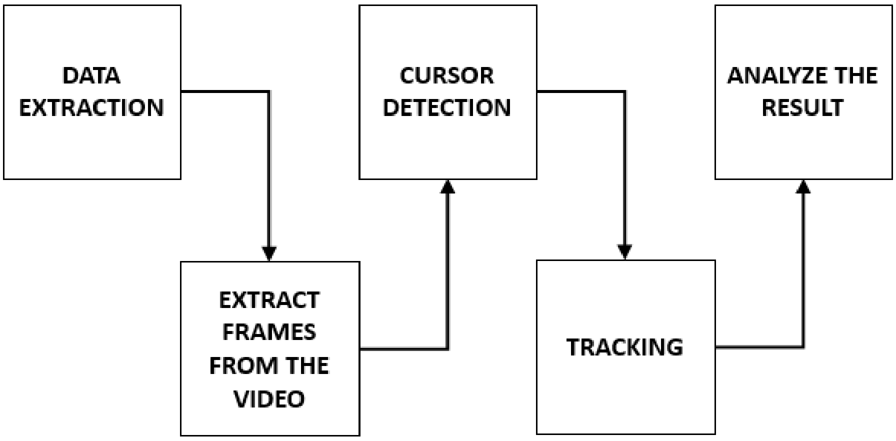


Figure 3.1 Block Diagram

Based on the figure block diagram above, explains the flow of this project research. All the dataset is taken from Hamlyn of Imperial Collage London website as requested by the supervisor. There are three types of dataset used: rotation, zoom in zoom out and liver deforming video. Dataset type rotation is used to do the initial analysis and obtain analysis for cursor detection. To extract frames from the video, dataset type rotation will be extracted and turned into photos. These obtained frames will be used to train and analyse the result. For cursor detection, colour detection method with three different approaches is used. It is HSV colour map array, K-means clustering and RGB colour specifier. In this project, there are 2 methods of cursor tracking are used. It is Blob Detection and Feature Point Tracking. Lastly, the dataset will be train and analyzed to observe which cursor tracking method is the best to be used to detect the cursor and to observe which method is the best.

3.4 Method of Solution

In this project, based on the objective stated in Chapter 1, the first objective is to detect the green colour cursor or pointer. Therefore, for the cursor detection, there are three types of colour detection are used. It is RGB Colour Space, HSV Colour Space, and K-Means Clustering. The second objective is to track the mark location that has been mark by the pointer or cursor. Therefore, there are two types of cursor tracking used. It is Blob Analysis and Features Point Tracking.

3.5 Flowchart Method of Solution

A flowchart is necessary to represent the process sequence required to execute the task. It used symbols to illustrate a process. Each stage of the process is symbolized by its symbol, which is accompanied by a brief description of the stage. The flowchart will begin with ‘Start’ at the top.

3.5.1 Cursor detection

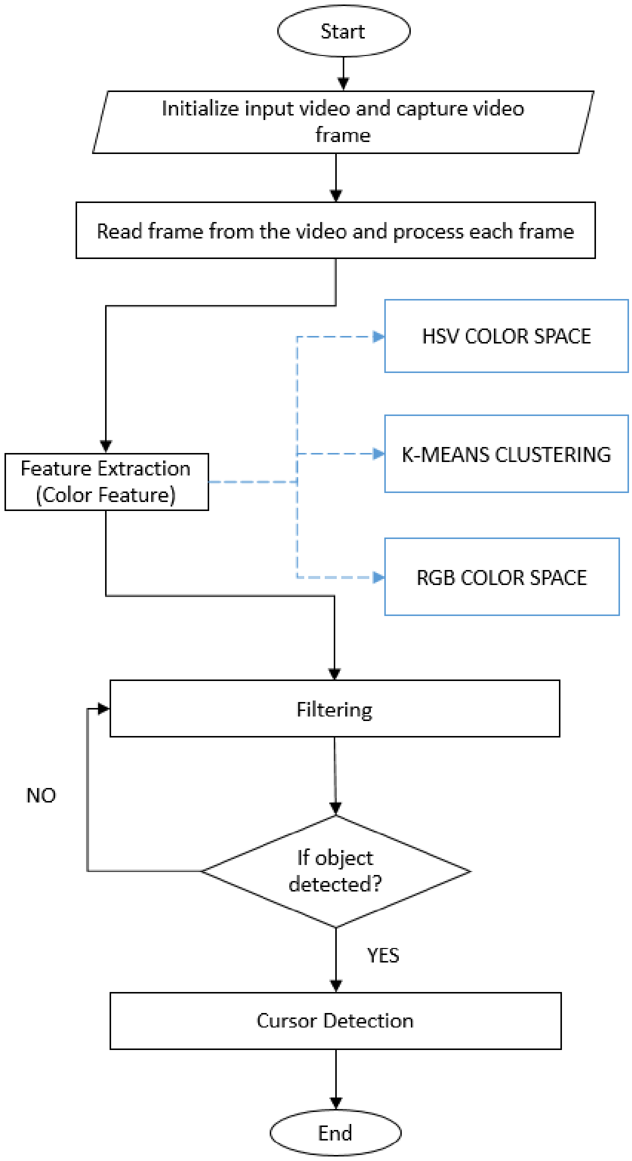


Figure 3.2 Flowchart for the cursor detection

For the process flow for the cursor detection, the flow begins with initializing the input video and capturing the video frame. After that, it will read the frame from the video and process each frame. The next flow is colour detection, it will detect the cursor. A colour detection algorithm searches an image for pixels with the same colour or colour range as a given colour. The colour of the detected pixels can be changed to isolate them from the rest of the image. The colour green was selected for the cursor in this study because it contrasts well with blood and tissue colour. As a result, it is important to recognise the green colour. There 3 types of methods used for colour detection. It is RGB Colour Specifier, HSV Colour map Array and K-Means Clustering. After colour detection, a mask or filter is used. Image filtering is a technique for hiding and revealing parts of an image. It is a non-destructive image altering method. It usually allows users to edit and adjust the mask later if necessary. It is frequently a more efficient and innovative method of picture manipulation services. An image alters a picture's size, colours, shading, and other features. After applying the filter or the mask, if NO means not detecting the cursor, it will filter the image more, and if YES, the final result will be cursor detection.

3.5.1.1 Flowchart RGB Colour Specifier

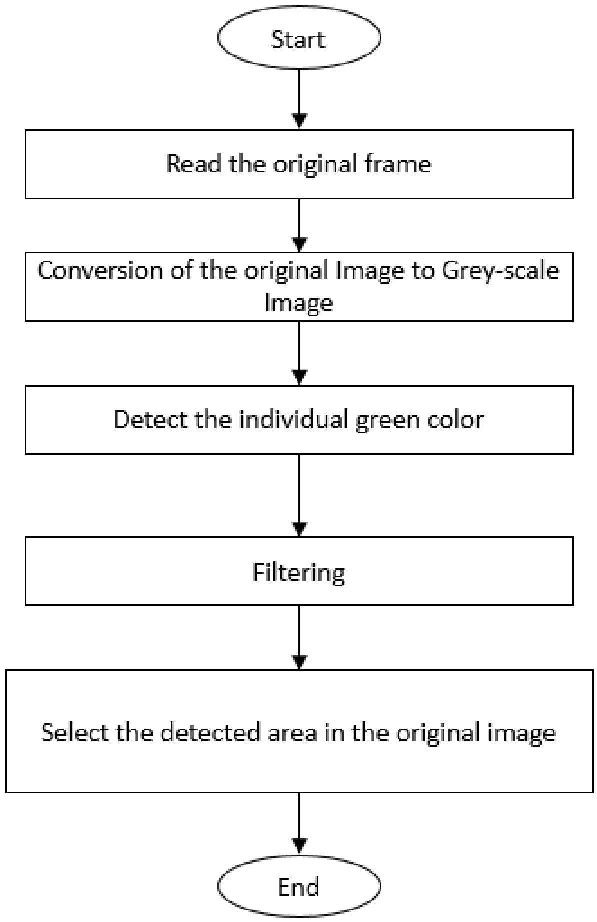


Figure 3.3 Flowchart for RGB Colour Specifier

For the process flow for the RGB Colour Specifier, the flow begins with reading the original frame. RGB colour detection method gets a subsection of the current image in the axes. It can extract the individual image by inserting the colour channel. To visualize a specific channel in this project is green colour, other colour channels need to set to zero. So to show the green channel, the blue and red channels need to be set to zero. The filter used in this method is imsubtract. This will subtract one image from another, or subtract a constant from an image.

3.5.1.2 Flowchart for HSV Colour map Array

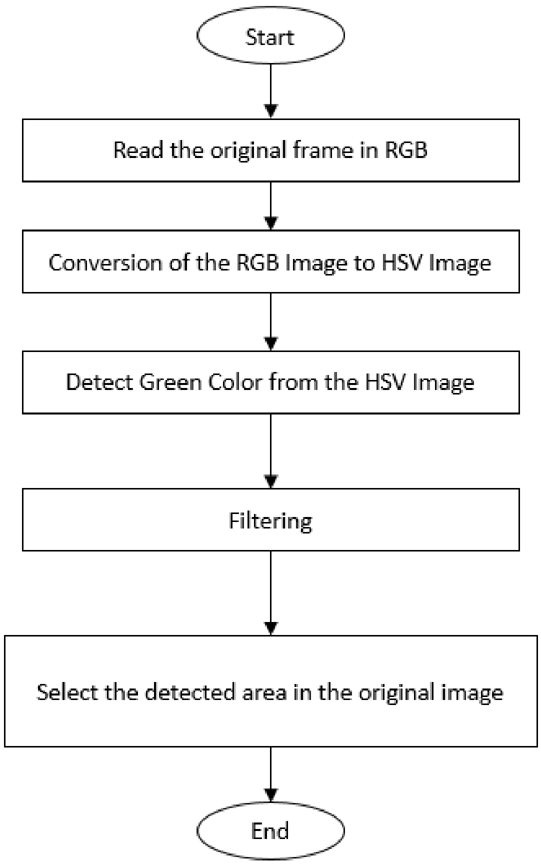


Figure 3.4 Flowchart for HSV Colour map Array

The HSV Colour map Array method is a method of extracting of H and S from HSV Colour map Array to detect colour. Hue Saturation Value (HSV) is an abbreviation for Hue Saturation Value. Hue defines the "colour," Saturation defines the colour’s dominance, and Value defines the brightness. The first step to convert the RGB image to the HSV image. Then, to detect the green colour, the saturation and hue value is important and need to be adjusted according to the HSV colour map array. The filter used in this method is imfill. On binary and grayscale images, the imfill function performs a flood-fill approach. This operation can help to remove unwanted imperfections from the images.

3.5.1.3 Flowchart for K-Means Clustering

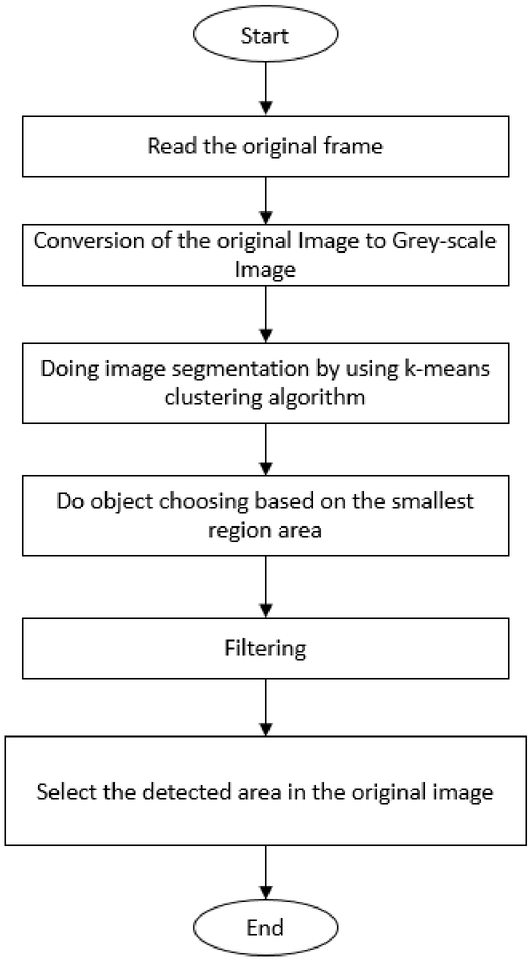


Figure 3.5 Flowchart for K-Means Clustering

K-Means Clustering is a colour separation technique or, to put it another way, a technique for separating colour groups. Each colour is treated as if it were a physical location in K-means clustering. The kmeans function separates data into k mutually exclusive groups and returns the cluster index for each observation. It finds partitions that keep colours within each cluster as near together as possible while keeping them as far apart as possible from colours in other clusters. The user must select the number of clusters to be partitioned and a distance metric to quantify how near two colours are to each other when using K-means clustering.

3.5.2 Cursor Tracking

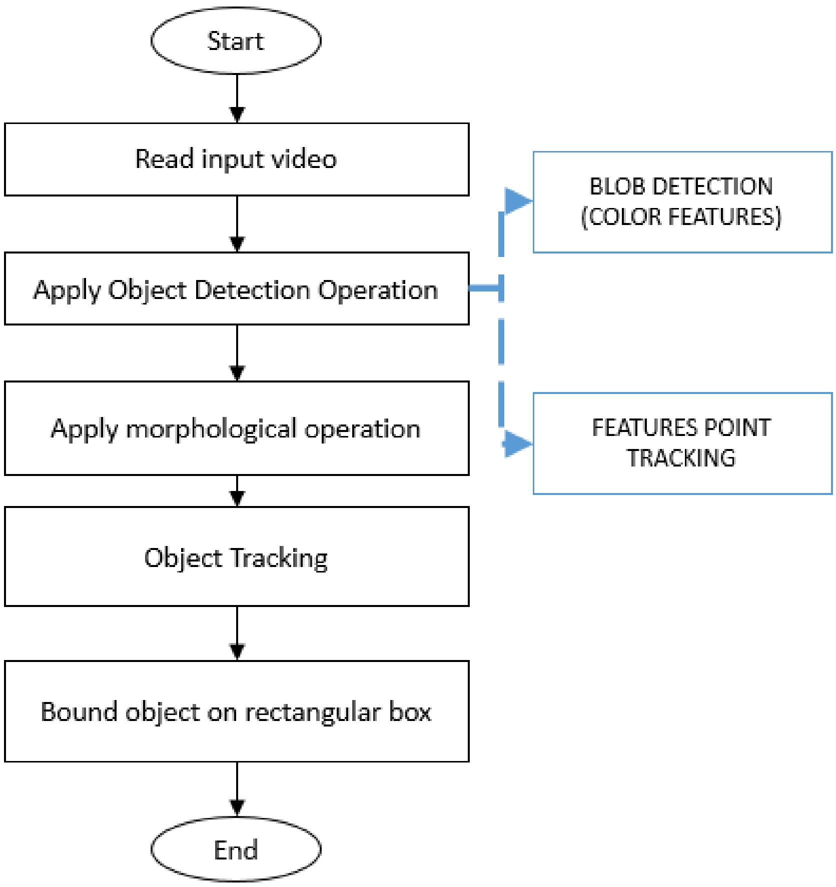


Figure 3.6 Flowchart for the Cursor Tracking

For the process flow for the cursor detection and tracking, the flow begins with the read input video. In this process, the system will read the frame from the video and process each frame. The next flow is object detection. In this step, the system will detect the green colour cursor. The method used to detect the cursor is by using Blob Detection (Colour Features) and Features Point Tracking. After cursor detection operation is applied, the morphological operation will be applied. Morphology is a term used to describe a group of image processing techniques that alter images based on their shape.

Morphological operations utilize an input image and apply a structural element to it, resulting in a similar-sized output image. In a morphological operation, each pixel value in the output image is determined by comparing the matching pixel in the input image with its neighbors. Apply the morphological operation, the system will track the cursor based on the condition applied for each method. Blob Detection (Colour Features) tracking the targeted object based on the detected colour. In this project, green colour is the aim of this method. As for Feature Point Tracking, the tracking processes the targeted object based on the minimum point detected. There will be a bounding box around the object as long as the system successfully detects and tracks the target object.

3.5.2.1 Flowchart for Blob Detection (Colour Features)

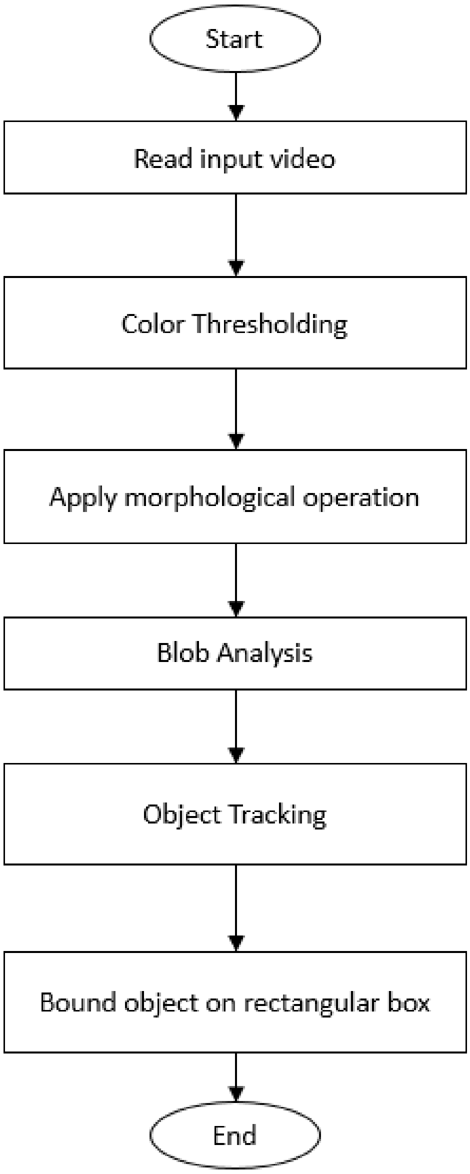


Figure 3.7 Flowchart for Blob Detection (Colour Features)

Blob detection is a technique for tracking the movements of objects within a frame using a computer system. Since this project aim to track the cursor, this technique is appropriate. Blob detection (Colour Detection) approaches aim to identify points or regions in an image that are brighter or more colourful than the surrounding area. Blobs are the term for these areas. A blob is a collection of pixels that can be used to identify an object. This detection process determines the blob's whereabouts in successive image frames is determined by this detection process. [20] Before any blob detection, the blob area must be created, where pixels with similar light or colour values are clustered together to find the blob. The goal of this project is to detect a green colour cursor. After colour detection and morphological operation, blob detection is used to locate the desired target in the image.

3.5.2.2 Flowchart for Feature Point Tracking

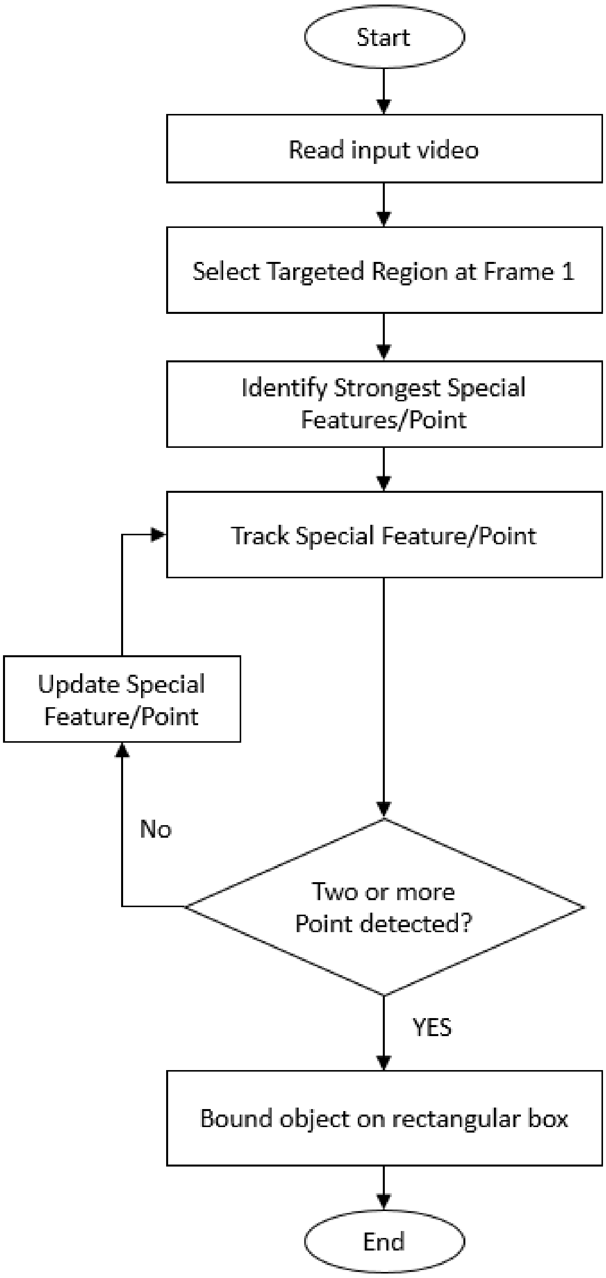


Figure 3.8 Flowchart for Feature Point Tracking

Feature Point Tracking is a method for evaluating abstractions of image information and determining whether or not there is an image feature of particular type at each image location based on the information on the reference image. A particular characteristic is extracted from the reference image before any feature detection and tracking. [21] Feature Point Tracking is a technique that uses matching methods to detect and track a pattern or specific structure in an image, such as a point, edge, or small picture patch. The Minimum Eigenvalue Algorithm is used to detect corners and return corner points in this project.

3.6 Design Software Development

It is regarded as one of the project's most important parts to achieve the objective and to make the system work. This subtopic will cover MATLAB R2021b Software and coding.

3.6.1 Simulation Software (MATLAB R2021b Software)

Figure 3.9 MATLAB R2021b Software

The MATLAB R2021b software is a high-level programming language and interactive environment for numerical computation, visualization and programming. It allows matrix manipulations, plotting functions and data, implementation of algorithms, creation of user interfaces, interfacing with programs written in other languages, analyses data, develops algorithms and creates models and applications. l possible.

3.6.2 Cursor Detection

3.6.2.1 Coding of RGB Colour Specifier

Refer to Appendix A.

3.6.2.2 Coding of HSV Colour map Array

Refer to Appendix A.

3.6.2.3 Coding of K-Means Clustering

Refer to Appendix A.

3.6.3 Cursor Tracking

3.6.3.1 Coding of Blob Detection (Colour Features)

Refer to Appendix B.

3.6.3.2 Coding of Feature Point Tracking

Refer to Appendix B.

3.7 Mathematical Equation

3.7.1 Accuracy

In object detection and classification, a method estimates a positive or negative class, and the predictions can be correct or false.

Any classification effort must include a validation for accuracy. It compares the categorize image to data from a trusted or ground-truth source..[22] It is important to use several frames and analyze each frame to gain an advantage and assist the development of a better solution for detecting and tracking the pointer.

Each method's accuracy is calculated manually using the formula below.

$$\text{Accuracy} = \frac{Number\ of\ correct\ prediction}{Total\ Number\ of\ predictions} \qquad 3.1$$

3.7.2 Euclidean Distance

The Euclidean distance is defined as the distance between two points in Mathematics. [23] In other words, the Euclidean distance between two points in Euclidean space is defined by the length of the line segment between them. The Pythagorean distance is named because the Euclidean distance can be calculated using coordinate points and the Pythagoras theorem. [24]

As previously stated, the Euclidean distance formula assists in making the distance between two line segments. Assume two locations in the two-dimensional coordinate plane, such as (x1, y1) and (x2, y2). As a result, the Euclidean distance formula is:

$$\text{Euclidean Distance} = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \qquad 3.2$$

In this project, Euclidean distance is used to calculate the error distance between the coordinate of the centroid in the previous frame and the coordinate of the centroid in the current frame.

3.8 Summary for Chapter 3

All of the subtopics were briefly explained in this chapter, as well as each of the component's rationale. Each component chosen for this project has been adequately studied to guarantee that it is appropriate, does not exceed the budget, and is long-lasting and sustainable. The block diagram and flow chart illustrated in this chapter can help the reader understand the entire system. This chapter includes a flowchart for each step to ensure that the reader understands the project.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

The predicted project results of Tracking Pointer at Endoscopic Images for Teleporter Remote Guided will be discussed in this chapter. First, it will describe the outcome of extracting frames from a video. Second, it will also describe the results of each colour detection method used for cursor detection. Finally, it will describe the results of each method for cursor tracking. The projected outcome of this machine is stated in light of the project's goal.

The reader will learn about the system's operation method in this chapter. The system must correspond to the flow and proper execution technique to ensure that it operates smoothly and successfully. The researcher will show how to extract the results in table and system information.

Rotation, zoom in zoom out, and liver deformation is three types of datasets utilised in this project. For each type of video, the full frame used for colour detection analysis is 497 frames. The determination of centroid values of the green colour cursor, which requires the image colour detection approach, is critical for the first objective. Then, every frame collected from each video is used to perform cursor tracking analysis. The object tracking technique determines the cursor's centroid values and the number of points matched in each frame, which is vital for the second objective.

The main software used in this project is MATLAB R2021b Software, which is used to code the system. The following subchapter will review every aspect of the methodology's outcomes and analysis.

4.2 Result Presentation

The results presentation section of the research paper is where the researcher presents the study's findings based on the information acquired as a result of the technique used.

4.3 Frame Extraction

Frames from a video can be extracted and turned into photos. Extracting frames from the rotation video, will be extracted and turned into photos. There are 711 frames obtained for this type of video. The coding used to obtain the result of frame extraction is attached in Appendix A.

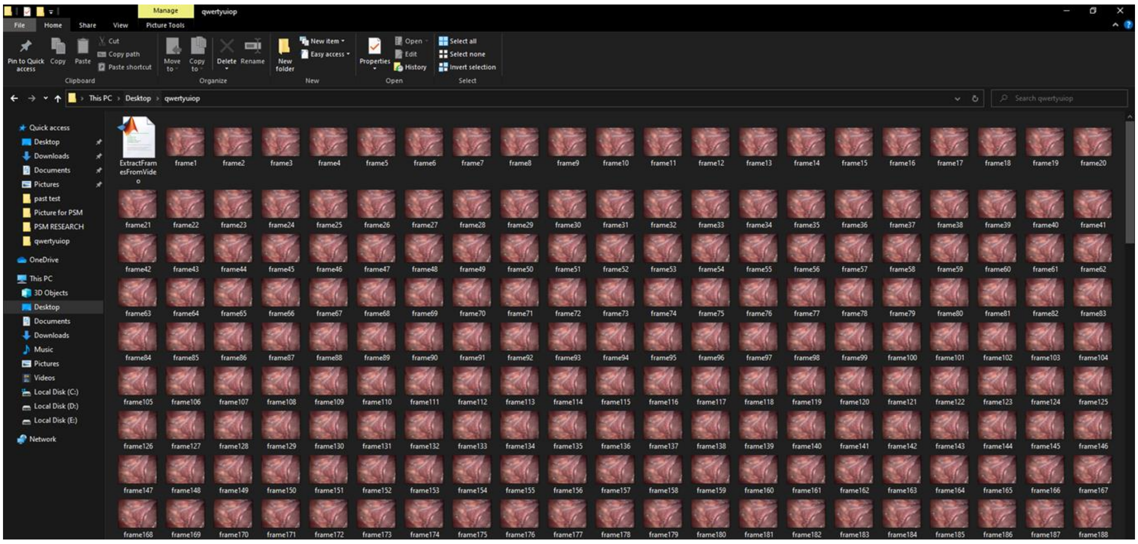


Figure 4.1 Frames Extracted Rotation video

Extracting frames from the zoom-in zoom-out video, will be extracted and turned into photos. There are 1236 frames obtained for this type of video. The coding used to obtain the result of frame extraction is attached in Appendix A.

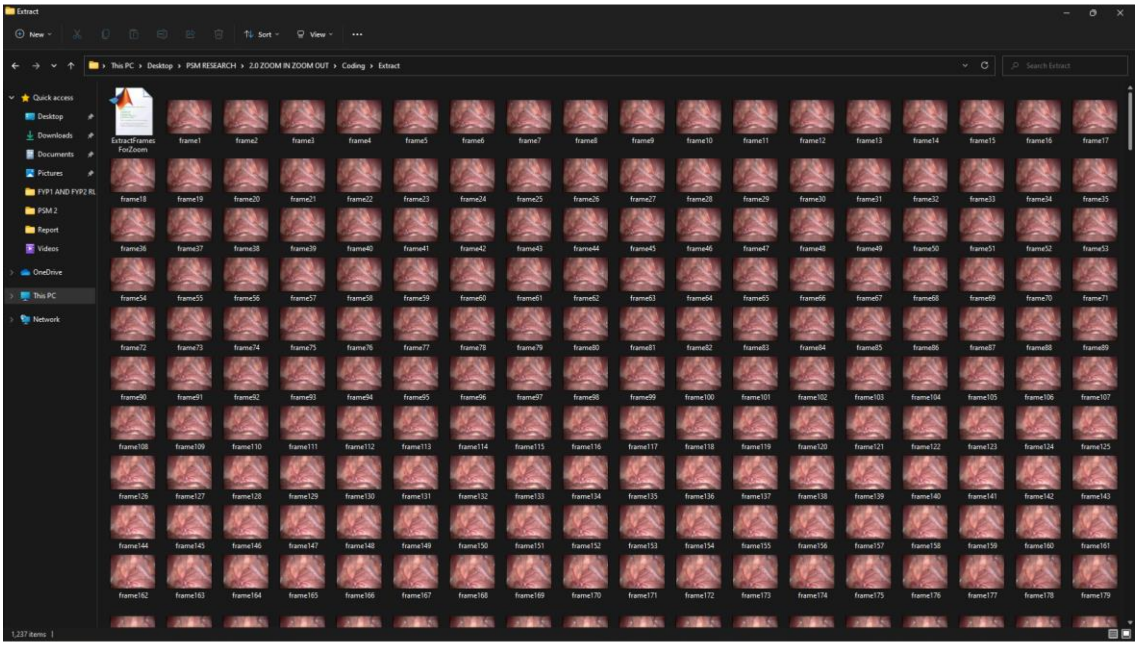


Figure 4.2 Frames Extracted Zoom in Zoom out type of video

Extracting frames from the liver deforming video, will be extracted and turned into photos. There are 804 frames obtained for this type of video. The coding used to obtain the result of frame extraction is attached in Appendix A.

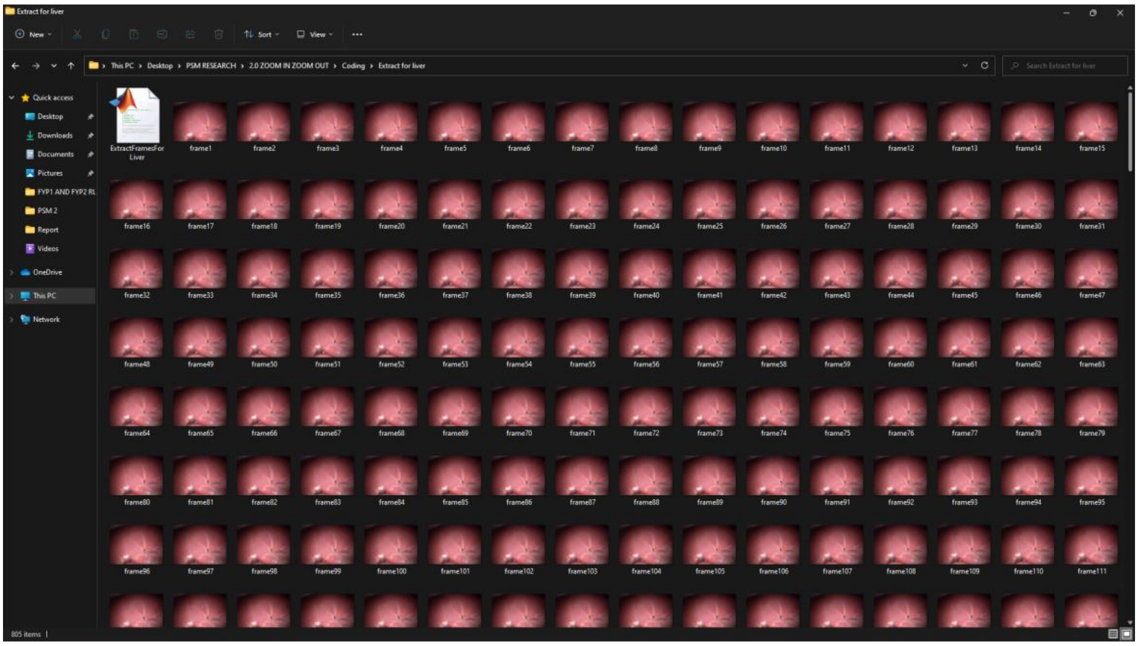


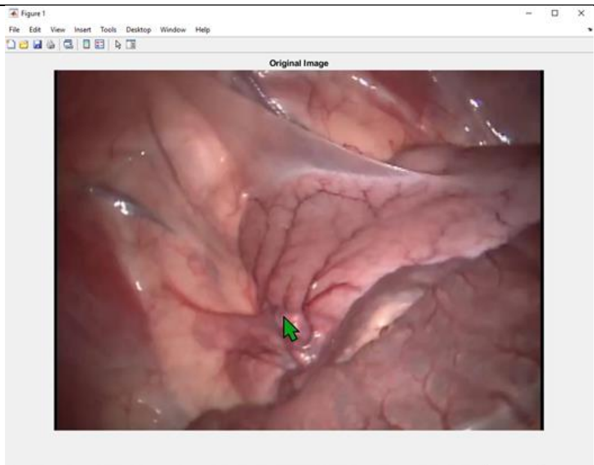
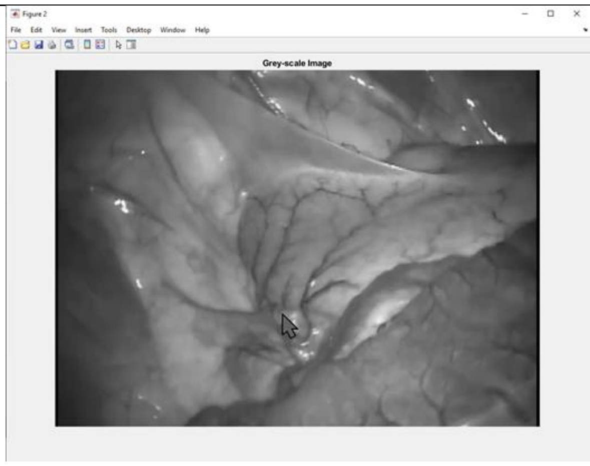
Figure 4.3 Frames Extracted Liver Deforming type of video

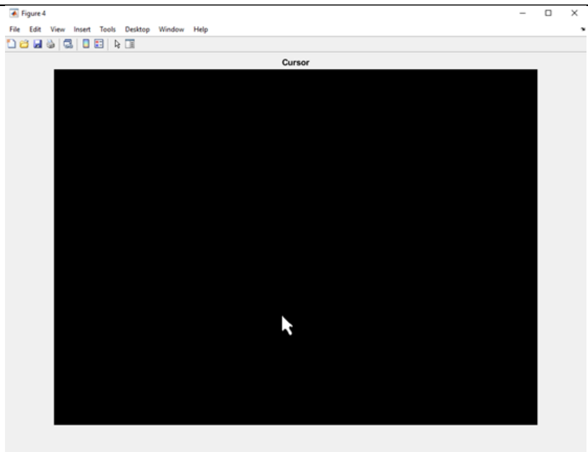
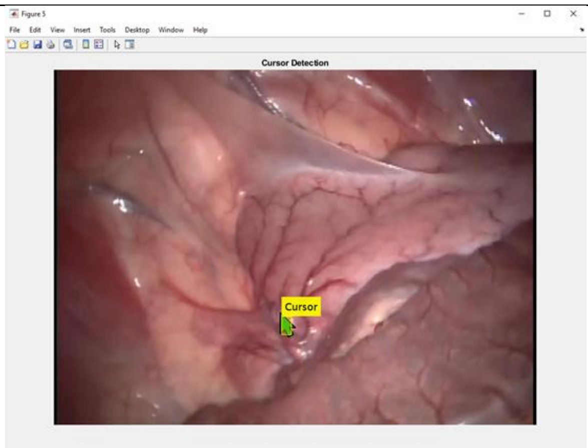
4.4 Cursor Detection

4.4.1 Result by using RGB Colour Space for Cursor Detection

Table 4.1 shows the result for each step in the RGB Colour Specifier method according to the flowchart discussed in the methodology in Chapter 3. The coding used to obtain the result using RGB Colour Specifier for Cursor Detection is attached in Appendix A.

Table 4.1 Result by using RGB Colour Specifier

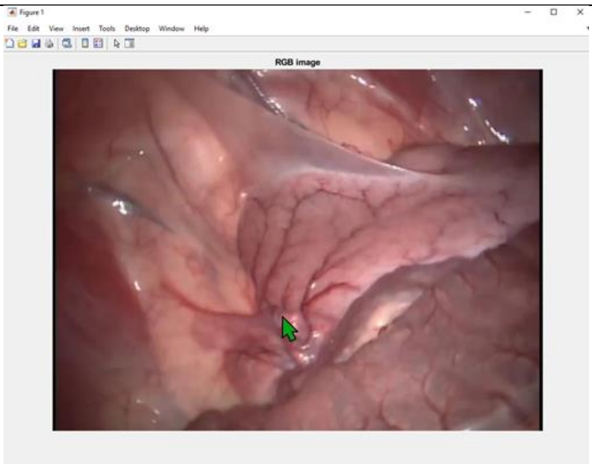
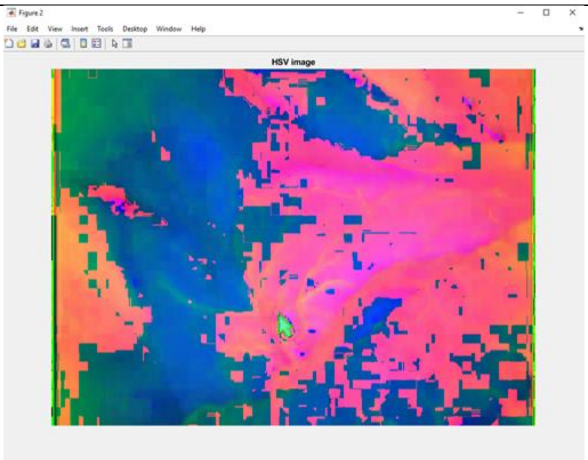
No	Step	Result/Outcome
1.	Read the original frame	
2.	Conversion of the original Image to Grey-scale Image	

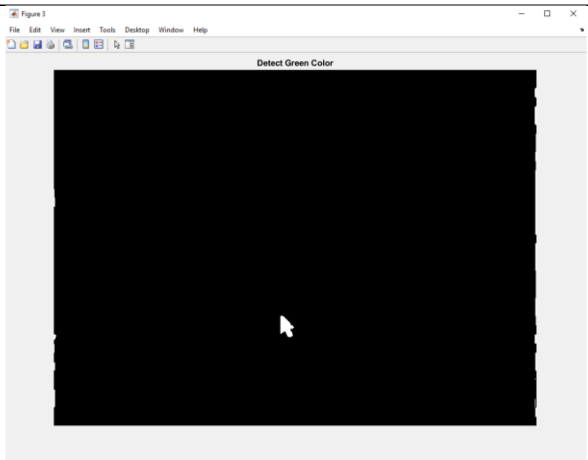
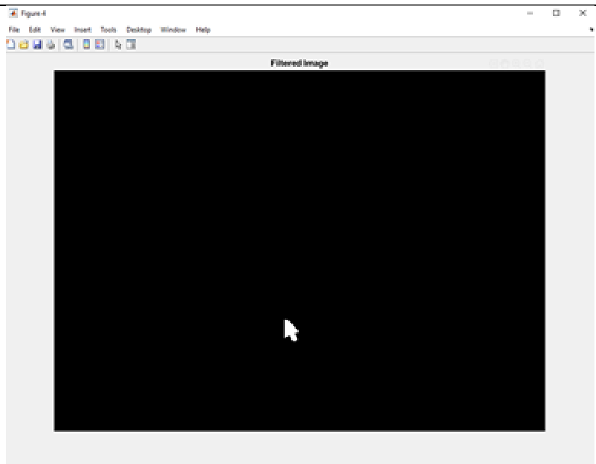
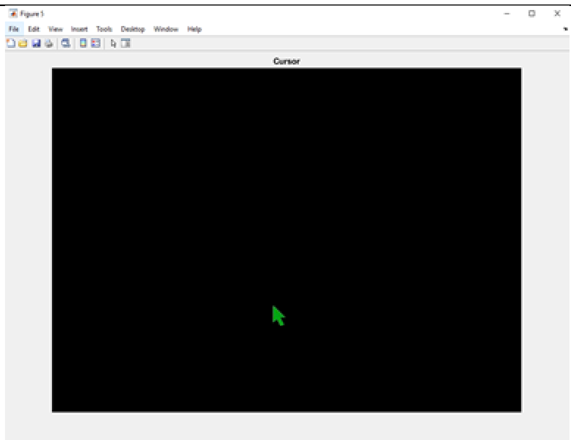
3.	Detect the individual green colour	
4.	Creating Mask/Filter	
5.	Select the detected area in the original image	

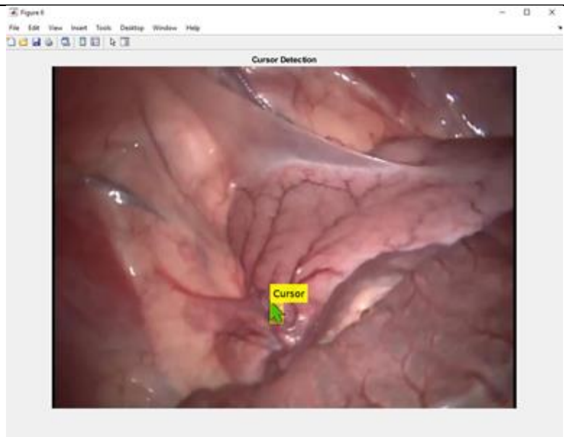
4.4.2 Result by using HSV Colour Space for Cursor Detection

Table 4.2 shows the result for each step in the HSV Colour map Array method according to the flowchart discussed in the methodology in Chapter 3. The coding used to obtain the result using HSV Colour map Array for Cursor Detection is attached in Appendix A.

Table 4.2 Result by using HSV Colour map Array

No	Step	Result/Outcome
1.	Read the original frame in RGB	
2.	Conversion of the RGB Image to HSV Image	

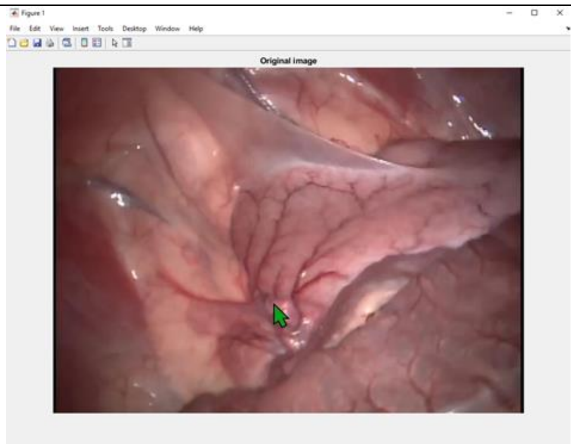
3.	Detect Green colour from the HSV Image	
4.	Creating Mask/Filter	
5.	Extract Green colour from the background	

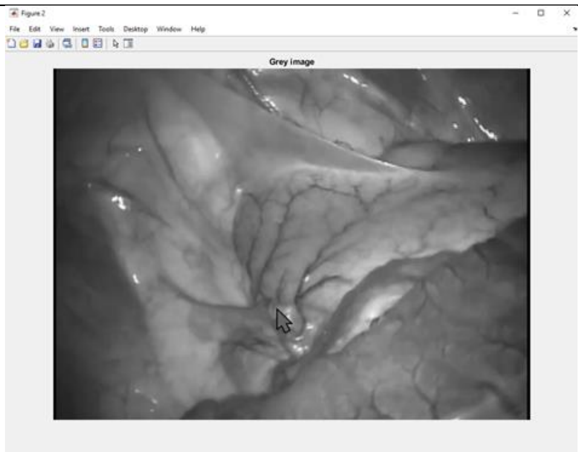
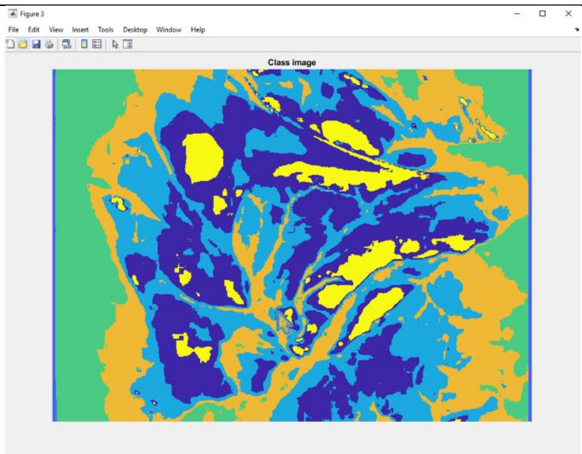
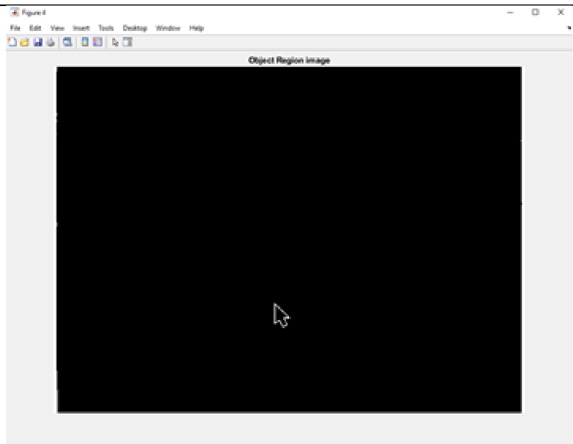
6.	Select the detected area in the original image	

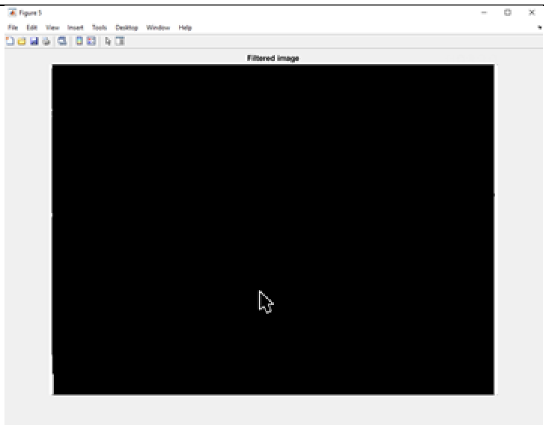
4.4.3 Result by using K-Means Clustering for Cursor Detection

Table 4.3 shows the result for each step in HSV Colour map Array method according to the flowchart discussed in the methodology in Chapter 3. The coding used to obtain the result using K-Means Clustering for Cursor Detection is attached in Appendix A.

Table 4.3 Result by using K-Means Clustering

No	Step	Result/Outcome
1.	Read the original frame	

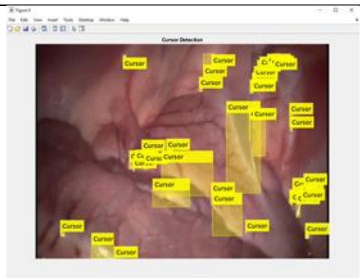
2.	Conversion of the original Image to Grey-scale Image	
3.	Doing image segmentation by using the K-Means clustering algorithm	
4.	Do object choosing based on the smallest region area	

5.	Creating Mask/Filter	
6.	Select the detected area in the original image	

4.4.4 Comparison between Detected and Not Detected Result

Table 4.4 shows the visual results when the cursor detection detects the green colour cursor and when the cursor detection fails to detect the green colour cursor.

Table 4.4 Comparison between Detected and Not Detected Results

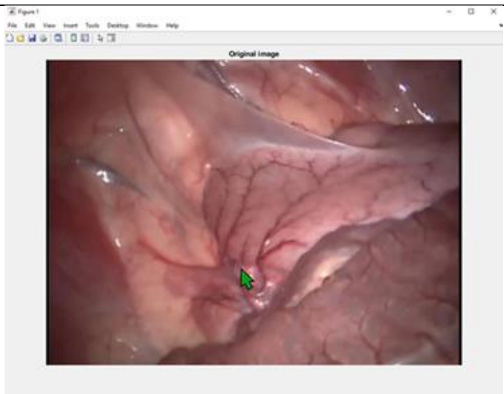
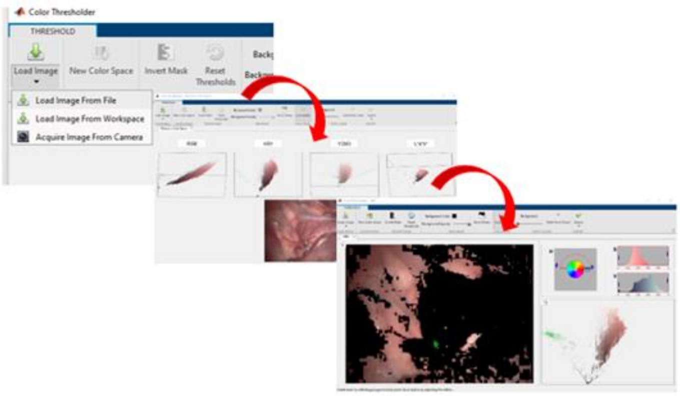
Detect	**Not Detect**
  	  

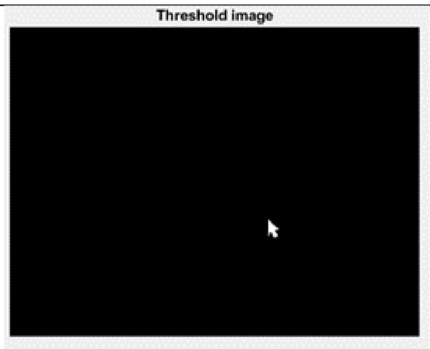
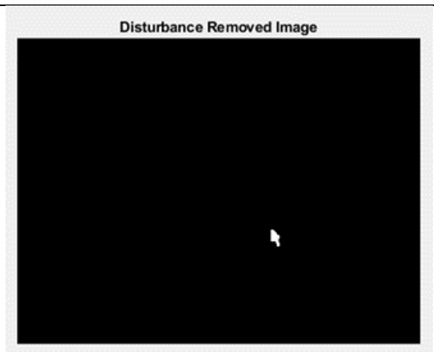
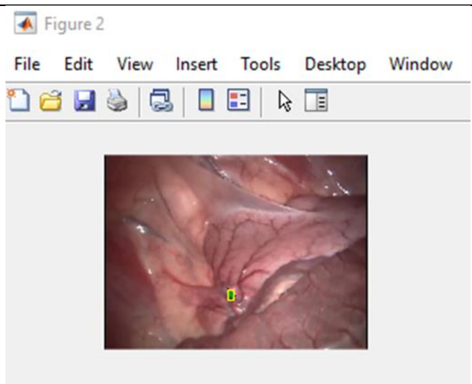
4.5 Cursor Tracking

4.5.1 Result by using Blob Detection (Colour Feature)

Table 4.5 shows the result for an important step in Blob Detection (Colour Feature) method according to the flowchart discussed in methodology in Chapter 3. The actual result of this cursor tracking is in the video. There will be a bounding box bound around the detected object. Here, only on image for important step is shown. The coding used to obtain the result using Blob Detection (Colour Feature) for cursor tracking is attached in Appendix B.

Table 4.5 Result by using Blob Detection (Colour Feature)

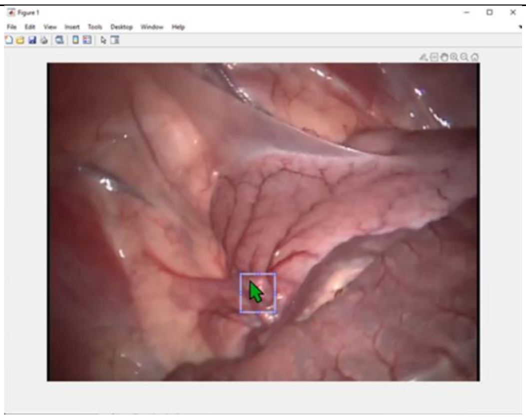
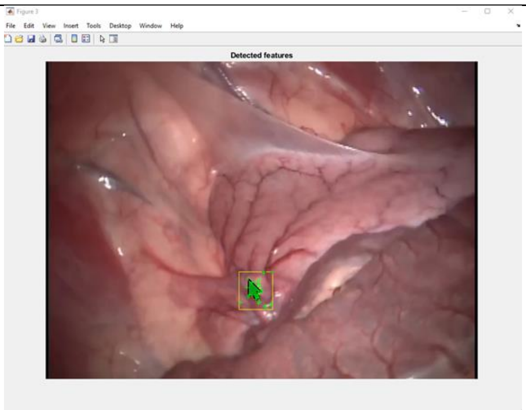
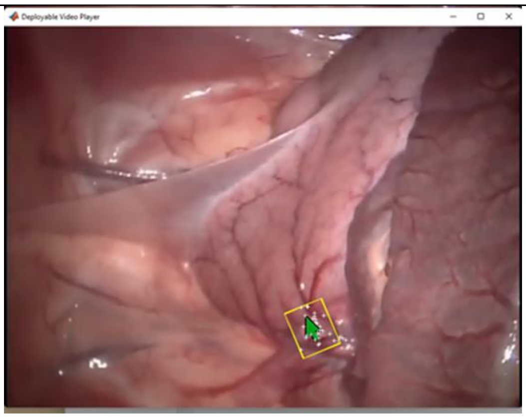
No	Step	Result/Outcome
1.	Read input video	
2.	Colour Thresholding (using Colour Thresholder App in MATLAB)	

		
3.	Apply morphological operation	
4.	Bound object on the rectangle box	

4.5.2 Result by using Feature Point Tracking

Table 4.6 shows the result for an important step in the Feature Point Tracking method according to the flowchart discussed in the methodology in Chapter 3. The actual result of this cursor tracking is in the video. There will a bounding box bounded around the detected object. Here, only an image for an important step is shown. The coding used to obtain the result using Feature Point Tracking for cursor tracking is attached in Appendix B.

Table 4.6 Result by using Feature Point Tracking

No	Step	Result/Outcome
1.	Select Targeted Region at Frame 1	
2.	Identify Strongest Special Feature/Point (using Minimum Eigenvalue Algorithm	
3.	Bound object on the rectangle box	

4.6 Description of Data

This part will show the result obtained and analysis of any information collected, observed, generated, or created to validate original research findings for each method for cursor detection and cursor tracking.

4.6.1 Cursor Detection

For cursor detection, 70% of the total frames extracted from the rotation video are used to analyse each colour detection method. The full frame used to be used for analysis for each method is 497 frames. The method observed for colour detection in this project is RGB Colour Space, HSV Colour Space, and K-Mean Clustering. The purpose is to know which method is the best for colour detection. The best method for colour detection will be used in Blob Detection (Colour Feature) for cursor tracking purposes.

The data obtained from each colour detection method stated before is the value of centroid value of the detected and tracked object. The correct colour detection is taken based on the number of centroid value found at each frame. If the frame has more than one value, it is considered unsuccessful.

4.6.1.1 RGB Colour Space

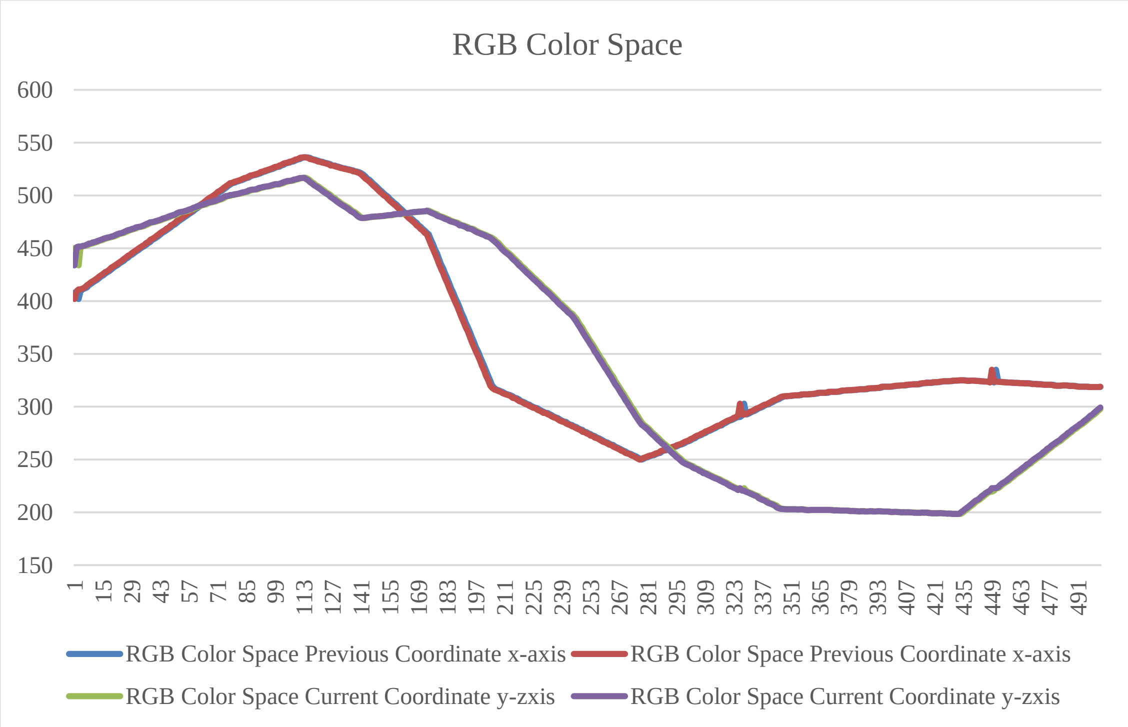


Figure 4.4 Data result using RGB Colour Space

Figure 4.4 is the graph plotting the value of the coordinate of the x-axis and y-axis for 497 frames extracted from the rotation video. Figure 4.4 shows that in most of the frames, only one centroid value is obtained, and the result shows that the value of the centroid between the current coordinates and previous coordinates of the x-axis and y-axis is close to each other. However, there is a spike on the graph, meaning there is an error in a specific frame.

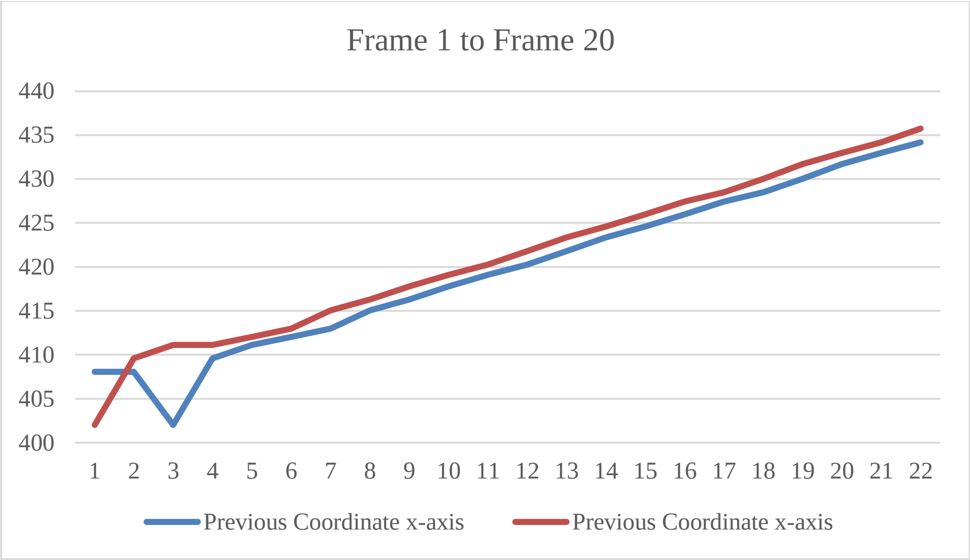


Figure 4.5 Data result using RGB Colour Space from Frame 1 to Frame 20

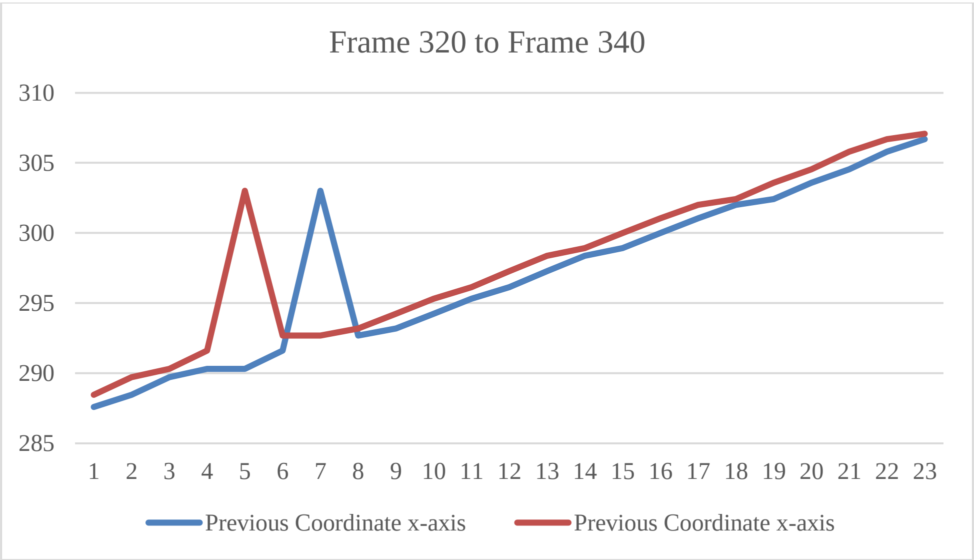


Figure 4.6 Data result using RGB Colour Space from Frame 320 to Frame 340

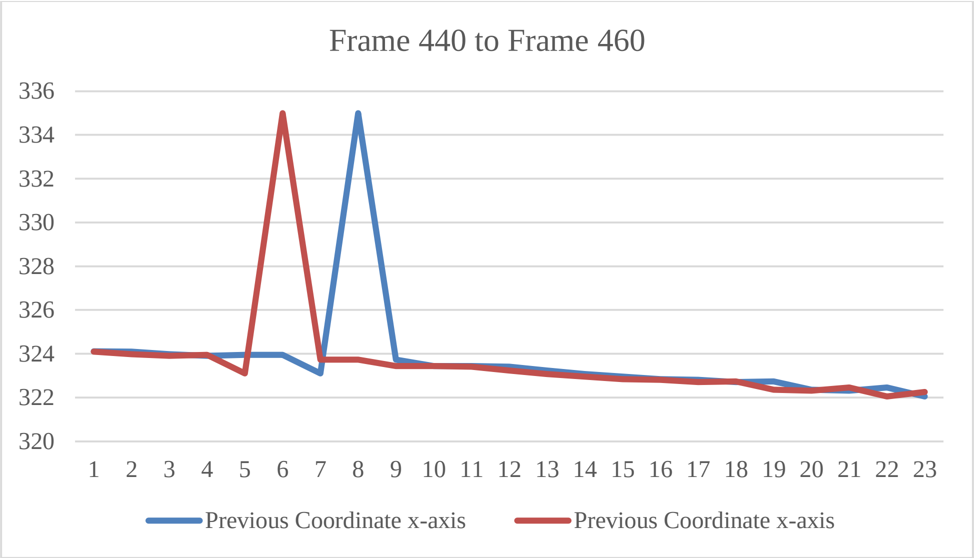


Figure 4.7 Data result using RGB Colour Space from Frame 440 to Frame 460

Figure 4.5 is a closer look where data from frame 1 to frame 20. Figure 4.6 is a closer look where data from frame 320 to frame 340. Figure 4.7 above is a closer look where data from frame 440 to frame 480. These images are taken as a closer of the x-axis of the graph in the Figure 4.4. These three images show that, there are 3 frames out of 497 frames which have more than one centroid value obtained because the line is not parallel to each other. The frame has more than one centroid value is Frame 2, Frame 324, and Frame 445.

4.6.1.2 HSV Colour Space

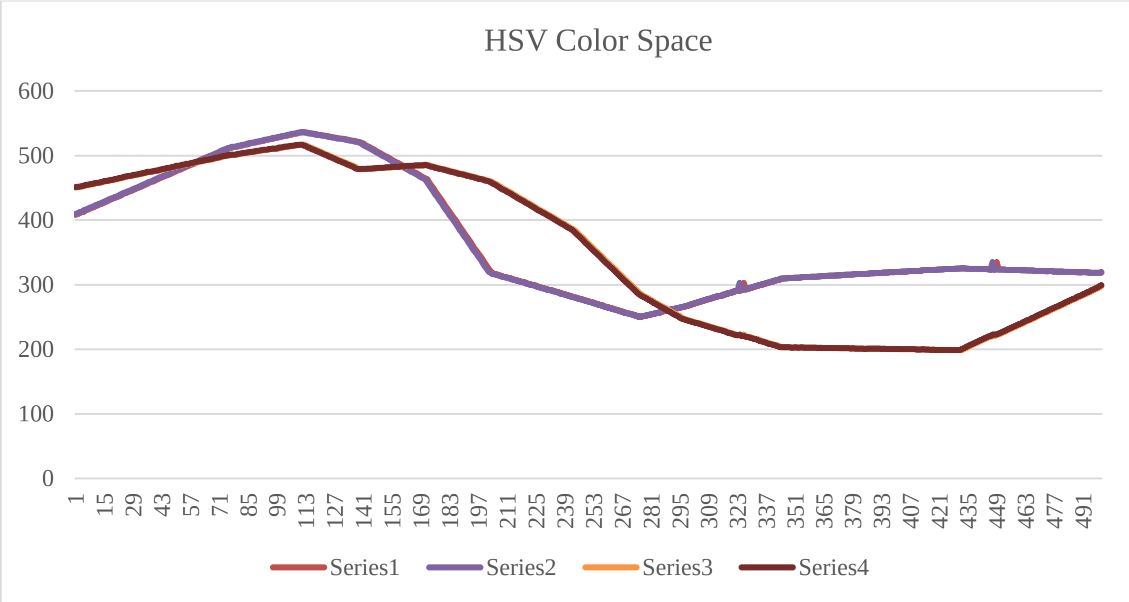


Figure 4.8 Data result using HSV Colour Space

Figure 4.8 is the graph plotting the value of the coordinate of the x-axis and y-axis for 497 frames extracted from the rotation video. Figure 4.8 shows that, in most of the frames, only one centroid value is obtained, and the result shows that the value of the centroid between the current coordinates and previous coordinates of the x-axis and y-axis is close to each other. However, there is a spike on the graph, means there is an error in a specific frame.

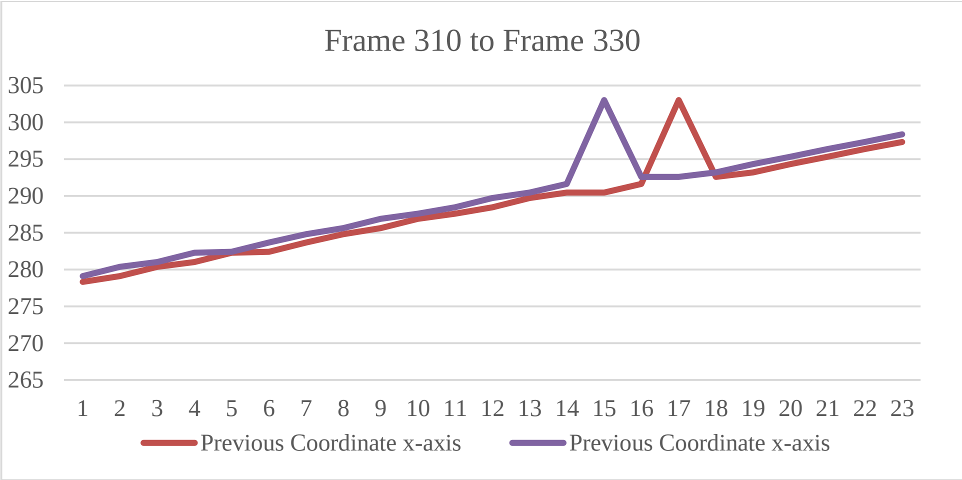


Figure 4.9 Data result using HSV Colour Space from Frame 310 to Frame 330

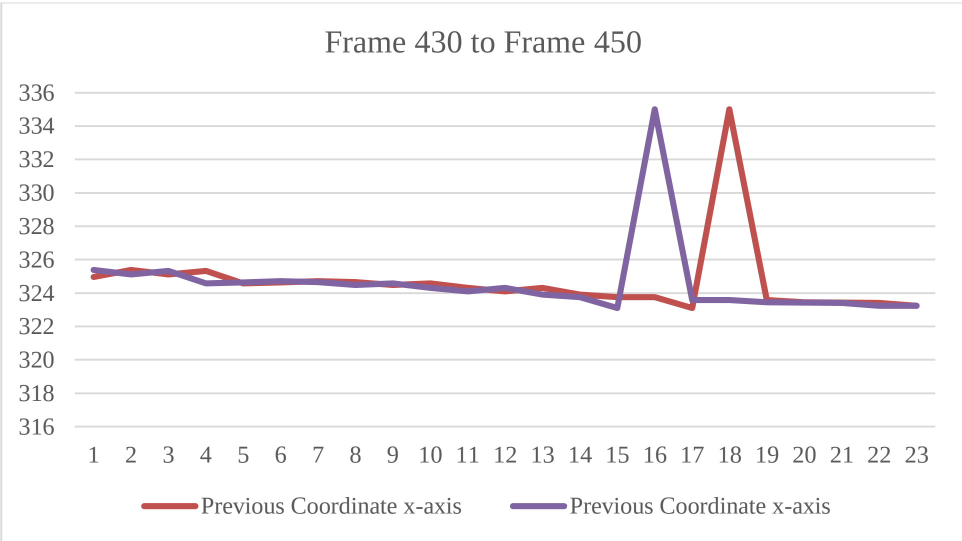


Figure 4.10 Data result using HSV Colour Space from Frame 430 to Frame 450

Figure 4.9 is a closer look where data from frame 310 to frame 330. Figure 4.10 is a closer look where data from frame 430 to frame 450. These images are taken as a closer of the x-axis of the graph in the Figure 4.8. These two images show that, there are 2 frames out of 497 frames which have more than one centroid value obtained because the line is not parallel to each other. The frame has more than one centroid value is Frame 324 and Frame 445.

4.6.1.3 K-Means Clustering

Table 4.7 Data result using K-Means Clustering

Frame	Number of centroid value					
	First Iteration		**Second Iteration**		**Third Iteration**	
	x-axis	**y-axis**	**x-axis**	**y-axis**	**x-axis**	**y-axis**
1	3	3	5	5	5	5
2	4	4	6	6	5	5
3	5	5	6	6	23	23
4	9	9	30	30	10	10
5	14	14	6	6	43	43
6	10	10	11	11	2	2
7	2	2	19	19	3	3
8	2	2	6	6	15	15
9	4	4	45	45	8	8
10	15	15	4	4	16	16

Based on Table 4.7, it shows that 10 frames are observed and the number of centroids detected is more than 1 at each frame. Three iterations are used to calculate the number of the centroid for each frame. The result shows that method K-Means Clustering is not suitable for detecting the green colour cursor.

It is because the K-Means clustering algorithm is an unsupervised algorithm used to segment the interest area from the background. It clusters, or partitions the given data into K-clusters or parts based on the K-centroids.

K-means is not suitable for strong illumination changes and it is complex compared to RGB and HSV color space. Every result is not stable. It is unstable because of the variety of lighting and objects that are moving rapidly and hard to find the center accurately.

4.6.1.4 Summary for Cursor Detection for each method

The number of frames used for each method is 497 frames. For this research project, the colour used for the cursor is green because it contrasts well with blood and tissue colour. So it is important to detect the green colour. The successfulness of the cursor detection is based on the number of centroid value found at each frame. If the frame has more than one value, it is considered as unsuccessful.

Figure 4.11 and Table 4.8 is the summary result for colour detection on 497 frames using three different types of previous method. The summary result from the chart images for each method is also explained in the previous subtopics. The parameter analyzed is the number of centroid values.

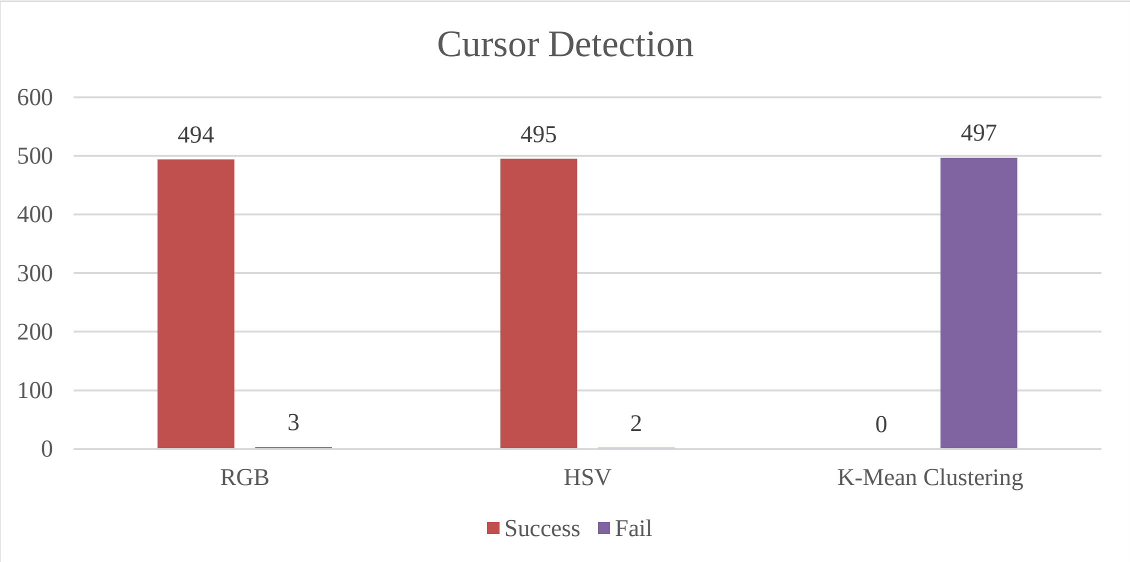


Figure 4.11 Summary for Cursor Detection for each method

Table 4.8 Summary for Cursor Detection for each method

Cursor Detection on 497 Frames	
Method Used	**Accuracy Percentage (%)**
RGB Colour Space	99.39
HSV Colour Space	99.59
K-Means Clustering	0

The result in Table 4.8 shows that for RGB Colour Space, 494 frames have only one centroid value while 3 frames have more than one centroid value. Therefore, for RGB Colour Space, the percentage of accuracy for cursor colour detection is 99.39%. For HSV Colour Space, 495 frames have only one centroid value while 2 frames have more than one centroid value. Therefore, for HSV Colour Space, the percentage of accuracy for cursor colour detection is 99.59%. For K-Means Clustering, all 497 frames have more than one centroid value. Therefore, for K-Means Clustering, the percentage of accuracy for cursor colour detection is 0%.

The final summary based on the result obtained and data analysis done shows that RGB Colour Space is the best method to detect green colour cursor for cursor detection. This method will be used in Blob Detection (Colour Feature) method for cursor tracking.

4.6.2 Cursor Tracking

4.6.2.1 Blob Detection (Colour Feature)

The data obtained from this method is the centroid value of the detected and tracked object. The successfulness of the detection and tracking using Blob Detection (Colour Feature) is based on the number of the centroid value. More than one centroid value is not a success. All the data is recorded manually in the Microsoft Excel to analyze the result. Three types of dataset in video format used are rotation, zoom in zoom out and liver deforming to do the analysis.

Figure 4.12 is the output for Blob Detection (Colour Feature) method. At the output, the detected object, the green colour cursor will be bounded by a bounding box. At the output, the detected and tracked object, the cursor will be bounded by a bounding box throughout the video. The system will return the centroid value for the detected object at each frame in the video. Figure 4.12 is the visual output for Blob Detection (Colour Feature) method. Figure 4.13 shows the output value obtained from the system representing the centroid value for the detected object.

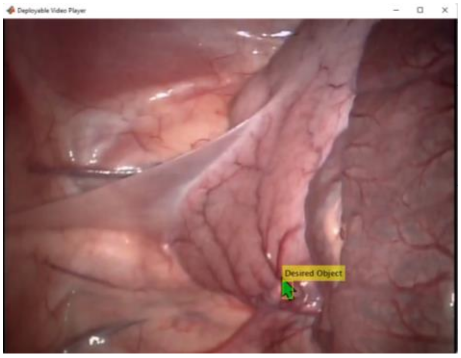


Figure 4.12 Visual output using Blob Detection (Colour Feature)

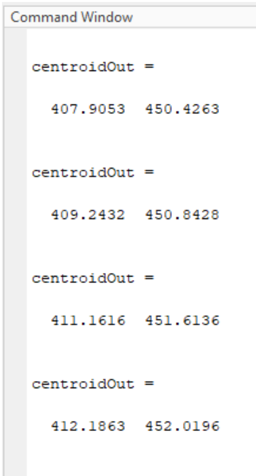


Figure 4.13 Output value obtained using Blob Detection (Colour Feature)

However, more specific observation is made to ensure the cursor detection's success using Blob Detection (Colour Feature). Observation is made by observing the centroid value in each frame at each video. The researcher observes whether each frames has only one value of centroid or not.

Figure 4.14 is the graph plotting the value of the coordinate of the x-axis and y-axis for all frames. For rotation video, all 711 frames are used to do the analysis. Figure 4.15 is the closer look where data from frame 1 to frame 10 are taken as a closer of the graph in Figure 4.14. The graph in Figure 4.14 shows that, at each frame, only one centroid value is obtained, and the result shows that the value of the centroid between the current coordinates and previous coordinates of the x-axis and y-axis is close to each other.

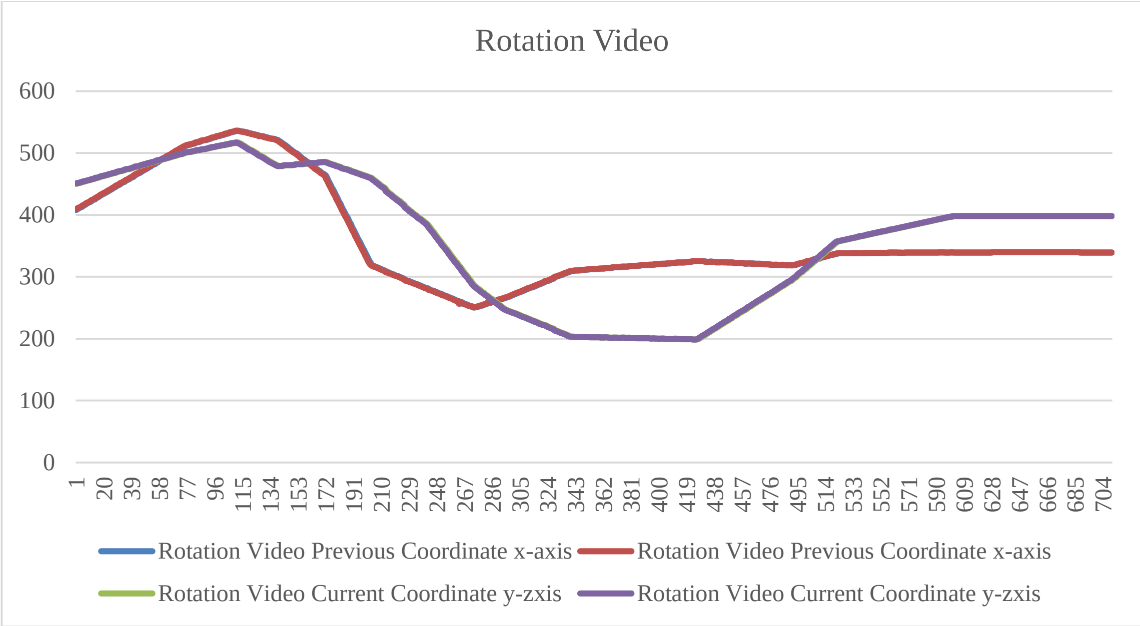


Figure 4.14 Data result for Rotation Video

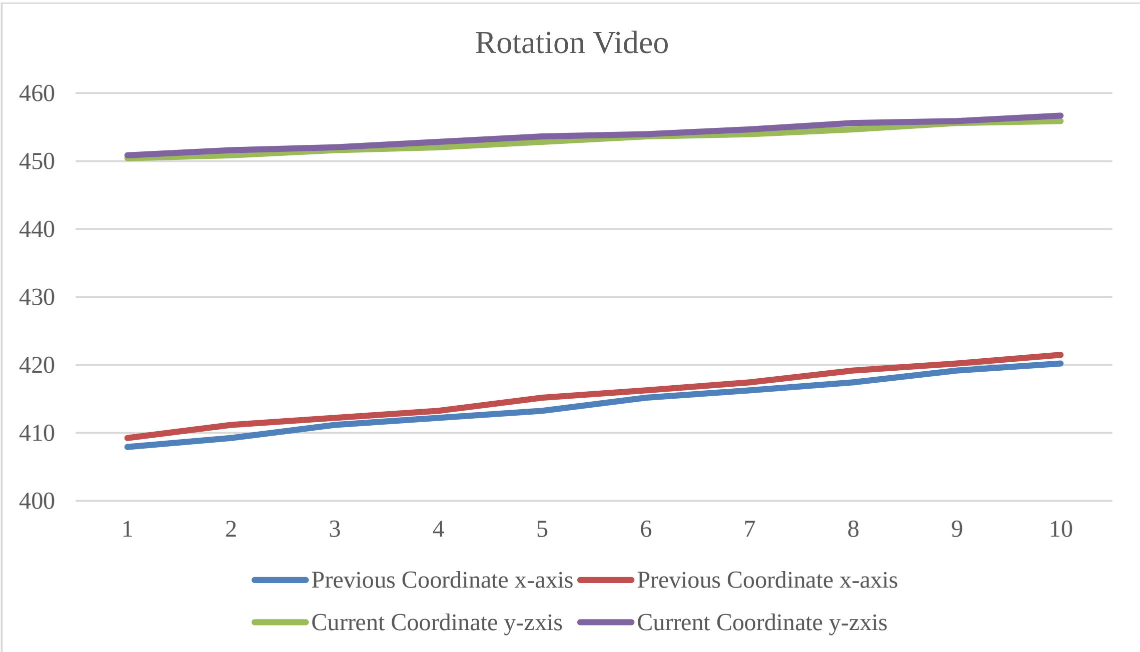


Figure 4.15 Data result for Rotation Video (Closer View)

Figure 4.16 is the graph plotting the value of the coordinate of the x-axis and y-axis for all frames. For zoom-in zoom-out video, all 1236 frames are used to do the analysis. Figure 4.17 is the closer look where data from frame 1 to frame 10 are taken as a closer of the graph in Figure 4.16. The graph in Figure 4.16 shows that, at each frames, only one centroid value is obtained and the result shows that the value of the centroid between the current coordinates and previous coordinates of the x-axis and y-axis is close to each other.

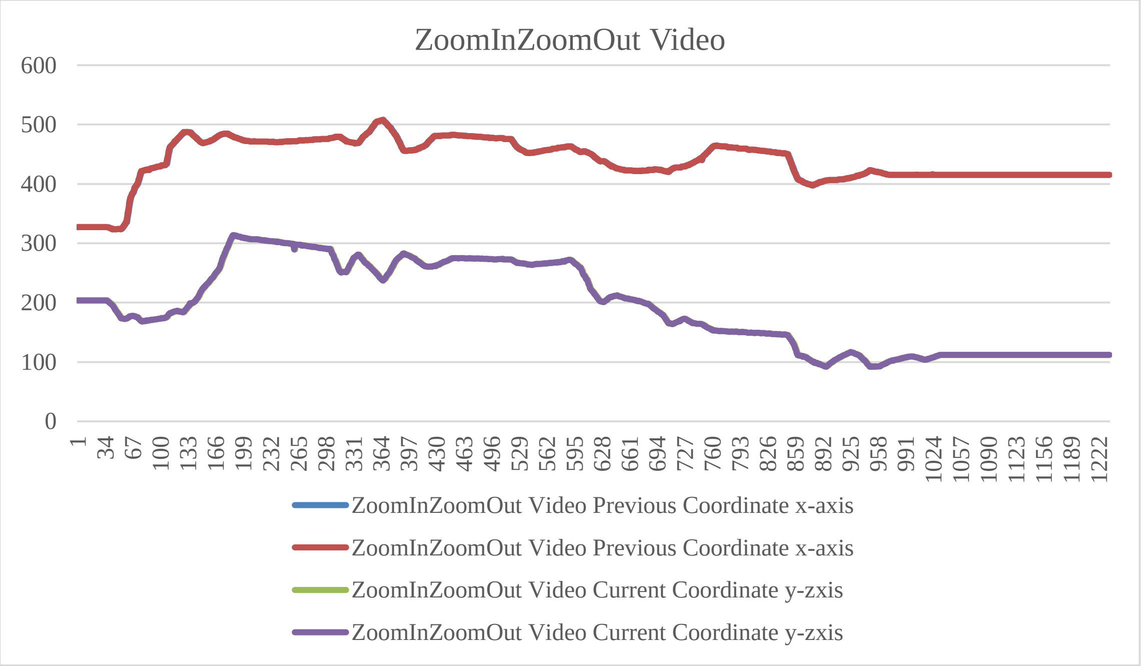


Figure 4.16 Data result for Zoom-In Zoom-Out Video

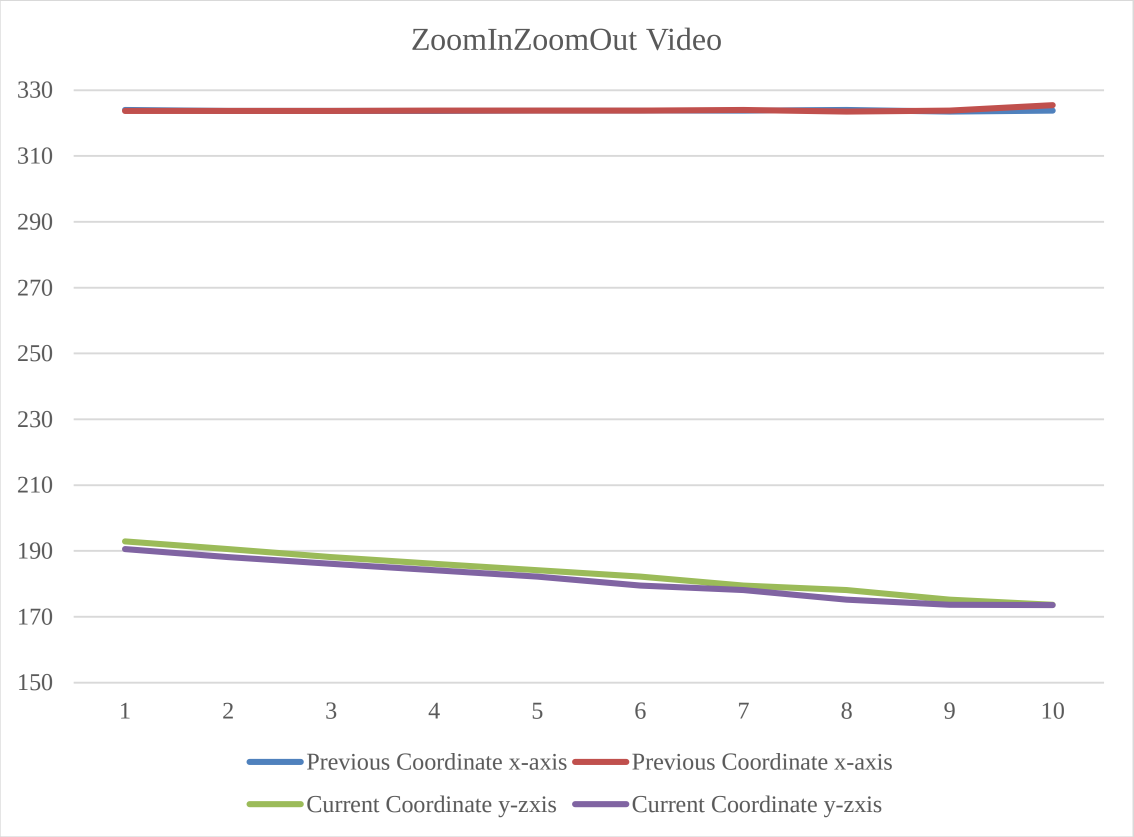


Figure 4.17 Data result for Zoom-In Zoom-Out Video (Closer View)

Figure 4.18 is the graph plotting the value of the coordinate of the x-axis and y-axis for all frames. For liver deforming video, all 804 frames are used to do the analysis. Figure 4.19 is the closer look where data from frame 1 to frame 10 are taken as a closer of the graph in Figure 27. The graph in Figure 4.18 shows that, at each frame, only one centroid value is obtained, and the result shows that the value of the centroid between the current coordinates and previous coordinates of the x-axis and y-axis is close to each other.

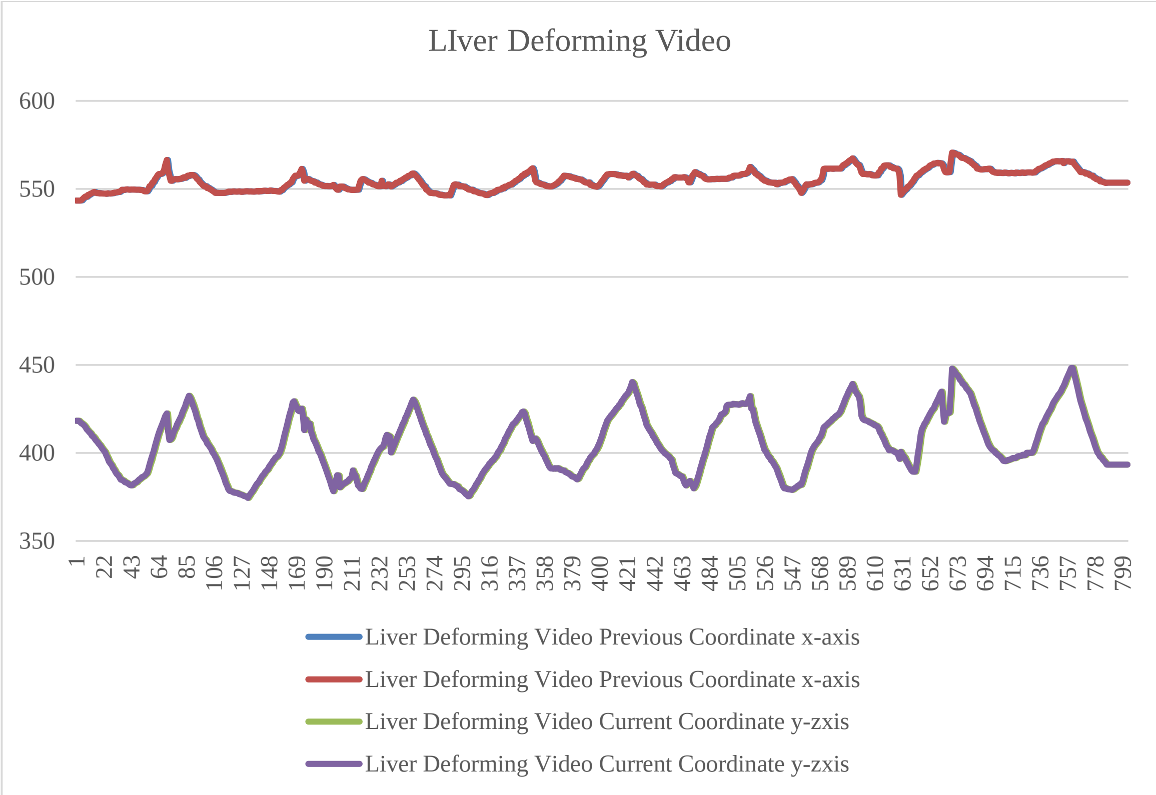


Figure 4.18 Data result for Liver Deforming Video

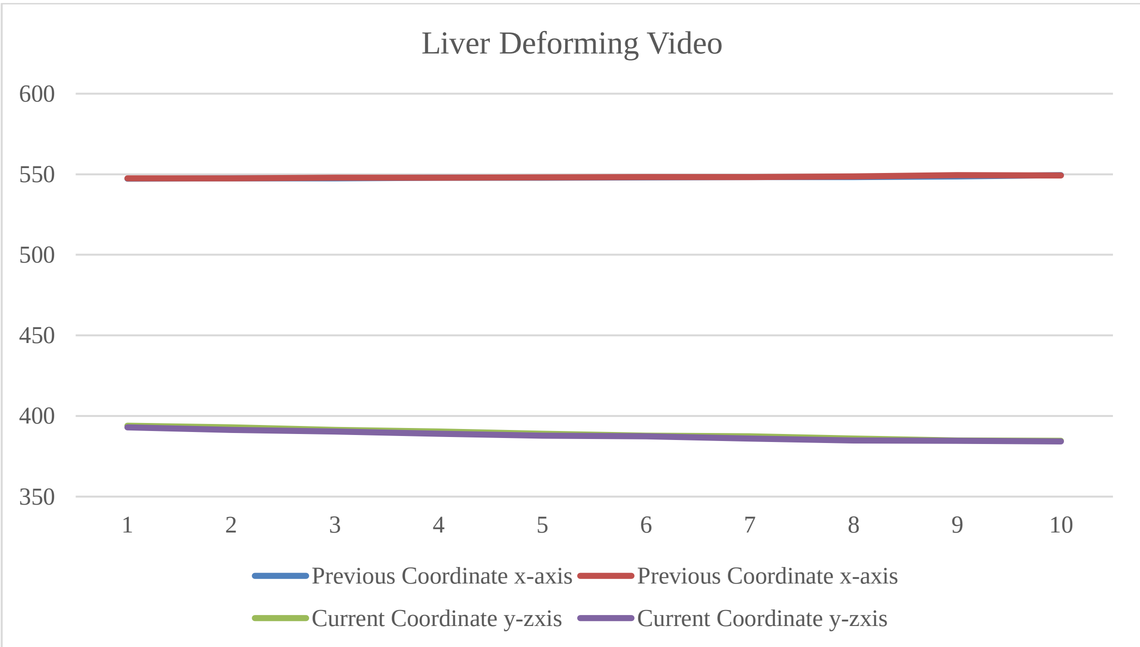


Figure 4.19 Data result for Liver Deforming Video (Closer View)

Figure 4.20 and Table 4.9 is the summary result for Blob Detection (Colour Feature) on the three sets of data: rotation video, zoom in zoom out video and liver deforming video. It also the summary result from the chart images for each video explained above. The parameter analyzed is the centroid value.

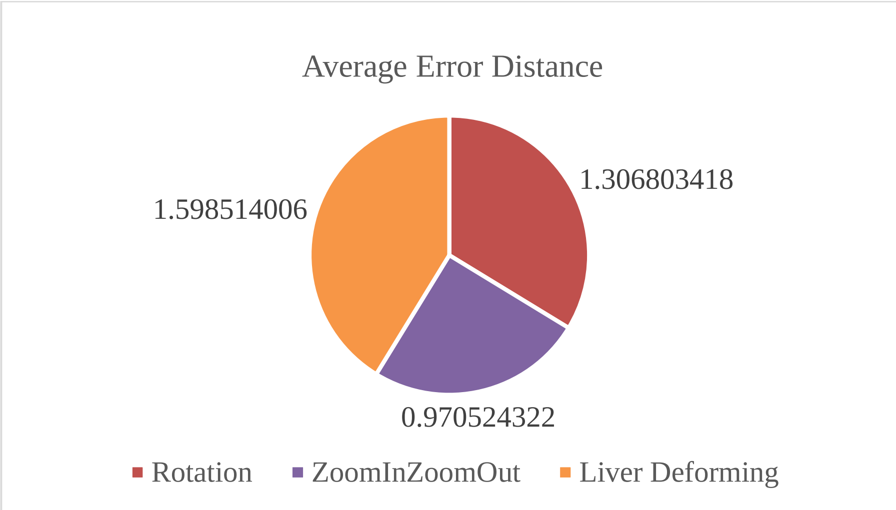


Figure 4.20 Summary result for Blob Detection (Colour Feature)

Table 4.9 Summary result for Blob Detection (Colour Feature)

	Rotation Video	**Zoom In Zoom Out Video**	**Liver Deforming Video**
	Centroid Value	**Centroid Value**	**Centroid Value**
Minimum Value	0	0	0
Maximum Value	5.256708449	9.559926712	15.10948524
Accuracy	100%	100%	100%
Number of frame observed	711 Frames	1236 Frames	804 Frames

Based on Figure 4.20 and Table 4.9, for rotation video, the number of frames observed is 711 frames. The minimum value of centroid between the current coordinates and previous coordinates is 0. This means the result shows that there are some of current coordinates and previous coordinates is the same. The maximum value of centroid between the current coordinates and previous coordinates is 5.256708449. There ae only one value of centroid at each frame, therefore, the accuracy of cursor detection and tracking using Blob Detection (Colour Feature) on rotation video is 100%, with an average value of 1.306803418.

For zoom in zoom out video, the number of frames observed is 1236. The minimum value of centroid between the current coordinates and previous coordinates is 0. This means the result shows that there are some current coordinates and previous coordinates is in the same coordinate. The maximum value of centroid between the current coordinates and previous coordinates is 9.559926712. There are only one value of centroid at each frame, therefore the accuracy of cursor detection and tracking using Blob Detection (Colour Feature) on zoom in zoom out video is 100%, with an average value of 0.970524322.

For liver deforming video, the number of frames observed is 804 frames. The minimum value of centroid between the current coordinates and previous coordinates is 0. This means the result shows that there are some current coordinates and previous coordinates is in the same coordinate. The maximum value of centroid between the current coordinates and previous coordinates is 15.10948524. There are only one value of centroid at each frame, therefore, the accuracy of cursor detection and tracking using Blob Detection (Colour Feature) on liver deforming video is 100%, with an average value of 1.598514006.

4.6.2.2 Feature Point Tracking

The data obtained from this method is the value of (x1, y1), (x2, y2), (x3, y3), and (x4, y4). That value represents the coordinate of each corner of the bounding box. The success is based on each frame, if it does produce more than one value at each coordinate per frame or does not have enough coordinates at each frame, it means there is no bounding box which leads to it failing to track the mark location. Thus, it will be considered as not accurate or the system fails to track. Second, the feature point detected in each frame must be equal to or more than two point matched. Three type of dataset in video format used: is rotation, zoom-in zoom-out, and liver deforming is used to do the analysis.

Figure 4.21 is the output for the Feature Point Tracking method. At the output, the detected and tracked object, the cursor, will be bounded by a bounding box throughout the video. Figure 4.21 is the visual output for the Feature Point Tracking method. Figure 4.22 is the output value obtained from the system representing the value of (x1, y1), (x2, y2), (x3, y3) and (x4, y4) represent the coordinate of each corner of the bounding box and also the value of matched point detected in each frame.

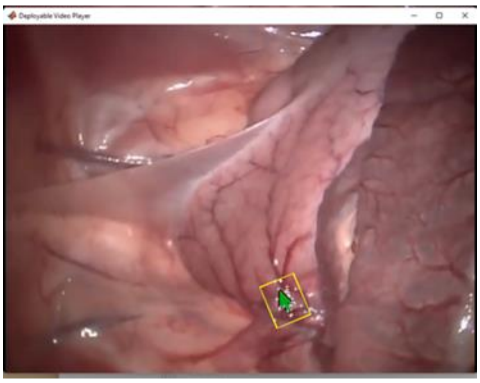


Figure 4.21 Visual output using Feature Point Tracking

```
Command Window
bboxPolygon =

  1×8 single row vector

  386.8603  420.5397  435.7598  420.9067  435.3478  475.7939  386.4484  475.4268


NumberofMatchedPoint =

    25


bboxPolygon =

  1×8 single row vector

  388.8520  420.8773  437.8212  421.8933  436.6808  476.8587  387.7116  475.8428


NumberofMatchedPoint =

    25


bboxPolygon =

  1×8 single row vector

  390.6683  421.3850  439.6205  422.8629  437.9616  477.8091  389.0095  476.3313


NumberofMatchedPoint =

    25
```

Figure 4.22 Output value obtained using Feature Point Tracking

Feature Point Tracking, the first condition or parameter to be observed is the number of (x1, y1), (x2, y2), (x3, y3), and (x4, y4) represents the coordinate of each corner of the bounding box, so it is crucial to observe whether this method meet the conditions set to determine the accuracy and successfulness for cursor detecting and tracking. Figure 4.23 is the graph plotting the value of detected and tracked of the coordinate of x-axis and y-axis for all frames. For rotation video, all 711 frames are used to do the analysis. For zoom in zoom out video, all 1236 frames are used to do the analysis. For liver deforming video, all 804 frames are used to do the analysis.

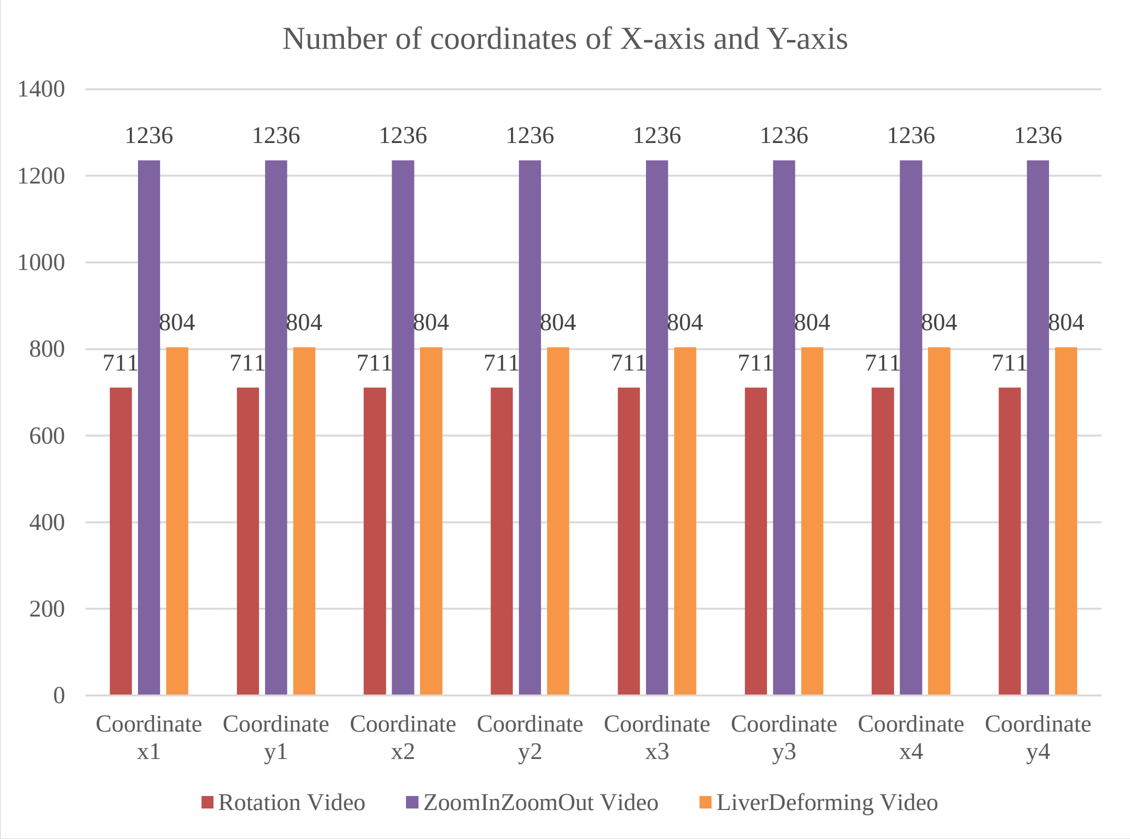


Figure 4.23 Graph plotting the value of detected and tracked of the coordinate of x-axis and y-axis for all frames

Based on the graph in Figure 4.23, for rotation video, the number of (x1, y1), (x2, y2), (x3, y3) and (x4, y4) detected is 711. Next, for zoom in zoom out video, the number of (x1, y1), (x2, y2), (x3, y3) and (x4, y4) detected is 1236. Next, for liver deforming, the number of (x1, y1), (x2, y2), (x3, y3) and (x4, y4) detected is 804. All three types of dataset observed, produce only one value at each coordinate per frame. The amount of each coordinate is the same as the number of original frames obtained from each video. For example, the number of frames for rotation video and the number of each coordinate (x1, y1), (x2, y2), (x3, y3) and (x4, y4) are the same, which is 711.

Feature Point Tracking, the second condition or parameter to be observed is that the feature point detected in each frame must be equal or more than two points matched, so it is crucial to observe whether this method meets the conditions set to determine the accuracy and successfulness of cursor detecting and tracking.

Table 4.10 Summary of number of feature points detected

Rotation Video		Liver Deforming Video		Zoom In Zoom Out Deforming	
Frame	**Number of Matched Point**	**Frame**	**Number of Matched Point**	**Frame**	**Number of Matched Point**
Frame 1	29	Frame 1	13	Frame 1	14
Frame 2-126	29	Frame 2-70	13	Frame 2-1236	14
Frame 127	26	Frame 71-201	12		
Frame 128-711	3	Frame 202-804	11		

Table 4.10 shows the number of feature points detected at each frame for each rotation video, zoom in zoom out video and liver deforming video. Based on the Table 4.10, for rotation video, the number of matched points at frame 1 during the current selection of cursor is 29 points. From frames 2 to frames 126, the number of matched points is 29 points. From frames 127, the number of matched points is 26 points. From frames 128 to frames 711, the number of matched points is 3 points.

For liver deforming video, the number of matched points at frame 1 during the cursor selection of is 13 points. From frames 2 to frames 70, the number of matched points is 13 points. From frames 71 to frame 201, the number of matched points is 12 points. Frames 202 to frames 804, the number of matched points is 11 points.

For zoom in zoom out video, the number of matched points at frame 1 during the selection of cursor is 14 points. From frames 2 to frames 1236, the number of matched points is 14 points.

Table 4.11 Summary of number of feature points detected

	Rotation Video	**Zoom in Zoom Out Video**	**Liver Deforming**
	Number of Matched Point	**Number of Matched Point**	**Number of Matched Point**
Minimum	3	14	11
Mode	3	14	11
Maximum	29	14	13
Accuracy	100%	100%	100%
Number of frame observed	711 Frames	1236 Frames	804 Frames

Table 4.11 is the summary result for the second condition or parameter on the three different data sets: rotation video, zoom in zoom out video and liver deforming video. For rotation video, the number of frames observed is 711 frames. The minimum number of matched points is 3. The maximum number of matched points is 29. The mode for matched point for rotation video is 3. This result shows that for rotation video, the accuracy and successfulness of cursor detecting and tracking is 100%. It because the number of matched points at each frame is more than two.

For zoom in zoom out video, the number of frames observed is 1236 frames. The minimum number of matched point is 14. The maximum number of matched point is 14. The mode for matched points for zoom in zoom out video is 14. This results shows that for zoom in zoom out video, the accuracy and successfulness of cursor detecting and tracking is 100%. It is because the number of matched points at each frame is more than two.

For liver deforming video, the number of frames observed is 804 frames. The minimum number of matched points is 13. The maximum number of matched points is 11. The mode for matched point for liver deforming video is 11. This result shows that for liver deforming video, the accuracy and successfulness of cursor detecting and tracking is 100%. It is because the number of matched points at each frame is more than two.

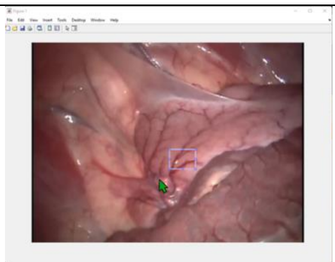
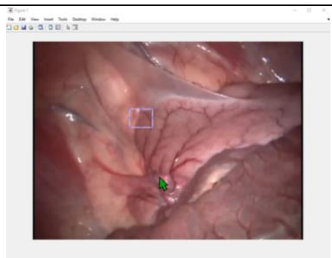
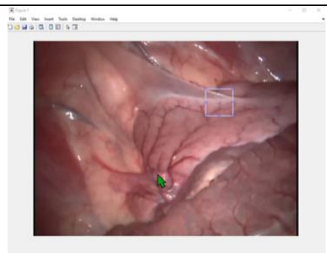
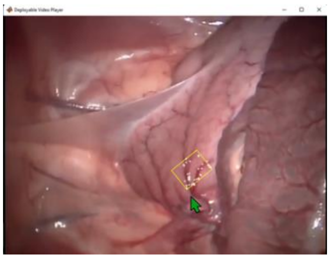
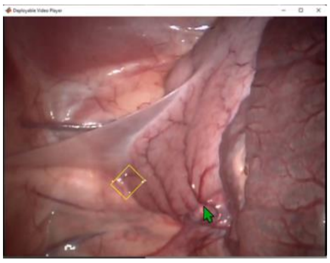
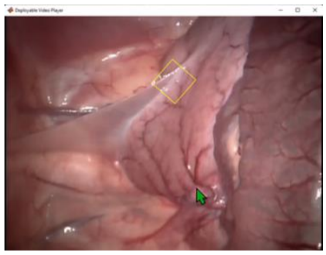
4.7 Analysis of Data

Although the facts based on the result shows that both method have 100% accuracy to detect and track the cursor, the Feature Point Tracking is the best method compared to Blob Detection in terms of flexibility for the user to choose region of interest (ROI).

The method of detecting and tracking objects by using Blob Detection (Colour Feature), is limited to one object as it detects the green colour so, it can only track the cursor. On the other hand, by using Feature Point Tracking, the user can freely choose any region of interest and also track the cursor if the user wants it to be tracked.

To support the statement above, the researcher analysed by selecting three different region of interest (ROI) other than the cursor. A rotation video is used for this analysis. The number of frames used for this analysis is 711 for each region of interest (ROI).

Table 4.12 Selection of different Region of Interest (ROI)

Method	**Other Region of Interest (ROI)**		
	Region of Interest (ROI) 1	**Region of Interest (ROI) 2**	**Region of Interest (ROI) 3**
Feature Point Tracking			
	Result: 	Result: 	Result: 

Based on the graph in Figure 4.24, for the region of interest (ROI) 1, the number of (x1, y1), (x2, y2), (x3, y3), and (x4, y4) detected is 711. Next, for the region of interest (ROI) 2, the number of (x1, y1), (x2, y2), (x3, y3), and (x4, y4) detected is 711. Next, for the region of interest (ROI) 1, the number of (x1, y1), (x2, y2), (x3, y3), and (x4, y4) detected is 711. All three types of region of interest (ROI) observed, produce only one value at each coordinate per frame. The amount of each coordinate is the same as the number of original frames obtained from each video. For example, the number of frames for rotation video and the number of each coordinate (x1, y1), (x2, y2), (x3, y3), and (x4, y4) are the same, which is 711.

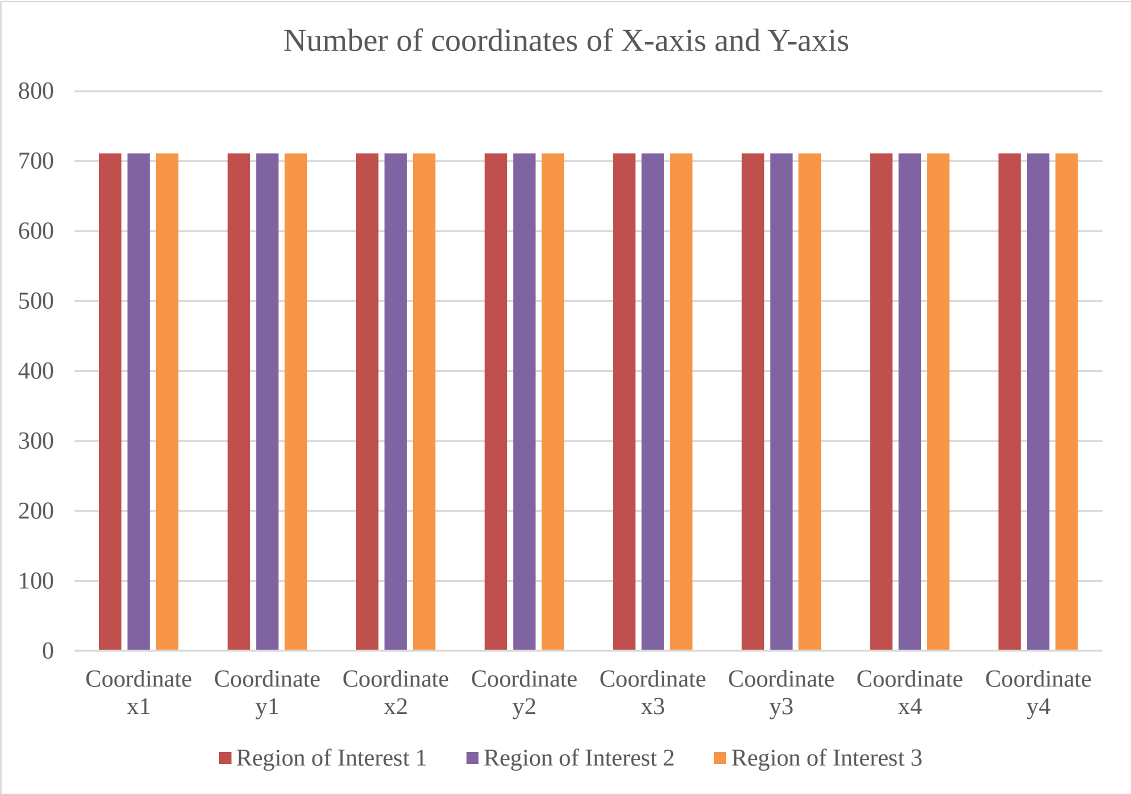


Figure 4.24 Graph plotting the value of detected and tracked of the coordinate of x-axis and y-axis for different ROI

Table 4.13 shows the number of feature points detected at each frame for each selected region of interest (ROI).

Table 4.13 Summary for number of feature point detected

Region of Interest (ROI) 1		**Region of Interest (ROI) 2**		**Region of Interest (ROI) 3**	
Frame	**Number of Matched Point**	**Frame**	**Number of Matched Point**	**Frame**	**Number of Matched Point**
Frame 1	25	Frame 1	11	Frame 1	26
Frame 2-252	25	Frame 2-711	11	Frame 2-45	26
Frame 253-509	24			Frame 46-127	25
Frame 510-590	23			Frame 128	21
Frame 591-711	22			Frame 129-711	19

Based on Table 4.13, for region of interest 1, the number of matched point at frame 1 during the cursor selection is 25 points. From frames 2 to frames 252, the number of matched points is 25 points. From frames 253 to frames 509, the number of matched points is 24 points. From frames 510 to frames 590, the number of matched points is 23 points. From frames 591 to frames 711, the number of matched point is 22 points.

For the region of interest 2, a number of the matched point at frame 1 during the selection of cursor is 11 points. In frames 2 to frames 711, the number of matched point is 11 points.

For the region of interest 3, a number of the matched point at frame 1 during the selection of cursor is 26 points. From frames 2 to frames 45, the number of matched points is 26 points. From frames 46 to frames 127, the number of matched points is 25 points. In frames 128, the number of matched points is 21 points. From frames 129 to frames 711, the number of matched points is 19 points.

Table 4.14 Summary for number of feature point detected

	Region of Interest (ROI) 1	**Region of Interest (ROI) 2**	**Region of Interest (ROI) 3**
	Number of Matched Point	**Number of Matched Point**	**Number of Matched Point**
Minimum	22	11	16
Mode	25	11	16
Maximum	25	11	26
Accuracy	100%	100%	100%
Number of frame observed	711 Frames	711 Frames	711 Frames

Table 4.14 is the summary result for the second condition or parameter on the three different data sets: which is Region of Interest 1, Region of Interest 2 and Region of Interest 3. For Region of Interest 1, the number of frames observed is 711. The minimum number of matched points is 22. The maximum number of matched points is 25. The mode for matched point for rotation video is 25. This result shows that of Region of Interest 1, the accuracy and successfulness of cursor detecting and tracking is 100%. It is because the number of the matched points at each frame is more than two.

For Region of Interest 2, the number of frames observed is 711. The minimum number of matched points is 11. The maximum number of matched points is 11. The mode for matched point for zoom in zoom out video is 11. This result shows that for

Region of Interest 2, the accuracy and successfulness of cursor detecting and tracking is 100%. It is because the number of the matched points at each frame is more than two.

For the Region of Interest 3 video, the number of frames observed is 711. The minimum number of matched points is 16. The maximum number of matched points is 26. The mode for matched point for liver deforming video is 16. This results shows that for Region of Interest 3, the accuracy and successfulness of cursor detecting and tracking is 100%. It is because the number of the matched points at each frame is more than two.

4.8 Summary for Chapter 4

The experimental results of the suggested system have been critically evaluated in this chapter. A total of 497 frames are used for cursor detection to test the system, compare each technique, and record the results. The frames are extracted from rotation video. This project reviewed and analysed recorded data in order to determine the best colour detection method for use in Blob Detection (Colour Space). To analyze the results, all of the data is manually recorded into Microsoft Excel. Researchers discovered that the study's hypothesis can be accepted based on the collected and analysed evidence. Three type of colour detection presented different results, with HSV highlighting more positive aspects and being more ideal for green colour cursor detection. HSV Colour Space can detect the green colour in each frame better compared to other methods. Therefore, HSV Colour Space will be chosen and will be used in Blob Detection (Colour Space).

Three types of datasets are utilized for cursor tracking. It is a rotation video, zoom in zoom out video, and liver deforming video, each consisting of 711 frames, 1236 frames, and 804 frames, respectively. To assess the results, all of the data is manually recorded into Microsoft Excel. Researchers discovered that the study's hypothesis could be accepted based on the evidence that was collected and analyzed. Both Blob Detection (Colour Feature) and Feature Point Tracking have 100% accuracy detecting and tracking the cursor. However, by using Blob Detection (colour feature), it is only limited to one object as it detects the green colour so it can only track the cursor. On the other hand, by using Feature Point Matching, the user can freely choose any region of interest and also track the cursor if the user wants it to be tracked.

CHAPTER 5

CONCLUSION

5.1 Introduction

This chapter focuses mainly on the research's conclusions. It also gives some recommendations for future advancements in medical detection and tracking applications.

5.2 Conclusion

In conclusion, in recent decades, medical endoscopy has evolved as a crucial tool for reducing mistakes during diagnostic procedures and mistakes during surgery in the abdomen, joints, and other body parts. With recent advancements in telemedicine technology, it is now possible to do closed-abdomen surgery on a remote patient's moving organ with a robotic instrument, reducing patient trauma and negative effects.

The experimental results of the proposed system have been critically evaluated in this chapter. To solve the problem statement mentioned, numerous methods and various types of data have been examined in this study. The purpose of this research aims to detect and track the mark location pointed by the cursor. There are three types of colour detection methods utilised for green colour detection which are RGB Colour Space, HSV Colour Space, and K-Means Clustering. Two cursor tracking methods are utilized for cursor tracking. Blob Detection (Colour Feature) and Feature Point Tracking are the two techniques. To determine the researcher's success, the researcher must return to this project objective to measure the system's success. MATLAB R2021b Software was used to complete all the methods and obtain all the data analysis results.

The first goal was achieved, with three distinct types of colour detection producing varied results, with HSV highlighting more positive aspects and more ideal for green colour cursor detection. The first objective requires the determination of the number of the centroid of the green colour cursor, which requires using an image colour detection approach.

The second objective is success. The system is capable of resolving the problem statement and achieving the goal. The system uses Blob Detection and Features Point Matching to detect and track the mark's location. The second goal requires using the mark locations tracking approach to determine the cursor's centroid values and the number of points matched in each frame. Based on the result, we found that Feature Point Matching is the best method compared to Blob Analysis in terms of flexibility for the user to choose the mark location.

Finally, the Blob Detection method was successfully measured and validated by observing the number of centroids and the distance between current and previous coordinated. Feature Point Matching method successfully measured and validated by observing the number of centroids and the number of points matched.

In conclusion, all of the outcomes were documented, the final product was outstanding, and everything was completed on time. All three objectives were met or surpassed in this project. This project has completed beautifully and on time. Overall, the objectives of this study were met by experimental work involving trial and error, which was completed with complete success.

5.3 Planning Management

For the financial planning of this research project, the researcher used MATLAB R2021b Software, an open-source software. The researcher had RM0 throughout this investigation since he used a vision-based system and image processing method, as demonstrated in Table 5.1.

Table 5.1 Planning Management

No	Items	Price (RM)	Quantity	Price (RM)
1.	MATLAB R2021b Software	RM0	1	RM0
2.	-	-	-	-
	Total Cost (RM)			RM0

5.4 Recommended and Future Research Directions

The findings of this study identified several areas for future research that can be improved upon based on the analysis conducted. Future recommendations are crucial for the next developer to recognize project constraints and problems to design a better system. This project has the potential to benefit society in Malaysia and around the world in terms of medical services.

For cursor detection, the future algorithm should be able to detect the required place or object accurately to improve detection accuracy. Second, the proposed future system should be capable of applying the proposed algorithm in a more complex environment, such as one with changing illumination.

For cursor tracking, the proposed algorithm should be implemented with various input data, including variable lighting, motion blur, lack of image texture, occlusion, and quick object movements in the proposed future system. Second, tracking algorithms can track objects at various scales and perspectives.

5.5 Chapter Summary

To summarize Chapter 5, the project’s goals were met and the project was completed effectively. The project’s success is evidenced and proved by the performance of the results.

REFERENCES

[1] R. A. Karim, N. F. Zakaria, M. A. Zulkifley, M. M. Mustafa, I. Sagap, and N. H. Md Latar, “Telepointer technology in telemedicine : A review,” *Biomed. Eng. Online*, vol. 12, no. 1, Mar. 2013, doi: 10.1186/1475-925X-12-21.

[2] “Colonoscopy prep drink: What it is, tips, and more.” https://www.medicalnewstoday.com/articles/colonoscopy-prep-drink#summary (accessed Jan. 14, 2022).

[3] G. S. D. Gordon *et al.*, “Quantitative phase and polarisation endoscopy applied to detection of early oesophageal tumourigenesis,” *J. Biomed. Opt.*, vol. 24, no. 12, pp. 1–13, Dec. 2018, doi: 10.1117/1.JBO.24.12.126004.

[4] A. Schneider, D. Wilhelm, U. Bohn, A. Wichert, and H. Feussner, “An evaluation of a surgical telepresence system for an intrahospital local area network,” *J. Telemed. Telecare*, vol. 11, no. 8, pp. 408–413, 2005, doi: 10.1177/1357633X0501100806.

[5] A. Schneider, D. Wilhelm, U. Bohn, A. Wichert, and H. Feussner, “An evaluation of a surgical telepresence system for an intrahospital local area network,” *J. Telemed. Telecare*, vol. 11, no. 8, pp. 408–413, 2005, doi: 10.1177/1357633x0501100806.

[6] S. Voros, J. A. Long, and P. Cinquin, “Automatic Detection of Instruments in Laparoscopic Images: A First Step Towards High-level Command of Robotic Endoscopic Holders:,” *http://dx.doi.org/10.1177/0278364907083395*, vol. 26, no. 11–12, pp. 1173–1190, Jul. 2016, doi: 10.1177/0278364907083395.

[7] D. B. Douglas, R. E. Douglas, C. Wilke, D. Gibson, J. Boone, and M. Wintermark, “A systematic review of 3D cursor in the medical literature,” *AIMS Electron. Electr. Eng.*, vol. 2, no. 1, pp. 1–11, Mar. 2018, doi: 10.3934/ELECTRENG.2018.1.1.

[8] N. Rajendra, R. N. Shree K, B. Rao, and A. Professor, “Cursor Movement by Object Detection Based on Image Processing,” *Int. J. Res. Publ. Rev.*, vol. 2, no. 5, pp. 18–22, 2021, Accessed: Jan. 14, 2022. [Online]. Available: www.ijrpr.com.

[9] H. Park, “A Method for Controlling Mouse Movement using a Real-Time Camera.”

[10] A. Shastrakar, J. Raman, M. Paul, N. Ramteke, and P. V. Sathwane, “Cursor Movement Control Using Colour Detection,” 2018.

[11] C. Cai and M. Lin, “Tire pattern similarity detection based on template matching and LBP,” *2019 IEEE 3rd Int. Conf. Electron. Inf. Technol. Comput. Eng. EITCE 2019*, pp. 419–423, Oct. 2019, doi: 10.1109/EITCE47263.2019.9095051.

[12] R. S. Sembiring, S. Efendi, and S. Suwilo, “Improving the accuracy of old and young face detection in the template matching method with Fuzzy Associative Memory (FAM),” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 725, no. 1, Jan. 2020, doi: 10.1088/1757-899X/725/1/012117.

[13] M. Noorjahan and A. Punitha, “An Efficient and Fast Partial Template Matching Technique-Enhancement in Normalized Cross Correlation,” *Int. J. Eng. Adv. Technol.*, no. 9, pp. 2249–8958, 2019, doi: 10.35940/ijeat.F8362.109119.

[14] P. Bose and S. Kumar Bandyopadhyay, “Human Face and Facial Parts Detection using Template Matching Technique,” *Int. J. Eng. Adv. Technol.*, no. 9, pp. 2249–8958, doi: 10.35940/ijeat.D6689.049420.

[15] M. D. Yusuf, R. Kusumanto, Y. Oktarina, T. Dewi, and P. Risma, “BLOB Analysis for Fruit Recognition and Detection,” *Comput. Eng. Appl. J.*, vol. 7, no. 1, pp. 25–36, 2018, doi: 10.18495/comengapp.v7i1.237.

[16] G. Salvi, “An automated vehicle counting system based on blob analysis for traffic surveillance,” *Proc. 2012 Int. Conf. Image Process. Comput. Vision, Pattern Recognition, IPCV 2012*, vol. 1, pp. 397–402, 2012.

[17] G. S. Lin and T. S. Tsai, “A face tracking method using feature point tracking,” *Proc. - 3rd Int. Conf. Inf. Secur. Intell. Control. ISIC 2012*, pp. 210–213, 2012, doi: 10.1109/ISIC.2012.6449743.

[18] S. Ratanasanya, J. Polvichai, and B. Sirinaovakul, “Feature point matching with matching distribution,” *Adv. Intell. Syst. Comput.*, vol. 361, pp. 9–18, 2015, doi: 10.1007/978-3-319-19024-2_2.

[19] Y. Chen, G. Wang, and L. Wu, “Research on feature point matching algorithm improvement using depth prediction,” *J. Eng.*, vol. 2019, no. 23, pp. 8905–8909, 2019, doi: 10.1049/joe.2018.9142.

[20] “BLOB Analysis (Introduction to Video and Image Processing) Part 1.” http://what-when-how.com/introduction-to-video-and-image-processing/blob-analysis-introduction-to-video-and-image-processing-part-1/ (accessed Jun. 23, 2022).

[21] “Introduction To Feature Detection And Matching | by Deepanshu Tyagi | Data Breach | Medium.” https://medium.com/data-breach/introduction-to-feature-detection-and-matching-65e27179885d (accessed Jun. 23, 2022).

[22] M. J. Carlotto, “Effect of errors in ground truth on classification accuracy,” *Int. J. Remote Sens.*, vol. 30, no. 18, pp. 4831–4849, 2009, doi: 10.1080/01431160802672864.

[23] A. Schmidt and K. G. Siebert, “Euclidian Distasnce Geometry and Applications,” vol.

LXX, no. 1, pp. 105–122, 2001.

[24] P. Barrett, "Euclidean Distance Whitepaper," *Tech. Whitepaper Ser. 6*, p. 26, 2005, [Online]. Available: http://www.pbarrett.net/techpapers/euclid.pdf.

APPENDIX A
CODING FOR CURSOR DETECTION

1. Coding for Cursor Detection by using Frames Extraction

```
%Coding for cursor detection (Final Year Project) [Extract Frame]
%Muhamad Hamizi Zaidi Bin Mohd Jonhanis
%EA18008

clc
clear all
close all

a=VideoReader('C:\Users\Acer\Desktop\PSM RESEARCH\Picture for
PSM\video\video6275992378272646564.mp4');

for img=1:a.NumFrames*(1/100)

    filename=strcat('frame',num2str(img),'.jpg');
    b=read(a,img);
    imwrite(b,filename);
end
```

2. Coding for Cursor Detection by using RGB Colour map Array

```
%Coding for cursor detection (PSM 1)- RGB COlour Specifier
%Muhamad Hamizi Zaidi Bin Mohd Jonhanis
%EA18008
clc
close all
clear all

%% to read the input image
I=imread('C:\Users\Acer\Desktop\PSM RESEARCH\2.0 ZOOM IN ZOOM
OUT\Coding\Extract\frame345.jpg');
%to read the input image
figure,imshow(I);
%to display the output image
title('Original Image')
%to give title to the output image

G=rgb2gray(I);
%to convert rgb to grey
figure,imshow(G);
%to display the output image
title('Grey-scale Image')
%to give title to the output image

%% detect green color using RGB color space
A=I(:,:,2);
figure,imshow(A);
%to display the output image
title('Detect Green Color')
%to give title to the output image
```

```
F=imsubtract(A,G);
%to filter other color than green
F=im2bw(F, 0.18);
%to make the image black and white scale
figure,imshow(F);
%to display the output image
title('Cursor')
%to give title to the output image

%% show the bounding box
s=regionprops(F,'BoundingBox','Centroid');
cent=cat(1,s.Centroid)
bbox=cat (1,s.BoundingBox);
RGBb=insertShape(I,'FilledRectangle',bbox,'Color','yellow','Opacity',0.3);
RGBb=insertObjectAnnotation(RGBb,'rectangle',bbox,'Cursor','TextBoxOpacity',0
.9,'FontSize',18);
figure, imshow(RGBb)
title('Cursor Detection')
```

3. Coding for Cursor Detection by using HSV Colour map Array

```
%Coding for cursor detection (Final Year Project) [HSV Method]
%Muhamad Hamizi Zaidi Bin Mohd Jonhanis
%EA18008
clc
close all
clear all

%% to read the input image
RGB=imread('C:\Users\Acer\Desktop\PSM RESEARCH\CODING\Extract frame
70%\frame1.jpg');
figure, imshow(RGB); %to display the output image
title ('RGB image')

%% conversion from RGB to HSV
HSV = rgb2hsv(RGB);
figure, imshow(HSV); %to display the output image
title ('HSV image')

 % Define thresholds for channel 1 based on histogram settings
    channel1Min = 0.262;
    channel1Max = 0.437;

    % Define thresholds for channel 2 based on histogram settings
    channel2Min = 0.183;
    channel2Max = 1.000;

    % Define thresholds for channel 3 based on histogram settings
    channel3Min = 0.237;
    channel3Max = 1.000;

%% Create mask based on chosen histogram thresholds
sliderBW = (HSV(:,:,1) >= channel1Min ) & (HSV(:,:,1) <= channel1Max) & ...
    (HSV(:,:,2) >= channel2Min ) & (HSV(:,:,2) <= channel2Max) & ...
    (HSV(:,:,3) >= channel3Min ) & (HSV(:,:,3) <= channel3Max);
BW = sliderBW;
```

```
figure,imshow(BW);
%to display the output image
title('Conversion Image')

%1. filling holes to fill in holes
BW=imfill(BW,'holes');
%2. area opening
BW=bwareaopen(BW,50);
figure,imshow(BW)
title ('Filtered Image')

%% show the box
s=regionprops(BW,'BoundingBox','Centroid');
cent=cat(1,s.Centroid) %Store the x- and y-coordinates of the centroids into
a two-column matrix.
bbox=cat (1,s.BoundingBox);
KOTAK=insertShape(RGB,'FilledRectangle',bbox,'Color','yellow','Opacity',0.3);
KOTAK=insertObjectAnnotation(KOTAK,'rectangle',bbox,'Cursor','TextBoxOpacity'
,0.9,'FontSize',18);
figure, imshow(KOTAK)
title('Cursor Detection')
```

4. Coding for Cursor Detection by using K-Means Clustering

```
%Coding for cursor detection (Final Year Project) - [k-Means Clustering]
%Muhamad Hamizi Zaidi Bin Mohd Jonhanis
clc
close all
clear
%% to read the input image
Img=imread('C:\Users\Acer\Desktop\PSM RESEARCH\CODING\Extract frame
70%\frame3.jpg'); %to read the input image
figure, imshow(Img);
%to display the output image
title ('Original image')
grayImage=rgb2gray(Img);
figure, imshow(grayImage,[]);
%to display the output image
title ('Grey image')

%% do image segmentation by using k-means clustering algorithma
numberOfClasses=6;
indexes=kmeans(grayImage(:),numberOfClasses);
classImage=reshape(indexes,size(grayImage));
figure, imshow(classImage,[]);
%to display the output image
colormap(gcf,parula)
title ('Class image')

%% do selection to get the object region
class=zeros(size(grayImage));
area=zeros(numberOfClasses,1);
for n=1:numberOfClasses
    class(:,:,n)=classImage==n;
    area(n)=sum(sum(class(:,:,n)));
end
[~,min_area]=min(area);
```

```
%do object choosing based on the smallest region area
object=classImage==min_area;
figure, imshow(object)
title ('Object Region image')

%% do morfologi to eliminate noise (titik titik kecik) to make the
segmentation nice
bw=medfilt2(object,[3 2]);
figure, imshow(bw)
title ('Filtered image')

%% do extraction boundng box to the object
s=regionprops(bw,'BoundingBox','Centroid');
cent=cat(1,s.Centroid)
bbox=cat (1,s.BoundingBox);
RGB=insertShape(Img,'FilledRectangle',bbox,'Color','yellow','Opacity',0.3);
RGB=insertObjectAnnotation(RGB,'rectangle',bbox,'Cursor','TextBoxOpacity',0.9
,'FontSize',18);
figure, imshow(RGB)
title('Cursor Detection')
```

APPENDIX B
CODING FOR CURSOR TRACKING

1. Coding for Cursor Tracking by using Blob Detection (Color Feature)

```
%Coding for Object Tracking ( Blob Detection [Color Feature] )
%Muhamad Hamizi Zaidi Bin Mohd Jonhanis
%EA18008

close all
clear
clc

%% Setup
%read  the input video
vidReader = vision.VideoFileReader('C:\Users\Acer\Desktop\PSM
RESEARCH\Picture for
PSM\video\Rotate_one_cursor_video6145690548050068645.mp4');
vidReader.VideoOutputDataType = 'double';

% Calculate detected objects’ area, centroid, major axis length and label
matrix by creating a Blob Ananlysis object to
hBlob = vision.BlobAnalysis('MinimumBlobArea',200,...
    'MaximumBlobArea',5000);

%Create VideoPlayer
vidPlayer = vision.DeployableVideoPlayer;

%% Run the algorithm in a loop
while ~isDone(vidReader)

    % Read Frame
    vidFrame = step(vidReader);

    % Convert RGB image to chosen color space
    Ihsv = rgb2hsv(vidFrame);

    % Define thresholds for channel 1 based on histogram settings
    channel1Min = 0.262;
    channel1Max = 0.437;

    % Define thresholds for channel 2 based on histogram settings
    channel2Min = 0.183;
    channel2Max = 1.000;

    % Define thresholds for channel 3 based on histogram settings
    channel3Min = 0.237;
    channel3Max = 1.000;

    % Create mask based on chosen histogram thresholds
    Ibw = (Ihsv(:,:,1) >= channel1Min ) & (Ihsv(:,:,1) <= channel1Max) & ...
        (Ihsv(:,:,2) >= channel2Min ) & (Ihsv(:,:,2) <= channel2Max) & ...
        (Ihsv(:,:,3) >= channel3Min ) & (Ihsv(:,:,3) <= channel3Max);


    % Create structural element for morphological operations to remove
disturbances
```

```
    diskElem = strel('disk',3);
     % Use morphological operations to remove disturbances
    Ibwopen = imopen(Ibw,diskElem);

    % Extract the blobs from the frame
    [areaOut,centroidOut,bboxOut] = step(hBlob, Ibwopen);

    % Draw a box around the detected objects

    Ishape = insertShape(vidFrame,'Rectangle',bboxOut);
    Ishape = insertObjectAnnotation(Ishape,'rectangle',bboxOut,'Desired
Object','TextBoxOpacity',0.5,'FontSize',15);
    centroidOut

    %Play in the video player
    step(vidPlayer,Ishape);
%     step(vidPlayer,result);

end

%% Cleanup
release(vidReader)
release(hBlob)
release(vidPlayer)
```

2. Coding for Cursor Tracking by using Feature Point Tracking

```
%Coding for Object Tracking (Feature Point Tracking)
%Muhamad Hamizi Zaidi Bin Mohd Jonhanis
%EA18008

close all
clear
clc

%% Read a video frame
videoReader = VideoReader('C:\Users\Acer\Desktop\PSM RESEARCH\Picture for
PSM\video\Rotate_one_cursor_video6145690548050068645.mp4');
videoFrame = readFrame(videoReader);
figure; imshow(videoFrame);

%% Draw the targetted region or object
objectRegion=round(getPosition(imrect))
videoFrame = insertShape(videoFrame, 'Rectangle', objectRegion);
figure; imshow(videoFrame); title('Detected');

%% Display the detected points using minimum eigenvalue algorithm
bboxPoints = bbox2points(objectRegion(1, :));
points = detectMinEigenFeatures(rgb2gray(videoFrame), 'ROI', objectRegion);
figure, imshow(videoFrame), hold on, title('Detected features');
plot(points);

pointTracker = vision.PointTracker('MaxBidirectionalError', 2);

%% Initialize the tracker with the initial point locations and the initial
% video frame.
points = points.Location;
```

```
initialize(pointTracker, points, videoFrame);

videoPlayer  = vision.DeployableVideoPlayer;
oldPoints = points;


while hasFrame(videoReader)
    % get the next frame
    videoFrame = readFrame(videoReader);

    % Track the points. Note that some points may be lost.
    [points, isFound] = step(pointTracker, videoFrame);
    visiblePoints = points(isFound, :);
    oldInliers = oldPoints(isFound, :);

    if size(visiblePoints, 1) >= 2 % need at least 2 points

        % Estimate the geometric transformation between the old points and
the new points and eliminate outliers
        [xform, inlierIdx] = estimateGeometricTransform2D(...
            oldInliers, visiblePoints, 'similarity', 'MaxDistance', 4);
        oldInliers    = oldInliers(inlierIdx, :);
        visiblePoints = visiblePoints(inlierIdx, :);

        % Apply the transformation to the bounding box points
        bboxPoints = transformPointsForward(xform, bboxPoints);

        % Insert a bounding box around the object being tracked
        bboxPolygon = reshape(bboxPoints', 1, []);
        videoFrame = insertShape(videoFrame, 'Polygon', bboxPolygon, ...
            'LineWidth', 2);


        % Display tracked points
        videoFrame = insertMarker(videoFrame, visiblePoints, '+', ...
            'Color', 'white');

        bboxPolygon
        NumberofMatchedPoint = length(visiblePoints);
        NumberofMatchedPoint

        % Reset the points
        oldPoints = visiblePoints;
        setPoints(pointTracker, oldPoints);
    end

    % Display the annotated video frame using the video player object
    step(videoPlayer, videoFrame);
end

% Clean up
release(videoPlayer);
release(pointTracker);
```

APPENDIX C

GAANT CHART FINAL YEAR PROJECT 1

No	Planning	Week													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Meeting with supervisor	■	■	■	■										
2	Research related to the project			■	■	■	■								
3	Summarize the article of research						■								
4	Project Report: Literature review							■	■						
5	Project Report: Introduction								■						
6	Apply cursor on the video sample					■									
7	Learn the detection using MATLAB R2021b Software							■	■						
8	Project progress with supervisor			■	■	■	■	■	■	■	■	■	■		
9	Simulation of the project									■	■	■			
10	Analysis of the output result										■	■	■		
11	updating project report											■	■	■	
12	Progress Presentation with panel												■		
13	Report and logbook submission														■
14	Upload report and logbook to coordinator														■

APPENDIX D
GAANT CHART FINAL YEAR PROJECT 2

No	Planning	Week													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Briefing about Final Year Project 2 with Panel	■													
2	Meeting and discussion with supervisor	■	■	■	■	■	■	■	■	■	■	■	■	■	■
3	Do research related to the project	■	■	■											
4	Construct the coding for the project	■	■	■	■	■	■								
5	Adjust the coding for project	■	■	■	■	■	■								
6	Get the result and collect data for this project.							■	■	■	■	■	■	■	■
7	Finish the thesis for Final Year Project 2 (FYP 2)												■	■	■
8	Update logbook for Final Year Project 2 (FYP 2)	■	■	■	■	■	■	■	■	■	■	■	■	■	■
9	Report and logbook submission														■
10	Upload report and logbook to coordinator														■