RAPTOR TRACKING AND COUNITING WITH BLUE SKY DECOLOURIZATION

By

Choo Chun Leong

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Perak Campus)

JANUARY 2013

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: **Raptor Tracking and Counting With Blue Sky Decolourization**

Academic Session: January 2013

I , CHOO CHUN LEONG

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_________________________ _________________________

(Author's signature) (Supervisor's signature)

Address:
52, Lorong Batu Nilam 19E,
Taman Bukit Tinggi 2,.
41200 Klang, Selangor.

____
Supervisor's name

Date: ______________________ **Date**: _____________________

DECLARATION OF ORIGINALITY

I declare that this report entitled “RAPTOR TRACKING AND COUNTING WITH BLUE SKY DECOLOURIZATION” is my own work except as citied in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _______________________

Name : _______________________

Date : _______________________

ACKNOWLEDGEMENT

First and foremost, I would like to express my appreciation to my supervisor, Mr. Tou Jing Yi who guided me in completing this project. Although I was new and inexperience with computer vision in the beginning, Mr. Tou was patiently explained the general idea of computer vision and some unfamiliar terminology to me. With his great knowledge in computer vision, he was able to give me useful advice in developing and implementing the method I proposed. Thanks to him, I was able to complete this project smoothly.

Taking this opportunity, I would like to express my gratitude to my senior Mr. Toh Chen Chuan for starting and completing this project so that I could improve this system to make it better. Without his hard work, I would probably need to create the system from scratch.

Lastly, I would also like to thank my friends and family for giving their support whenever I lost my direction in the progress. Even though they did not give me technical or topic related advices, they still encouraged me and give me more strength to overcome my problems.

ABSTRACT

This is a project that analyzed on another paper's method in tracking flocks of migratory raptors using optical flow. Optical flow is a popular motion detection algorithm created around 1970s and had been used in detecting and tracking objects by many researchers. One of the major problems discovered in the paper was high FAR (false acceptance rate) result in some videos which contained heavy portions of clouds. FAR of 70% in tracking process was too high and unacceptable. To track flocks of raptors with reduced FAR, this project performed a colour decolourization on blue sky background. Other than that, this project proposed a chance giving algorithm to redetect raptors that lost during their tracking process. With the same concept, optical flow was still being used but not in detecting moving raptors. It was used to predict the future location of raptors so that lost tracked raptors can still be tracked. Extra evaluation was done in this project to test the occlusion tolerance while tracking multi raptors.

In this project, a prototype was developed to compare the results between the referenced paper and proposed solution. Two major comparisons were accuracy and FAR. The development environment used in this project will be MATLAB which is same with previous one but the version is updated. Same video footages will be used to test the new solution proposed on tracking raptors.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

MATLAB	Matrix Laboratory
FAR	False acceptance rate
ID	Identification number
fps	Frame per second

CHAPTER ONE

INTRODUCTION

1.1 Project Background

In recent decades, raptor conservation had become a global issue to ecologists due to significant change in climate and other human factors. As part of the biodiversity, populations of raptors (bird of prey) have great impact and play an important role in the ecosystem. Due to their characteristics, they often place at top tier in the food chain and a little change in their population will bring a significant change to those at lower tier. According to statistics, their populations continue to decline and some of them even close to extinction such as Philippine eagle and Indian vulture (BirdLife International 2013). To prevent this from happening, raptor tracking activities had increased around the world to monitor and observe migration of raptors.

By tracking raptor, ornithologists or other raptor researchers are able to identify their habitat, the way they migrate, their behavior when migrating, their migration route, and effects to the ecosystems. Besides, raptors can be treated as natural bio-indicators that indicate the health of particular environment (Borgnis 2010). This information is documented and further study to find out the way to conserve them. With their help, many raptor populations are positively increasing nowadays. Many organizations had existed to help conserving raptors such as Hawk Mountain Sanctuary Association, The Peregrine Fund, BirdLife International, etc. Most of them are non-profit organizations which aim to conserve raptors and still working hard on it.

1.2 Problem Statement

Based on Tou & Toh (2012), cloud had become the major contributor to the high FAR especially in those videos with heavy portions of cloud contain solid outlines. This is due to difficulty for morphological operations to differentiating moving cloud with solid outlines from flying raptors. Apart from that, traditional thresholding method was not working well in segmenting the bird apart from the scene. Using a static thresholding value did not work well in real time where color of the sky will change according to time. Other than that, the processing speed is slow causing the solution disqualified for real time implementation.

1.3 Justification and motivation

The reason for conducting this research is to improve method proposed in (Tou & Toh 2012). Raptor counting and tracking using automation system is very beneficial to raptor researchers but not much people are working on it. Compare to raptors, tracking humans and cars are much more common in object tracking field. There are only few papers published on raptor tracking and this made this project more interesting and challenging.

The method used to track raptors in video was based on optical flow. Improvement is needed due to high FAR presented when tracking experiment was carried out. This problem is much more significant in videos with heavy portions of cloud. Having high FAR is very deadly in object tracking field. False track of raptors causes inaccurate data collection and further affect the raptor conservation progress. Some improvements were believed could achieve to make a reliable and real time implementable raptor tracking system. A successful system not just helping in raptors conservation, but also contribute to the object tracking field.

1.4 Project Objective

The main objective of this project is to create a new solution which improvement will be made on three important aspects: FAR, processing speed and bird tracking accuracy. Details are as following:

- To decrease the overall FAR to below 10% especially in those videos contain heavy portions of cloud.
- To improve the processing speeds to 30 fps for video with resolution of 320 x 180.
- To improve the bird tracking accuracy of previous solution to above 90%

1.5 Project Scope

The scope of this project covered:

- Detail review on previous solution
  Revision and analysis must be done the prototype implemented in previous raptor tracking solution to find out source of problem that cause high FAR, slow processing speed and tracking accuracy.

- Solve major problem causing FAR.
  One of problems had been identified which is misinterpretation of cloud as objects (raptor). New raptor tracking system should be able to handle this problem to lower the FAR by using suitable threshold value.

- Improve detection method
  Based on observation on (Tou & Toh 2012), optical flow was used to detect object of interest. This may causes other moving objects like clouds to be detected. This project will separate of foreground and background using thresholding.

CHAPTER TWO

LITERATURE REVIEW

2.1 What is Raptors?

Raptors are subspecies of bird that had different special characteristics such as hooked bills, powerful feet with sharp claws, and keen vision. Their hunting abilities were strong enough to tear down preys that are larger than their own size (Debus 2012).

Migration of raptors often occurs before the winter season to find better environment for breeding and surviving. Some of the raptors may travel lesser than 100km while most of them travel globally with great a distance (BildStein 2006).

Raptors usually migrate during the days by using thermal soaring. Thermal is hot air rising up from the ground into the sky and it allows raptors to soar up high and glide to next thermal spot.

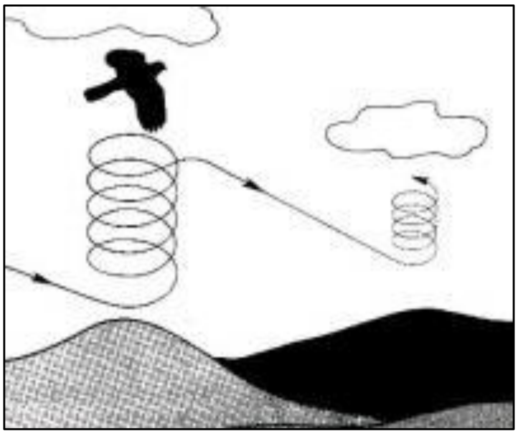


Figure 2-1-1: Raptor fly using thermal soaring (Raptor Migration, n.d.).

2.2 Video Tracking: An Introduction

Object tracking is an important task within the field of computer vision and it was started out around 1970s. Due to inexpensive video recording equipments, the need for automated video analysis had rise the need for better object tracking algorithm. Object tracking can be defined as estimation of trajectory of moving objects in the sequence of images. Object tracking has been applied to many different fields and they can be group in six main areas which include media production and augmented reality, medical applications and biological research, surveillance and business intelligent, robotics and unmanned vehicles, tele-collaboration and interactive gaming, and art installation and performances (Maggio & Cavallaro 2011).

Tracking objects can be complex due to loss of information caused by projection of the 3D world on a 2D image, noise in images, complex object motion, non-rigid or articulated nature of objects, partial and full object occlusions, complex object shapes, scene illumination changes, and real-time processing requirements (Maggio & Cavallaro 2011). While facing these challenges, many researchers in this field had proposed new idea in designing object tracking algorithm. Although none of them is perfect, but effectiveness and efficiency of object tracking had improved a lot compare to last few decades.

2.3 Video Tracking: General Process

Although researchers may have their own unique solution, but there are still three key processes common in most of them which are object detection, object tracking, and object identification (ed. Goszczynska 2011).

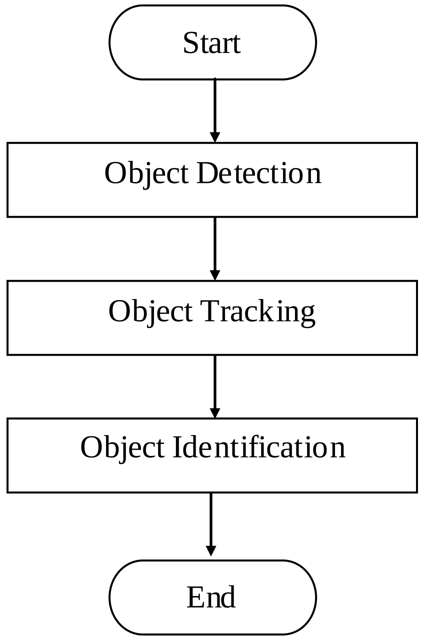


Figure 2-3-1: Common processes in object tracking.

2.3.1 Video Tracking: Object Detection

Most of the tracking system starts with object detection where object(s) of interest are identified and separate from the background image. The very first step in this process is getting the input which maybe video recorded by video camera or existing video (ed. Goszczynska 2011). There are a lot of methods that use to identify the object of interest either in every frame or when the object first appears in the video. There is no perfect solution but depends on the situation and complexity in analyzing image. These methods can be divided into 4 main categories (Yilmaz, Javed, & Shah 2006).

2.3.1.1 Point Detection

Interest-point detectors are used to find interest points in images that are reliable, when illumination and pose may change in an image sequences. Interest point sensitivity to illumination and camera viewpoint changes will determine its quality and usefulness. Different point detectors may have their own way of interpreting image intensity or other features. Commonly used interest point detectors include KLT detector, and SIFT detector.

2.3.1.2 Background Subtraction

Modelling can be performed on an image and then classify whether it is background or foreground. This kind of method is often suitable for static background scene such as surveillance camera at car park. However, it is sensitive to illumination changes. Any significant change in an image region that cannot be explained by the background model will be identifying as foreground or moving object. Estimating the change of background had made this method more challenging compare to point detection (Maggio & Cavallaro 2011).

2.3.1.3 Segmentation

Image segmentation is process of partitioning the image into group of pixels which having similar characteristic such as pixel value. Defining the characteristic of every partition is the major problem faced by segmentation algorithm because it affects the correctness of object detection (Szeliski 2010).

2.3.1.3.1 Otsu Thresholding

Thresholding is known for its simplicity in image segmentation. It converts a grayscale image into a binary image which only contains black and white. In traditional thresholding, a fix threshold value is use to differentiate background pixels and foreground pixels. However, fix threshold value let to lesser robustness and adaptability in real time.

In year 1979, Nobuyuki Otsu introduced a threshold selection method from gray-level histogram and this method was name Otsu's method. Otsu's method chooses the threshold to minimize the intraclass variance or maximize the interclass variance of gray-level histogram. It works well for image whose histogram show clear bimodal or multimodal distribution (Ng 2006).

2.3.1.4 Supervised Learning

Through learning some examples of object of interest, this method gain high robustness to the change of object's pose and view. However, a lot of templates need to increase the efficiency of object detection. Thus, large memory is required to store a complete set of templates.

2.3.2 Video Tracking: Object Tracking

After successful detection of object either single object or multiple object, object tracker will be the next thing to worry about. The aim of an object tracker is to generate the trajectory of an object over time by locating its position in every frame of the video. Object tracker may also provide the complete region in the image that is occupied by the object at every time instant. The tasks of detecting the object and establishing correspondence between the object instances across frames can either be performed separately or jointly. For tracking methods, they are divided into three categories.

2.3.2.1 Point Tracking

Objects of interest identified in object detection process represented by points. These points are often set at the centre of objects. When the object location changes from frame to frame, the points will be also move signify the motion of object. To track this motion, object detection had to be done every frame.

2.3.2.1.1 Kalman Filter

Kalman Filter is an optimal estimator that used to estimate the state of linear system. It was named after Rudolf E. Kalman in 1960 and was widely applied in technology field such as aerospace navigation systems. By minimizing the mean of square error of estimated parameters, Kalman Filter can estimate state of past, present and future very well under one condition where noises have Gaussian distribution. Which mean the more noises are Gaussian distributed, the better the Kalman Filter estimation. Basically, Kalman Filter consists of two steps which are prediction and correction. Prediction is the process of predicting new state from current state model whereas correction measures the current predictions to improve those predictions and update the object's state. Since Kalman Filter works recursively, only present state is needed and does not rely on past states (Welch & Bishop 2006).

2.2.2.1.2 Particle Filter

Particle Filter is part of sequential Monte Carlo Method used for Bayesian filtering. Particle Filter is the opposite of Kalman Filter in term of state variables since Particle Filter does not need the state to be Gaussian distributed. It is very popular with state without linear system and Gaussian distribution. In Particle Filter, any probability density function can be represented as a set of samples or known as particles. With this method, any arbitrary distribution can be represented making better than Kalman Filter under non-Gaussian state. To select the appropriate sample, sample scheme like importance sampling is used (Chin 2005).

2.2.2.2 Kernel Tracking

Objects of interest are represented by kernel such as rectangle and circle. Kernel refers to the object shape and appearance. Not just geometric shape can be use, but template of object is also applicable. Object motion can be identify by computing the change of kernel with previous kernel. This motion is usually in the form of a parametric transformation such as translation, rotation, and affine.

2.2.2.3 Silhouette Tracking

In real time situation, objects may have complex shape that hard to be described using geometric shape. Silhouette tracking try to find out the exact shape of complex object by using object edge or other ways. Motion of the object could be seen by observing the changes of the edges. Image segmentation is one of the methods to identify the shape of object in every frame.

2.4 Non-linear Filtering

There are many different types of noise filtering and reduction algorithm used in image processing. They can be categorized into linear filtering and non-linear filtering (Szeliski 2010). Compare to linear filtering, non-linear filtering can performs better on non-Gaussian images. In real time, it was advised to use combination of linear and non-linear filter to produce better results.

2.4.1 Morphological Operation

Morphological Operation which is originated from mathematical morphology is widely applied in binary image processing as a noise filtering or image enhancement. There are four kind of operation which includes dilation, erosion, opening and closing. Dilation is the process of filling small hole and connecting disjoint objects by expanding the objects. The other way round, erosion shrinks the objects by eroding their boundary. With the combination of dilation and erosion, opening and closing operation are formed. Opening performs erosion first followed by dilation while closing is vice versa of opening. All these four operation can be represent in Boolean consists of AND (erosion) and OR (dilation) (ed. Bovik 2005).

2.4.2 Median Filters

Median filter is a common and widely-used non-linear filtering in removing noises in image. It was able to preserve edges well and better than linear filtering. Median filter works by moving through pixels and replacing those pixels with median value of neighbouring pixels. A fixed-size K x K window was set to slide over the whole image. All values within the window are sorted and median value is selected to replace the middle pixel in the windows (Szeliski 2010).

2.5 Feature in Object Tracking

Selecting feature in object tracking is the very first step and could possibly affect the choice of tracking algorithms. Feature selection is depending on the characteristics of object interested. Colour is the most common feature used since colour gives much more information compare to other features. However, analyzing colour usually increases the complexity and time-consuming. To reduce the complexity, edge detector can be used to detect edges. As mid-level feature (Maggio & Cavallaro 2011), edges are used to represent the objects boundary. One of the advantages of edges is that it is less sensitive to illumination changes. Thresholding an image can easily generate edges of objects. Other than detecting using current information, object of interest can also be detected using motion as feature. Objects of interest are usually in moving and have different location over time. By computing dense optical flow, moving object can be easily detected.

2.5.1 Optical Flow

Optical flow is a field of displacement vector which show the movement of every pixel from one image to another. Using the low-level feature - motion as basic, optical flow is able to tell any moving objects in a sequence of image. Optical flow was computed using brightness constraint and three assumptions was made which include brightness consistency, spatial coherence and temporal persistent. Optical flow can be use in image segmentation, tracking, and estimation of object (Yilmaz, Javed, & Shah 2006).

Brightness consistency assumes that the brightness in a region of image should remain the same although the location may change from frame to frame. Spatial coherence states that neighbouring points of object in a image should belong to same surface and hence have similar motion. Temporal persistence assumes that the object's motion change gradually over time (Thrun 2006).

Optical flow can be further classified into two groups which are local method and global method. Local methods such as Lucas-Kanade technique are often more robust under noise but does not give dense flow field, while global methods such as Horn-Schunck technique yield dense flow field but is sensitive to noise. A lot of research had been done such as Bruhn, Weickert& Schnorr (2005) that try to overcome their weaknesses.

2.6 Related Work

A number of bird tracking methods have been proposed in recent years. Brau *et al.* (2011) presented Gaussian processes as smooth motion for multiple target tracking. By using Gibbs sanpler as tracking approach, results of Gaussian processes was outperformed the traditional method which was using linear dynamical systems as smoothing motion. Gaussian process was not performing well with non-linear type of motion where objects move in different way that hard to model by linear dynamical systems.

Zhang *et al.* (2008) used a foreground subtraction to detect the bird and a MCMC filter with no move types to track manoeuvrable birds. This method is robust to abrupt scene change but not suitable for scene with background of ground and buildings. The algorithm also is not able to cope with bird's occlusion well when the number of birds exceeds ten.

Nan, Morishita & Yokoi (2011) proposed a particle filter based clustering method to track multiple chickens. The proposed method did not perform well as expected when number of chickens increased. The tracking results produced are very dependable on the clustering results and a single clustering failure caused continual tracking failure.

Spampinato *et al.* (2008) combined multiple algorithms in detecting and tracking fish. Moving average algorithm and adaptive Gaussian mixture model were used to identify the background. The detection provided a satisfactory result in different kind of underwater scenes. The combination of blob shape matching and histogram matching tracking algorithm were able to achieve about 90% accuracy. The only weakness of the system was failure to detect the same fish in frames and caused fish over-counted.

Similar with (Tou & Toh 2012), Kalafatic, Ribaric & Stanisavljevic (2001) developed a system for tracking laboratory animals based on optical flow and active contours. The system was first threshold the frames to identify object contours and then used optical flow to solve object occlusion problem. To enhance the robustness, a supervision module was added to detect tracking failures and reinitializes the lost contours. But the system may not perform well in natural environment such as forest due to edge detection of other non-interest objects.

Dealing with wild environment, Tweed & Calway (2002) proposed a system using CONDENSATION tracker, a special particle filtering approach which used in situation where the state density can be multimodal and the state dynamics nonlinear. Although the system was able to track multiple animals and resistant to occlusion, the observation model still not sophisticate enough to detect in wildlife footage that contained severe motion blur and well-camouflaged animals

2.7 Problem Analysis

A detail analysis was done on (Tou & Toh 2012)'s prototype to identify the problem causing FAR and other problems. The figure shows the processes workflow of the prototype.

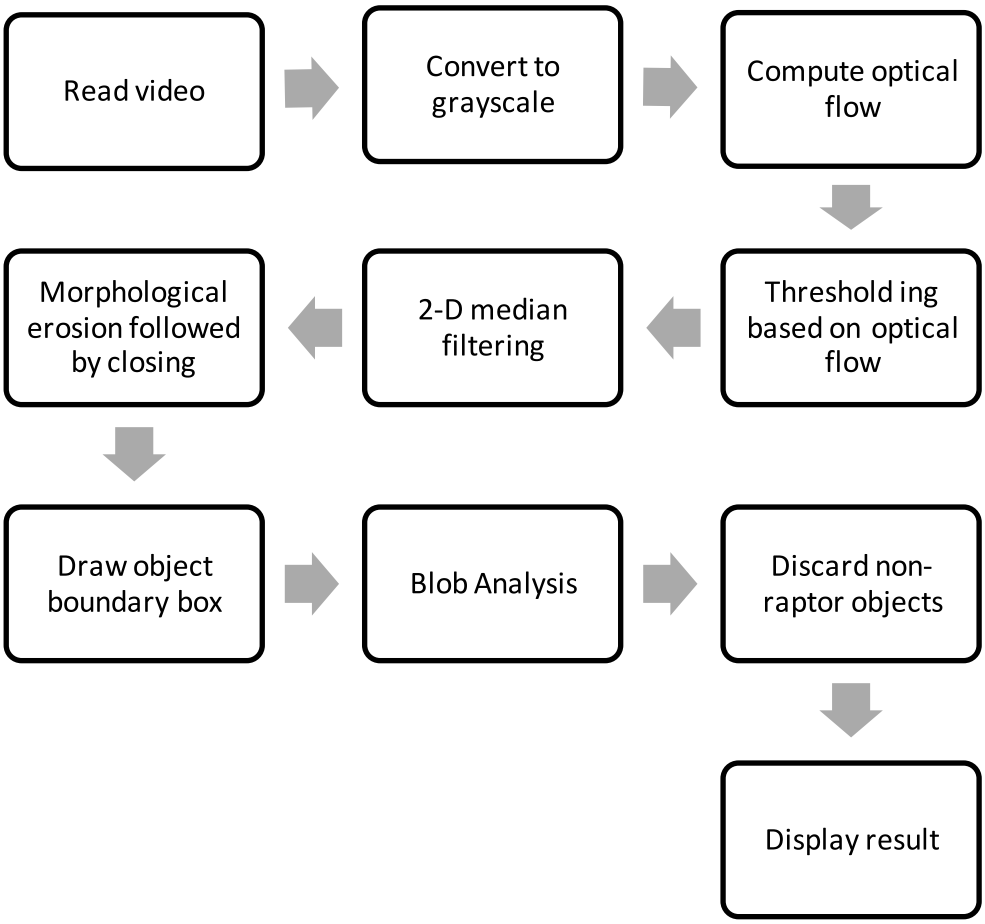


Figure 2-7-1: Diagram of the (Tou & Toh 2012) proposed solution's prototype process workflow.

2.7.1 Optical Flow as Detection Feature

From the prototype, Tou & Toh (2012) was using Horn-Schunck optical flow to detect raptor within the video. Optical flow was used to detect all motion within the video. Thus, result in detecting moving cloud as interest object.

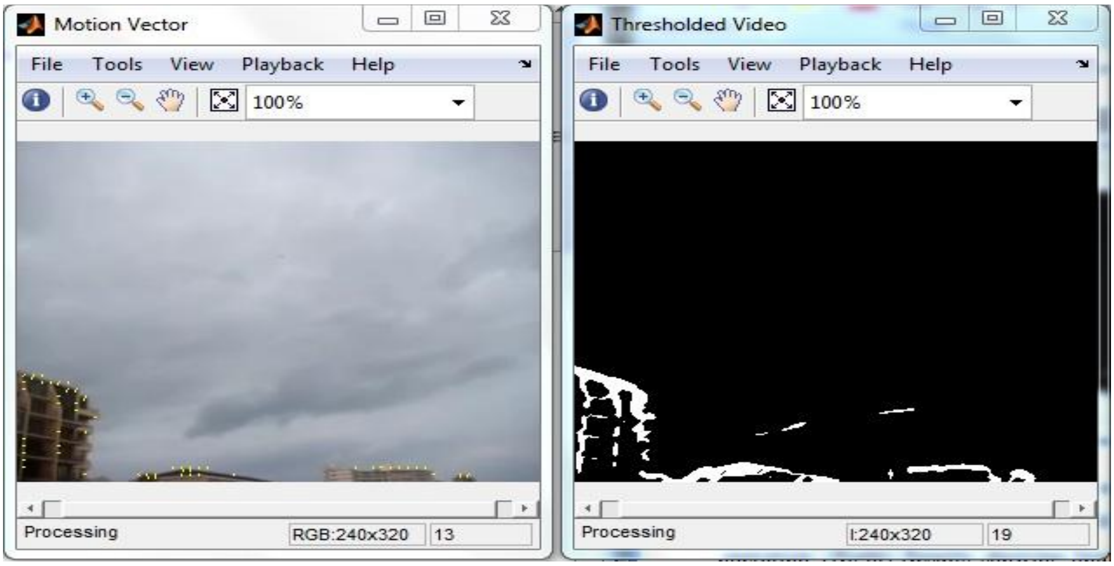


Figure 2-7-2: Detection of clouds and buildings as object of interest.(Left) Motion vector compute from optical flow. (Right) Results showing clouds and bulidings detected as object of interest (Zija777 2009).

2.7.2 Disadvantage of Morphological Operation

Morphological Operation performed on thresholded image was helpful in eliminating background noise but at the same time, it combined two close blobs. As results, raptors that fly close to another were treated as one object.

2.7.3 Flaw in Blob Analysis

Blob analysis done on thresholded image after performing morphological operation was excluding blob larger than value specified. The maximum blob area set in the prototype was only 200 pixels which was too small for some blobs. It may be able to filter out cloud but the value was not optimized to suit the situation.

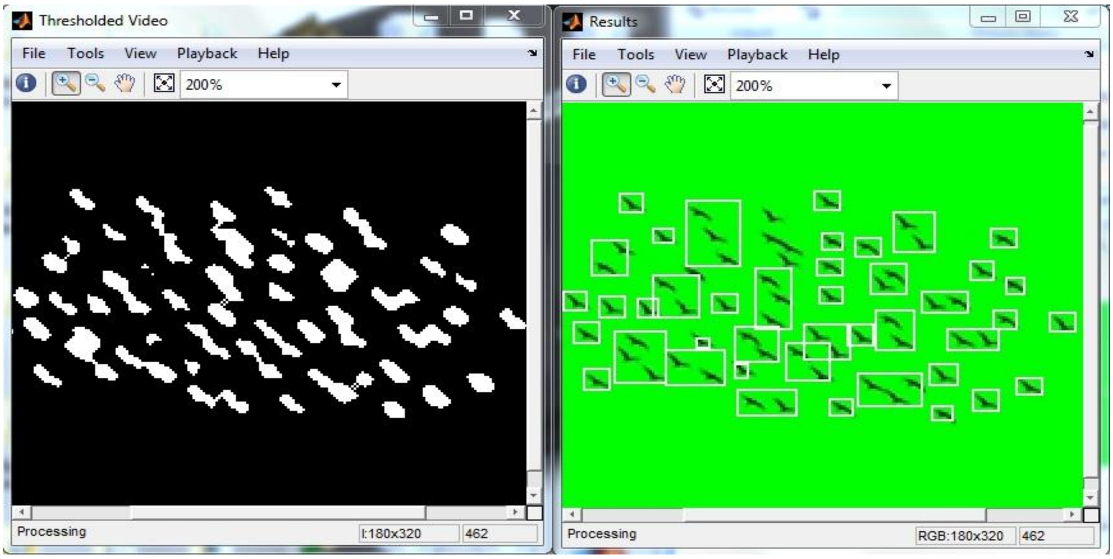


Figure 2-7-3: Morphological operation performed on thresholded video (Dermitdemmikro 2010a). (Left) After morphological operation. (Right) Results showing a few bird silhoutte detected as one.

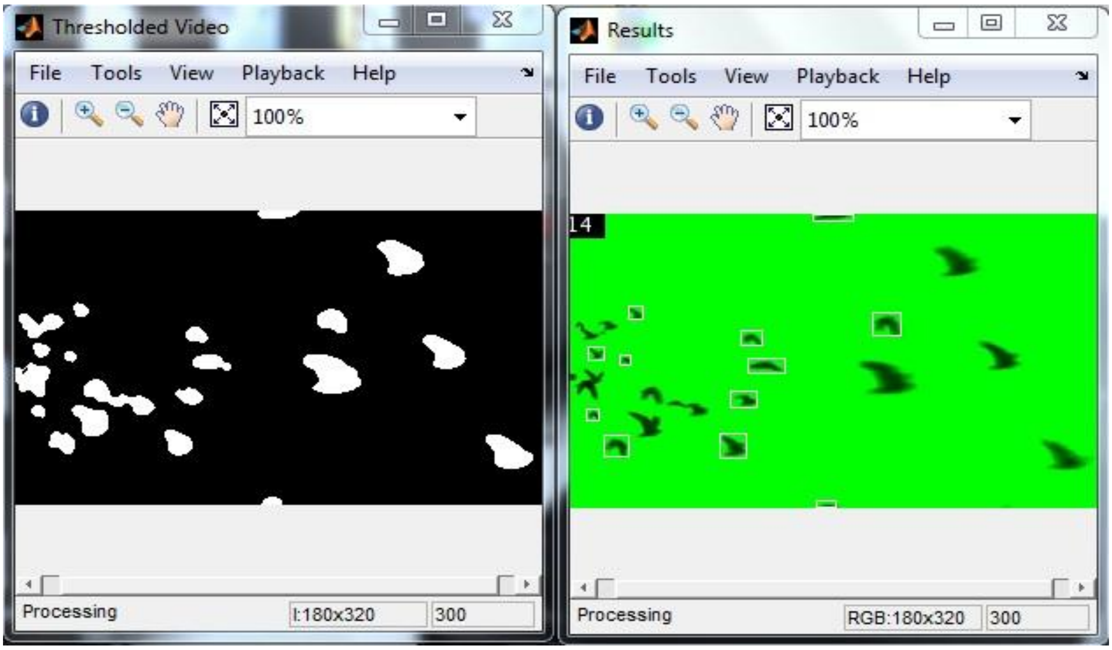


Figure 2-7-4: Testing blob analysis (Dermitdemmikro 2010b). (Left) After morphological operation. (Right) Results showing some bird silhoutte was not detected due to the size of blobs.

2.7.4 Fail in Discard Non-raptor Objects

The algorithm used to exclude other objects which are non-raptors by analyzing ratio between area of the blob and the area of the bounding box. This method is not really suitable in this situation where clouds also produced same blobs like raptors.

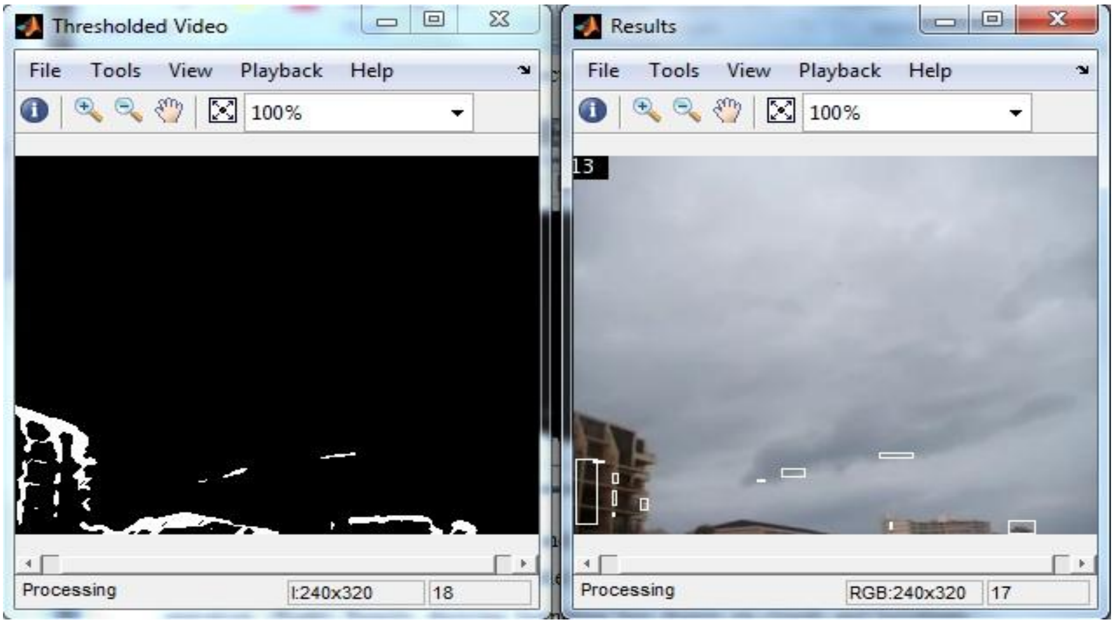


Figure 2-7-5: Elinimation of non-raptor objects. (Left) Thresholded video. (Right) Result showing failure in removing clouds and buildings (Zija777 2009).

CHAPTER THREE

METHODOLOGIES AND TOOLS

3.1 Development Tool

MATLAB is the development tool chosen for developing this counting and tracking raptor system. MATLAB is a popular software for it wide range of application in professional field such as complex numerical computation, data analysis, estimation algorithm, computational biology and image processing. MATLAB is now the standard for these kinds of problems hence a lot of people can help in troubleshooting one's problem in algorithm. With the help of tutorials, beginner can adapt to the MATLAB environment relatively fast and simple. It had a simple GUI that can show statistical data and result visually. Compare to other software, it provides large set of libraries which contains ready-to-use function which can reduce user works on those algorithms. In this project, the Image Processing Toolbox will be used. It provides some basic image processing function such as grayscale conversion and thresholding. Other than these toolbox, many other user around the world often share some new algorithm like KLT and particle filter which will be helpful in this project.

3.2 System Methodology Introduction

The project used optical flow to track and estimate the raptors in every frame. The methodology is mainly separated in into two phases: segmentation phase and tracking phase.

For segmentation part, a simple adaptive pre-processing was done to decolourize the sky colour before thresholding was used to separate the sky and raptors. Then the noises in thresholded frame were removed using morphological operation. After successful segmented the background and foreground, blob analysis was performed to obtain blob statistics.

For tracking phase, optical flow was then used to estimate the motion vector of each interest objects (raptors and noises). Each object that qualified for tracking was evaluated in each frame to remove non-raptor objects. After segmentation and tracking processes was done on one frame, the output was shown and the whole process was repeated on next frame until finish. The number of objects tracked in the particular frame and their respective ID was shown in the output video.

3.2.1 System Flow Diagram

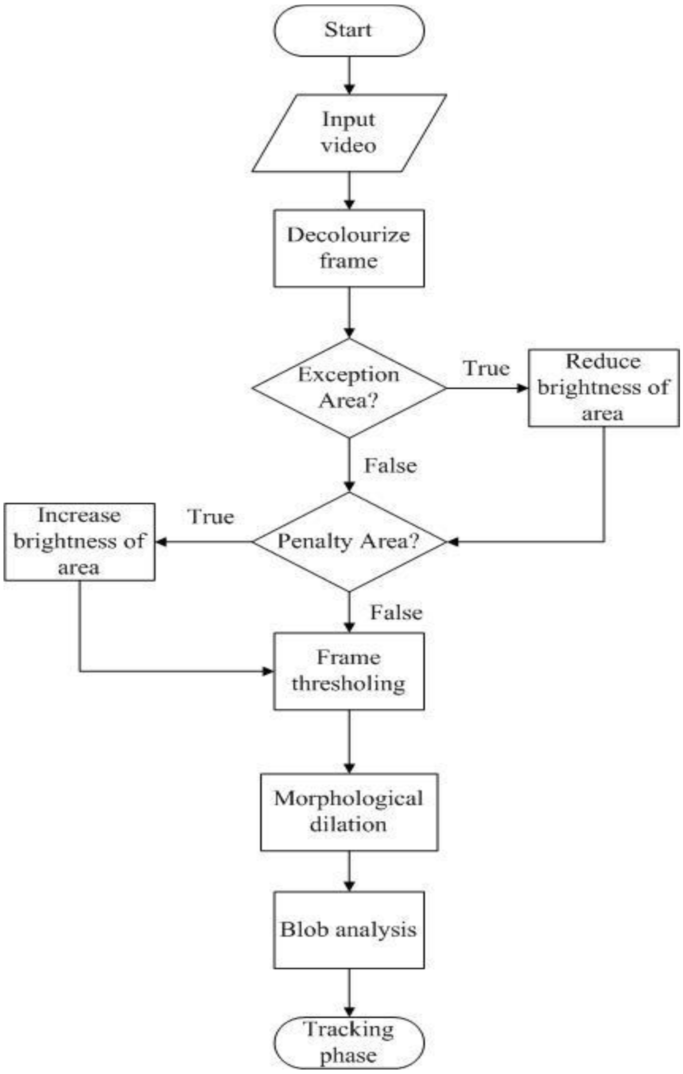


Figure 3-2-1: General flowchart of segmentation phase.

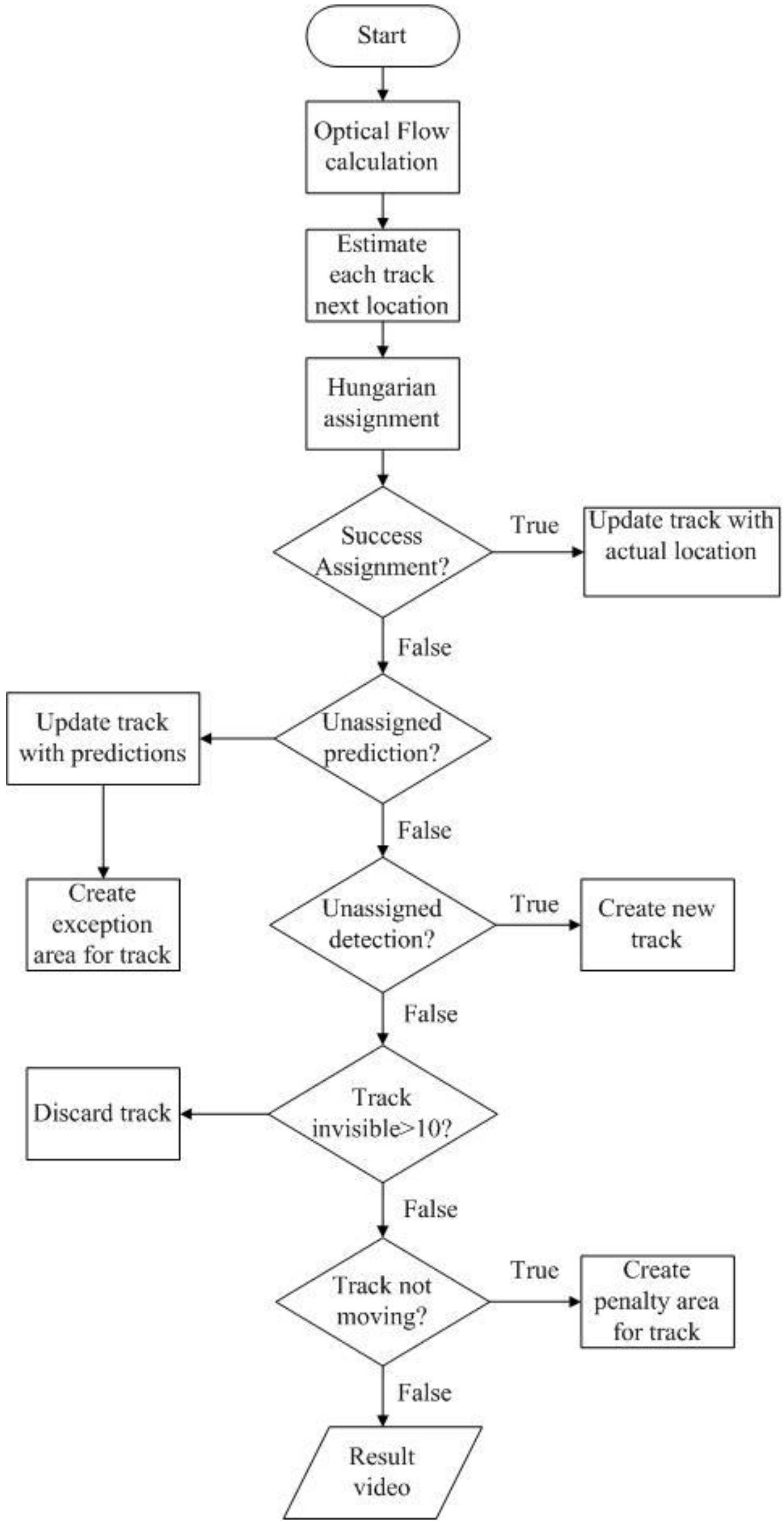


Figure 3-2-2: General flowchart of tracking phase.

3.3 Segmentation Phase

3.3.1 Pre-processing: Decolourize

According to Tou & Toh (2012), high percentage of FAR was caused by the little movement of clouds. Cloud itself may not seem to be culprit since cloud was appeared white in the dataset videos. They could be easily filtered out using appropriate threshold value. After an amount of experiments, FAR was actually caused by 1.Gap between clouds that show the blue sky and 2.Thickness of the cloud within the gap. When thresholding the frames with static threshold value, most of the gaps were remained as objects of interest. Setting lower threshold may help but there was risk of filtering true raptors in the process. In video footage, some raptors RGB value was close to edge of clouds. In this proposed method, the edge of cloud was treated as object of interest and further tracking it. Distinguishing them from true raptors was covered in Section 3.4.3 (pg.31) in the tracking phase. Primary goal of pre-processing was to reduce the false detection and directly affect the FAR. On the other side, it also decreased the number of objects that required evaluation in tracking phase and improved the processing speed. Therefore, filtering these gaps was the work of pre-processing.

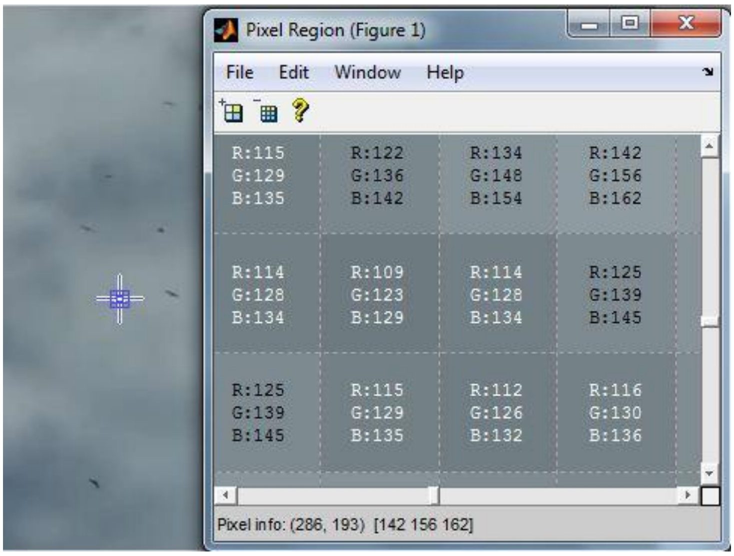


Figure 3-3-1: Example pixels value of raptor.

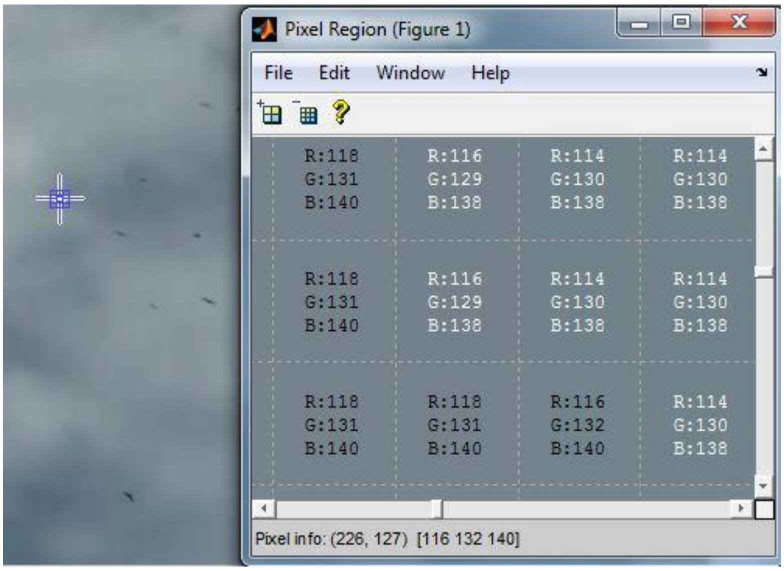


Figure 3-3-2: Example pixels value of edge of cloud.

In actual implementation, the colour of sky was detected using blue colour channel and based on the depth of sky colour, different thickness of cloud was added to cover the sky. This pre-processing was adaptively implemented which decided the overall thickness of cloud needed to add to eliminate most gaps in the grayscale frame. As result, most gaps between clouds were filled up.

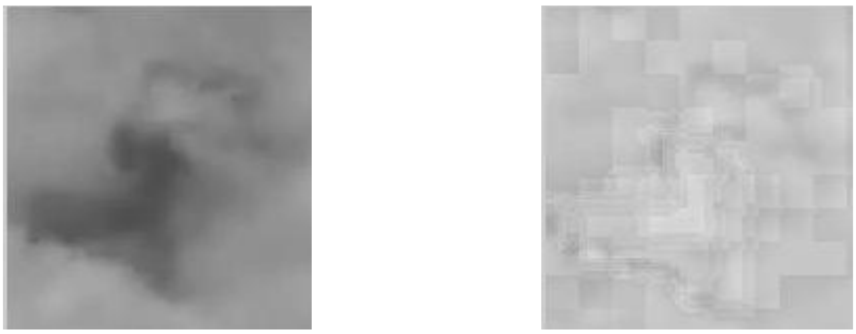


Figure 3-3-3: Example of pre-processing outcome. (Left) Grayscale frame without pre-processing. (Right) Grayscale frame with pre-processing.

3.3.2 Frame Thresholding

Thresholding is the simplest way to segment out object of interest from frame. In grayscale frame, every pixel is represented by value from 0 to 255. A value was being set to compare with each pixel in frame where

$$p(x,y) = \begin{cases} 0, & p(x,y) \leq \text{value} \\ 1, & p(x,y) > \text{value} \end{cases} \quad (1)$$

This value was known as threshold value.

The output from pre-processing which in grayscale was being converted in to binary form consists of 1 and 0. In binary form, 1 (white) will be background and 0 (black) is the object of interests. The static threshold value set was based on the adaptive value set in pre-processing and the value is 127. This binary frame was then passed to blob analysis to obtain information on the object of interest.

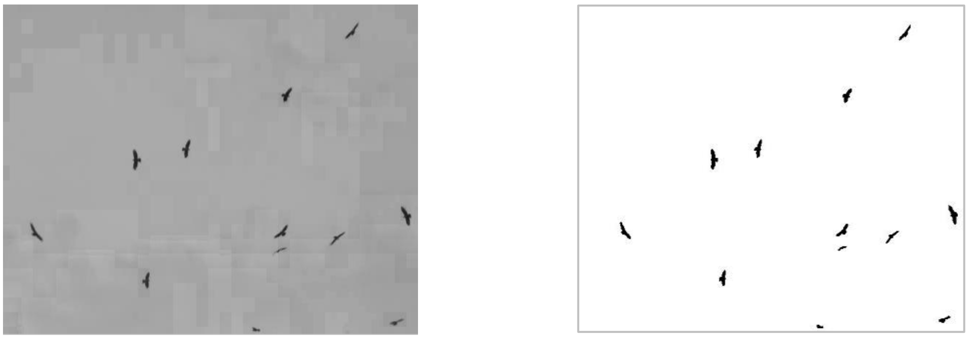


Figure 3-3-4: Example of thresholding outcome. (Left) Grayscale frame after pre-processing. (Right) After thresholing.

3.3.3 Morphological Operation

Sometimes, a pixel of raptor may be strays away due to noise or raptor's flapping movement. Since the blob analysis implemented in this project counted the single pixel as one blob, a noise filtering was needed to solve this situation or it will increase the FAR.

After thresholding, a morphological operation: dilation was used to join those strayed pixels with closest blob. A minimal structuring element or window was created for morphological dilation to minimize the effect of occlusion. If the structuring element is large, it will causes occlusion to occur more often. The structuring element of morphological dilation was:

0	**1**	**0**
1	**1**	**1**
0	**1**	**0**

Table 3-3-1: The disk-shape structuring element used in this project.

Figure 3-3-5: Example of morphological dilation on synthetic image. (Left) Before morphological dilation. (Right) After morphological dilation.

3.3.4 Blob Analysis

Blob analysis is a method provided by MATLAB to compute the statistic for connected white pixels in binary frame. Since it detects white pixels as object of interest, the output from previous thresholding process was reversed using matrix NOT operation. Now, 1 (white) will be object of interests and 0 (black) is the background. All object of interest was now known as blob and statistics such as blob area, centroid, bounding box were returned as parameters. These parameters were very vital in later tracking phase.

Compared to (Tou & Toh 2012), all blobs were accepted and qualified rather than using blob analysis to filter out non-raptors objects. Filtering blob based on maximum and minimum area may not be effective since the size of raptors vary from 1 pixels to full screen assuming the worst case where a raptor fly through just in front of video camera. But this kind of cases were rare and at least it was not happening often in dataset video. Still, there were chances of filtering small and large raptors and cause reduction in accuracy. Due to this reason, median filtering and morphological operation: closing and opening were not implemented in this project. Another method introduced by Tou & Toh (2012) that discarded non-raptor using ratio is not encouraged as reason stated in Section 2.7.4 (pg. 17).

3.4 Tracking Phase

3.4.1 Optical flow Prediction

In this project, optical flow was used to predict the location of blobs in next frames by estimating the speed and direction using previous frame as reference. Lucas-Kanade method provided by MATLAB is used to compute the optical flow. Lucas-Kanade was chosen instead of Horn-Schunck due to the computation cost of Horn-Schunck is about 2 times greater than Lucas-Kanade. This is because Horn-Schunk solves the horizontal and vertical optical flow using iteration until the flow field converge. Iteration may goes up to 100 to reach a converge state and requires more computation cost. On the other hand, Lucas-Kanade divided the image into smaller sections and used a weighted least-square fit to constant model for each section. Computation cost is required according to the number of sections divided from original image. Lucas-Kanade solves optical flow by minimizing the following equation (Barron & Thacker 2005):

$$\sum_{x \in \Omega} W^2 \left[I_x u + I_y v + I_t \right]^2 \qquad (2)$$

- I_x, I_y and I_t are the spatiotemporal image brightness derivatives.
- u is the horizontal optical flow.
- v is the vertical optical flow.
- W is a window function that emphasizes the constraints at centre of neighbourhood.
- Ω is the sections.

If a blob suddenly undetected in next few frames, prediction will be use as substitute for actual location until the blob reappears. Since the prediction was only based on optical flow of previous frame, the prediction may be incorrect over long time interval. So, if a blob was undetected within 10 frames, the blob was assumed that it had leaved the scene and not longer available for tracking.

Initially, the optical flow only calculated the speed and direction for blob only to reduce the computational cost, but optical flow provided by MATLAB is more suitable to compute flow field for whole frame. In MATLAB, the size of input image was not tuneable and performing optical flow on every single blob required more time than whole image. Theoretically, skipping all background pixels will greatly reduce the computational cost.

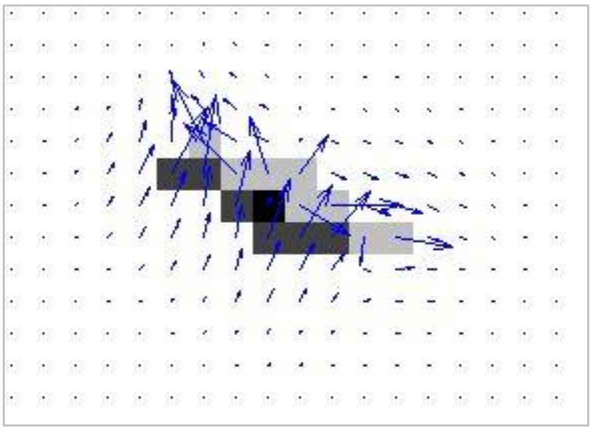


Figure 3-4-1: Example of MATLAB optical flow outcome. Gray region is where the raptor moves while black region is initial location.

3.4.2 Hungarian Assignment Algorithm

Hungarian assignment algorithm was needed due to incapable of blob analysis provided by MATLAB in tracking same blob ID. Blob analysis was performed according to columns then rows, figure 3-4-2 show how blob ID changes. Blob ID is an important value to identify the blob's current information so that correct update on new location, optical flow, age, etc could be done.

Figure 3-4-2: Failure tracking blob ID using blob analysis. (Left) First frame. (Right) After 10 frame.

In MATLAB, there is a built-in Hungarian assignment algorithm created by James Munkre's. To implement the algorithm a table of predictions × detections is drawn and distance between each prediction and each detection is calculated. Predictions are the estimated location of current frame's blobs in next frame while detections are the true location of blob in next frame. The distance computed is known as cost in the algorithm. Sample table shown below.

Detection / Prediction	D1	D2	D3
P1	16.65	13.54	19
P2	36.8	27.56	5.98
P3	6.3	15.88	25.21

Table 3-4-1: Example of Hungarian assignment algorithm.

As seen in table, the closer the distance, the detection is more likely to be assign to a prediction. Unfortunately, a blob may not always be detected in next frame, a maximum distance value need to be set so that prediction or detection remain unassigned. This value is known as cost of non-assignment. Therefore, there are 3 different cases produced by this algorithm which were successful assignment, unassigned predictions and unassigned detections.

3.4.2.1 Case 1: Successful Assignment

Successful assignment also means that the prediction does match with the future outcome. When there is some distance between current blob prediction and blob in next frame that does not exceed the cost of non-assignment and it is the lowest compare to other distances, the particular prediction is assigned to lowest distance detection. Then, the blob's information will be updated with information of current detected blob.

A successful assignment does heavily depend on the cost of non-assignment. If the cost is set at high value, for example, 100 (in pixels) and the shortest distance is 90 (in pixels). Since 90 is smaller than 100 and it is the shortest distance, the detection will be assign to the prediction. Theoretically, it may seem correct but a raptor can't move 90 (in pixels) just in one frame in this project. Through observation, raptors only move at maximum of 5 pixels per frame or sometimes no movement at all.

Even assignment may be successful, but there are still some cases where assignment is incorrect. Especially when two blob in one frame cross-over each other or occlusion between blobs happens. When cross-over happens, two blobs ID may update with other information and their path are recorded wrongly. Since this project focus on tracking and counting raptors, this problem doesn't really cause much influence to the results.

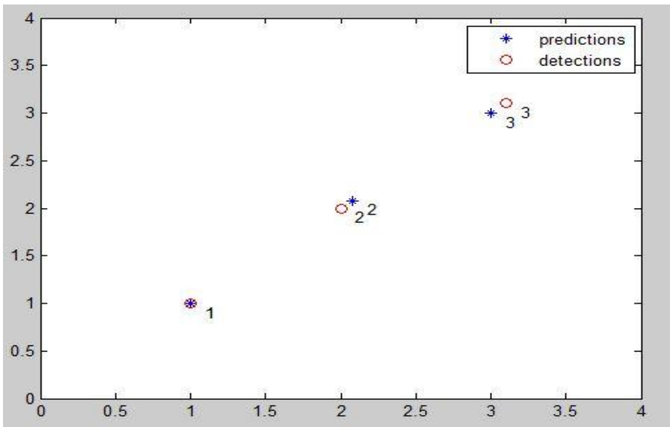


Figure 3-4-3: MATLAB example of successful assigns all detections to predictions.

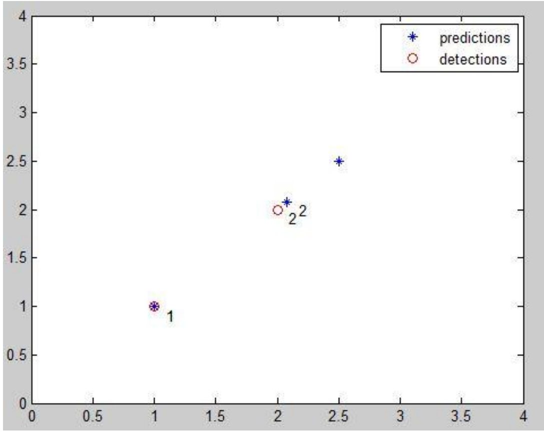


Figure 3-4-4: MATLAB example of one unassigned prediction.

3.4.2.2 Case 2: Unassigned Predictions

Unassigned predictions only happen when a blob in tracking process suddenly disappear from the scene. Disappearance may cause by 1. The blob exit the scene (natural disappearance) or 2. The blob is not detected in segmentation phase but appeared in video screen (segmentation disappearance). Natural disappearance occurred in dataset video can further divide into 3 categories.

The first one is the blob exit the through the borders of video camera screen. First problem could be partially solved by connecting the borders of video camera with others to form a larger screen. This is an incomplete solution since raptor that moves in one direction will eventually exit the screen unless video cameras is placed and connected all around the world which is costly. Next, the blob exits the scene by hiding behind thick/dense cloud. To counter this problem, different angles of a scene could be recorded so that raptor behind the cloud could be seen. Finally, the blob moves far enough that the video camera only captures a small black dot that is not moving. One of the easiest ways to deal with this problem would be following the raptor with the video camera. It may sounds stupid but effective and cost wise. Make sure the video camera can follows the speed or raptors. The common solution to natural disappearance would be placing more video camera and distributing them to different place.

Rather than solving natural disappearance, this project is focusing on solving the segmentation disappearance. The solution discussed above that involves hardware will not be implemented in this project. Segmentation disappearance mostly happens when raptor hides behind the cloud and thus making the raptor whiter. When undergo thresholding in Section 3.3.2, the raptor will be filtered out affecting the accuracy of this project. Using the dataset video in this project, cause of segmentation disappearance is also known as illumination changes since cloud is white in colour. Raptor hides behind cloud can refer as increasing of brightness.

First way to solve this problem is using optical flow predictions mentioned in Section 3.4.1 which specially developed to solve segmentation disappearance. Before that, the raptor is assumed that it must, at least been detected once before prediction could be done. Raptor that disappeared in the tracking process will be predicted for at most 10 frames. Tracking process could be continued if the particular raptor is detected within 10 frames and predicted location is close to it. In most case, raptor will disappear more than 10 frames and thus ended its tracking process. To solve this, this project introduced another method which called exceptional area.

3.4.2.2.1 Exceptional Area

Exceptional area is same like chance giving. When a blob is using predicted location to update its information, a chance will be given to the blob when performing decolourization in Section 3.3.1. By using the blob’s bounding box information calculated by blob analysis, exceptional area could be set during decolourization. When adding cloud to the frame, this exceptional area will had lesser cloud added compare to other to decrease the brightness of the blob. Thus, enable the blob to be detected in segmentation phase. If the blob successfully detected, the exceptional area will be discarded.

3.4.2.3 Case 3: Unassigned Detection

When a blob is detected and unassigned to any prediction, this blob is considered as new blob and started to track it in next frame onward. To start a new track, a set of information need to be initialize such as blob ID, centroid, bounding box, age of track, etc. If a blob is discarded from tracking process and redetected in segmentation phase, it will be treated as another new blob. This project tries to defend the blob from being discarded so that the blob remains the same ID using optical flow predictions and exceptional area.

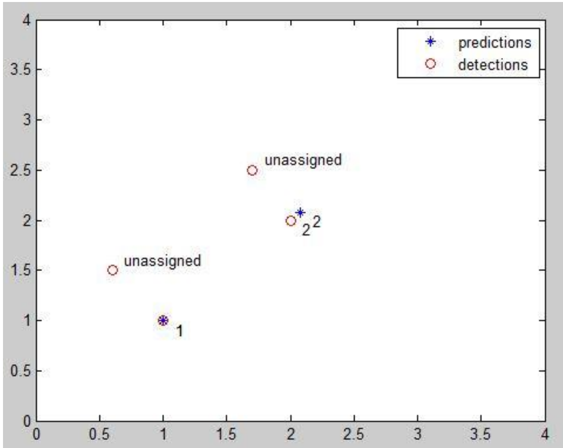


Figure 3-4-5: MATLAB example of two unassigned detections.

3.4.3 Filtering Non-raptor Blob

Compare to (Tou & Toh 2012) that used blob area and ratio of bounding box, this project is using consecutive invisible count and distance from original location to filter non-raptor blobs.

Consecutive invisible count is a counter that increases by 1 for every time a blob used prediction to update its information. But the counter will reset to 0 when the blob used detection to update its information. When a particular blob's counter increase to 10, it means the blob is not detected for 10 frames and thus discarded from tracking process.

Calculating distance from original location is an easy way of distinguishing moving objects or raptors from non-raptors. In dataset video, the edge of cloud is usually detected as blob and almost no movement at all, unless there is some camera movement. Without any camera movement, the edge of cloud is not going to move thus the distance from original location will be 0. On the other hand, raptor can't stay at fixed location while gliding in sky, the distance will be always larger than 0. However, only best case where no camera movement while recording the video can use distance of 0 to identify raptors. Moreover, raptors with small movement may have distance of 0 in some frame and caused false rejection. To solve this problem, this method is using distance between original location and location after 10 frames to discard the non-raptor tracking process. The distance must be at least 10 pixels to qualify the blob for tracking.

Removing the non-raptor from tracking process may not be effective since the non-raptor blob could be redetected in segmentation phase and start a new tracking process. This infinite loop of discard and redetect will increase the FAR as long as the non-raptor blob remains in the scene. To end this infinite loop, this project introduction another method called penalty area.

3.4.3.1 Penalty Area

Penalty area works exactly the opposite of exceptional area in Section 3.4.2.2.1. Instead of decreasing the brightness in certain area, the brightness is going to increase in penalty area. This causes the blob to be filtered out during thresholding. If a blob is detected in the penalty area, the blob has high chance to be a raptor and the penalty area will be removed.

3.5 Counting Phase

The number of blob that remained in tracking process is counted for every frame. In this project, the blob exited the scene will still be under tracking process for 10 frames. This data is only show the total number of blob under tracking in the particular frame and does not represents the total number of raptors tracked.

3.6 Additional Feature: Travel Path

Other than labelling ID for every blob, this project also draws out the travel path of every blob. If the ID is discarded, then the path will also removes from the output video. All paths are stored in the blob ID information and discarded ID information will remains in the prototype program for further process. The correctness of path will be discussed.

CHAPTER FOUR

DATASET AND TEST ENVIRONMENT

4.1 Video Footages of Raptor

The videos of raptor flocks used in this project were same in (Tou & Toh 2012). Initially, there is a total of 29 videos recorded and stored in m2ts file format, H264 codec, 50 frame rate and 1920x1080 resolutions. Due to the large resolution and incapable of MATLAB in supporting m2ts file, they are converted into avi file format, H264 codec, 30 frame rate and both 320x240 resolution and 640x480 resolution. Two resolutions are needed to test the proposed method robustness towards sharp and blur frames. The conversion process is done by using Freemake Video Converter (Freemake.com) with one-pass encoding.

4.2 Criteria of Selecting Dataset for Testing

While selecting dataset for testing, different kind of characteristics were used to describe the videos. These characteristics are essential and can affect the test result independently. Dataset were collected based on characteristics described in next sub section.

4.2.1 Camera Movement

Camera movement is also describing how fast the background moving in videos and there are 5 kind of movements.

- High – The background moving in one direction faster than most raptors in the scene.
- Low – The background moving in one direction slower than most raptors in the scene.

- High jitter – The background heavily shaking or moving back and forth faster than raptors in the scene.
- Low jitter – The background slightly shaking or moving back and forth slower than raptors in the scene.
- No movement – The background remain the same for whole video.

The 2 camera movement: high and high jitter was not being selected as dataset since this project was focusing on developing autonomous raptor counting and tracking system. Assumption was made that the camera is not embedded with moving capabilities. Low movement was still acceptable specially those unnoticeable one.

4.2.2 Raptor Distance

Raptor distance describe how far the distance of raptor from the camera. Rather than using value in meter to measure the distance, this project was using the average size of raptors in the video as criteria.

- Far – The average raptor size is lesser than 30 pixels
- Moderate – The average raptor size is between 30 to100 pixels
- Close – The average raptor size is more than 100 pixels

4.2.3 Number of Raptors

Describe the maximum number of raptors appeared in one scene of the video. When there are many raptors, occlusion will occur more often.

- Many – More than 30 raptors
- Moderate – Between 10 to 30 raptors.
- Few – Lesser than 10 raptors.

4.2.4 Cloud Outline

Cloud outline describes whether the cloud thickness is well distributed or poor distributed in the video. Well distributed cloud will be flat and fewer gaps between clouds will be seen while poor distributed is the opposite.

- Solid – The cloud is poorly distributed.
- Flat – The cloud is well distributed.
- No cloud.

4.3 Dataset Selection

Different combinations of dataset were selected based on criteria in section 4.2 to include different situations and constraints for testing the proposed method's robustness and accuracy. Some videos were being cropped to fit into the criteria such as camera movement. The dataset was as following:

No.	Frames	Characteristics			
		Camera movement	**Raptor distance**	**Number of raptors**	**Cloud outline**
1	1038	low jitter	moderate	many	solid
2	679	low	far	moderate	solid
3	549	low jitter	far	few	solid
4	549	low jitter	moderate	few	solid
5	535	no	far	many	flat
6	1075	no	far	many	flat
7	837	no	far	moderate	flat
8	781	no	far	few	flat
9	333	low	close	few	flat

Table 4-3-1: Characteristics of dataset.

4.4 Raw Data Collection

When running tests in MATLAB, raw data are collected for further evaluation of the results. Most of the data were collected by counting manually since there were no correct outputs provided. Data may be little difference from actual correct outputs. Types of raw data collected were described in next sub section.

4.4.1 Raptors, R

The actual number of raptors should be tracked. If a raptor leaves and re-enter the scene, it counts as another new raptor. This value is counted manually.

4.4.2 Raptors tracked, Rt

The number of raptors successfully tracked. A raptor is successfully tracked if and only if an ID or bounding box is labelled on it for at least 90% of its age in the video. Age is known as the actual number of frame the raptor appeared in the scene. This value is counted manually.

4.4.3 Non-raptors tracked, $X\text{-}Rt$

The number of non-raptors allowed for tracking in the video. If a non-raptor is tracked more than 10 frames from the first moment it is detected, it is said to be allow for tracking. For example, a non-raptor is detected in frame No. 178 and if it is not discarded before frame No. 188, it is known as non-raptors tracked. This value is counted manually.

4.4.4 Objects Detected, Od

Objects detected are also known as the current value of ID. Whenever there is a new detection, an incremental ID is assigned to it. New detection is not restrict to new raptor enter the scene, raptor can lost track and redetected as another ID. The ID is unique for every new detections and can use for counting total number of objects detected. Objects detected can include detection of raptors, non raptors, and failure in tracking. This value is obtained from MATLAB.

4.4.5 False Detection, Fd

When the detection is not a raptor, it is known as false detection. In the dataset, there are no other non-raptor objects and only cloud can cause false detection. There are chances where a raptor is detected as two objects and one of them is known as false detection. If the false detection is tracked, it will be a non-raptor tracked. This value is counted manually.

4.4.6 Occlusion, Occ

An occlusion can occurred when two or more objects or raptors overlap each other. This will lead to either successful or unsuccessful remains its correct ID after occlusion. In this project, if two raptors overlap each other, it is counted as two occlusions. This is because when one raptor may remains its ID while the other may not. The number of occlusion is counted manually.

4.4.6.1 ID Remained, $ID\text{-}R$

This data represents the number of raptors successfully remained its ID after occlusion. This value is counted and observed manually.

4.4.6.2 ID Exchanged, *ID-E*

This data represents the number of raptors unable to remain its own ID and used the ID of another raptor in the occlusion. This value is counted and observed manually.

4.4.4.1 New ID, *ID-N*

The last case of occlusion is happened when the raptors used a new ID that does not belongs to any raptors in the occlusion. This data contributed to the number of new objects detected. This value is counted and observed manually.

4.4 Result Evaluation

To measure the performance, several types of evaluation are performed on the test results. The first one is accuracy and the formula is:

$$accuracy = \frac{Rt}{R} \times 100\% \quad (3)$$

Accuracy indicates the how close the number of raptors tracked compares to the number of actual raptors appeared in the whole scene. The best accuracy will be 100% where all raptors successfully tracked.

The second evaluation is false track rate (FTR). The formula is:

$$FTR = \frac{X\text{-}Rt}{Rt + X\text{-}Rt} \times 100\% \quad (4)$$

False track rate indicates within all objects tracked, how many percentage is non-raptors. The best FTR will be 0% where all tracked objects are raptors.

The third evaluation is false acceptance rate (FAR). The formula is:

$$FAR = \frac{Fd}{Od} \times 100\% \qquad (5)$$

False acceptance rate indicates among all objects detected, how many of them is not a raptors. The best FAR will be 0% where all objects detected are raptors.

The forth evaluation is false rejection rate (FRR). The formula is:

$$FRR = \frac{(Rt \times 10\%) + ((R - Rt) \times 189\%)}{R \times 2} \qquad (6)$$

False rejection rare indicates how many percentages of each raptor are not detected during its appearance in the scene. The value is calculated using average of worst case and base case. Base case is that tracked raptors have 0% while non- tracked raptors have 89% false rejection. The worst case is that tracked raptors have 10% and non-tracked raptors have 100% false rejection. In a case where all raptors detected, the minimum FFR will be 5%.

The fifth evaluation is occlusion tolerance rate (OTR). The formula is:

$$OTR = \frac{ID\text{-}R}{Occ} \times 100\% \qquad (7)$$

Occlusion tolerance rate indicates how well the prototype deals with occlusion. The best OTR will be 100% where all occlusion occurred were successfully solved.

The last evaluation is ID change rate (IDC rate). The formula is:

$$IDC\ rate = \frac{Od + ID\text{-}E - (Rt + X\text{-}Rt)}{Rt + X\text{-}Rt} \tag{8}$$

IDC rate indicates for each objects tracked (including raptors and non-raptors), how many times the ID changes in the whole scene. IDC rate does not measure in percentage, but the average time of ID changes for each objects tracked. The best IDC rate will be 0 where no ID changes occurred. If the value is more than 1, the travel path of raptors will be unreliable.

4.5 Test Environment

In this project, the development of prototype and testing was done under the same hardware and software specifications. There are no changes in the specifications from the beginning of development until the ending of testing and experiment. The details are as following:

Specification	Details
Processor	Intel Core i5-2410M CPU@ 2.30Ghz, 2 Core(s), 3 MB L3 Cache
Motherboard	Asus K43SV, Intel HM65
Memory	Samsung 4GB DDR3 @ 1333 MHz
System Drive	Hitachi 500GB SATA 5400rpm
Graphics	Nvidia GeForce GT 540M with 1GB DDR3 VRAM
Operating System	Microsoft Windows 7 Home Premium 64-bit SP 1
Test Environment	MATLAB R2012b

Table 4-5-1: Development and testing environment specifications.

CHAPTER FIVE

TESTS AND RESULTS

5.1 First test

The purpose of first test is to evaluate the robustness and accuracy of method under different circumstances. All dataset defined and selected in Section 4.3 is evaluated here.

No.	*R*	*Rt*	*X-Rt*	*Od*	*Fd*	*Occ*	*ID-R*	*ID-E*	*ID-N*
1	174	170	0	299	0	321	92	155	74
2	23	23	0	59	7	91	30	48	13
3	54	38	0	135	2	16	3	7	6
4	37	37	0	96	4	26	8	9	9
5	54	52	0	102	2	70	27	35	8
6	145	145	0	260	3	541	196	236	109
7	33	33	0	52	0	48	15	26	7
8	26	25	0	34	0	9	3	5	1
9	57	57	0	978	0	26	6	11	9

Table 5-1-1: Raw data collected from test with resolution of 640x360.

No.	*R*	*Rt*	*X-Rt*	*Od*	*Fd*	*Occ*	*ID-R*	*ID-E*	*ID-N*
1	174	166	0	235	0	295	99	162	34
2	23	18	0	143	10	85	32	42	11
3	54	31	0	93	1	10	4	4	2
4	37	37	0	73	0	24	9	12	3
5	54	47	0	135	0	64	28	27	9
6	145	143	0	327	0	472	171	205	96
7	33	31	0	107	0	45	18	20	7
8	26	23	0	75	0	11	9	2	0
9	57	57	1	81	3	26	4	9	13

Table 5-1-2: Raw data collected from test with resolution of 320x180.

No.	Acc (%)	FTR (%)	FAR (%)	FRR (%)	OTR (%)	IDC Rate
1	97.70	0	0	7.06	28.66	1.67
2	100.00	0	11.86	5.00	32.97	3.65
3	70.37	0	1.48	31.52	18.75	2.73
4	100.00	0	4.17	5.00	30.77	1.84
5	96.30	0	1.96	8.31	38.57	1.63
6	100	0	1.15	5.00	36.23	2.72
7	100.00	0	0	5.00	31.25	1.44
8	96.15	0	0	8.44	33.33	0.56
9	100	0	0	5.00	23.08	17.81
Average Rate:	**95.61**	**0**	**2.29**	**8.93**	**30.40**	**3.78**

Table 5-1-3: Results calculated from raw data collected in resolution of 640x360.

No.	Acc (%)	FTR (%)	FAR (%)	FRR (%)	OTR (%)	IDC Rate
1	95.40	0	0	9.11	33.56	2.39
2	78.26	0	6.99	24.46	37.65	9.28
3	57.41	0	1.08	43.12	40.00	2.13
4	100.00	0	0	5.00	37.50	1.30
5	87.04	0	0	16.60	43.75	2.45
6	98.62	0	0	6.23	31.06	3.08
7	93.94	0	0	10.42	40.00	3.10
8	88.46	0	0	15.33	81.82	2.35
9	100.00	0	3.70	5.00	15.38	0.58
Average Rate:	**88.79**	**0**	**1.31**	**15.03**	**40.08**	**2.96**

Table 5-1-4: Results calculated from raw data collected in resolution of 320x180.

5.1.1 Result Discussion

Based on the results, this prototype produced a high accuracy in tracking which represent most raptors could be track for 90% of the time in the dataset video. There are still some lost of raptors due to fast moving raptors in the scene that caused the FFR. When the raptors move too fast, optical flow will not be able to calculate the magnitude and direction. Another reason of lost track is due to the brightness of the raptors. Raptors hidden behind cloud are often brighter than normal ones. This causes the raptor to be filter out in thresholding. The accuracy in 320x180 was reduced especially in video no. 3. This is because most raptors tracked in 640x360 were in at the bottom line of being filtered. When the raptor itself was already blurred in high resolution, it will be harder to track in lower resolution. In video no.4, both resolutions obtained 100% accuracy. This means that the raptors still remained sharp in low resolution.

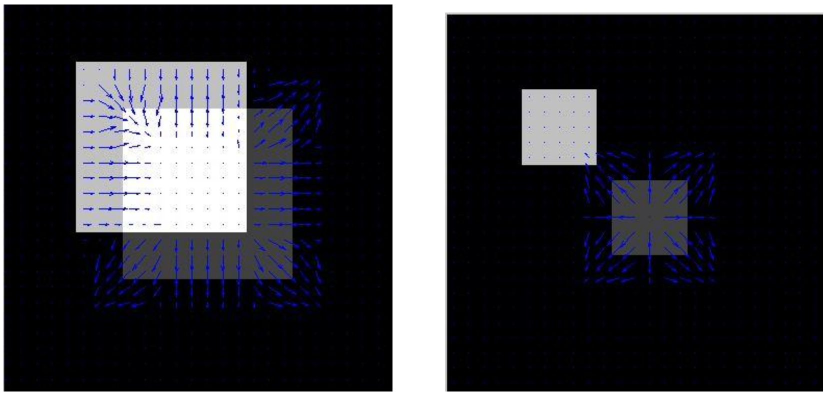


Figure 5-1-1: Example of optical flow on synthetic image. (Left) Overlapping in 2 frames. (Right) Non-overlapping in 2 frames.

Since the project proposed a penalty area, unmoving detection is not tracked in any of the dataset and thus resulting 0% FTR. The FAR is also low because most of the non-raptor objects are filtered out by the pre-processing and thresholding. In high resolution video, the sharp frame caused the false objects remained after thresholding and being detected. The small FAR remained is caused by the exception area. The exception area is applied in square box shape but the raptor is not in square shape. When exception area applied on raptors, the remaining area may detect the solid cloud coincidently. However, the non-moving cloud will be filter by penalty area.

OTR presented in the result was quite low. When two raptors fly in opposite direction and occluded, the occlusion had low percentages of remaining its true id. Indeed, the successful occlusion often occurs when only part of the raptors occluded such as only the wing of raptors occluded.

The IDC rate is also very high in the results causing the travel path drawn is not reliable. When two raptors occluded each other, the Hungarian Assignment Algorithm interpreted the situation as one detection and two predictions. The prediction caused the raptors to exchange ID in FIFO (first come first serve) basis. IDC rate was not solely cause by occlusion; the fast moving raptors also contributed much in this IDC rate. Since fast moving raptors cannot be predict using optical flow, raptors will be redetect continuously as long as it remain its speed. Most of time, a fast raptor can changes its ID 10 times in the scene. One of the good examples is video no.9.

5.2 Second test

Second test was done to determine the difference between processing speed of different resolution and the factors affecting the processing speed.

No.	Frames	Time completed (s)	Processing Speed (fps)
1	1038	99.25	10.46
2	679	55.65	12.02
3	549	43.52	12.61
4	549	41.92	13.10
5	535	50.14	10.67
6	1075	109.12	9.85
7	837	67.31	12.43
8	781	55.72	14.02
9	333	26.35	12.64
Average speed:			**11.98**

Table 5-2-1: Average frame per second for 640x360 resolutions.

No.	Frames	Time completed (s)	Processing Speed (fps)
1	1038	34.13	30.41
2	679	17.69	38.38
3	549	14.01	39.19
4	549	13.60	40.36
5	535	16.93	31.61
6	1075	38.92	27.62
7	837	21.68	38.60
8	781	17.78	43.92
9	333	9.19	36.24
Average speed:			**36.26**

Table 5-1-3: Average frame per second for 320x180 resolutions.

5.2.1 Results Discussion

From the results above, the proposed solution runs 3 times faster when the resolution scaled down by a factor of 4. Other than that, proposed solution had lower fps in no.1 and no.6 where both of them shared the same characteristic: many raptors in the scene. This is because each object detected needed to be evaluate to decide whether they are raptors. Exceptional and penalty area also needed to apply on any raptors that lost track. Therefore, when the number of raptors increases the processing speed will be lower. At the same time, lesser raptors will result in faster processing speed such as no.3, no.4, no.8 and no.9.

5.3 Results Comparison

5.3.1 Processing Speed

To compare the processing speed, 3 videos different in number of raptors is selected. This is because number of raptors would affect the proposed solution processing speed. Below is the processing speed of (Tou & Toh 2012) prototype.

No.	Frames	Number of raptors	Time completed (s)	Processing Speed (fps)	Comparison (fps)
1	1038	Many	116.60	8.90	10.46
2	679	Moderate	77.92	8.71	12.02
3	549	Few	63.15	8.69	12.61
Average speed:				**8.77**	**11.60**

Table 5-3-1: Average frame per second of (Tou & Toh 2012) for 640x360 resolutions. The comparison is referring to proposed solution fps in **Table 5-2-1.**

No.	Frames	Number of raptors	Time completed (s)	Processing Speed (fps)	Comparison (fps)
1	1038	Many	29.31	35.41	30.41
2	679	Moderate	19.25	35.27	38.38
3	549	Few	15.77	34.81	39.19
Average speed:				**35.16**	**35.99**

Table 5-3-2: Average frame per second of (Tou & Toh 2012) for 320x180 resolutions. The comparison is referring to proposed solution fps in **Table 5-2-2**.

The comparison above showed that the proposed solution is faster than (Tou & Toh 2012) in 640x360 resolution but about the same in 320x180 resolution. This indicates that the previous solution's processing speed was more affected by the video resolution and not affected by the number of raptors in the scene. Previous solution was able to achieved 35.16 fps due to the improvement in CPU computation time.

5.3.2 Accuracy

Three different distances of raptor were selected to compare the previous solution with proposed solution.

No.	Distance of raptors	R	Rt	Accuracy (%)	Comparison (%)
3	Far	54	47	87.04	70.37
4	Moderate	37	25	67.57	100.00
9	Close	57	5	8.77	100
Average rate				**54.46**	**90.12**

Table 5-3-3: Accuracy of (Tou & Toh 2012) for 640x360 resolutions. The comparison is referring to proposed solution accuracy in **Table 5-1-4**.

From the results above, there was a clear difference in the average accuracy. When raptors are small and bright, proposed solution will filter them out during segmentation phase. On the other hand, previous solution was using optical flow to track flying raptors. As long as there are movements, optical flow will able to track it.

When the raptors are moderate size and dark, proposed solution was able to segment them successfully. Since the larger raptors produced incorrect and inconsistent optical flow movements, previous solution will had hard time tracking raptors. Same goes to the larger raptors size.

5.3.3 FAR

Since the FAR are caused by moving clouds that have solid outline, three videos with heavy portion of clouds are used to do the comparison. Since the prototype of previous solution do not function like proposed solution, FAR could not be calculated using the same method. A simple calculation on the number of false detection is done for comparison.

No.	False Detection	Comparison
2	295	7
3	31	2
4	275	4

Table 5-3-4: False detection of (Tou & Toh 2012) for 640x360 resolutions. The comparison is referring to proposed solution false detection in **Table 5-1-1**.

Result above clearly showed the reduction in false detections. When dealing with solid cloud outline, proposed solution was able to filter out most cloud that caused the false detections. On the other hand, optical flow was very sensitive to movement and thus performed badly when camera jittered.

CHAPTER SIX

CONCLUSION

5.1 Conclusion

This project was able to reduce the FAR to average of 2.29% for video of 640x360 and 1.31% for video of 320x180 with the assumption of the background or camera is not moving intensively. This assumption is valid since the main goal of this project is to create an automation process of tracking multiple raptors in the sky. The static camera can be install at somewhere and tracks raptors without any human labour needed.

The proposed solution also achieved an average accuracy of 95.61% for video of 640x36o and 88.79% for video of 320x180. The difference in accuracy of two resolutions video was not large and still within the expectation. Processing the dataset with lower resolution might be better since it achieved 36.26 fps while the higher resolution only achieved 11.98 fps processing speed. 30 fps is the minimal requirement needed in real time implementation. The extra 6.26 fps can be used to implement some algorithms to achieve a better result. However, the lower resolution also caused higher FFR (15.03%) which lead to reduction in accuracy.

This project had taken the first step into solving occlusion problem and yet the occlusion tolerance only achieved a low rate of 30.40%. Most of the occlusions were unsolved and caused high IDC rate. The extra feature: travel path provided by this project was unreliable due to the high IDC rate.

5.2 Future Work

This solution may produce a high accuracy, but there are still many improvement can be done on solving the occlusion. Occlusion is a very vital factor that needs to be considered in tracking and count flock of raptors in sky. When two raptors occluded, they should be count as two instead of one. At the same time, occlusion also caused the travel path unreliable. Modification on Hungarian assignment or proposing another better method is encouraged to solve the IDC rate.

Other than that, improvement also can be done on the prediction method on raptors future location by analyzing the travel path of raptors. As a prerequisite, a reliable travel path must be provided for better prediction. This flight path can also be use to decide whether the raptor entering the scene is new or old one. With this, the assumption of treating re-enter raptor as another that made in this project can be cancel out.

In this project, the all dataset provided had no ground truth and the results produced may be biased to human factor. If the solution is good in segmentation, it will be hard to detect the lost tracked raptors with human eyes. Some modules such as counting number of raptor entering scene, number of occlusion and number of raptor in every frame can be produced to compute the ground truth of dataset so comparison can be done on test results and actual results.

Finally, this project should be test under actual environment in the future. The current dataset are selected under constraint and assumption and may not be robust in the actual world. There are still many other tests could be done such as different sky colour, include mountain and tree in background, and performance under raining scene. There are still many tests need to be done to increase the solution robustness and accuracy in actual environment.

REFERENCES

Barron, JL & Thacker, NA 2005, Tutorial: Computing 2D and 3D Optical Flow, Tina Memo, Available from: < http://www.tina-vision.net/docs/memos/2004-012.pdf >. [20 March 2013].

BildStein, KL 2006, *Migrating raptors of the world: their ecology & conservation*, Comstock Publication Associates, New York.

BirdLife International 2013, *Welcome to BirdLife Preventing Extinction Programme*. Available from: <http://www.birdlife.org/extinction/index.html>. [20 March 2013].

Borgnis, E 2010, *Bird communities as bioindicators of stream degradation*. Available from: <http://conservationmaven.com/frontpage/bird-communities-as-bioindicators-of-stream-degradation.html >. [20 March 2013].

Bovik, A (ed.) 2005, *Handbook of image & video processing*, 2nd edn, Elsevier Academic Press, London.

Brau, E, Barnard, K, Palanivelu, R, Dunatunga, D, Tsukamoto, T & Lee, P 2011, 'A generative statistical model for tracking multiple smooth trajectories', *proceedings of 2011 IEEE conference on computer vision and pattern recognition*, pp. 1137-1144.

Bruhn, A, Weickert, J & Schnorr, C 2005, 'Lucas/Kanade meets Horn/Schunk: combining local and global optical flow methods', *International Journal of Computer Vision*, vol. 61, no. 3, pp. 211-231.

Chin, KH 2005, *Short introduction to particle filter*, Institute for Infocomm Research. Available from: <http://www1.i2r.a-star.edu.sg/~hock/Publications/particle_filter.pdf>. [20 March 2013].

Debus, S 2012, *Birds of prey of Australia: a field guide*, 2nd edn, CSIRO PUBLISHING, Australia.

Dermitdemmikro 2010a, ‘Bird flock static full-hd’, video, YouTube, 7 January, viewed 30 March 2013, <http://www.youtube.com/watch?v=mTMOpdm96bo&list=UU0JWit876qdpQO1EpclBtCA&index=18>

Dermitdemmikro 2010b, ‘Bird flock moved full-hd’, video, YouTube, 8 January, viewed 30 March 2013, <http://www.youtube.com/watch?v=TOxFtytQ_6k&list=UU0JWit876qdpQO1EpclBtCA&index=17>

Freemake.com, computer software 2013. Available from: <http://www.freemake.com/free_video_converter/>. [28 March 2013].

Goszczynska, H (ed.) 2011, *Object Tracking*, Intech, Croatia.

Kalafatic, Z, Ribaric, S & Stanisavljevic, V 2001, ‘A system for tracking laboratory animals based on optical flow and active contours’, *proceedings of eleventh international conference on image analysis and processing*, pp.334-339.

Maggio, E & Cavallaro, A 2011, *Video tracking: theory and practice*, Wiley, Chichester.

Nan, D, Morishita, S & Yokoi, H 2011, ‘A tracking method of multiple moving objects for animal watch sensor systems — measures for random mobility and the problem with connection and separation’, *proceedings of 2011 SICE annual conference*, pp. 1991-1994.

Ng, HF 2006, ‘Automatic thresholding for defect detection’, *Pattern Recognition Letters*, vol. 27, no. 14, pp. 1644-1649.

Raptor Migration, n.d. Available from: <http://www.raptorcorridor.org/files/images/Thermalsoaring.jpg>. [20 March 2013].

Spampinato, C, Cheh-Burger, YH, Nadaraian, G & Fiaher, R 2008, ‘Detecting, tracking and counting fish in low quality unconstrained underwater videos’, *proceedings of third international conference on computer vision theory and applications*, vol. 2, pp. 512-519.

Szeliski, R 2010, *Computer vision: algorithm and application*, Springer London, Washington. Available from: <http://szeliski.org/Book/>. [20 March 2013].

Thrun, S 2006, Lecture 7 optical flow, lecture notes distributed in Computer Vision at The Stanford University, California on 6 February 2006.

Tou, JY & Toh, CC 2012, 'Optical flow-based bird tracking and counting for congregating flocks', in *Intelligent Information and Database Systems*, eds Pan, JS, Chen, SM & Nguyen, NT, Springer Berlin Heidelberg, Kaohsiung, pp. 514-523.

Tweed, D & Calway, A 2002, 'Tracking multiple animals in wildlife footage', *proceedings of sixteenth international conference on pattern recognition*, vol. 2, pp 24-27.

Welch, G, Bishop, G 2006, *An introduction to kalman filter*, lecture notes distributed in Computer Science at The University of North Carolina, Chapel Hill on 24 July 2006.

Yilmaz, A, Javed, O & Shah, M 2006, 'Object tracking: a survey', *ACM Computing Surveys*, vol. 38, no. 4, article 13, pp.1-45.

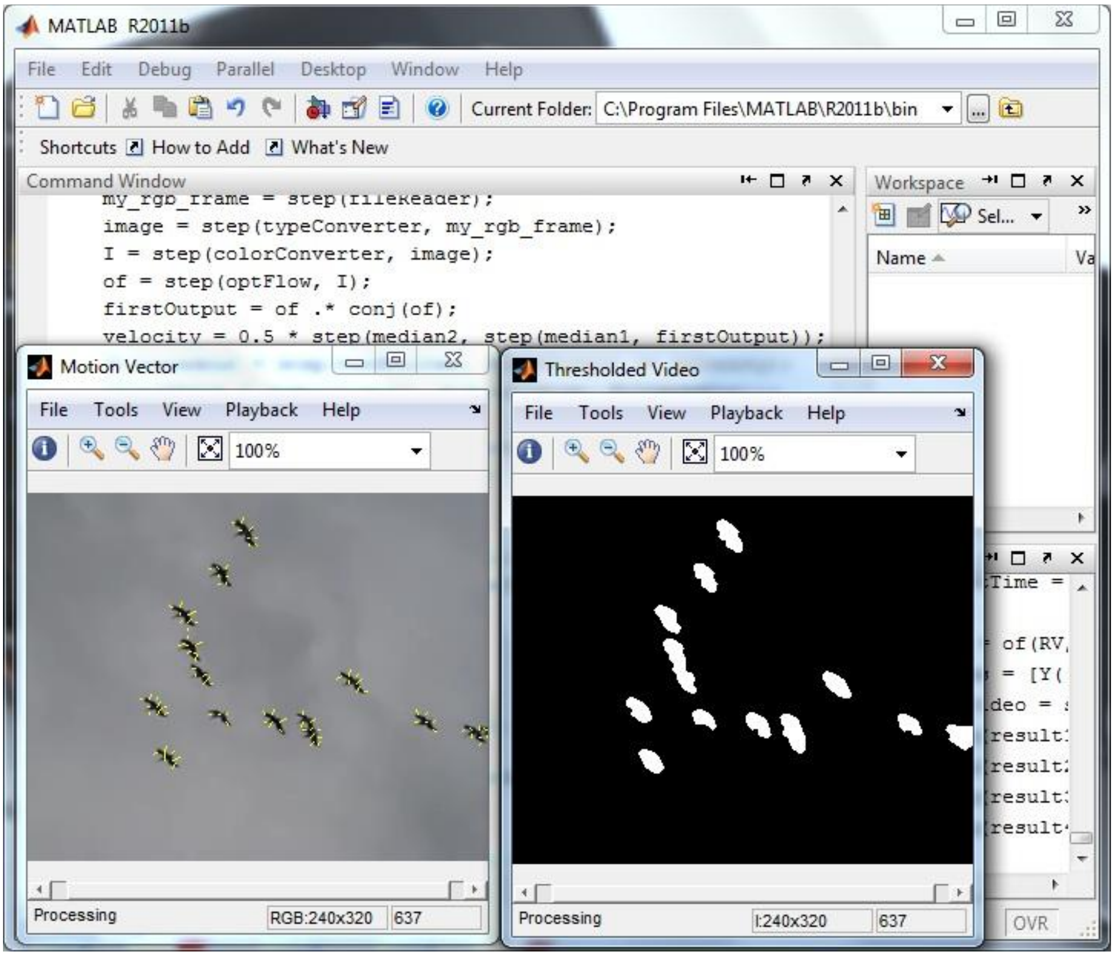
Zhang, XY, Wu, XJ, Zhou, X, Wang, XG & Zhang, YY 2008, 'Automatic detection and tracking of maneuverable birds in videos', *proceedings of 2008 international conference on computational intelligence and security*, pp. 185-189.

Zija777 2009, 'Bird flying over jax beach', video, YouTube, 6 December, viewed 30 March 2013, <http://www.youtube.com/watch?v=wouDAQ7UFnE>

,

APPENDIX A

DEVELOPMENT TOOLS



APPENDIX A-1 MATLAB R2011b GUI