2012

Birds/bats motion tracking with infrared radiation camera for wind farm applications

Lai Wei

The University of Toledo

Follow this and additional works at: http://utdr.utoledo.edu/theses-dissertations

Recommended Citation

This Thesis is brought to you for free and open access by The University of Toledo Digital Repository. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of The University of Toledo Digital Repository. For more information, please see the repository's About page.
A Thesis

entitled

Birds/Bats Motion Tracking with Infrared Radiation Camera for Wind Farm Applications

by

Lai Wei

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the

Master of Science Degree in Electrical Engineering

Dr. Mohsin M. Jamali, Committee Chair

Dr. Peter V. Gorsevski, Committee Member

Dr. Ezzatolah Salari, Committee Member

Dr. Patricia R. Komuniecki, Dean
College of Graduate Studies

The University of Toledo

December 2012
An Abstract of

Bird/Bats Motion Tracking with Infrared Radiation Camera for Wind Farm Applications

by

Lai Wei

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Master of Science Degree in Electronic Engineering

The University of Toledo
December 2012

This study aims to design a robust and efficient monitoring system for analyzing the bird migration data for wind farm applications. This system is able to process thermal image sequence data and output accurate results suitable for interpretation by wild life biologists. An IR video processing algorithm has been proposed which meets the IR video processing requirement for this project. The propose algorithm consists of background subtraction and consecutive frame subtraction, frame selection, 3-D region labeling and breakpoint recovery. It has been simulated for performance evaluation. It is then used to process spring 2011 bird migration data that has been collected in Ottawa National Wildlife Refuge in Ohio. Results from this study will be useful for wild life biologists to make intelligent decision for siting of wind turbines. It will also help policy makers to develop an appropriate public policy for wind farm development in an area with extensive avian activity.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>i</td>
</tr>
<tr>
<td>Contents</td>
<td>ii</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Applications of Motion Tracking</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Objective of Birds/Bats Monitoring via IR Camera</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Approaches</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Thesis Outline</td>
<td>5</td>
</tr>
<tr>
<td><strong>2 Background Information and Previous Work</strong></td>
<td>8</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>8</td>
</tr>
<tr>
<td>2.2 IR Data</td>
<td>8</td>
</tr>
<tr>
<td>2.3 Background Knowledge of Motion Tracking System</td>
<td>9</td>
</tr>
<tr>
<td>2.4 Previous Work</td>
<td>10</td>
</tr>
<tr>
<td><strong>3 IR Motion Tracking</strong></td>
<td>12</td>
</tr>
<tr>
<td>3.1 Information on IR Processing</td>
<td>12</td>
</tr>
</tbody>
</table>
3.2 Motion Tracking Techniques ............................................................ 13
  3.2.1 Algorithms for Motion Tracking ............................................. 13
  3.2.2 Noise Filtering ..................................................................... 24
  3.4.3 Image Thresholding ............................................................... 31
  3.3 Summary ................................................................................... 38

4 Target Motion Tracking System Design ........................................ 41
  4.1 Motion Detection ....................................................................... 42
  4.2 Frame Selection ........................................................................ 45
  4.3 3-D Region Labeling ................................................................ 47
  4.4 Breakpoint Recovery ................................................................. 49
  4.5 Target Information Analysis and Results Output ...................... 51
  4.6 System Comparison .................................................................. 55
    4.6.1 Adaptive Background Subtraction System Design .............. 55
    4.6.2 PCA Based Tracking System .............................................. 55
    4.6.3 Proposed Target Tracking Algorithm ................................. 56
4.7 Summary ...........................................................................................................57

5 IR Video Processing ..........................................................................................59

5.1 Study Area and Study Time ...........................................................................59

5.2 Data Collection Time and Techniques .........................................................59

5.3 Video Data Processing ...................................................................................60

5.4 Validation .........................................................................................................62

5.5 Results ...............................................................................................................62

5.5.1 Results of Sample with One High Velocity Object .................................62

5.5.2 Results of Sample with Multiple Objects ..................................................65

5.5.3 Computation Time Comparison ....................................................................67

5.6 Migration Data Analysis ..................................................................................68

5.7 Summary ...........................................................................................................74

6 Conclusion and Future Research ......................................................................84

References .............................................................................................................86
Chapter 1

Introduction

1.1 Applications of Motion Tracking

Moving object tracking has received considerable attention in the past few years, mainly because of the wide range of its potential applications. Many studies and works have made great contribution to the development of this new technology. It was primarily applied for detecting movements in surveillance system. In past ten years, object tracking with new techniques has been used ubiquitously. Generally speaking, the motion tracking systems can be applied wherever efficient monitoring is needed; especially in the security-sensitive areas such as banks, department stores, parking lots, and so on. There are obviously a large amount of potential users because of one of its most important application domains, which is smart surveillance. The “smart” systems do not only detect the moving targets which alone might lead to false alarm, but also analyze and classify their behaviors to find out the suspicious actions.

Furthermore, face recognition is another important application domain of motion tracking [2]. The object tracking system can recognize human face by identifying different facial characteristics. In this way, a large number of cameras already installed in security-sensitive areas could be used as a tool for efficient real-time automated or semi-automated surveillance. For example, when using a videophone, tracking and locating a
user’s face in the video would be more efficient if object’s features are used for computation, storage and transmission.

On the other hand, face recognition is very useful in the entertainment industry. Movie or gaming industry can also benefit a lot from the modern visual technology. For example, the animation or science fiction movie modeling and action capturing in video games can all be based on this technology.

In addition, object tracking technology can also be adopted as an important and efficient way of studying other relevant fields. By analyzing the recorded videos, tracking system can capture the movement and classify objects. Also, it can provide researchers with the trajectories of objects and related information such as velocity, density, distance and object size. For instance, a traffic monitoring system can analyze the traffic flow in a specific area by recording videos. In some cases, the motion tracking can be applied with moving cameras, such as in the area of robotics. Robots can automatically follow an object by capturing and locating it. This is commonly seen in exploration for critical environment which does not allow manned operation.

1.2 Objective of Birds/Bats Monitoring via IR Camera

As many algorithms of object tracking have been developed, there are a variety of purposes for studies and research related to object motion tracking. This project aims to assist wildlife biologists in the observation of birds’/bats’ behaviors in the vicinity of wind turbines. We have developed a system which can be used to track the birds’/bats’ flying trajectories and analyze their behaviors, such as direction, velocity, travel distance and straightness. The algorithm deals with IR camera data which is recorded from the
birds’/bats’ activities during nocturnal migration. This goal of this research is to help and
guide the mitigation process which may reduce unwanted effects on birds/bats.

A good algorithm should not only be robust for a particular application, but also
should have shorter computation time. In this project, the algorithm is required to provide
an efficient way of tracking all the birds’/bats’ activities and responding with accurate
information of target behavior; results of data processing will lead to the classification
works. Therefore, the algorithm should perform well under various weather conditions
and provide accurate results.

1.3 Approaches

Challenges of tracking in a dynamic, adversarial and noisy environment are as follows:

1) **Limited visual depth:** The camera has relatively wide Field of View (FOV), but
limited depth of field. It causes difficulties in separating the targets from the
background because some time their intensity is similar to the background.

2) **Complex background noise:** There are multiple factors which may cause
difficulties in separating the background with targets of interests. The weather
condition can be different. Clouds could have higher intensity which may be
close to the intensity of objects. These two reasons are challenging when targets
are occluded under the clouds, insects and other random noise.

3) **Acquiring target information:** The algorithm should not only identify the
moving targets, but also accurately acquire other related information. The results
can be affected by noise. An unavoidable problem is that when we are doing the
denoising some useful information could get removed as well; the target
information should consider results from previous frames. So it is important to recover the data from imperfect outputs.

4) **Multiple targets appear in the scene:** When targets are traveling through the camera field of view, more than one target occasionally appears at the same time. The algorithm should be able to handle this situation. So the problem for us is to separate the targets from each other and collect the target information respectively.

The project is focused on analysis of behaviors of birds in different conditions, so above challenges need to be addressed. The goal is to develop solutions that can be efficient and robust. Following are some of the ideas that can be exploited in this work.

1) **Appropriate selection of the threshold:** An accurate threshold needs to be determined for optimum object segmentation. Dynamic determination of threshold values is also desired.

2) **Adaptive background model and noise reduction:** Noise can come from different sources therefore an appropriate noise reduction technique is desired. Firstly, the background model needs to be adjustable during the tracking process. This could be helpful for the relatively stable noise (such as clouds and trees). In regards to the other types of noise, different techniques for tracking could be employed to reduce noise.

3) **Break point recovery:** For the problem of gaining target information (primarily caused by unexpected object removal), a method has been developed to connect the break points. This algorithm can acquire the target information from possible objects, calculate the similarity, identify and locate the break points of objects.
The algorithm can then reconnect and obtains possible trajectory. This method can deal with the breaks caused by noise reduction as well as object segmentation.

4) Three dimensions region labeling with frame pre-selection: For the multiple targets problem, we apply the region labeling into three dimensions. By doing this, the labeling job will not only be based on the current frame, but also on the previous frame as well as the next frame. So this method will mark the pixels in the video all together instead of every single frame. But this method would cause two small problems: memory redundancy and progressing inefficiency. To deal with them, we brought in another idea which is just selecting the frames containing useful information, and packing them respectively. So when performing three-dimensional region labeling, the algorithm will only go through the groups of selected frames. This approach increases the processing speed by a large margin.

Based on the above facts, challenges can be overcome by exploiting existing techniques and creating new algorithms. Therefore, we claim that the algorithms about monitoring and analyzing birds/bats behaviors will be robust and reliable for target tracking.

1.4 Thesis Outline

This thesis is organized as follows:

In chapter 2, some basic knowledge of motion tracking will be introduced along with previous works on this field.
In chapter 3, we will introduce the information of Infrared Radiation (IR) camera and its data since the data for this project is captured by IR camera. Since there are differences between IR camera and normal camera, the processing techniques will also be different. The usage of the technique for target monitoring will be introduced as well. Moreover, motion tracking, noise filtering and image thresholding will be presented. These algorithms are essential factors in a motion tracking system.

In chapter 4, a new video processing algorithm has been proposed which is appropriate for IR video processing in this project. In chapter 5, some information about our study, such as the study area, study time and data processing method will be generally introduced in the beginning. Then we will continue with the data collection time and techniques. In the third section, migration data analysis and other results will be presented.

In chapter 6, we will conclude our work and future research directions will be provided.
Chapter 2

Background Information and Previous Work

2.1 Introduction

Various techniques for bird monitoring already exist and some of them are more time-consuming and expensive than the others. Direct observation is the simplest and oldest technique and may differentiate the migratory birds based on sizes, colors, songs and flight characteristics. Slight modifications also allow night observation by applying terrestrial vertical light beam or ceilometer techniques. Another approach for bird detection is the use of acoustic-based technology to identify bird specific signatures such as drums from woodpeckers. Also radar-based monitoring technology has been used for bird detection, although the coarse spatial resolution is a limiting factor.

This project utilizes IR camera for monitoring of birds and bats in the vicinity of wind turbines. The use of passive infrared cameras (measuring avian body heat) allows cloudy night observation and reduces disturbances due to artificial light. Radio tracking or telemetry is another method to monitor birds. A small radio transmitter is attached to the bird that sends a periodic beep signal which can be tracked down.

2.2 IR Data

The thermal camera is a sophisticated thermal imaging camera that provides excellent night visibility and situational awareness, even in absolute darkness. According to the observation time of our study, from one hour after sunset to one hour before
sunrise, the equipment is required to be able to work not only in low-light condition, but also complete darkness.

The IR data in this project is recorded by FLIR SR-19 camera with “White Hot” palette in which the objects with higher temperature appear white or brighter than the black or dark shades. This IR camera system has standard resolution Focal Plane Array (FPA) of 320 (H) x 240(V) pixels. The detector frame rate is thirty frames/second. The IR camera was installed pointing up vertically; the top of the field of view was rotated 19.75º from north. Figure 2-1 gives a picture of FLIR SR-19 camera.

![Figure 2-1: FLIR SR-19 thermal camera](image)

The data collected in this project has simpler background than the one used in traffic monitoring applications. Major sources of noise are clouds and insects. However, there are some difficulties in processing data due to lot of white noise that can reduce the quality of data. There are few sharp differences of intensity between some objects and
their background which may cause some errors by ignoring the minor objects. Therefore, an important issue of this project is to identify the moving objects in an effective way.

2.3 Background Knowledge of Motion Tracking System

Motion tracking system consists of several different steps to achieve the desired objective. Each of them plays an important role in the entire system. Different data or project requires different functions of the system, which leads to different combinations of parts. Generally, there are three essential parts for any tracking system regardless of the requirement of the project: (1) motion segmentation and tracking, (2) noise filtering and (3) image thresholding.

In any motion tracking system, motion segmentation and tracking is the core algorithm because it identifies and locates the moving objects and reads the related information. The selection of tracking technique will directly affect the performance of the tracking system. Depending on the objectives of a project, one or more tracking algorithms could be applied.

The noise filtering appears in most signal processing systems. Noise is the undesired information of the data/source signal that needs to be removed. The quantity of noise is a main factor that affects the quality of outputs. Many different types of noise filters could be employed into computer vision analysis systems. In this thesis, three widely used filters are introduced.

Image thresholding is the key issue of motion tracking and defines the performance of the system more than the tracking algorithm. Image thresholding refers to a technique used for converting RGB or gray-scale images into binary images. It is important for motion tracking system because binary image sequence has only two values: 1 and 0 to
represent moving objects and background. So the function of image thresholding is to separate the moving objects from their background. A good threshold selection could be extremely helpful for a system. Image thresholding could be classified as fix thresholding and auto thresholding. Both of these techniques will be discussed in this thesis.

2.4 Previous Work

In motion tracking, the precondition is that the velocity or acceleration is slow relative to the frequency of image acquisition, allowing each feature to be tracked according to its spatiotemporal continuity. There would be little or no ambiguity in matching when the location of objects in a new frame could be predicted by the previous frames. By averaging properties of the over-determined systems, it is possible to tolerate occasional incorrect matches as long as the errors are limited in size by a small search window. So such systems can achieve reliable performance in the frame-to-frame motion up to several pixels.

The system described in this thesis demands an acceptable robustness when processing data under challenging conditions. The thermal image sequence we utilized to record the birds’/bats’ behaviors can provide us with purer color information than normal videos. Since there are only gray-scale images in the video, the system requires high accuracy when identifying moving targets. In order to efficiently segment motion information, a fast and accurate auto thresholding technique is expected to be used in our project. Otsu’s method (Otsu, 1979) [19] which is borrowed from a concept of statistics provides a good way of handling the images with uneven height in different peaks and wide bottoms. This method has laid a solid foundation for the following image thresholding techniques. Su, Lu and Tan (2010) [27] developed another method based on
Otsu’s method and made it more tolerant to the uneven illumination and other types of degradation. The video data for our project is captured under multiple weather conditions, which may affect the quality due to random noise. Thus, one important task for this system is denoising. (Dimitri, 2003) [26] developed a fuzzy image filter for denoising. This algorithm can decrease the quantity of white noise by averaging neighborhood values. Unlike median filter, it avoids blurring edges of the objects when denoising, which makes it more suitable for thermal image data. Object detecting and segmentation have been proposed by Chen & Zhu (2010) [1]. The technique allows background subtraction and consecutive frames subtraction to work cooperatively in a given system. This method takes advantages of combining both background subtraction and consecutive frames subtraction. Pandey, Singh and Tripathi (2011) [20] borrowed the concept called Principle Component Analysis (PCA) from statistics and provided an alternative way of tracking objects from video sequence. It is popular in video processing field because it is fast and easy to apply. This algorithm looks very promising and can be used in many different ways.
Chapter 3

IR Motion Tracking

3.1 Information on IR Processing

The techniques for IR data processing are quite different from the normal video processing techniques. IR processing has some unique features that can be exploited. IR camera usually captures video with less background information and highlights objects with higher temperature. This helps in subtracting objects from the simple background model. IR camera could record the objective features much better than normal cameras. Furthermore, IR camera is less sensitive to the changes in illumination and makes the background model more stable.

However, the challenge of IR processing is also obvious. Most studies of motion tracking depend on color information to recognize the moving objects. Since in IR data there are only gray-scale images (although they are recorded in RGB format), the color information is not very useful for identifying different objects in details. For example, in normal videos, birds and bats could be quite different in colors; in IR data, they just look the same with the similar intensity. Another problem for IR processing is that the data shows the heat emitted by the objects instead of their actual shapes. This would blur the object shapes and provides inaccurate information of the size and shape, which makes it more challenging for object classification.
3.2 Motion Tracking Techniques

The function of human eyeballs and brain allows us to track moving objects from any video. However, in computer vision field, how does a system perform motion tracking in video? We can uncover this secret by deeply studying the procedure of motion tracking system. This chapter will introduce the essential parts of motion tracking system and discuss currently used techniques. Various techniques will be compared and their advantages and disadvantages will be pointed out in the following sections.

3.2.1 Algorithms for Motion Tracking

A typical vision-based object tracking is to estimate the state of a moving object from a sequence of camera images. In our case, the techniques applied on observing birds/bats belong to the 2-D rigid object tracking [32]. 2-D rigid object tracking is intended to determine the motion of the projection of one or more rigid objects on the image plane. This motion is induced by the relative movement between the camera and the observed scene.

A basic assumption of 2-D rigid motion tracking is that there is only one rigid, relative movement. This is the case of, for example, a moving car. This assumption rules out the articulated objects such as a moving human body, or deformable objects such as a piece of cloth.

Color information proves to be effective in motion tracking, because it enables fast processing that can lead to real-time tracking (e.g., 20-30 frames per second) while providing robust results. Color segmentation is the core of color-based object tracking algorithms. It can be very useful tracking information, if the color of the target object can
be modeled efficiently and distinguished from the color of other objects in the scene and its background.

This section provides further information about several widely used color-based object tracking algorithms. With the comparison and discussion, the advantages and disadvantages of each algorithm will be highlighted.

3.2.1.1 Background Subtraction

In each color-based motion tracking techniques, color modeling is a key issue. The difference of building color model serves different purposes. For example, tracking a particular person from a large scene with other moving objects requests the specific color information of the objective person. So the color of skin or clothing is the important information when building the color model.

However, in our project we are required to track all moving objects instead of one specific object. In this case, a different color model needs to be built. We take the background as a model rather than a specific object in order to distinguish every target as output for the next step. This algorithm is called background subtraction. Figure 3-1 shows an example of background subtraction.

![Figure 3-1: Background subtraction and binary output](image)

In the most naive implementation of this approach, the model of scene is a single image that is acquired without the presence of any moving object during an initialization.
step. Each new frame is subtracted from the scene model to segment any foreground (i.e., moving) objects. This approach provides a fast, simple method to segment moving objects. It is popular when applying in real-time tracking system with stable background (e.g., parking lots). The background subtraction function is shown in Equation (3.1):

$$D_k(x,y) = |f_k(x,y) - B_{k-1}(x,y)|$$  \hspace{1cm} (3.1)

where $f_k$ is current frame, $B_{k-1}$ is the background model. The condition of thresholding operation is:

$$G_k(x,y) = \begin{cases} 
0, & D_k(x,y) < T \\
1, & Otherwise 
\end{cases}$$  \hspace{1cm} (3.2)

where T is the binary threshold. If $G_k(x,y)$ is zero then it will be considered as background.

The major issue of background subtraction is how to provide robustness against changes in illumination and unexpected noise. This can be achieved, for example, by controlling the illumination conditions, but it is infeasible in a real world environment especially for outdoor scenes. Alternatively, illumination invariance or color correction can be employed.

### 3.2.1.2 Modified Background Subtraction

To overcome the disadvantages of classic background model, all kinds of modified background model can be taken into consideration. The core of these algorithms is that instead of taking one single constant image as model, the algorithms update information from a current frame into model. This helps the background subtraction improve the robustness with minor reduction of the processing speed. In our project, two algorithms have been tested, namely running average and adaptive background.
In running average background model, we take the mean value of previous ten frames to build a model for the current frame subtraction. The algorithm is shown in Equation (3.3):

\[ B_{ave}(i) = \frac{1}{n} \sum_{i-n}^{i-1} B(i) \]  \hspace{1cm} (3.3)

where \( B_{ave}(i) \) is the background model for current frame and \( i \) is the frame number.

This algorithm updates every time when \( i \) is changing. At the beginning of the video (before the 10\(^{th}\) frame), it takes the constant background as model. The updating keeps going on from the 11\(^{th}\) to the end of video. This algorithm is a fast, direct-improved method based on background subtraction. It is useful when number of targets is not very large. When there are a large number of objects, or even some slowly moving object appeared in the scene, this algorithm may give an inaccurate background model which directly affects the robustness of results.

The adaptive background model [4] is built on the video with slow changes in illumination and minor noise. Instead of taking previous constant number of frames to build model, adaptive background only updates the last frame information into the video. The algorithm is shown as Equation (3.4):

\[ B_{apt}(i) = p \times C(i) + (1 - p)B(i - 1) \]  \hspace{1cm} (3.4)

where \( C(i) \) is the current frame, \( p \) is the adaptive updated rate.

Adaptive background offers an easily computable algorithm with reliable background model. It is simple to segment objects from the background regardless of their moving speed. It is also fast and easy to apply. However, the disadvantage is that it
may introduce unexpected errors with rapid changes in illumination and high random noise.

3.2.1.3 Mixture Gaussian Background Subtraction

The two previously discussed algorithms for background subtraction have some common drawback that is they are not robust when there is relatively high random noise in the videos. This problem is obviously more prominent in the outdoor video sequences. In continuous image sequence, we can obtain more frames. If taking every pixel of the frames as a random process, we can build Gaussian models for the color value [11]. If the video is true color meaning it has RGB value for every pixel, we can build a 3-D Gaussian model for pixels and is known as Mixture Gaussian Background Subtraction (MGBS) [12]. This model is continuously updated within the actual operation of the tracking system by using a simple adaptive filter. For each pixel of video sequence, the likelihood that its color stems from the color distribution of the pixel with the same location in the background model is evaluated. If the likelihood is small the pixel is labeled as foreground pixel (i.e., the object). For a more complex condition, the algorithm has even three to five models for each pixel. Each of these Gaussians can represent either a background or a foreground “pixel process”. For example, in an outdoor video scene, one particular pixel can belong to different objects in different time. It may belong to a leaf of a tree for this second or a roof of a house for the next second. Normally, for one specific pixel $x$, the probability density $P(x)$ is represented as in Equation (3.5):

$$P(x) = \sum_{i=1}^{K} \omega_i \ast \eta(x|\mu_i, \sigma_i^2)$$

(3.5)
where $x$ is a $D$-dimensional continuous-valued data vector, $\omega_i$ is the mixture weights, and $\eta(x|\mu_i\sigma_i^2)$ is the component Gaussian densities. Each component density is a Gaussian function of the form [13]:

$$
\eta(x|\mu_i\sigma_i^2) = \frac{1}{(2\pi)^{D/2}|\sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(x - \mu_i)'\sigma_i^{-1}(x - \mu_i)\right\}
$$

(3.6)

with mean vector $\mu_i$ and covariance matrix $\sigma_i$.

Mixture Gaussian manifests itself as the most robust approach to handle very complex conditions. It prevails in the industry which requires accuracy more than time efficiency. The disadvantage of Mixture Gaussian is clear as well as its processing time is much longer than other algorithms.

### 3.2.1.4 Consecutive Frames Subtraction

In consecutive frame subtraction, the different feature from background subtraction class is that there is a specific model. Consecutive frame subtraction takes two continuous or adjacent frames for subtraction [17]. Each output of subtraction is a new image that is used for further processing. The algorithm is as follows:

$$
D_k(x, y) = |f_k(x, y) - f_{k-1}(x, y)|
$$

(3.7)

$$
G_k = \begin{cases} 
0, & D_k(x, y) < T \\
1, & \text{Otherwise} 
\end{cases}
$$

If $G_k$ is zero then it will be considered as background. Consecutive frame subtraction is an effective way to deal with the constant or slowly changing noise and easy to program and compute; but it is not as robust as background subtraction. An extended consecutive frame subtraction can be used [18]. In this method, we take three or more frames instead of two for subtraction. After that, we apply logic calculation to the results and acquire
new results. Given that there is an array of binary image sequence $\{I_m\}$, the output of two continuous frames subtraction is $D_m$ represents as follows:

$$D_m^T(i,j) = |I_m(i,j) - I_{m-1}(i,j)|$$  \hspace{1cm} (3.8)

Thus the output of two previous frames of frame $m$ is:

$$D_{m-1}^T(i,j) = |I_{m-1}(i,j) - I_{m-2}(i,j)|$$

If a pixel $(i,j)$ belongs to an object in both $D_m^T(i,j)$ and $D_{m-1}^T(i,j)$, it will be taken as a pixel of object and be counted in result $D_m^T(i,j)$. The logic will be as follows:

$$DD^T_m(i,j) = \begin{cases} 1, & D_m^T(i,j) = 1 \\ 0, & D_{m-1}^T(i,j) = 1 \cdot D_m^T(i,j) = 1 \\ \text{Otherwise} & \end{cases}$$  \hspace{1cm} (3.9)

where $T$ is binary threshold. If $DD^T_m(i,j)$ is zero then it will be considered as background.

The extended consecutive frames subtraction is more robust than the previous approach and easy to program and process. The drawback is that the results occasionally have “holes” which may prevent further track steps. Although considered as the classic consecutive frames subtraction, the objects may not be completely segmented as well.

3.2.1.5 PCA Motion Tracking

Principal Component Analysis (PCA) [21] is a standard tool in modern data analysis in diverse fields from neuroscience to computer graphics - because it is a simple, non-parametric method for extracting relevant information from confusing data sets. This widely used method is based on the statistics. The key issue of PCA is extracting the principal component from the data, which in motion tracking means extracting the moving object from the background and noise.
We suppose that our input data is a naive basis \( X \) and \( X \) is a vector recording corresponding object position \( \{x_1, x_2, \ldots, x_n\} \). For each \( x_i \) there is an \( m \)-dimensional column vector:

\[
x_i = \begin{bmatrix}
    a_1 \\
    a_2 \\
    a_3 \\
    \vdots \\
    a_m
\end{bmatrix}
\]  

(3.10)

In our project, for each column vector, there are only two dimensions which represent the coordinates of pixels. Thus \( x_i \) in this project is

\[
x_i = \begin{bmatrix}
    x_{obj} \\
    y_{obj}
\end{bmatrix}
\]

If in an image, there are 10,000 object pixels, and then there are 10,000 of \( x_i \) vectors.

The core idea of PCA is extracting the principal components from data \( X \) by linear transformation with principal component matrix \( P \). Given that \( Y \) is the objective data transformed from \( X \), the transformation is represented as follows:

\[
\begin{bmatrix}
P_1 \\
\vdots \\
P_m
\end{bmatrix}
\begin{bmatrix}
x_1 \\
\vdots \\
x_n
\end{bmatrix} = \begin{bmatrix}
y_{11} & \cdots & y_{1n} \\
\vdots & \ddots & \vdots \\
y_{m1} & \cdots & y_{mn}
\end{bmatrix}
\]

\[
P X = Y
\]  

(3.11)

Also let us define the following quantities:

- \( p_i \) are the rows of \( P \)
- \( x_i \) are the columns of \( X \)
- \( y_i \) are the columns of \( Y \).

Therefore, the following equations can be derived:

\[
Y = \begin{bmatrix}
P_1 \cdot x_1 & \cdots & P_1 \cdot x_n \\
\vdots & \ddots & \vdots \\
P_m \cdot x_1 & \cdots & P_m \cdot x_n
\end{bmatrix}
\]
We note the form of each column of $Y$.

$$y_i = \begin{bmatrix} p_1 \cdot x_i \\ \vdots \\ p_m \cdot x_i \end{bmatrix}$$

We recognize that each coefficient of $y_i$ is a dot-product of $x_i$ with the corresponding row in $P$. In other words, the $j^{th}$ coefficient of $y_i$ is a projection on to the $j^{th}$ row of $P$. Therefore, the rows of $P$ are a new set of basis vectors for representing columns of $X$.

According to the above discussion, the remaining question is simple: What is a good choice of basis $P$ in order to represent objective matrix $Y$? The answer depends on what features we would like $Y$ to exhibit.

Figure 3-2 shows a data cloud with signal and noise variances. The noise in the objective matrix $Y$ must be low; otherwise tracking results will not be acceptable. Accordingly, we have a ratio to measure the quality of signal, which is called Signal-to-Noise Ratio (SNR).

![Figure 3-2: Simulated data of $(x, y)$. The signal and noise variances $\sigma^2_{signal}$ and $\sigma^2_{noise}$ are represented by the two lines subtending the cloud of data.](image)
\[ SNR = \frac{\sigma_{signal}^2}{\sigma_{noise}^2} \]  

(3.12)

For different values of SNR, there could be different correlation between \( x \) and \( y \). A high redundancy cloud of data, \( SNR \gg 1 \) indicates a high precision measurement, in which \( x \) and \( y \) are highly correlated. A circle \((SNR=1)\) or even worse cloud of data will not provide needed information due to high noise. In such cases, the correlations of \( x \) and \( y \) will be relatively low. The covariance measures the degree of the linear relationship between two variables. A large positive value indicates positively correlated data. For two row vectors \( a = [a_1 \ a_2 \ \cdots \ a_n] \) and \( b = [b_1 \ b_2 \ \cdots \ b_n] \), the covariance is

\[ \sigma_{ab}^2 = \frac{1}{n} \sum_i a_i b_i \]

\[ \sigma_{ab}^2 = \frac{1}{n} ab^T \]  

(3.13)

The input data we have is the \( m \times n \) matrix where \( m \) is the dimension of input data and \( n \) is the number of frames. The covariance matrix \( C_X \) can be defined as:

\[ C_X = \frac{1}{n} XX^T \]  

(3.14)

The matrix \( C_X \) the \( ij^{th} \) element is considered as the dot product between the vector of \( i^{th} \) measurement type and \( j^{th} \) measurement type. Properties of \( C_X \) can be summarized as:

- \( C_X \) is a square symmetric \( m \times m \) matrix.
- The diagonal terms of \( C_X \) are the variance of particular measurement types.
- The off-diagonal terms of \( C_X \) are the covariance between measurement types.

\( C_X \) has the covariance between all possible pairs which reflect the noise and redundancy in our measurements. Large diagonal values correspond to signal of interest.
and off diagonal terms are mostly noises. The objective output covariance matrix $Y$ must be optimized to maximize the signal while reducing the noise. Thus the expected covariance matrix $C_Y$ should be:

- All off-diagonal terms in $C_Y$ should be zero. In other words, $C_Y$ is diagonal matrix.
- Each successive dimension in $Y$ should be rank-ordered according to variance.

This problem can be solved by the eigenvector decomposition. According to the variables and matrices we mentioned above, the goal is summarized as:

Find some orthonormal matrix $P$ in $Y = PX$ such that $C_Y \equiv \frac{1}{n}YY^T$ is a diagonal matrix. The rows of $P$ are the principal components of $X$.

$$
C_Y \equiv \frac{1}{n}YY^T = \frac{1}{n}(PX)(PX)^T = \frac{1}{n}PX^TP^T = P\left(\frac{1}{n}XX^T\right)P^T = PC_XP^T \quad (3.15)
$$

For symmetric matrix $A$ which could be decomposed as $A = EDE^T$, where $D$ is a diagonal matrix and $E$ is the matrix of eigenvectors of $A$. Now if we select the matrix $P$ to be a matrix where each row $p_l$ is an eigenvector of $C_X$, which makes $P = E^T$. Since $P$ is the orthogonal matrix, it has property of $P^{-1} = P^T$. So $C_Y$ can be expressed as:

$$
C_Y = PC_XP^T = P(E^TDE)P^T = P(P^TDP)P^T = (PP^T)D(PP^T)
$$

$$
= (PP^{-1})D(PP^{-1})
$$

$$
C_Y = D \quad (3.16)
$$

It is evident that the choice of $P$ diagonalizes $C_Y$.  

23
3.2.2 Noise Filtering

One of the most important stages in image processing applications is noise filtering [20]. The importance of image sequence processing is constantly growing with the ever increasing use of digital television and video systems in consumer, commercial, medical, and communicational applications. The noise filtering plays an important role in this project, not only for improving the quality of the image sequence but also for being used as a preprocessing stage in further algorithms.

The noise filtering methods will be introduced in this part including: edge detection filtering, fuzzy filtering and morphological filtering [22].

3.2.2.1 Edge Detection Filtering

Edge detection solves some of the problems inherent in linear filtering techniques and employs a non-linear mapping to reduce the noise [24]. Edge detection anticipates that the high frequency component of signal power is slightly larger than the noise power. For example, low intensity pixels in a high-passed filtered image can be noise, whereas high intensity pixels are most likely to be signal. It then makes a significant effect on attenuating low intensity pixels and leaving high intensity pixels unchanged. This is the essential idea of edge detection filter.

Since uncorrelated noise power spreads evenly in all directions and edges rather than to align them directionally, edges need not be attenuated as much in order to achieve the same overall noise reduction. Edge detection can definitely handle such situation in a fairly acceptable way. In addition, as edge detecting functions are chosen intuitively, it may have problems when dealing with more complex images.
3.2.2.2 Fuzzy Filtering

The general idea of the fuzzy filter is to average a pixel by using other pixel values from its neighborhood, but simultaneously to take care of important image structures such as edges [25]. The main concern of the proposed filter is to distinguish between local variations due to noise and image structure.

This can be accomplished by first checking for the edges. The fuzzy rules need to be applied to obtain a robust estimation.

The approach will derive a value from each pixel that expresses the degree in the derivation in a certain direction. Given that one particular pixel in the image is \((x, y)\); the eight-direction neighborhood of \((x, y)\) is expressed in direction initials as displayed in Figure 3-3 (a). The neighborhood array is \(\begin{bmatrix} NW & N & NE \\ W & (x,y) & E \\ SW & S & SE \end{bmatrix}\).

\[(3.17)\]

![Figure 3-3](image)

Figure 3-3: (a) Neighborhood of a central pixel \((x, y)\). (b) Pixel values indicated in gray and used to compute the “fuzzy derivative” of the central pixel for NW direction [25]
A simple derivative at the central pixel position \((x, y)\) in the direction \(D\) is defined as the difference between the pixel at \((x, y)\) and its neighborhood \(D\). This derivative value is donated by \(\nabla_D(x, y)\). For example:

\[
\nabla_N(x, y) = I(x, y - 1) - I(x, y)
\]

\[
\nabla_{NW}(x, y) = I(x - 1, y - 1) - I(x, y)
\]

Next, the principle of the fuzzy derivative is based on the following observation. Consider an edge passing through the neighborhood of a pixel \((x, y)\) in SW-NE direction. The derivative value \(\nabla_{NW}(x, y)\) can be large and the derivative value of neighboring pixels perpendicular to the edge’s direction can be expected as a large number as well. For example, in NW-direction we can calculate the values \(\nabla_{NW}(x, y), \nabla_{NW}(x - 1, y + 1)\) and \(\nabla_{NW}(x + 1, y - 1)\) as shown in Figure 3-3 (b). This idea is to cancel out the effect of one derivative value which turns out to be high due to the noise. Therefore, if two out of three derivative values are small, it is safe to assume that no edge is present in the considered direction. This observation is the core idea of building a fuzzy rule to calculate the fuzzy derivative values.

To compute the value that expresses the degree to which the fuzzy derivative in a certain direction is small, we will make use of the fuzzy set \(small\). The membership function \(m_K(\mu)\) for the property \(small\) is given in Equation (3.18):

\[
\begin{align*}
    m_K(\mu) &= \begin{cases} 
        1 - \frac{\mu}{K}, & 0 \leq \mu \leq K \\
        0, & \mu > K 
    \end{cases}
\end{align*} 
\]  
(3.18)

where \(K\) is an adaptive parameter and \(\mu\) is homogeneity which is used to estimate the noise density.
If there is a block in a certain image \( B \), we compute the rough measure for the homogeneity of this block by considering the maximum and minimum pixel value.

\[
\mu = 1 - \frac{\max_{(x,y) \in B} I(x,y) - \min_{(x,y) \in B} I(x,y)}{L}
\]  (3.19)

This measure is commonly used in the context of fuzzy image processing. The membership function can be applied to compute the fuzzy derivative \( \nabla_{NW}^F(x, y) \) for pixel \((x, y)\) in NW-direction using following rules:

if \( (\nabla_{NW}(x, y) \text{ is small and } \nabla_{NW}(x - 1, y + 1) \text{ is small}) \) or \( (\nabla_{NW}(x, y) \text{ is small and } \nabla_{NW}(x + 1, y - 1) \text{ is small}) \) or \( (\nabla_{NW}(x + 1, y - 1) \text{ is small and } \nabla_{NW}(x - 1, y + 1) \text{ is small}) \), pixel \((x, y)\) is considered as background.

Eight such rules are applied, each computing the degree of membership of the fuzzy derivatives \( \nabla_{D}^F(x, y), D \in \text{dir.} \) to the set \textit{small}. These rules are implemented using the minimum to represent the AND-operator, and the maximum for the OR-operator.

The robustness in this filter is the result of combining multiple simple derivatives around the pixel and by making the parameter \( K \) adaptive. The rules we applied to every pixel in eight directions. It can specify if the pixel belongs to an edge or a single peak noise.

The advantage of fuzzy filter stands out when dealing with sharp white noise. It takes the average value to analyze if the pixel belongs to a noise. The fuzzy rule approach can eliminate chances of an accidental removal of edges.

One of the most important improvements of fuzzy filter is to extend the definition of neighborhood from 2-D to 3-D. It is very meaningful because not only the neighborhood pixels in current frame but also in previous and next frames can be incorporated. It is also helpful when dealing with videos with minor details. Filtering noise in only 2-D may
remove the details of little parts which may be considered as noise and lead to an unexpected removal of useful information.

3.2.2.3 Morphological Filtering

Image morphological filter is widely implemented in grey-scale and binary images [22]. In object motion tracking techniques, the binary image sequence is an essential procedure for segmenting objects from their background. Image morphological filter is very suitable for this case. By applying morphological filters, most salt and pepper noise can be removed and the object can be reshaped. This procedure can be helpful for following tracking steps, especially region labeling. Image morphological filters include dilation, erosion, opening and closing.

Given a sample binary image signal \( f[x] \) with value 1 for the image object and 0 for the background, typical image transformations involving a moving window set \( W = \{y_1, y_2, ..., y_n\} \) of \( n \) sample indexes would be

\[
\psi_b(f)[x] = b(f[x - y_1], ..., f[x - y_n])
\]

(3.20)

where \( b(v_1, ..., v_n) \) is a Boolean function of \( n \)-variables. The mapping \( f \rightarrow \psi_b(f) \) is called a Boolean filter. By varying the Boolean function \( b \), a large variety of Boolean filters can be obtained. For example, choosing a Boolean AND for \( b \) would shrink the input image object, whereas a Boolean OR would expand it. Numerous other Boolean filters are possible, since there are \( 2^{2n} \) possible Boolean functions of \( n \)-variables. The main applications of such Boolean image operations have been in biomedical image processing, character recognition, object detection, and general 2-D shape analysis.

Among the important concepts offered by mathematics, morphology is used to represent binary images and set operations to binary image transformation. For a given
binary image $f$, a many-to-one binary or Boolean function $h$, and a window $B$, the Boolean-filtered image $g = h(f)$ is given by

$$g(n) = h[Bf(n)]$$

at every $n$ over the image domain. Thus, at each $n$, the filter collects local pixels according to the geometrical role into a windowed set, performs a Boolean operation on them, and returns the single Boolean result $g(n)$.

The most commonly used Boolean operations are AND, OR, and MAJ. They are used to create the following simple, yet powerful morphological filters. These filters act on the objects in the image by shaping them as expanding, shrinking, smoothing, and eliminating too-small features.

The dilation filter is defined as $g(n) = OR[Bf(n)]$. The erosion filter is defined as $g(n) = AND[Bf(n)]$.

The dilation filter expands the size of the foreground, object or high-value regions in gray-scale image $f$. The process of dilation can also smooth the boundaries of objects, removing gaps or bays of too-narrow width as well as object holes of too-small size.

The erosion filter shrinks the size of foreground, object or high-valued region in the gray-scale image $f$. Alternately, it expands the size of the low-value region. The erosion processing smoothes the boundaries of objects as well, but in a different way from dilation: it removes high-valued objects of too-small size. Generally, an isolated object can be eliminated if the dilation window cannot fit into it.

Figure 3-4 shows the example of dilation and erosion filter applied to a gray-scale image. With $B = BALL(5)$, the results of dilation and erosion are shown in Figure 3-4(b) and Figure 3-4(c) respectively.
In binary images, dilation and erosion filters work in the same way as in gray-scale images. Since there are only 1 and 0 (which means white and black) in binary, the filters have smoothing and removing effect even better than gray-scale images. An important and common misconception is that although dilation and erosion filters shrink and expand the sizes of objects in images, they are not inverse operation of one another. Dilating an eroded image very rarely yields the original image. In particular, dilation cannot recreate peninsulas, fingers, or small objects that have been eliminated by erosion. Likewise, erosion cannot un-fill holes filled by dilation or recreate gaps or bays filled by dilation. Even without these effects, erosion generally will not exactly recreate the same shapes that have been modified by dilation, and vice versa.
One thing must be noticed is that both dilation and erosion filters can change the sizes of objects as well as smoothing them. However, in most conditions, it is not desirable. Although erosion and dilation are not the inverse operations of one another, they are approximate inverses in the sense that if they are performed in sequence on the same image with the same window $B$, then remaining object and holes can be returned to their approximate sizes. We thus define the size-preserving smoothing morphological operators as open filter and close filter, as follows:

$$
\begin{align*}
open(f, B) &= \text{dilate}(\text{erode}(f, B), B) \\
close(f, B) &= \text{erode}(\text{dilate}(f, B), B)
\end{align*}
$$

As we can find out, the difference between open filter and close filter depends on the order of dilation and erosion operation. The open and the close have the same smoothing properties as the dilation and the erosion respectively, but they do not change the size of sufficiently large objects. Figure 3-5 shows the open and close filters applied to the original with the same window size.

![Figure 3-5: Gray-scale images after applying open filter (a) and close filter (b)](image)

3.4.3 Image Thresholding

Image thresholding is a technique for converting gray-scale images into binary images. It plays an indispensable role in object motion tracking. The objective of binary
conversion is to define the foreground and background. This is always processed according to the results from motion segmentation.

In many applications of image processing, the gray levels of pixels belonging to the object are quite different from the gray levels of the pixels belonging to the background. Thresholding becomes then a simple but effective tool to separate objects from the background. The output of the thresholding operation is a binary image whose gray level of 1 (white) indicates a pixel belonging to a print, legend, drawing, or target and whose gray level of 0 (black) indicates the background.

If we pick a threshold value \( \mu \), and all the pixels with greater value than \( \mu \) would be assigned value 1, otherwise would be assigned value 0. For gray-scale image \( f \), it can be expressed as:

\[
p(f) = \begin{cases} 
1 & \text{if } p(f) > \mu \\
0 & \text{if } p(f) \leq \mu 
\end{cases}
\]  

(3.24)

For each image sequence, the threshold value \( \mu \) is an important issue for the binary conversion. An inappropriate selection of \( \mu \) would lead to the failure of conversion and even all following steps.

Figure 3-6 shows different values of threshold \( \mu \) and resulting images corresponding to threshold value. The effect of threshold can be seen in this figure.

In some application, the thresholds could be manually selected or experimented. This usually happens to fix thresholds. However, for some video sequences which may be recorded in a variety of illumination environment such as outdoor; fix threshold will not be appropriate and will require an automatic or adaptive thresholding method.
Figure 3-6: A gray-scale image [34] (a) convert to binary images with different thresholds [0.3(b), 0.5(c), 0.7(d)]

3.2.3.1 Fix Thresholding Method

Fix thresholding is a fundamental, easily to use for image thresholding. In most gray-scale images, picking an appropriate threshold relies on the image histogram. Image histogram is a type of histogram used to display the distribution in a digital image. It plots the number of pixels of each tonal value. The histogram for picture Figure 3-6(a) is shown in Figure 3-7:
Figure 3-7: Image histogram of picture in Figure 3-6(a)

In this histogram, the x-axis is in the range of [0 to 255] representing 8-bit gray-scale image pixel values and will result in 256 bins. When the number is closer to zero then the color will be closer to black. Otherwise, the value will produce white color. The y-axis number shows the number of pixels in each bin. Therefore, histogram directly displays the distribution of color in each picture. According to the histogram values, it is obvious that there are peaks and valleys which indicate the high-distributed and low-distributed color values respectively. The threshold usually is picked from the low-distributed color.

In the previous subsection, we mention that motion tracking techniques rely on the comparison of current frame against background/object model (previous frames in consecutive frames subtraction). By subtracting the model and current frame, we receive a high-contrast image. This makes the histogram more polarized into two parts.

In Figure 3-8, the subtraction between background (a) and object frame (b) outputs the result (c). Usually, our objective is to segment the object from Figure 3-8 (c) by binary thresholding. The histogram of Figure 3-8 (c) is shown in Figure 3-9.
Figure 3-8: Example pictures of background object frame (b) and output (c)

Figure 3-9: Histogram (a) of Figure 3-8 (c) and binary output (b)
In this histogram, it is clear that most of pixels are distributed at the black side of x-axis, which means those pixels belong to background. Then we can pick the threshold as shown by the red line. The binary output is shown in Figure 3-9(b).

3.2.3.2 Auto Thresholding method

The fix threshold selection method is appropriate for still images. For the applications containing video sequence, the fix threshold will be less efficient and inaccurate. In order to deal with video sequence involving a huge number of frames, auto thresholding technique proves to be very useful. A simple and accurate auto thresholding approach called Otsu’s method is introduced in the following section [19].

The ideal case of auto thresholding would be that there is a deep and sharp valley between two peaks representing objects and background respectively in the histogram. An appropriate threshold can be chosen as the bottom of this valley. However, in more precise words, especially in such cases as when the valley is flat and broad, embedded with noise, or when the two peaks are extremely unequal in height, often producing no traceable valley.

Let the pixels of a given picture be represented in $L$ gray levels $[1, 2, \ldots, L]$. The number of pixels at level $i$ denoted by $n_i$ and the total number of pixels by $N = n_1 + n_2 + \cdots + n_L$. The gray-scale histogram is normalized and regarded as probability distribution:

$$p_i = \frac{n_i}{N}, \quad p_i \geq 0, \quad \sum_{i=1}^{L} p_i = 1 \quad (3.25)$$

We assume there are two classes $C_0$ and $C_1$ which represent background and foreground by thresholding at level $k$. Therefore, $C_0$ denotes pixels in level $[1, \ldots, k]$ and
$C_1$ denotes pixels in level $[k + 1, ..., L]$. Then the probabilities of class occurrence $\omega$ and the class mean level $\mu$ respectively are given as:

$$
\omega_0 = \Pr(C_0) = \sum_{i=1}^{k} p_i = \omega(k)
$$

$$
\omega_1 = \Pr(C_1) = \sum_{i=k+1}^{L} p_i = 1 - \omega(k)
$$

(3.26)

and

$$
\mu_0 = \sum_{i=1}^{k} iPr(i|C_0) = \sum_{i=1}^{k} i p_i / \omega_0
$$

$$
\mu_1 = \sum_{i=k+1}^{L} iPr(i|C_1) = \sum_{i=k+1}^{L} i p_i / \omega_1
$$

(3.27)

In order to evaluate thresholding at level $k$ accurately, the following discriminant criterion measures can be used:

$$
\eta = \sigma_B^2 / \sigma_T^2
$$

(3.28)

where

$$
\sigma_B^2 = \omega_0 (\mu_0 - \mu_T)^2 + \omega_1 (\mu_1 - \mu_T)^2 = \omega_0 \omega_1 (\mu_1 - \mu_0)^2
$$

$$
\sigma_T^2 = \sum_{i=1}^{L} (i - \mu_T)^2 p_i
$$

$\sigma_B^2$ and $\sigma_T^2$ are the between-class variance and total variance of levels respectively. Now the next step is to optimizing the problem to search for a good threshold $k$ that would maximize the function $\eta$.

It is simply noticed that total variance $\sigma_T^2$ is independent of $k$, which means the maximum value of $\sigma_B^2$ is the maximum value of $\eta$ as well. The range of $k$ would be restricted to $1 \leq k \leq L$. Thus, the optimal threshold $k^*$ is
\[ \sigma^2_B(k^*) = \max_{1 \leq k \leq L} \sigma^2_B(k) \quad (3.29) \]

As a comparison, Figure 3-10 shows the result from fix thresholding result (a) and auto thresholding from Otsu’s method (b). From the comparison we can see, the result of Otsu’s method has less details as well as the noise. Auto thresholding method will be more appropriate in this work.

![Figure 3-10: Thresholding results from Fix thresholding method (a) and Otsu’s auto thresholding method (b)](image)

3.3 Summary

This chapter gives a brief introduction to the features of motion tracking system based on IR camera and some important techniques in motion tracking systems.

Section 3.1 describes the differences between IR camera and normal camera, and the important features of IR data. Section 3.2 focuses on the processing of IR data and introduces the main advantages and challenges in IR processing. Section 3.4 presents three important techniques in motion tracking: Motion Tracking Algorithm, Noise Filtering and Image Thresholding.

For motion tracking algorithm, we have briefly introduced several commonly used 2-D object motion segmentation and tracking algorithm and compared them in robustness and efficiency. Mixture Gaussian has the best robustness and least efficiency. Running
average and adaptive background subtraction have reliable robustness in most conditions. Consecutive frames subtractions have the fastest processing speed but have some obvious problems as well. Besides mixture Gaussian, other algorithms have the abilities to handle real-time tracking. At the end of this chapter, PCA is introduced in details. This algorithm is widely used in a variety of fields. We have discussed the application in motion tracking regarding to our project.

For Noise filtering, three commonly used filters are introduced. Edge detection filter has faster processing speed than fuzzy filter and morphological filter, but it is not able to handle complex images. Fuzzy filter is an advance filter modified from neighborhood averaging filter, and has better ability to deal with more detailed pictures, especially with more edges. Morphological filters are well-known and popular in noise filtering and classified into four different types. The specialty of morphological filters is that they are not only capable of filtering the noise, but also could smooth the shape of objects and fill gaps and holes, which is considerably useful to some tracking system.

For image thresholding, we have introduced two basic classes of thresholding techniques: fix thresholding and auto thresholding. Fix thresholding is fast, easy to implement and can meet the needs of some video with small changes in illumination and stable noise. But it has less robustness than auto thresholding. Auto thresholding is a general name which includes many methods under this topic. The method that we have emphasized in this chapter is called Otsu’s method. It is a new and improved method comparing to the traditional auto thresholding methods. Otsu’s method borrows an idea from statistic and finds a good solution to the histograms with flat and broad valley which
is difficult for other methods to find a perfect threshold. Moreover, it has robustness against noise.
Chapter 4

Target Motion Tracking System Design

Based on the knowledge from previous chapters, a target motion tracking system has been designed for our project. The goal of this project is to analyze birds’/bats’ activity during nocturnal migration. The information will be useful for wildlife biologist to determine behavior of birds/bats in the vicinity of wind turbines. We are required to focus on the information of the target trajectories. Therefore, some information such as contour or color is not important in this work. The flow chart of the proposed procedure is shown in Figure 4-1.

![Figure 4-1: Proposed Target Tracking Algorithm](image-url)
The tracking techniques that been used in this system is as follows:

- **Algorithm for motion tracking**: Consecutive frames subtraction followed by modified background subtraction
- **Noise filtering**: Fuzzy filtering followed by image morphological filtering;
- **Image thresholding**: Otsu’s auto thresholding method.

### 4.1 Motion Detection

This system is designed for thermal image sequence processing. For thermal images, there is less complex color information than normal videos. Only the objects with sufficient emitted heat can be recorded into the thermal image sequence. Thus, there are no special needs for algorithms like Mixture Gaussian. Since our goal is not focused on one single bird but almost every target in the image sequence, the background subtraction meets our needs. However, the video data contains some steady noise which appears and stays on the specific position. It can be removed by background subtraction. Since consecutive frame subtraction is an effective method to reduce steady noise, we design the system with Background subtraction and Consecutive Frames Subtraction (BGCFS). The procedure of BGCFS is shown in Figure 4-2.
The BGCFS algorithm uses following processing steps:

- **Initialize background model**: Take first ten frames in the image sequence and calculate the average value for each pixel. The output of this step is an image which is used as the original background model.

- **Consecutive frames subtraction**: Take two adjacent frames for subtraction and detect the changed regions. To decide which region is changed, we need a threshold $T_1$ for binary conversion. In this project, threshold $T_1$ is selected by Otsu’s method. Given that the location of pixels are represented by coordinates $(i, j), f_k$ is the $k^{th}$ frame, The discriminant criterion is
\[
|f_k(i,j) - f_{k+1}(i,j)| < T_1 \quad B_1(i,j) = f_{k+1}(i,j) \tag{4.1}
\]
\[
|f_k(i,j) - f_{k+1}(i,j)| \geq T_1 \quad M_1(i,j) = f_{k+1}(i,j) \tag{4.2}
\]

Where \(B_1(i,j)\) is the first background region, \(M_1(i,j)\) is the changed region. The first background region pixels are updated into background model by updating rate \(p_1\). Since the probability of \(B_1(i,j)\) of being background is large, update rate \(p_1\) has to be in lower range. In this case, \(p_1 = 0.25\) is preferred; the first changed region \(M_1(i,j)\) is prepared for processing the next step.

- **Moving object detection:** In this step, firstly apply the fuzzy filter to region \(M_1(i,j)\) to reduce the salt or pepper noise. Then compare \(M_1(i,j)\) with the background model and evaluate the results with likelihood threshold \(T_2\). Assume in this step, \((i,j)\) belongs to \(M_1(i,j)\) and \(B_M(i,j)\) is the background model. The discriminant criterion is:

\[
\frac{f_{k+1}(i,j)}{B_M(i,j)} < T_2 \quad B_2(i,j) = f_{k+1}(i,j) \tag{4.3}
\]
\[
\frac{f_{k+1}(i,j)}{B_M(i,j)} \geq T_2 \quad M(i,j) = f_{k+1}(i,j) \tag{4.4}
\]

Where \(B_2(i,j)\) is the second background region, and \(M(i,j)\) is the object we need. \(B_2(i,j)\) is the region covered by the object in previous frame, so that it can be detected as a changed region in consecutive frames subtraction.

- **Background re-update:** The last step in this procedure is updating the background model for the second time. Since the second background region \(B_2(i,j)\) is the changed region in previous frame and more likely to become a moving object in the next frame, the update rate for \(B_2(i,j)\) should be larger than \(p_1\). In this project, the update rate is \(p_2 = 0.75\).
In this BGCFS algorithm, the key components are the auto-selected binary threshold $T_1$ and the likelihood threshold $T_2$. These two thresholds determine that which region is the moving region in the final result. $T_1$ and $T_2$ are chosen by experience and project requirement. Generally, if illumination is relatively steady, $T_1$ preferred to be close to 0. In this project, the value of $T_1$ is 0.027. $T_2$ is calculated by auto thresholding algorithm.

After the object region is selected, the fuzzy filter is implemented in the algorithm to filter the noise. According to the property of fuzzy filter, it could filter the noise without changing the shape of desired objects. The output results from this step lay the foundation for the following procedure.

4.2 Frame Selection

For our project, there are possibilities that multiple targets appear in the same frame. The typical motion tracking algorithm which uses blobs as region labeling method can only analyze the regions in 2-D. However, this would cause errors in our project. We suppose that a pixel $(i,j)$ which belongs to object 1 in frame $f_k$ is represented by $f_k(i,j) \in Obj_1$; in frame $f_{k+2}$, the pixel belonging to object 2 is represented by $f_{k+2}(i,j) \in Obj_2$. If we label every frame respectively, the pixel in frame $f_{k+2}$ may still be considered as object 1; this would make the algorithm label object 2 and object 1 by mistake. To avoid this happening, we need to bring labeling into 3-D. Based on our tests, this labeling would need more memory or waste of memory and slow down the processing speed. We notice an important feature of our videos which is that there are only a small number of frames with useful information (birds/bats). Frames with useful information may be approximately 7% of the whole video. To deal with this circumstance, we have incorporated a frame selection scheme into our system.
The frame selection allows the system to only process the frames with useful information and ignore the rest. This process plays an important role in optimizing the processing speed. The steps of frame selection process are as follows:

- **Filtering noise and reshape the objects**: Before selecting useful frames, we need to make sure that no noise is taken as minor objects. Furthermore, reshaping the objects could make them seem closer to the origins, since object might be partially removed with noise by filter. This is exactly how morphological filters work. We choose close filter because it can fill holes and gaps without extending the boundary of objects. Boolean window size is

\[
se = \text{square}(5)
\]  

Bigger the window we choose, the more shaping effects work on object. This window size is a tradeoff between noise filtering and maintains the shape of the objects.

- **Minor object removal**: Most of the noise has been removed by the close filter. But if there is some noise larger than the Boolean window, it would remain in the frame as noise. Moreover, some small objects are actually insects and unnecessary in our project. Since the size of the noise or insects is much smaller than birds/bats, we remove the objects that are less than 50 pixels.

- **Frame selecting and packing**: After filtering noise and removing minor objects, frames with target object regions would be taken as useful information. Pack all binary frames with objects together and make it a new video sequence named as *BWVideo*. This would enormously reduce the size of the original video. In addition, there is an array named “Call(i)” which maintains the sequence
number of selected frames in the original video.

4.3 3-D Region Labeling

After the BGCFS procedure and frame selection, the new video sequence “BWVideo” is built only by binary images with 0 for the background and 1 for the objects. 2-D blob region labeling does not provide a good solution to multiple targets in same frames. Applying 3-D region labeling method would label the objects respectively without confusion.

Region labeling is based on neighborhood detection. The number of neighbors being considered in each detection process is called connectivity. For 2-D region labeling, the connectivity is 4 or 8. For 3-D region labeling, connectivity is 6, 18 or 26. So 3-D region labeling can detect neighbors in previous and next frames, which is especially effective in video sequence. Figure 4-3 shows the neighbors of a pixel under connectivity 6 and 18, and Figure 4-4 shows the neighbors of a pixel under connectivity 26.

Figure 4-3: The neighbor dots (circles) of central pixel (star) with connectivity 6 (a) and 18 (b)
In this project, the labeling connectivity is 26. The pixels in neighborhood with the same value (0 or 1) as the objective pixel are labeled as the same object number. After the labeling process, all the “1”s in video are labeled as different numbers for different objects.

Even though we have reduced the original video size by selecting frames, there are still many frames in BWVideo sequence and labeling them all together would still require large memory storage; thus we label the frames group by group. The output from frame selection has an array which stores the frame number of selected frames in the original video. It can help us with labeling. For separating groups, the system reads the frame number from array Call(i). If Call(i + 1) > Call(i) + 1, it means that there are frames without any target activity between frame Call(i) and frame Call(i + 1). For
example, $Call(i) = 61$ means the useful frame is the 61\textsuperscript{st} frame in the original video. If $Call(i + 1) = 87$, it means that the next useful frame in the original video sequence is the 87\textsuperscript{th} frame and there are 26 frames between them without any detected objects. Therefore, we group the frames up to the $i^{th}$ frames in $BWVideo$ (the 61\textsuperscript{st} in original video) and prepare for region labeling.

In some cases, there may be a situation that small objects are covered by clouds and removed by the filters. Thus the objects seem to disappear from the video in several frames and appear again. In order to overcome this problem, a window is used for grouping. A window size of three has been selected and thus the condition for grouping is:

$$\text{if } Call(i + 1) - Call(i) \leq 3 \quad \text{then } i \leftarrow i + 1;$$

$$\text{if } Call(i + 1) - Call(i) > 3 \quad \text{stop grouping; } brk \leftarrow i;$$

where $brk$ is the last frame number of the current group and next grouping starts with $Call(brk + 1)$.

The outputs of this step are $XAxis$ for objects x-coordinates and $YAxis$ for object y-coordinates. For example, the position of $i^{th}$ object in $j^{th}$ frame is $[XAxis(i, j), YAxis(i, j)]$. Additionally, an intensity matrix $Int(i, j)$ is created. It stores intensities in gray-scale values (0-255) and size array $Objectsize(i)$ for object size.

4.4 Breakpoint Recovery

In previous procedure, the region labeling process returns the coordinates of objects and the intensity values. The expected results of the tracking system consist of two parts: trajectory images and target information spreadsheet. In ideal case, the outputs $XAxis$, $YAxis$ and $Int$ should be good enough to process the results. However, as discussed in
Chapter 1, some objects are still removed from several frames because of unavoidable reason (such as covered by clouds). Although we allow a picking window for grouping, it cannot deal with the object displacement caused by accidental removal. Moreover, the limit of labeling connectivity is 26. Thus the object trajectories which should belong to one object may not be labeled together and break into two or more parts. Figure 4-5 gives example when trajectory breaks.

![Figure 4-5: One object track (a) is broken into two tracks (b) and (c)](image)

Based on our calculation, the probability of breaking occurrence is 32%, which means for ten objects the system returns thirteen or more results. This leads to a big inaccuracy. In order to avoid this effect, we bring in the breakpoint recovery procedure. Breakpoint recover procedure includes three parts: Breakpoint detection, Coordinates recovery and Blank group notification.

- **Breakpoint detection:** The objective of breakpoint detection is analyzing where the breaks happened. This process would go through the coordinates matrix \( XAxis \) and \( YAxis \) and measure each ending point of object and the beginning point of the next object. According to the width of the field of view of our camera, birds look much different in size. Therefore, we set thresholds in size difference and displacement distance to determine whether the two trajectories belong to the
same objects. Assuming that the size difference threshold is $T_S$ and the displacement distance threshold is $T_D$, if there is $|S_i - S_{i+1}| < T_S$ & $|D_{diff}| < T_D$, we mark the object $i$ and $i + 1$ as the same.

- **Coordinates recovery:** After identifying the breakpoints, coordinates recovery connect the breakpoints by extending object $i$ coordinates with the coordinates of object $i + 1$. A similar process can apply to intensity matrix $Int(i,j)$. For object size, array elements are redefined by $Objectsize(i) = Objectsize(i) + Objectsize(i + 1)$. After this recovery step, information of object $i + 1$ can be removed from the matrixes and array.

- **Blank group notification:** In some condition, after completing two previous steps, there still remains object information that is out of our interest. This may be caused by minor objects such as insects. However, as the objective is tracking birds/bats, this information is out of our concern. Usually insects are closer to the camera and have smaller size and faster speed than birds (because a bird flying higher than insects, which makes it looks slower in camera), so they only appear in several frames in the camera view. If an object has only two-frame appearance in video, we can remove that and send a blank group notification. After the system gets the information, the steps of target information analysis and results output would be skipped.

### 4.5 Target Information Analysis and Results Output

The results consist of two parts that are spreadsheet with target information of each object and trajectory images for each object. The analysis is based on coordinates matrixes $XAxis$, $YAxis$, intensity matrix $Int(i,j)$ and object size array $Objectsize(i)$. 
Target information includes six factors: object size, object intensity, distance, velocity, straightness index and direction.

- **Object size:** The object size information can be collected from array \( Objectsize(i) \). For each element in \( Objectsize(i) \), it stores the total number of pixels of the connected components. Thus the size of object \( i \) is the average size in each frame. The expression is

\[
Size(i) = Objectsize(i)/n_i
\]  

where \( n_i \) is the number of frames with the appearance of object \( i \).

- **Object intensity:** For acquiring the object intensity values, the system backtracks and locates the objects by the coordinates obtained from the region labeling, and then reads the intensity values of objects from each frame. The values are stored in the matrix \( int(i,j) \) where \( (i,j) \) indicates the number of object and corresponding frame number. The average intensity value \( Aveint(i) \) is calculated by

\[
Aveint(i) = [int(i,1) + int(i,2) + \cdots + int(i,j)]/j
\]

- **Distance and velocity:** Distance is the total distance of an object traveled in camera view regardless of how curved the trajectory is. Most bats have curvy or zigzag trajectories but birds always have a straight line. In order to get the accurate distance, the system calculates the distance between every two continuous frames and adds them up. The velocity would be the result of average distance for one second. The distance \( SumDist(i) \) and velocity \( V(i) \) are expressed as:
\[
SumDist(i) = \sum_{k=1}^{j-1} \sqrt{XAxis_{diff}(i,k)^2 + YAxis_{diff}(i,k)^2} \quad (4.8)
\]
\[
V(i) = 30 \times SumDist(i)/j \quad (4.9)
\]

where
\[
XAxis_{diff}(i,k) = XAxis(i,k+1) - XAxis(i,k) \quad (4.10)
\]
\[
YAxis_{diff}(i,k) = YAxis(i,k+1) - YAxis(i,k) \quad (4.11)
\]

- **Straightness index**: The results of this tracking system are used for analyzing birds’/bats’ behaviors. In order to analyze targets respectively, all the objects being tracked need to be classified. Straightness index is an essential factor for evaluation and objects classification. The straightness index \( StrIndex(i) \) is calculated by straight-line distance against real distance \( SumDist(i) \).

\[
StrIndex(i) = \frac{\sqrt{XAxis_{diff}(i,j)^2 + YAxis_{diff}(i,j)^2}}{SumDist(i)} \quad (4.12)
\]

where
\[
XAxis_{diff}(i,j) = XAxis(i,j) - XAxis(i,1) \quad (4.13)
\]
\[
YAxis_{diff}(i,j) = YAxis(i,j) - YAxis(i,1) \quad (4.14)
\]

- **Direction**: Direction is the most important information for this project. Direction information helps us study the birds’ migration in the specific season. In this tracking system, directions are summarized by vectors \( XAxis_{diff}(i,j) \) and \( \cot(\theta) \).

\[
\cot(\theta) = \frac{XAxis_{diff}(i,j)}{YAxis_{diff}(i,j)} \quad (4.15)
\]

Figure 4-6 shows the relationship between \( XAxis_{diff}(i,j) \) and \( \theta \).
Figure 4-6: Relationship between $\theta$ and $X_{axis_{diff}}$

Factor $\theta$ defines which direction the object is heading. The system compares $\theta$ with coordinate system as shown in Figure 4-7 and finds out which region $\theta$ belongs to.

In this coordinate system, the direction wheel is rotated clockwise with respect to the rotation of camera. Since the camera points up vertically, the left-hand side of the camera view is “East” instead of “West”.

Figure 4-7: Direction coordinate system
4.6 System Comparison

In order to achieve the objective of tracking birds/bats from videos, two other systems were designed to illustrate the differences with the proposed tracking system. These systems will be applied to the same data as the proposed system, and their performances will be evaluated. Results will be shown in the next chapter.

4.6.1 Adaptive Background Subtraction System Design

The features of adaptive background subtraction system are as follows:

1) 3-D Adaptive Gaussian background with edge detection filter
2) Fix threshold selection
3) Morphological noise filtering (Dilate filter)
4) 3-D region labeling without frame selection

This system is designed according to the most widely used tracking system. It is very robust and accommodates changes in illumination, difficult weather condition and constant noise.

4.6.2 PCA Based Tracking System

PCA has many advantages compared to the traditional algorithms. By extracting the principle components, PCA can filter the noise while analyzing and labeling objects. Like most tracking methods, PCA analysis is based on the motion segmentation outputs. In order to compare the difference, BGCFS is also applied in our PCA system for motion segmentation.

As introduced in previous chapter, PCA is an algorithm incorporated from statistics. It can be applied in many different ways for different objectives. The main purpose of applying PCA is to generate the target information from the noise. In this project, it is
used to minimize and locate the objects from each frame in a 2-D environment. Alternatively, PCA can be applied when objects are labeled for tracking and analysis. It can extract locations and plot trajectories when applied in 3-D. However, when applied in 3-D it performs as an ordinary algorithm without any advantage in time-efficiency or robustness. This PCA based system consists of following computational steps:

1) BGCFS with fuzzy filter
2) Frame selection
3) PCA object labeling and tracking
4) Breakpoint recovery

4.6.3 Proposed Target Tracking Algorithm

The difference between this proposed algorithm and others is that this algorithm is designed to suit the special needs of the birds/bats monitoring project. It fully takes the advantage of thermal image data and optimizes without reducing the accuracy and the quality of results.

The frame selection and breakpoint recovery modules are new and unique, they are originally designed for the birds/bats monitoring project. These two algorithms are appropriate for our system and provide better computational efficiency.

This proposed algorithm has following computational steps:

1) BGCFS with fuzzy filter
2) Frame selection
3) 3-D region labeling
4) Breakpoint recovery
4.7 Summary

This chapter provides details of various processing steps in the target motion tracking system. It presents a clear picture of how the system works and helps us understand the system better by deeply discussing the function and importance of each step.

Section 4.1 introduced the core module of motion tracking: Motion detection. Based on the techniques that have been talked in Chapter 3, we combined the background subtraction and consecutive frame subtraction to produce a new algorithm which can better fulfill our needs. This algorithm takes both advantages from background subtraction and consecutive frame subtraction and provides better robustness and output results.

Section 4.2 presents a unique algorithm which is called “Frame selection”. Since there are lot of data need to be processed, frame selection will extremely accelerate the processing.

Section 4.3 and Section 4.4 talked about the labeling techniques we used in this system. 3-D labeling would provide us a more accurate output than 2-D labeling, and breakpoint recovery could rebuild the connections that has been cut off by noise and other reasons.

Section 4.5 give us the idea that how the target information are analyzed. According to the outputs from previous modules, target information can be computed and outputted as results in images or spreadsheet.

In Section 4.6, two tracking systems are designed with different techniques. These systems will be used to process the same data and compare the advantages and
disadvantages with proposed system. In this way, we can find which system could better fulfill the needs of our project. Simulation results and comparison is provided in the following chapter.
Chapter 5

IR Video Processing

5.1 Study Area and Study Time

The IR camera used in this project was deployed at Ottawa National Wildlife Refuge in Ohio during spring bird migration. Wild life biologists in the project team have recommended this site for collection of data for nocturnal bird migration in the Western Lake Erie basin. This area has been sighted for development of potential on-shore and off-shore wind farms. IR camera was deployed along with marine radar and acoustic recorders for both birds and bats. The data was collected from three different sources (IR camera, marine radar and acoustic recorders) continuously from May to July 2011, starting in one hour after the sunset to one hour before the sunrise. The selection of time and dates coincides with the spring nocturnal bird migration in this area. Figure 5-1 shows the location of Ottawa National Wildlife Refuge, in which the red marker “A” refers to the location.

5.2 Data Collection Time and Techniques

In our project, the data has been collected by a FLIR SR-19 thermal camera and was recorded. A system has also been developed for wireless transmission or IR data to the radar signal processing laboratory at the University of Toledo, Toledo, Ohio. The time stamps have been accurately labeled in each video data allowing for synchronization of the separate video feeds. The thermal camera positioning objective has called for the
design of a system to position the IR camera from a remote computer. The camera can be positioned within one degree of tolerance without any overcompensation or oscillations.

![Map of Ottawa National Wildlife Refuge](image)

**Figure 5-1: Location of Ottawa National Wildlife Refuge**

### 5.3 Video Data Processing

The target motion tracking system that has been designed in this project is programmed with Matlab R2010b which has a processing speed of approximately ten frames/second. The videos are recorded and stored as .asf format. In this case, there are 63000 frames in 1 GB data. Since the total size of data for one season is approximately 205 GB, there are 12,915,000 frames that need to be processed. This is a large amount of computational work for a normal computer. In order to accelerate the processing, facilities of the Ohio Supercomputer Center (OSC) has been considered.

According to the requirement of OSC, we clipped videos into 15 minute length duration and converted them into .avi files; the OSC doesn’t support image outputs, so we divided the program into two parts namely Supercomputer code and Local
computer code. Instead of output images, the Supercomputer code processed all the data and saved the results in a .mat file. Local computer code reads .mat file as input and will output results in a format that is desired by wildlife biologists. The User Interface (UI) of OSC supporting software is shown in Figure 5-2.

![Figure 5-2: UI of OSC supporting software](image)

This software is designed for data transfer and storage of results. The codes of our system are under the folder with the username (utl0310). The data will wait in the “Input” folder and the results will be placed into the “Output” folder.

The Supercomputer code could read all the video files from Input folder and process them one-by-one. Once we have all the codes and video files transferred into objective folders, we can start processing. By executing commands in Matlab, a connection to OSC will be built after user verification and authentication. Then a job handle is initialized in Matlab which allows the customer to monitor the progress of
submitted computational job. Once the job is finished, we can read results from Output folder by using the supporting software.

For each video file, the result from OSC has two parts: (1) A .mat file for further processing by Local computer code, and (2) a spreadsheet in which target information is recorded. The Local computer code will read the .mat file, import data into Matlab and output the images. Each image will be labeled with corresponding target information and saved as a .jpg file.

5.4 Validation

The system is designed for both accuracy of results and computational efficiency. Moreover, it is developed with unique functions to better address our special needs in target information analysis. In order to illustrate advantages of our system, we have simulated two previously developed algorithms namely adaptive background system and the PCA based system. The proposed algorithm for birds/bats tracking first has been simulated under similar conditions. Finally, it has also been used for data processing of spring 2011 migration data. Results are compared in terms of computational efficiency and accuracy.

5.5 Results

This section presents several tests based on different samples. The objective of this part is to compare the advantages and disadvantages of different systems.

5.5.1 Results of Sample with One High Velocity Object

The first sample data for this comparison is a clip from the IR video data of our project. It is seven seconds (219 frames) long and contains one object and a clear
background. In the camera view, the object is traveling from the middle of the bottom to the upper left-hand side. A frame shot of this sample is shown in Figure 5-3.

![Figure 5-3: The frame shot of the sample video (original)](image)

There is a clear track in the sample data which can be captured by tracking system. The object velocity is faster than most of other objects. The output track from the proposed target monitoring system is shown in Figure 5-4. It clearly tracked this target and its trajectory is shown.

![Figure 5-4: Trajectory output by target monitoring system for sample (1)](image)

The adaptive background subtraction system was also used and its results are shown in Figure 5-5.
Figure 5-5: Trajectories output by adaptive background subtraction system for sample

As we can see from Figure 5-5, the trajectories which are supposed to belong to one single object break into two parts. This issue happens when there is no breakpoint recovery in the system and the object moves at higher speed. Furthermore, there is some pepper noise in results. It would lead to inaccurate analysis of target information.

For PCA based tracking system, the trajectory output is shown in Figure 5-6.

Figure 5-6: Trajectory output by PCA system for sample (1)

The most apparent difference between the result of PCA and the others’ is that the object in the image is represented in separated square blocks. The reason is that PCA algorithm not only removes the noise but also rotates the object from every single frame. This would change the shape of the object and prevents further analysis. However, the result of PCA shows good robustness. The trajectory is clear and noise was also removed.
This algorithm is a fast and reliable algorithm when there are no other target information in the system is required.

5.5.2 Results of Sample with Multiple Objects

The second data example contains more than one objects traveling through the camera view in the same time. Tracking these objects separately is one of the most important issues in our project. This video is six seconds (173 frames) long and involves two objects as well as a clear background. In the camera view, the object is traveling from the left-hand side to the upper right-hand side. The frame shot for this sample is shown in Figure 5-7.

![Figure 5-7: The frame shot of the sample video (original)](image)

For the target monitoring system, the trajectory is shown as in Figure 5-8.

![Figure 5-8: The trajectories outputs by target monitoring system for sample (2)](image)
The trajectories output from the target monitoring system shows that there is no cross-over with each other. The edges of the objects are not parallel because the birds change their shapes while flying. This may also be caused by wings flapping. As we can see in Figure 5-8, there are two clear tracks in the result, which would help the system for labeling separately.

For the adaptive background subtraction system, the results are shown in Figure 5-9.

![Figure 5-9: The trajectories output by adaptive background subtraction system for sample](image)

For the adaptive background subtraction system, the output image is similar to Figure 5-8. As the result in Figure 5-5, the noise is unfiltered and appears more due to the complex background.

For PCA based tracking system, the trajectory output is shown in Figure 5-10.
The output from PCA system provides a clear background with no noise, which makes the result easier to adopt for further processing. Unfortunately, there are many cross-overs in the trajectories of the two birds. This would definitely make it more challenging for further processing. This is a drawback of the PCA algorithm.

5.5.3 Computation Time Comparison

In order to compare computation time, we have selected a long sequence of data for comparison purposes. This data has 12562 frames (approximate 7 minutes) and the frame rate is thirty frames per second. The number of objects in this video is 34.

This test only focuses on the object motion segmentation, computation time and tracking. It ignores other target information such as direction, velocity, intensity, object size and straightness index. This is because the motion segmentation and tracking procedure take up more than 90% of the total processing time. Furthermore, the time for target information analysis depends on the number of objects being tracked. Table 5-1 shows computation time, number of objects detected and average processing speed for all three systems.
PCA has computational advantage and takes the shortest computation time. This is due to the fact that it is the simplest algorithm. However, it tracks less number of objects than the other two systems. It may be caused by occlusion due to the changes of the object shapes.

Adaptive background system has tracked 25 objects, which is close to the target monitoring system. The disadvantage is obviously the processing time which is 244 minutes. Proposed target monitoring system has tracked the most of the objects with acceptable processing time. Although it is a little bit slower than PCA system, it can be ignored due to its capability of tracking multiple targets. Moreover, PCA changes the object size and shape, which lead to wrong output from target information analysis.

<table>
<thead>
<tr>
<th></th>
<th>Processing time</th>
<th># of objects</th>
<th>Average speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target monitoring</td>
<td>17’42’</td>
<td>32</td>
<td>12 frames/s</td>
</tr>
<tr>
<td>Adaptive background</td>
<td>244”22’</td>
<td>25</td>
<td>0.86 frames/s</td>
</tr>
<tr>
<td>PCA</td>
<td>14”57’</td>
<td>18</td>
<td>14 frames/s</td>
</tr>
</tbody>
</table>

Table 5-1: The comparison of processing time for three systems

5.6 Migration Data Analysis

The proposed target monitoring system has been applied to process the IR video data for birds/bats monitoring. The data were collected in spring, 2011 in Ottawa National Wildlife Refuge. Data was recorded from April 13th to May 11th, the length of entire season data was 113 hours.

The total number of targets that have been tracked is 4808. Figure 5-11 and Table 5-2 shows the distribution of objects by nights.
In order to analyze the behavior of targets, the passage rates could help to get more reliable results than using distribution of objects only. The passage rate is calculated by averaging the number of objects by hours, which means how many objects appeared in
one hour. The distribution of passage rate in this season is shown in Figure 5-12 and Table 5-3.

![Passage rate of each night](image)

**Figure 5-12: Distribution of passage rate by nights**

<table>
<thead>
<tr>
<th>Nights</th>
<th>Passage rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/13-04/14</td>
<td>113</td>
</tr>
<tr>
<td>04/21-04/22</td>
<td>16</td>
</tr>
<tr>
<td>04/22-04/23</td>
<td>3</td>
</tr>
<tr>
<td>04/23-04/24</td>
<td>199</td>
</tr>
<tr>
<td>04/24-04/25</td>
<td>11</td>
</tr>
<tr>
<td>04/26-04/27</td>
<td>165</td>
</tr>
<tr>
<td>04/29-04/30</td>
<td>40</td>
</tr>
<tr>
<td>04/30-05/01</td>
<td>41</td>
</tr>
<tr>
<td>05/01-05/02</td>
<td>108</td>
</tr>
<tr>
<td>05/04-05/05</td>
<td>59</td>
</tr>
<tr>
<td>05/05-05/06</td>
<td>61</td>
</tr>
<tr>
<td>05/08-05/09</td>
<td>38</td>
</tr>
<tr>
<td>05/09-05/10</td>
<td>1</td>
</tr>
<tr>
<td>05/10-05/11</td>
<td>10</td>
</tr>
</tbody>
</table>

**Table 5-3: Distribution of passage rate by nights**

The system provides observed objects passing through but also analyzes and records their related information. One type of target information is the direction; it helps people
to understand the direction of bird migration. After processing one season data, the objects direction summary is shown in Figure 5-13.

![Figure 5-13: Overall directions distribution of objects](image)

<table>
<thead>
<tr>
<th>South</th>
<th>West</th>
<th>North</th>
<th>East</th>
<th>Southwest</th>
<th>Southeast</th>
<th>Northwest</th>
<th>Northeast</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>240</td>
<td>708</td>
<td>592</td>
<td>987</td>
<td>264</td>
<td>411</td>
<td>1100</td>
<td>506</td>
<td>4808</td>
</tr>
</tbody>
</table>

Table 5-4: Overall directions distribution of objects

The results of spring 2011 migration data will shed light and contribute in determining the behaviors of birds and bats. There are more target information (velocity other than directions has been captured and recorded. This data will be very useful for future classification work.

The result outputs from the proposed target monitoring system consist of two parts: A spread sheet which stored all the target information corresponds to the each individual object; and an image which has the object trajectory with related target information labeled.

Table 5-5 shows part of the spreadsheet of one night result.
<table>
<thead>
<tr>
<th>Image Name</th>
<th>size(pix)</th>
<th>Heat</th>
<th>Distance (pix)</th>
<th>Velocity (pix/sec)</th>
<th>Straightness</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>144.875</td>
<td>144.8125</td>
<td>684.3322</td>
<td>1283.1228</td>
<td>0.99815</td>
<td>East</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>423.8571</td>
<td>140.8571</td>
<td>551.2489</td>
<td>1181.2477</td>
<td>0.99836</td>
<td>Northwest</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>120.7778</td>
<td>117.7778</td>
<td>427.6083</td>
<td>1425.3611</td>
<td>0.99972</td>
<td>North</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>3060</td>
<td>158.25</td>
<td>467.2111</td>
<td>3504.0833</td>
<td>0.99961</td>
<td>Northwest</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>133.95</td>
<td>131.65</td>
<td>478.863</td>
<td>718.2945</td>
<td>0.935</td>
<td>North</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>85</td>
<td>130.3333</td>
<td>383.4037</td>
<td>3834.0373</td>
<td>0.99882</td>
<td>East</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>710.3684</td>
<td>159.6316</td>
<td>513.2669</td>
<td>810.4215</td>
<td>0.98403</td>
<td>North</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>250</td>
<td>124.8889</td>
<td>448.2987</td>
<td>1494.3289</td>
<td>0.99992</td>
<td>East</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>2986.2105</td>
<td>173.6316</td>
<td>481.4009</td>
<td>760.1066</td>
<td>0.98207</td>
<td>Southwest</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>75</td>
<td>149</td>
<td>200.5284</td>
<td>546.8956</td>
<td>0.98693</td>
<td>Southwest</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>150.1667</td>
<td>168.6667</td>
<td>124.422</td>
<td>622.1099</td>
<td>0.99992</td>
<td>East</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>460.8537</td>
<td>162.4634</td>
<td>742.6872</td>
<td>543.4297</td>
<td>0.85964</td>
<td>Southwest</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>283.7778</td>
<td>159.5556</td>
<td>214.5247</td>
<td>715.0824</td>
<td>0.99482</td>
<td>East</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>578.0476</td>
<td>166.7143</td>
<td>527.1484</td>
<td>753.0692</td>
<td>0.99881</td>
<td>South</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>406.3333</td>
<td>145</td>
<td>351.6582</td>
<td>1758.2912</td>
<td>0.9963</td>
<td>East</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>253</td>
<td>142.6667</td>
<td>487.4201</td>
<td>974.8401</td>
<td>0.99944</td>
<td>Northeast</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>173.7</td>
<td>189.925</td>
<td>445.9179</td>
<td>334.4384</td>
<td>0.99704</td>
<td>Northwest</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>96.3333</td>
<td>132</td>
<td>137.6183</td>
<td>458.7278</td>
<td>0.99286</td>
<td>Northeast</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>177.4667</td>
<td>138.8667</td>
<td>692.274</td>
<td>1384.5479</td>
<td>0.99874</td>
<td>East</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>958.75</td>
<td>162.25</td>
<td>201.2803</td>
<td>1509.6022</td>
<td>0.99987</td>
<td>Northwest</td>
</tr>
<tr>
<td>20110505_21-03-47(1).jpg</td>
<td>746.1111</td>
<td>166</td>
<td>392.6329</td>
<td>1308.7762</td>
<td>0.99919</td>
<td>West</td>
</tr>
</tbody>
</table>

Table 5-5: Part of spreadsheet result from night 05/05/2011-05/06/2011
Figure 5-14 to Figure 5-24 give examples of different objects trajectories. They are classified as:

- birds in high altitude
- birds in low altitude
- bats in high altitude
- bats in low altitude
- insects
- unknown
- vertebrate (unable to classify as birds or bats)

Figure 5-14: Trajectory of a bird in high altitude
Figure 5-15: Trajectory of a bird in low altitude

Figure 5-16: Trajectory of a bat in high altitude
Figure 5-17: Trajectory of a bat in low altitude

Figure 5-18: Trajectory of an insect
Figure 5-19: Trajectory of an unknown object in low altitude

Figure 5-20: Trajectory of a Vertebrate in high altitude
Figure 5-21: Trajectory of an unknown interesting object

Figure 5-22: Trajectory of an unknown interesting object
Figure 5-23: Trajectory of an unknown interesting object

Figure 5-24: Trajectory of an unknown interesting object
Some results when objects have occlusions will be shown in following figures:

Figure 5-25: Trajectories of two occlusion objects (1)
Figure 5-25: Trajectories of two occlusion objects (2)
Figure 5-25: Trajectories of two occlusion objects (3)
When objects have occlusions, it would cause problem for tracking. For all the single-camera projects, there is no perfect solution for this issue. That’s one important reason why multi-camera system is desired. However, the proposed system could handle occlusions as long as the objects appear on the same location in different time. This could significantly improve the robustness of system.

5.7 Summary

This chapter has introduced the motion tracking system for our project in details. This chapter begins with the debriefing of the study area and study time, and then leads to the data collection time and techniques.

Section 5.3 introduces the method of processing with supercomputing system. It starts with discussing the difficulty of normal processing method and explains the importance of the supercomputing system. This section demonstrates the procedure step-by-step and gives a clear picture of how the system works in supercomputing system. Part of the video data has also been processed by a new high speed personal computer.

Section 5.4 and 5.5 focused on discussing the results. In Section 5.4, we have presented two other systems to measure and compare the performance of target monitoring system. The adaptive background subtraction system is widely used and is considered as a classic system; PCA system is an advanced system based on the PCA algorithm. Section 5.5 has discussed three types of test systems with samples from different aspects: the ability of tracking fast and small objects, the ability of tracking multiple objects at the same time, and time efficiency of each system.

The results of the tests show that the proposed target monitoring system provides the best performance in our project. Being a latest algorithm, PCA has the best processing
speed (14 frames/s) but weak performance when handling multiple targets. The adaptive background subtraction system tracks the most of the objects from the video, but has breakpoints sometimes. The processing time is also much longer than the other two systems due to a lack of frame selection.

In section 5.6, the summary results of season spring 2011 were shown. They provided that the target tracking system is able to provide desired results and it is computationally efficient. These results will help to develop future classifications and guide public policy in regard to interaction of wildlife with wind turbines.
Chapter 6

Conclusion and Future Research

Birds monitoring has a widespread potential that it can be applied in numerous fields such as ecology, climatology, and avian related infections such as avian influenza. Moreover, a better understanding of the flyways of various avian species can be gained by monitoring bird migration. A number of research papers show that various techniques have been developed and applied but some of them are very time-consuming and expensive. Among them, direct observation is considered as the oldest but simplest method. By observation and data recording, the migratory birds can be differentiated based on their different characteristics of flight. In order to better serve the project, a system of IR camera has been applied for reporting the migratory birds as an effective tool, as IR camera and other more sophisticated cameras were previously used for other monitoring objectives, such as traffic monitoring, security applications and military use. It allows nocturnal observation, which is very helpful to monitor any interaction with other objects during migration.

In this thesis, we have not only described a target monitoring systems but also presented the basic knowledge of IR camera as well as the experimental results. It is worth mentioning that we have designed several algorithms for our system in order to meet the requirement of precise analysis of birds’ behaviors. An IR video processing algorithm has been developed which meets the need of our project, provides
computational efficiency, reliability and accuracy. The algorithm performs better than a human eye.

In the future studies, researchers are suggested to focus on several aspects as follows:

- The observation of the avian migration could be done with more cameras installed and working together. With the slight difference made by the cameras, more information could be abstracted from the data.

- The camera could be replaced by a long-range camera to provide more details of birds in different altitudes. The data with more information could be analyzed by the updated system.

- The PCA algorithm is so promising in the video processing field that it could be used for various purposes such as locating objects in image sequences. This could be completed by abstracting principle components in 3-D.

- Both BGCFS algorithm and PCA algorithm function well with good processing speed in our system. By developing the procedure, there is a great possibility of applying them in real-time tracking system.

- This system is quite robust at handling data of steady background. It could be used in other objectives as well, such as parking lot monitoring. With a more appropriate model, it would be able to monitor outdoor environment.

- Efforts should also be directed to make this algorithm more computationally efficient and exploit graphic processing based hardware for real time processing.

With extended works, motion tracking systems are believed to be more accurate and more feasible in the future. Many types of applications have already been widely seen due to the development of motion tracking techniques.
References


[27] Su, Bolan et al, Binarization of Historical Document Images Using the Local


[34] http://www.bitchslap.nl/
