

Faculty of Engineering and Built Environment

**School of Engineering**

**Mechanical Engineering Project B**

**Semester 1 - 2019**



**PROJECT TITLE**

Optical Flow Based Obstacle Avoidance for a Fixed Wing UAV

**NAME**

Patrick Prell

**SUPERVISOR**

Dr Christopher Renton



**FINAL YEAR PROJECT**



# Optical Flow Based Obstacle Avoidance for a Fixed Wing UAV

Final Year Project Report - MECH4841 Part B

June 2019

Patrick Prell<sup>1</sup>

<sup>1</sup> *Student of Mechatronics Engineering,  
The University of Newcastle, Callaghan, NSW 2308, AUSTRALIA  
Student Number: 3204734  
E-mail: [Patrick.Prell@uon.edu.au](mailto:Patrick.Prell@uon.edu.au)*

---

## **Abstract**

Remember that executive summary may include the following information:

- Defines the intention of the report.
- Places the report in context so the reader knows why it is important to read it.
- Why is it important?
- What problem is addressed?
- Briefly states the results
- Briefly presents the implications and recommendations

## **Acknowledgements**

You may like to say thank you to someone that helped you with your project.

## Contents

<b>1. Introduction</b>	<b>5</b>
<b>2. Background</b>	<b>5</b>
2.1. Biological Inspiration . . . . .	6
2.1.1. Visual Navigation . . . . .	6
2.2. Optic Flow . . . . .	6
2.2.1. Horn & Schunk . . . . .	6
2.2.2. Lucas - Kanade . . . . .	6
2.2.3. What is Observable in Optic Flow . . . . .	6
2.3. Camera Calibration . . . . .	7
2.3.1. Camera Models . . . . .	7
2.3.2. Projective Transform . . . . .	8
2.3.3. Homogenius Coordinates . . . . .	8
2.3.4. Rodrigus Transform . . . . .	8
2.4. Inverse Optimal Control . . . . .	8
2.4.1. Sudo Control Algorithm for Insects . . . . .	8
2.5. Approach . . . . .	8
2.5.1. Inverse Depth Calculation . . . . .	8
2.5.2. Unity Simulation . . . . .	8
2.5.3. Assumptions . . . . .	8
2.5.4. Simulations . . . . .	9
2.5.5. Real World Data . . . . .	9
<b>3. Optical Flow</b>	<b>10</b>
3.1. Algorithms . . . . .	10
3.2. Measurement Model . . . . .	10
3.3. Simulation . . . . .	10
3.4. Farnback Algorithm - OpenCV . . . . .	10
<b>4. Camera Calibration</b>	<b>11</b>
<b>5. Vector Field Divergence for Object Avoidance</b>	<b>12</b>
<b>6. Flow Magnatude for Object Avoidance</b>	<b>13</b>
<b>7. Discussion</b>	<b>14</b>
<b>8. Conclusion</b>	<b>15</b>
<b>9. Recomendations</b>	<b>16</b>
<b>A. Journal</b>	<b>18</b>

## 1. Introduction

To organise your introduction section you can use the following structure:

- **Position:** Show there is a problem and that it is important to solve it.
- **Problem:** Describe the specifics of the problem you are trying to address
- **Proposal:** Discuss how you are going to address this problem. Use the literature to back-up your approach to the problem, or to highlight that what you are doing has not been done before

The rest of the report is organised as follows. Section 2 describes items related to the core content. Section 8 concludes the report. Appendix ?? shows an example of how to make a Table.

## 2. Background

To the average human or most mammals for that matter, visual tasks such as identifying objects and interpreting visual cues come as second nature. However, implementing algorithms that mimic this ability is infuriatingly difficult. What is often forgotten is that for all of its amazing abilities approximately 1/3 of the human brain is dedicated to analysing the information collected from our eyes. This statistic helps to put into perspective how challenging the field of computer vision can be.

The concept of computer vision has been explored since the beginning of artificial intelligence research in the late 1960s, with the goal of extracting the 3-Dimensional structure of a scene from images. The following decade saw research attempting to extract more information from a sequence of images using visual cues from the motion of a camera. This brought about work on visual navigation as a viable method to navigate an environment. Using Visual Navigation on a robot to move through an unstructured environment with no prior knowledge of the map is often based on optic flow (Altshuler and Srinivasan, 2018), where a motion field is estimated from fusing the motion of the robot with the motion of the pixels of a video feed. Furthermore, biologically inspired visual navigation techniques have been a topic for research. By observing the way that animals and insects utilise their visual systems to move about and map their environments, the processes for which this is achieved can be replicated and used on a robotic system for similar tasks. This section outlines the planar and spherical model of optic flow, the tools used to infer motion between frames, and how honeybees use Visual Navigation (Srinivasan, 2011).

## **2.1. Biological Inspiration**

### **2.1.1. Visual Navigation**

#### **2.1.2.**

#### **2.1.3.**

## **2.2. Optic Flow**

The first use

### **2.2.1. Horn & Schunk**

### **2.2.2. Lucas - Kanade**

### **2.2.3. What is Observable in Optic Flow**

Pose rate estimation

- Angular velocity,
- reflect on the work presented,
- make recommendations,
- suggest future work or improvements.

## 2.3. Camera Calibration

Computer vision begins with the camera detecting light from a scene. This light has travelled from its source, reflecting from an object, through the camera lens and is recorded on the image sensor. For these rays of light that make up an image to be useful for representing the real-world, we must understand the geometry that transforms these rays from the three-dimensional real world to a two-dimensional image plane i.e. world coordinates to image coordinates.

The camera calibration process describes the camera with two sets of parameters, namely the intrinsic parameters, and the extrinsic parameters. The intrinsic parameters describe the geometric relationship between the camera model and its projection. And the extrinsic parameters describe the location and orientation or pose, of the camera with respect to the world frame.

This project uses a GoPro Hero 7 to test and verify the calibration routine. This section will cover camera models and outline why a planar model is used for the GoPro. It covers the mathematical tools used to capture the different effects that make our camera imperfect for representing the real world, and how we can correct for these deviations.

### 2.3.1. Camera Models

The most elementary way of representing a camera is with the pinhole model. Light enters the aperture (pinhole) after reflecting from objects in a scene. For an ideal pinhole camera, only one ray enters the aperture from any point in the scene. The light is then projected onto a surface behind the aperture resulting in an image of the scene that is scaled proportional to the focal length  $f$ .

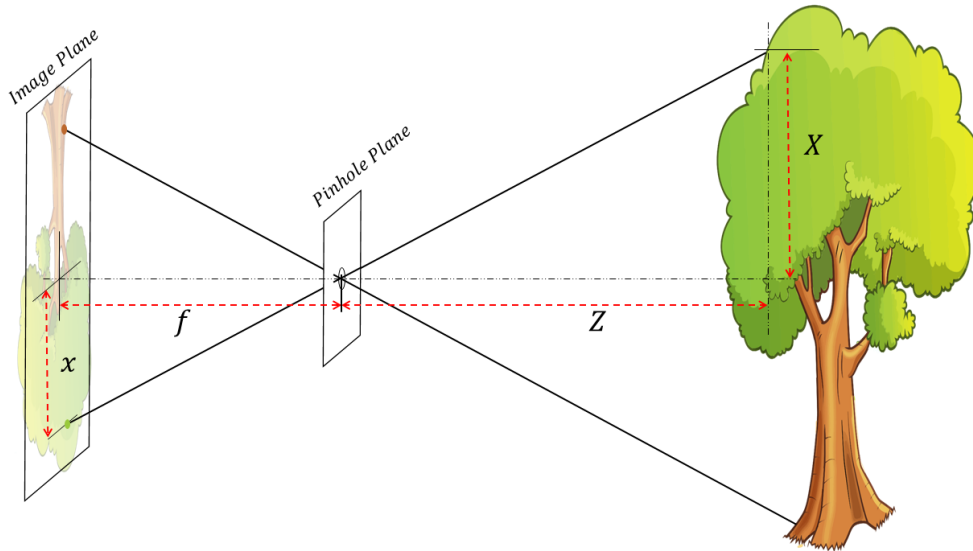


Figure 1: Elementary pinhole model of a camera: In practice, the aperture must be big enough to let light in, allowing a narrow ray of light from each point in the scene blurring the image. Therefore there is a tradeoff between brightness and sharpness.

Figure 1 shows an ideal pinhole camera model and its parameters, where  $f$  is the focal length,  $Z$  is the distance along the optical axis to an object,  $x$  is the size of the projected object and  $X$  is the size of

the object. For this model, (2.1) is the relationship between the projected image and the object

$$\frac{-x}{f} = \frac{X}{Z} \quad (2.1)$$

Due to how light enters the camera, the image of a pinhole model is represented upside down. This issue is fixed in a planar camera model. The planar model is similar to a pinhole model, however rather than passing through an aperture, light travels from each point in the scene, intersects the image plane and converges to a single point known as the *Centre of projection*. this concept is best described in Figure 2, where

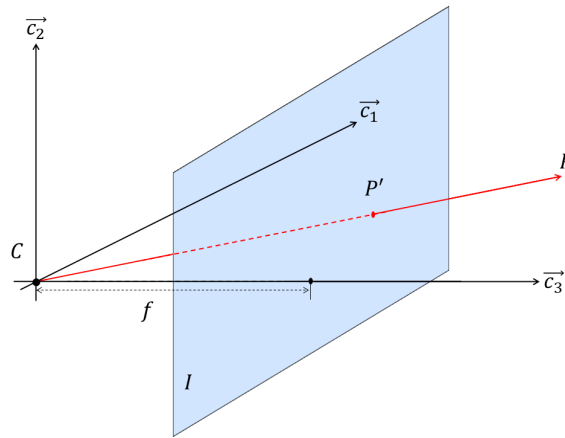


Figure 2

### 2.3.2. Projective Transform

### 2.3.3. Homogenous Coordinates

### 2.3.4. Rodrigus Transform

## 2.4. Inverse Optimal Control

### 2.4.1. Sudo Control Algorithm for Insects

### 2.4.2.

## 2.5. Approach

### 2.5.1. Inverse Depth Calculation

### 2.5.2. Unity Simulation

### 2.5.3. Assumptions

Static environment using GoPro Simple aircraft kinematic model



#### **2.5.4. Simulations**

Kinematic data collected Used ray cast to gather depth information

#### **2.5.5. Real World Data**

The project is primarily simulation based. However, the camera calibration routine (explored in following sections) was tested and evaluated on real world images collected from a GoPro Hero 7. Were this project to be implemented on a real world fixed wing UAV, the kinematic data would need to be obtained through sensors on the UAV. Additionally, the control algorithm U-Blox RTK GPS module High Performance MEMS IMU (i.e. ) GoPro Hero

### **3. Optical Flow**

#### **3.1. Algorithms**

#### **3.2. Measurement Model**

#### **3.3. Simulation**

#### **3.4. Farnback Algorithm - OpenCV**

## **4. Camera Calibration**

## **5. Vector Field Divergence for Object Avoidance**

## **6. Flow Magnatude for Object Avoidance**

## **7. Discussion**

## **8. Conclusion**

## **9. Recommendations**



## References

## **A. Journal**