Faculty of Engineering and Built Environment

School of Engineering

Mechancial Engineering Project B

Semester 1 - 2019

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

PROJECT TITLE

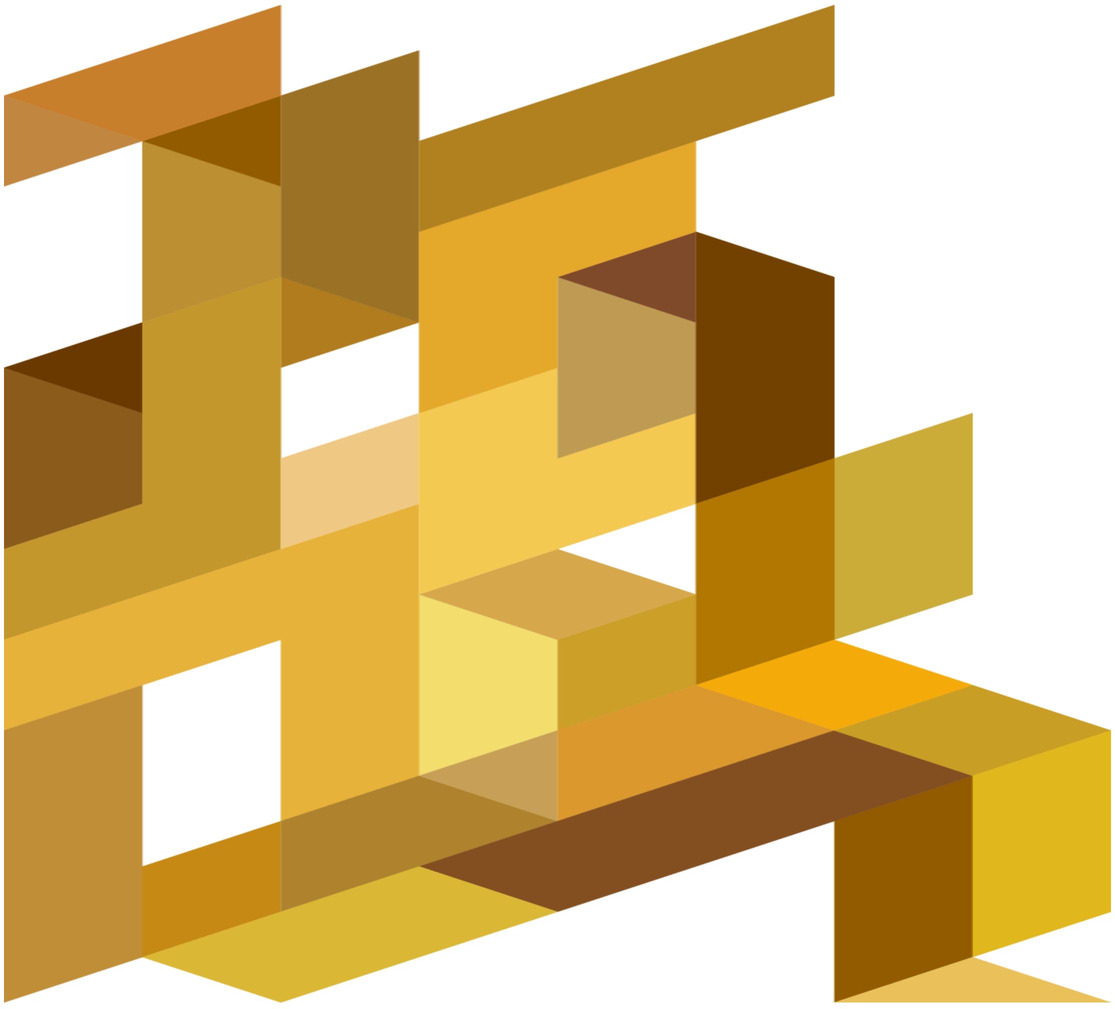
Optical Flow Based Obstacle Avoidance for a Fixed Wing UAV

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 [1]

[1] *Student of Mechatronics Engineering,*
The University of Newcastle, Callaghan, NSW 2308, AUSTRALIA
Student Number: 3204734
E-mail: 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

1. Introduction

Exploring and navagating unknown environments is an essential instinct in biology and avoiding obstacles is an major component. For robotic vehicles to perform high level tasks, they must also have these instincts.

- **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 seem like a trivial task. However, implementing algorithms that mimic a mammalian visual cortex is infuriatingly difficult and largely remains an unsolved problem (Hartley and Zisserman, 2003). One fact often overlooked is that for all of its amazing abilities, approximately 1/3 of the human brain is dedicated to analysing 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 inception of artificial intelligence in the late 1960s. Corresponding with a rise in electronic computer power, allowing manipulation of large data sets such as those collected from a camera. The following decade saw research attempting to extract information from a sequence of images using visual cues embedded in the motion of an image. This brought about research on visual navigation as a viable method for a mobile robot 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 optical flow. A motion field is estimated from fusing the motion of the robot with the motion of the pixels of a video feed. Furthermore, visual navigation techniques have been inspired by biology. 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 implemented on a robotic system (Altshuler and Srinivasan, 2018).

The following section builds the background knowledge required to utilise visual cues as input to an obstacle avoidance system. In particular, a history of visual navigation is outlined; the geometry of a camera and the tools for calibrating it is introduced; and the kinematics of vision optical flow and optical flow on the view sphere. Finally, this section discusses the approach taken to validate the obstacle avoidance algorithm.

2.0.1. Biomimicry

Biology so often holds a wealth of inspiration for engineers. From the material science of the microstructures in a butterfly wing to the computational fluid dynamics of a Kingfisher diving into water (Benyus, 1997). visual sensors are found so frequently in nature, making it very productive to look at nature and tap into the eons of knowledge distilled by evolution.

2.0.2. Visual Navigation

2.0.3. Sudo Control Algorithm for Insects

2.0.4.

2.1. Optical Flow

In computer vison we are constanly running into the problem of dimentional compression. A lot of information is lost in the projection from 3 dimentions to a 2D plane, and when that projection is inverted, the lost data needs to be inferred. In the case of optical flow, the lost data comes mainly from different assumptions that are made to make the optic flow algorithm solvable. Each Optical flow algorithm has its own set of assumptions and constraints

2.1.1. Horn & Schunk

Optical flow is an ill-posed problem, meaning that there is only one independent measurement available from the image sequence and we require 2 variables to describe the flow at a point. B.K.P. Horn and B.G. Schunck developed an algorithm for computing optical flow in 1980 (Horn and Schunck, 1981) by adding a second constraint known as the brightness constancy. The brightness constancy constraint assumes that the pixel intensities remain constant over time.

$$\frac{d}{dt}I(x(t), y(t), t) = 0 \tag{2.1}$$

where $I(x(t), y(t), t)$ is the pixel intencity of the image at time t.

Horn and Schunck's optical flow algorithm was a huge step forward in computer vision research, although this algorithm is limited in that it typically can only estimate small motions, the method fails in the presence of large motion when the gradient of the image is not smooth enough (Meinhardt-Llopis et al., 2013).

2.1.2. Lucas - Kanade

The Lucas-Kanade method is a widely used method for calculating optical flow. This method operates under the assumption that the flow is effectively constant in a small cluster of pixels, the method can then solve the optical flow equation for all the pixels in that neighbourhood using least squares regression. The Lucas-Kanade method improves on previous methods by using the spatial intensity gradient to find the disparity vector h, reducing the algorithms calculation time to $O(M^2 log(N))$ for the average case.

2.1.3. What is Observable in Optical Flow

Pose rate estimation

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

2.2. 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.

2.2.1. 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 proportionally to the focal length f.

Figure 4 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, (2.2) is the relationship between the projected image and the object

$$\frac{-x}{f} = \frac{X}{Z} \tag{2.2}$$

Due to how light enters the camera, the image projection from a pinhole model is projected 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*. Figure 2 shows an ideal planar camera model. It better represents the geometry of a camera with a lens, albeit a camera with an impossibly perfect lens.

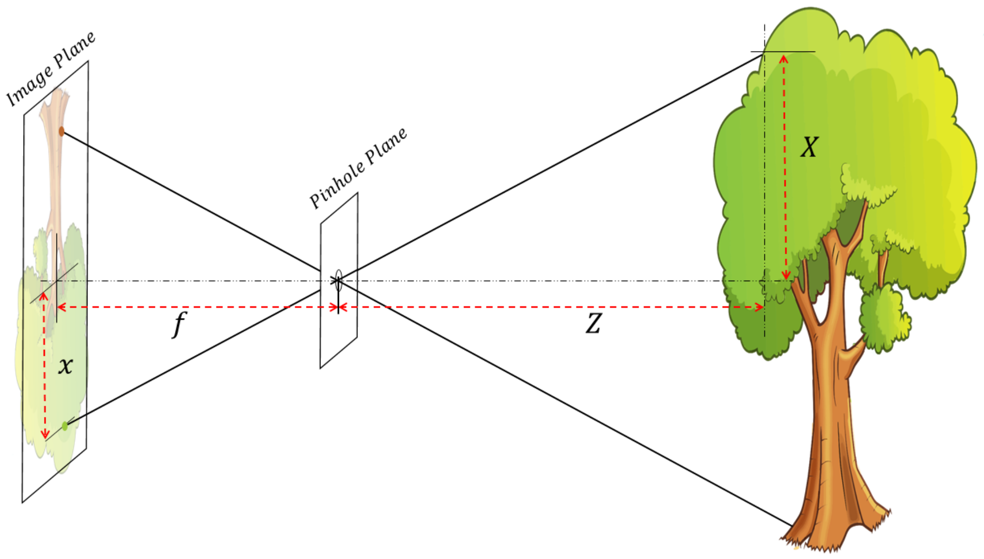


Figure 1: Elementary pinhole model of a camera: In practice, the appeture must be big enough to let light in, allowing a narrow ray of light from each point in the scene blurring the image. Therefore there is a tradeoff between brightness and sharpness.

The idea behind modelling and calibrating a camera is to identify and cancel out imperfections caused by slight defects in the manufacturing process. Firstly, the image sensor is often mounted askew from the optical centre of the lens. Therefore, we introduce α_x and α_x to model a displacement from the optical centre. Furthermore, each pixel is not necessarily square causing a small discrepancy between the focal length in the $\vec{u}$ and $\vec{v}$ directions, captured in the f_u and f_v parameters. The following equations 2.2.1 project a point P_{c_1,c_2,c_3} to image sensor co-ordinates u, v:

$$u = f_u \times \frac{\vec{c_1}}{\vec{c_3}} + \alpha_u$$
$$v = f_v \times \frac{\vec{c_2}}{\vec{c_3}} + \alpha_v$$

2.2.2. Projective Geometry

Projective geometry is a way to describe the relationship that transforms a point from world coordinates (n_1, n_2, n_3) into image coordinates u, v. When applying such transformations, it is convenient to add an extra dimension to each vector in the transform. for instance, a point that lies in a n dimensional space is represented in homogeneous coordinate as a $n+1$ dimensional vector. this co-ordinate system is known as *homogenius coordinates*. for example, a point lying on a plane at (x, y) is represented by the homogenous coordinates (x, y, z). It is important to note that in homogeneous coordinates, any two vectors that share a common factor represents the same point in Euclidean geometry. This has the effect that a point in Euclidean geometry can be represented by an infinite number of homogeneous vectors. for instance $\alpha * (q_1, q_2, q_3) = (q_1, q_2, q_3)$. The original primitive point is recovered by dividing through by q_3.

C	: Centre of projection
$\{c\}$	: Camera co-ordinates
P	: Point in world space
P'	: Point on image plane
$\vec{u}, \vec{v}$	: Image co-ordinates
f	: Focal length
I	: Image plane

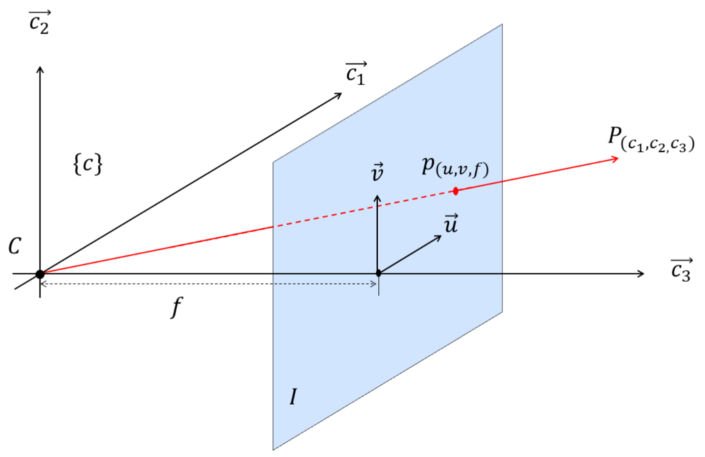


Figure 2: Ideal camera model. A point P in camera co-ordinates $\{\vec{c_1}, \vec{c_2}, \vec{c_3}\}$ is projected and displayed as a point p in image co-ordinates $\{\vec{u}, \vec{v}\}$

2.2.3. Lens Distortion

2.3. Approach

2.3.1. Unity Simulation

2.4. Divergence

2.4.1. Assumptions

As discussed in

Static environment using GoPro Simple aircraft kinematic model

2.4.2. Simulations

Kinematic data collected Used ray cast to gather depth information

2.4.3. Real World Data

The project is primaritly simmulation 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 GoPro Hero

3. Optical Flow - Simulation

3.1. Measurement Model

3.2. Farnback Algorithm - OpenCV

4. Camera Calibration

As discussed in section 2.2 it is impericive to understand and account for the construction adn imperfections in a camera, if the measurments takne need to be relevent outside the camera basis C. This section outlines the procedure used in this project to correct a gopro hero 7 using a planar camera model and a 6th order equation to correct for distortion parameters.

4.1. Calibration Boards

Theoretically, to calibrate a camera any object could be used as a calibration object, so log as it is appropriately characterised. However it is much more practical to use a well defined and easily defined object. For this reason, a flat rigid checker pattern was chosen. The geometry of the calibration pattern can be seen in 3:

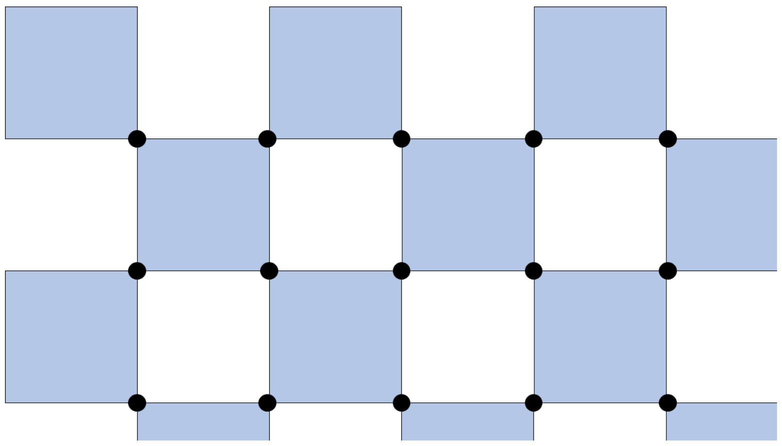


Figure 3: Write something here

4.2. Calibration Tool

To calibrate the GoPro Hero 7 for in this project, A camera calibration tool was written in C# using WPF as a graphical user interface(see Appendix B)).

5. Vector Field Divergence for Object Avoidance

6. Flow Magnatude for Object Avoidance

7. Discussion

8. Conclusion

9. Recomendations

References

Altshuler, D. L., Srinivasan, M. V., 2018. Comparison of visually guided flight in insects and birds. Frontiers in neuroscience 12, 157.

Benyus, J. M., 1997. Biomimicry: Innovation inspired by nature.

Hartley, R., Zisserman, A., 2003. Multiple view geometry in computer vision. Cambridge university press.

Horn, B. K., Schunck, B. G., 1981. Determining optical flow. Artificial intelligence 17 (1-3), 185–203.

Meinhardt-Llopis, E., Pérez, J. S., Kondermann, D., 2013. Horn-schunck optical flow with a multi-scale strategy. Image Processing on line 2013, 151–172.

A. Journal

B. Calibration Tool

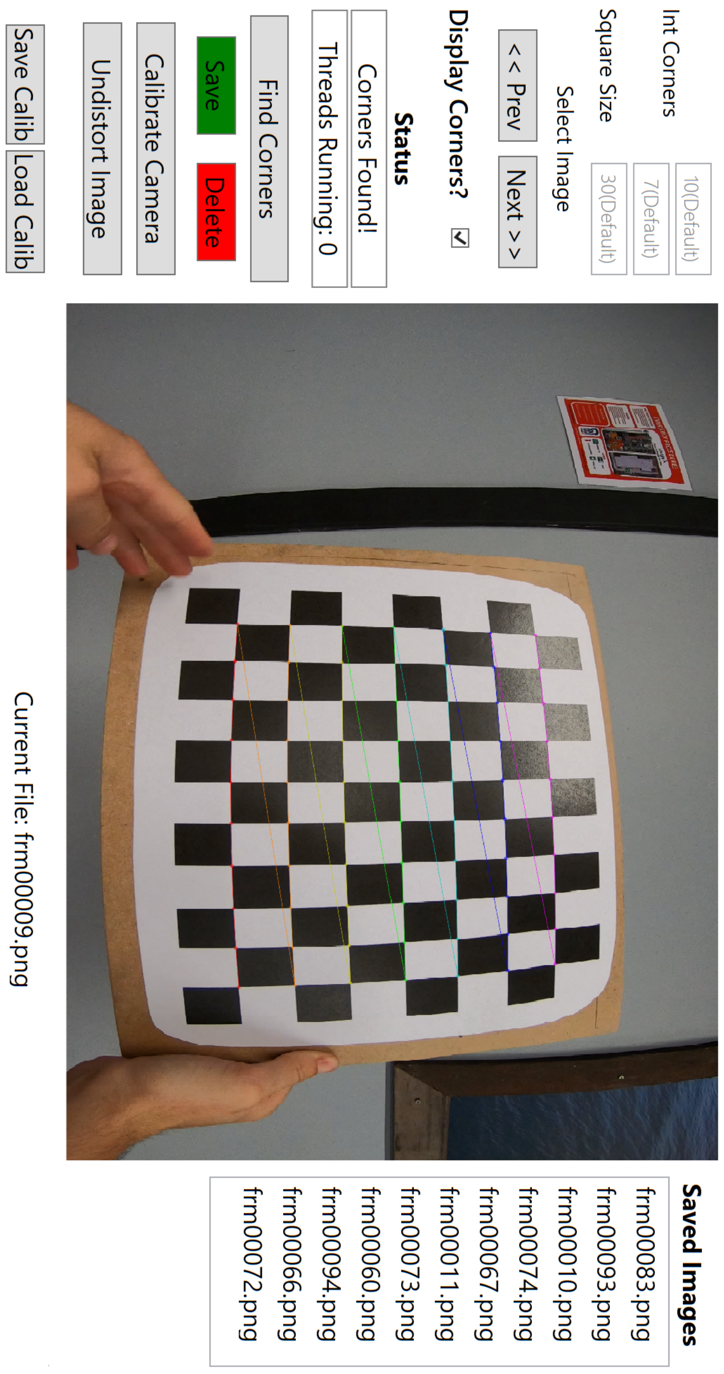


Figure 4: Write something here