A Novel Method to Locate Targets Using Active Vision and Robotic Inertial Navigation Data

Matthew James Simone

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
In
Electrical Engineering

Dr. Pushkin Kachroo, Chair
Dept. of Electrical Engineering

Dr. Charles Reinholtz
Dept. of Mechanical Engineering

Dr. Alfred Wicks
Dept. of Mechanical Engineering

May 11, 2006
Blacksburg, Virginia

Keywords: target acquisition, waypoint navigation, autonomous robots, pattern matching, image processing

A Novel Method to Locate Targets and Areas of Interest Using Active Vision and Robotic Inertial Navigation Data

Matthew Simone

(ABSTRACT)

Unmanned vehicles are increasingly being used for mobile sensing missions. These missions can range from target acquisition to chemical and biological sensing. The reason why these vehicles are increasingly being used is because they can carry many different types of sensors and can function as a cheap platform for carrying these sensors. The sensing that will be explained in this thesis is target acquisition. Target acquisition is the act of locating the exact position of an "area of interest." Currently this task can be completed with different types of complex range sensors. This thesis presents a type of target acquisition scheme for unmanned vehicles that will use a combination of cheap, simple vision sensors and robot inertial navigation data in order to accurately measure the location of a target in real world coordinates.

This thesis will first develop an accurate waypoint driving algorithm that will either use dead reckoning or GPS/ compass sensors. We will then develop a robust target extraction algorithm that will be able to pick out a target in an image. After this is completed we will develop an algorithm that will be used to find the distance to the target from the robot. This algorithm will be based on a type of active vision system.

Finally we will integrate all of these algorithms together in order to develop a target extraction technique that will be able to accurately find the distance to the target. With the distance we can then find the real world location of the target.

Acknowledgements

The author first would like to thank Dr. Pushkin Kachroo for all his help and guidance throughout this journey. Second the author would like to thank my committee members Dr. Reinholtz and Dr. Wicks for all of their guidance and support. Without their help I would not have been able to enter this exciting field of unmanned systems.

The author would like to also thank the IGVC team for letting me use Johnny 5 as the test robot. This work would not have been able to be completed without this “robust” robot.

Finally, the author would like to thank my family for all of their support. Without them I would not be able to achieve this goal.

Contents

Chapter 1 ..... **1**

1.1. Motivation 1
1.2. Target Extraction 2
1.3. Previous Research 2
1.4. Contributions of this Thesis 3
1.5. Organization of this Thesis 3

Chapter 2 ..... **5**

2.1 Vision Sensors 5
2.2 Range Finder Sensors 12

Chapter 3 ..... **16**

3.1. Robot Description 16
3.2. Programming Language Description 17
3.3. Robot Modeling 19
3.4. Open Loop Control 20
3.5. Closed Loop Control with Encoders 22
3.6. Closed Loop Control with GPS 26
3.6.1. GPS System Characteristics 27
3.7. Closed Loop Control with Sensor Fusion 32

Chapter 4 ..... **36**

4.1. Target Extraction Techniques 36
4.1.1. Hough Transform 37
4.1.2. Pattern Matching 40
4.2. Comparison of the Two Techniques 44

Chapter 5 ..... **46**

5.1. Camera Calibration 46
5.2. Target Distance Measurement Methods 47
5.2.1. Method 1 47
5.2.2. Method 2 49
5.3. Test Results 52
5.3.1. Method 1 Test 52
5.3.2. Method 2 Test 53

Chapter 6 ..... **56**

6.1 Open-Loop Target Acquisition 56
6.2 Closed-Loop Target Acquisition 58
6.3 Propagation of Uncertainty 59

Chapter 7 ..... **64**

7.1. Concluding Remarks 64
7.2. Future Work 65
7.2.1. Waypoint Controller 65

7.2.2. Target Extraction with Robotic Control 65
Appendix A. 68
A.1. Experiment2.m 68
A.2. Error_Propagation.m 69

List of Figures

Figure 2.1 Fire-I™ Board Digital Camera [6] 6
Figure 2.2 Basler A102FC Series Camera [7] 7
Figure 2.3 Axis 213 PTZ Network Camera [8] 8
Figure 2.4 Point Grey Research's Bumblebee™ [10] 10
Figure 2.5 TYZX DeepSeaV2 [11] 11
Figure 2.6 SICK PLS LRF [12] 12
Figure 2.7 Opti-Logic RS100 [13] 14
Figure 3.1: Picture of Johnny 5 16
Figure 3.2: Stripped Down Version of Johnny 5 17
Figure 3.3: Sample Labview Program Screenshot 18
Figure 3.4: Sample Labview Program Front Panel 18
Figure 3.5: Johnny 5's Kinematics Model 19
Figure 3.6: Waypoint Navigation Description [16] 23
Figure 3.7: Heading Error [16] 24
Figure 3.8: Johnny 5 Encoder Waypoint Navigation Test 25
Figure 3.9: Johnny 5 Waypoint Test Heading Plot 26
Figure 3.10: GPS Constellation 27
Figure 3.11: Novatel ProPack GPS Receiver 28
Figure 3.12: Scattered GPS Measurements 29
Figure 3.13: Johnny 5 GPS Waypoint Navigation Test 30
Figure 3.14: Johnny 5 GPS Waypoint Navigation Heading Plot 31
Figure 3.15: Extended Kalman Filtering System 32
Figure 4.1: IARC Target [20] 36
Figure 4.2: Canny edge image of IARC target 38
Figure 4.3: Three dimensional Hough Space [24] 38
Figure 4.4: Hough Transform Result 40
Figure 4.5: Block Diagram of the Pattern Matching Algorithm 41

Figure 4.6: Target Image Threshold 42
Figure 4.7: Matched Region of IARC Target 44
Figure 5.1: Axis 213 PTZ Network Camera [8] 46
Figure 5.2: Experiment 1 Set-up 47
Figure 5.3: Image Plane Projection 48
Figure 5.4: Stereo Vision Set-up 49
Figure 5.5: General Stereo Vision Setup 51
Figure 5.6: Method One Test 53
Figure 5.7: Parallel Image Plane Test 54
Figure 5.8: General Image Plane Test 55
Figure 6.1: Open Loop Control Experiment Set-up 56
Figure 6.2: Closed Loop Control Experiment Set-up 58

List of Tables

Table 2.1 Fire-I™ Specification Table [6] 6
Table 2.2 Basler A102FC Specification Table [7] 7
Table 2.3 Axis 213 PTZ Network Camera [8] 9
Table 2.4 Point Grey Research's Bumblebee™ [10] 10
Table 2.5 TYZX DeepSeaV2 [11] 11
Table 2.6 SICK PLS LRF [12] 13
Table 2.7 Opti-Logic RS100 [13] 14
Table 3.1: Comparison of Mode Accuracies 28
Table 4.1: Results of the Two Different Techniques 45
Table 6.1: Open Loop Experiment Results 57
Table 6.2: Closed Loop Experiment Results 59
Table 6.3: Experimental Data with Two Standard Deviation Error 62
Table 6.4: Experimental Data with Greater Error Bounds 62

Chapter 1

Introduction

Mobile robots are increasingly becoming more complex and robust. They can now navigate waypoints and complicated paths. They can even navigate through rough terrain with obstacles as demonstrated in the latest DARPA Grand Challenge [1]. Since they can achieve these goals more emphasis are starting to be applied to different types of missions. One mission is target acquisition. Target acquisition is the process of identifying a target and determining the location of the target in the real world.

1.1. Motivation

There are many reasons why mobile robots are increasingly being used for target acquisition. The most important reason is that it takes people out of harms way. If the robot was not used for this mission then a person would have to enter the dangerous situation. The next reason is that people can send these robots out and ideally they would complete their task without human interaction. This would let a person complete a more valuable task while the robot is completing its own mission.

The next reason why mobile robots can be used for target acquisition is that they have the capability of carrying different types of payloads. These payloads can range from relatively large laser range and Electro-Optic/ Infrared (EO/IR) sensors to small stereo vision systems. These types of payloads can be very expensive and are not financially viable on small cheap mobile robots. In order to reduce these payload costs and still complete the same target acquisition task new types of algorithms have to be developed that will use cheap payload sensors.

1.2. Target Extraction

The goal of this thesis is to develop a target extraction algorithm that can use cheap sensors. These sensors can be vision sensors like CCD cameras or in the case of targeting an RF source they can be RF sensors. Since we only have access to CCD cameras the type of sensor and algorithm that will be used will be based on CCD cameras.

1.3. Previous Research

There has been significant research in locating the global pose of a robot by cameras. These systems use many cameras located about a room in order to triangulate the position of a robot. This is the complete opposite to what is proposed in this thesis. We are going to use a robot with known location and a camera for target acquisition.

Target acquisition can be thought of in two ways. One way would be to acquire a target and just know where it is in the image. The other way would be to acquire the target and locate it with respect to the world coordinate frame. For example this could be a way to find the targets GPS location.

There has been some effort into this, for example, [2] used two different ways to acquire a target. They first found the target in the image plane and aimed the gun at it. This method just obtained the two dimensional information of the target. They also found the three dimensional position of the target using the location of the target laser and the camera. With these locations they found the location of the target just by simple triangulation.

Target acquisition by imaging can be thought of to be 3-d stereo vision. The reason being is that target acquisition is the process of finding the position of a target. This is what stereo vision accomplishes. Stereo vision can be implemented in many different ways. The most known way is having two cameras side by side. Another example of a stereo vision system is in [3]. This system uses a single camera and optics in order to create two virtual cameras that can be used for stereo vision. Another stereo vision method is found in [4]. This method uses a camera mounted to a turn table with an offset from center. A single image is acquired then the camera is rotated on the turn table and a second image

is acquired. Since the camera is mounted on the turn table with an offset the camera not only rotates but it also changes location. This change of location can be used to calculate the 3-d information of the scene.

1.4. Contributions of this Thesis

The following contributions were made during this thesis work.

- Two different vision processing techniques were compared.
    - The first technique is the Hough transform.
    - The second technique is pattern matching.
- Three different waypoint navigation techniques were developed and compared.
    - The first technique is closed loop control using encoders.
    - The second technique is closed loop control using GPS and a digital compass.
    - The third technique is closed loop control using an Extended Kalman filter.
- A stereo vision system algorithm using a single camera and robot movement was developed.

1.5. Organization of this Thesis

This thesis is organized as follows:

- Chapter 2: This chapter describes the different vision and laser range sensors that are on the market.
- Chapter 3: This chapter describes two different robot control laws that can be used.
    - An open loop controller that uses encoder values to let the robot know where it is at.
    - A closed loop controller that uses GPS and a digital compass to in order to drive waypoints.

- Chapter 4: This chapter describes two different types of algorithms that can be used to pick a target out of an image.
    - One algorithm is the Hough transform that can be used to detect circular objects in an image.
    - Another algorithm is a pattern matching algorithm that is used to detect a pattern in an image.
- Chapter 5: This chapter describes two different types of techniques that can be used to extract target distance from a set of images.
    - The first technique is to take an image of a target straight on and calculate the distance to it.
    - The second technique is to take two pictures from different locations and use stereo vision calculations in order to calculate the distance to a target.
- Chapter 6: This chapter will discuss the real world results of the two different types of experiments.
    - The first experiment is target acquisition with open loop control.
    - The second is target acquisition with closed loop control.

Chapter 2

Sensor Background

This chapter describes the different types of sensors on the market that can be used for target acquisition. This is compromised of vision sensors and laser range finder sensors. We will describe the different types of sensors and list their specifications for comparison.

2.1 Vision Sensors

There are many different vision sensors out in the market place that are ideal for picking out the location of objects. They range from cheap single lens charged couple device, (CCD) cameras to expensive 3-D stereo vision systems. These vision sensors also include infra-red (IR) and night vision cameras. Among these types of sensors, stereo vision systems produce the most information about an objects location. The reason being is that stereo vision systems can acquire the location of an object in an image plane as well as the distance away from the camera.

The CCD camera is the most frequently used camera for vision. The CCD chip in the camera is an array of "tiny light sensitive diodes which accumulate electrons (electrical charge) when they are hit with light particles (photons)" [5]. These accumulations of electrons are then converted to an image. There are many different types of CCD cameras on the market that can be used as sensors for unmanned vehicles. These range from very basic functionality cameras like the Fire-I^{TM} board digital camera to the Axis^{TM} 213 Pan Tilt Zoom network camera. The camera shown below in Figure 2.1 is the Fire-I^{TM} board digital camera.

Figure 2.1 Fire-I^{TM} Board Digital Camera [6]

The important specifications for this camera are shown below in Table 2.1.

Specification	Value
Overall Size (L x W x H)	2.12 in x 2.05 in x 0.68 in
Color Resolution	640 x 480 pixels
Max Color Frame Rate (fps) At Max Resolution	30 fps
Connection Type	IEEE-1394 Firewire
Price	$105.00 (Sept. 2005)

Table 2.1:Fire-I^{TM} Specification Table [6]

As shown in the above table, the Board Digital Camera is a CCD camera that has a color resolution of 640x480 with a maximum frame rate of 30 frames per second. It also has a fairly small footprint measuring roughly two inches on each side. Since the camera comes in a board form, any enclosure can be designed for the camera. This makes it an ideal camera for small unmanned vehicle applications. This camera also uses an IEEE-1394 Firewire connector as its interfacing connection, thus making this camera fairly easy to interface with many types of computers that have Firewire capability. Another option that this camera has is the choice between many different types of lenses. A lens can be chosen for any focal length that is needed. A lens for different horizontal viewing angles

can also be found to fit this camera. Finally the price for the Digital Board Camera is one hundred and five dollars in September 2005.

The next digital camera that will be reviewed is the Basler A100 Series camera which is shown below in Figure 2.2.

Figure 2.2 Basler A102FC Series Camera [7]

The specifications for this camera are shown below in Table 2.2.

Specification	Value
Overall Size (L x W x H)	2.42 in x 2.42 in x 1.90 in
Color Resolution	1392 x 1040 pixels
Max Color Frame Rate (fps) At Max Resolution	15 fps
Connection Type	IEEE-1394 Firewire
Price	$3,500.00 (Sept. 2005)

Table 2.2: Basler A102FC Specification Table [7]

The Basler A102FC digital camera is a higher end camera then the Fire-I^{TM} Board Digital Camera. This higher end classification means that this camera has better resolution then the previous camera. This higher resolution is better for machine vision because it can pick out objects better. The downside to this higher resolution is the frame rate. As shown the Basler camera has a frame rate of only 15 frames per second, while the board camera

has a frame rate of twice that. This high resolution also hampers the vision processing, meaning that it will take a longer time to accomplish vision processing if this camera was chosen. The Basler A102FC costs $3,500 without the added cost of lenses. The lenses that can be added range in the price from $100 to a couple of thousand dollars. But just about any lens type can be added to this camera meaning that this camera can be altered to different needs. Finally this camera also uses an IEE-1394 interface as its interfacing connection.

The Axis 213 PTZ Network Camera is the final CCD camera that will be reviewed here. PTZ in the name stands for Pan Tilt Zoom. This camera is shown below in Figure 2.3.

Figure 2.3 Axis 213 PTZ Network Camera [8]

Listed below in Table 2.3 are the specifications for this camera.

Specification	Value
Overall Size (L x W x H)	4.09 in x 5.12 in x 5.12 in
Color Resolution	768 x 576 pixels
Max Color Frame Rate (fps) At Max Resolution	30 fps
Connection Type	RJ-45 Ethernet
Price	$1,629 (Sept. 2005)

Table 2.3: Axis 213 PTZ Network Camera [8]

The Axis PTZ camera is unique for a couple of different reasons. The first reason is that it has the ability to pan, tilt, and zoom. This ability can be controlled from a viewing computer. The angles that the camera pans and tilts can also be measured from a controlling computer. This camera has a pan angle range of 340 degrees and a tilt angle range of 100 degrees. The Axis 213 PTZ also has an IR illuminator so that the camera can be used at night. The interface to the controlling computer is an RJ-45 Ethernet connection. This means that as long as this camera is on a network, it can be controlled by any computer that is running the controlling program.

The next types of cameras are able to acquire vision in three dimensions. These types of cameras are stereo vision cameras. Stereo vision cameras work by finding the same features in each of the two images, then, by vision processing, measuring the distances to objects containing the features by triangulation [9]. This process enables the camera to provide images to the robotic system in 3-d. These types of images are useful for navigation since a robot can acquire a target and know exactly where it is, in three dimensions, unlike regular CCD cameras.

The first type of stereo vision system is Point Grey Research's Bumblebee^{TM} system. Figure 2.4 below shows a picture of the camera.

Figure 2.4 Point Grey Research's Bumblebee™ [10]

Shown below in Table 2.4 are the specifications for this stereo system.

Specification	Value
Overall Size (L x W x H)	6.30 in x 1.57 in x 1.97 in
Color Resolution	640 x 480 pixels
Max Color Frame Rate (fps) At Max Resolution	30 fps
Connection Type	IEEE 1394 Firewire
Price	$3,000 (October 2005)

Table 2.4: Point Grey Research's Bumblebee™ [10]

The Bumblebee uses two color CCD cameras, as seen in Figure 2.4, to obtain the images for three dimensional data extraction. It also comes as a full development kit, with the camera head, interface card and software. The Bumblebee is also pre-calibrated for camera misalignment and lens distortion. This is important because any variation in these variables means an error in the three dimensional accuracy. Finally the left and right cameras are aligned to a singular focal point within 0.05 RMS pixel accuracy.

The final stereo vision system that will be reviewed is TYZX's DeepSeaV2 Development System. This camera is shown below in Figure 2.5.

Figure 2.5 TYZX DeepSeaV2 [11]

This camera also has the same characteristics of the Bumblebee system, which is that this system also uses two color CCD cameras to obtain images to extract three dimensional information from the scenes. These color cameras have a fixed focal point that is set by TYZX. This systems has an onboard stereo computation engine, meaning that the robots computer will be able to use this system and not be burdened by the intense image processing that is needed to extract three dimensional information from images. Listed below in Table 2.5 is a list of specifications for this system.

Specification	Value
Overall Size (L x W x H)	5 in x 2 in x 1.5 in
Color Resolution	512 x 480 pixels
Max Color Frame Rate (fps) At Max Resolution	30 fps
Connection Type	PCI card
Price	Unknown

Table 2.5: TYZX DeepSeaV2 [11]

This camera system is just slightly smaller then the Bumblebee. This small size means that it can be attached easily to an existing robotic system and be attached to smaller

robotic systems. The DeepSea system uses a PCI interface to the robotic computer system.

2.2 Range Finder Sensors

The next set of sensors that will be reviewed is range finder sensors. These sensors include laser range finders, ultrasonic sensors, and sonar sensors. Laser range measures a distance to a target by shooting a laser beam out at a target and measuring the time it takes for the reflection to come back. This time can then be used to calculate the distance to that target. Ultrasonic and sonar sensors operate on the same principle except they use sound.

The first type of range sensors that will be reviewed is a laser range finder. The specific type of laser range finder (LRF) that will be reviewed is the SICK PLS LRF. This is shown below in Figure 2.6.

Figure 2.6 SICK PLS LRF [12]

The laser range finders have to be reviewed differently then the CCD cameras. The reason being is that they operate differently. Below in

Table 2.6 is a list of important specifications for the SICK PLS LRF.

Specification	Value
Overall Size (L x W x H)	6.14 in x 6.10 in x 7.23 in
Scanning Range	50 meter radius
Range Precision	$\pm$ 50 mm
Scanning Angle	180º max
Scanning Rate	$\geq$ 80 ms
Interface	RS-232 Serial
Price	$4,350.00 (April 2006)

Table 2.6 SICK PLS LRF [12]

The first specification in this table is the overall size of the range finder. As can be seen in this table, this LRF is fairly large for a sensor. Since this sensor is fairly large it can only be used on robots that are large enough to carry it. It is also fairly easy to integrate with a controlling computer because it uses RS-232 for communication. The scanning range of this LRF is 50 meters with a range precision of $\pm$ 50 millimeters, which is fairly far and accurate and thus an adequate range for robots. The scanning angle of this LRF is 180º max which means that it can scan everywhere in front of the vehicle. The final specification in this table is the Scanning Rate, which is $\geq$ 80 ms. This can be a draw back for applications where a robot can travel quickly. The reason why this is a drawback is if the robot is covering a large amount of distance quickly, this LRF will not be able to scan completely in front of the vehicle quick enough and thus can get “blind spots” in front of the vehicle.

The next type of laser range finder is quite different then the SICK PLS LRF. This laser range finder is called the RS100 and is made by the Opti-Logic Corporation. A photo of this product is shown below in Figure 2.7.

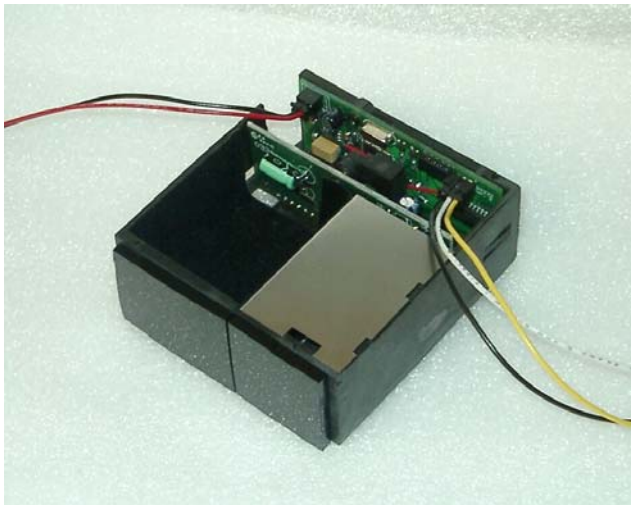

Figure 2.7 Opti-Logic RS100 [13]

As can be shown in this picture this laser range finder is not as neatly packaged as the SICK laser range finder. A table of the specifications for this laser range finder is shown below in Table 2.7.

Specification	Value
Overall Size (L x W x H)	1.26 in x 3.07 in x 3.31 in
Scanning Range	100 yards max
Range Precision	± 1 m
Scanning Angle	N/A
Scanning Rate	100 ms
Interface	RS-232 Serial
Price	$485.00 (April 2006)

Table 2.7: Opti-Logic RS100 [13]

As can be seen from Table 2.7, this laser range finder is a completely different sensor then the SICK LRF. The first noticeable difference is the size. The RS100 is about the quarter of the size of the SICK LRF meaning that this laser range finder can be attached to smaller robots. The next difference is that the RS100 has a longer range then the SICK

LRF. This LRF can detect objects from 100 yards away, but there is a drawback to this range, the range precision is not as accurate as the SICK LRF. The major difference between the RS100 and the SICK LRF is the fact that the RS100 does not have a scanning angle. This means that the RS100 can only detect objects where the laser is facing; it cannot scan over an area automatically like the SICK LRF. A way to get around this would be to put the RS100 on a motor and rotate it in order to sweep over an angle. As shown in the above comparisons the laser range finder would be the best sensor to use if we wanted to acquire a targets distance. They would not be the best sensor if we wanted to use a cheap sensor. These LRF sensors cost considerably more then the vision sensors. Since the goal of this thesis is to develop a target acquisition algorithm using a cheap sensor we will use a CCD camera as the primary sensor.

Chapter 3

Robot Control Techniques

In this chapter different types of robot control laws will be developed and discussed. This chapter will include open loop control as well as closed loop control algorithms.

3.1. Robot Description

The robot that will be used in this thesis is Johnny 5. Johnny 5 is a robot designed for the Intelligent Ground Vehicle competition. A picture of Johnny 5 is shown below in Figure 3.1.

Figure 3.1: Picture of Johnny 5

As shown in the photo, Johnny 5 is a differential drive robot. This means that the robot is controlled and moves by just two wheels. The reason why this type of robot is advantageous for this thesis is that the control for this type of robot is not difficult. Since it is a differential drive robot, it can turn about its center point and basically make any type of turns.

Johnny 5 is also used because it is a durable robot. Johnny 5 has two servo motors that are attached to each wheel. It also has an independent motor controller for these motors as well as 16000 pulses per revolution encoders that are attached to each motor shaft. Finally, a 10:1 gear head is attached to each motor. Figure 3.2 shows a striped down version of Johnny 5 as well as where some of its components are located.

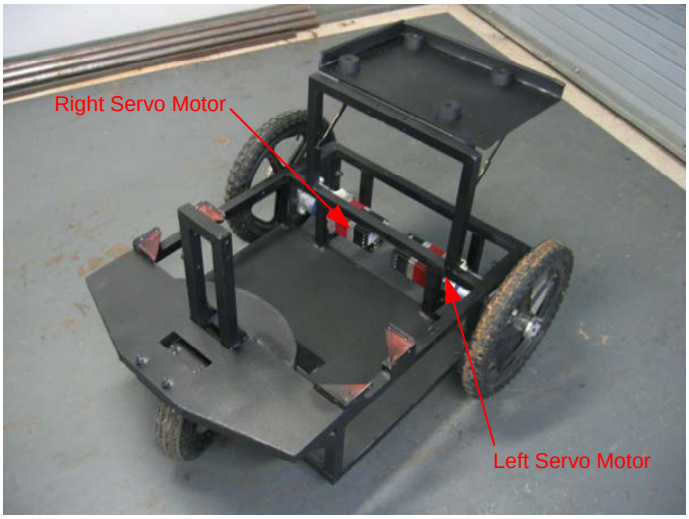

Figure 3.2: Stripped Down Version of Johnny 5

Finally the camera will be mounted on the mast of the vehicle. Since the camera will be mounted at a height of about five feet it will have a greater field of view.

3.2. Programming Language Description

Before we start describing the process of developing a controller and target extraction algorithm we need to briefly describe the programming language that we will be using. The type of programming language that we will be using in this thesis is National Instruments Labview. Labview is a set of graphical development software tools. What makes Labview different from other text based programming languages like C++ is the fact that Labview programs look like block diagrams. Below in Figure 3.3 is a screenshot of an example program.

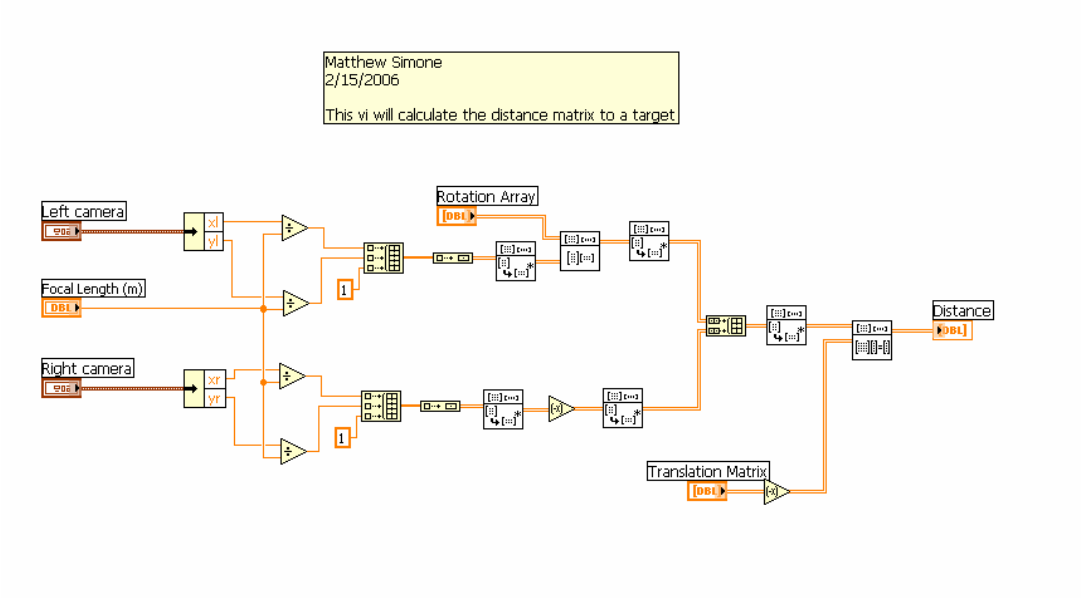

Figure 3.3: Sample Labview Program Screenshot

Labview also has impressive graphical user interface (GUI) properties as well. The front panel is what national instruments call the GUI. Below in
Figure 3.4 is a screenshot of the front panel/GUI.

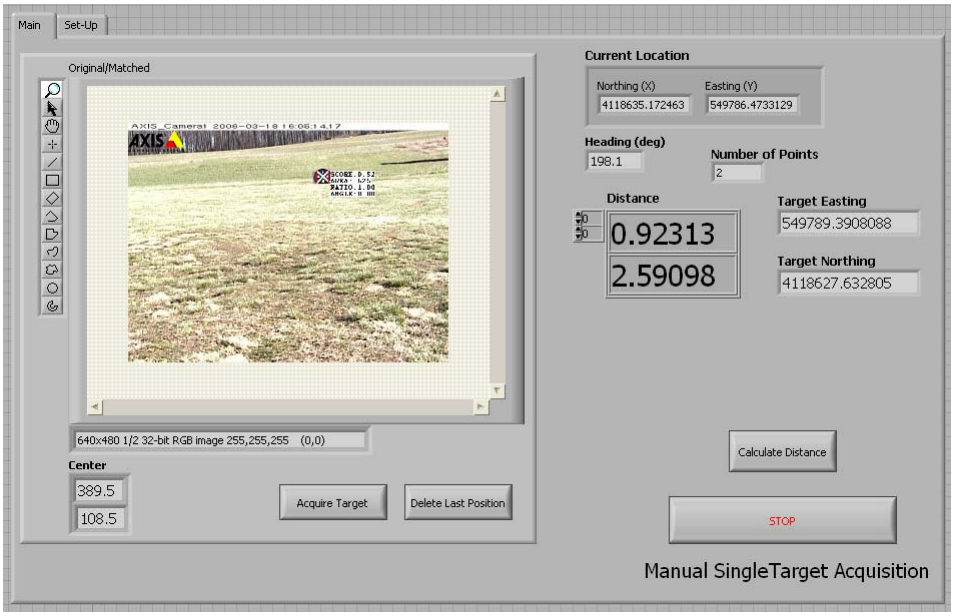

Figure 3.4: Sample Labview Program Front Panel

This type of programming makes it very easy for beginning programmers to develop very advanced programs. Labview also has embedded functions within it that will reduce

development time. One of the main reasons why we choose Labview for this thesis is because there are embedded functions within Labview itself that will let us easily set-up and communicate with serial ports. Labview also has some vision processing functions in it, thus reducing the development time and letting the programmer focus more time on algorithms.

3.3. Robot Modeling

The mathematical model that will be used for developing the robots controller is a kinematics model. A kinematics model deals with the relationship between control parameters and the behavior of a system in state space [14]. A kinematics model is also one where only a robots state in the world is used, not its dynamics. This type of model for Johnny 5 is shown below in Figure 3.5.

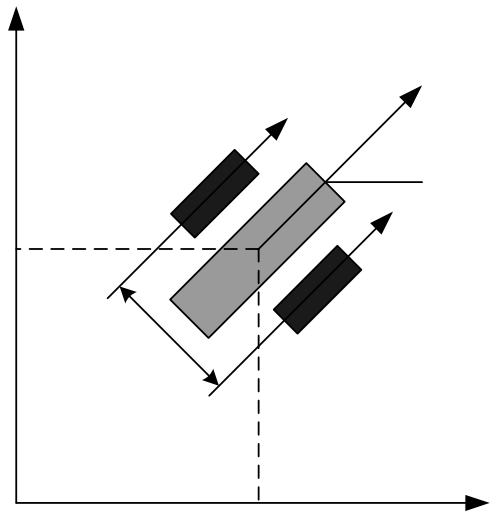

Figure 3.5: Johnny 5's Kinematics Model

Where V_l and V_r is the left and right wheel's velocity and V velocity of the robots center point. X and Y is the robot's location in the world coordinate frame while θ is the robot's heading in the real world coordinate frame where counterclockwise is positive angles. Finally, l is the distance between the robot's two wheels. With the forward

kinematics model shown above, we can now derive forward kinematics equations for a differential drive robot.

$$V = \frac{V_r + V_l}{2} \tag{3.1}$$

$$\dot{\theta} = \frac{V_r - V_l}{l} \tag{3.2}$$

Equations (3.1) and (3.2) can then be used to find the state-space kinematics model of a differential drive robot.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\cos\theta \\ \frac{1}{2}\sin\theta \\ \frac{1}{2l} \end{bmatrix} V_r + \begin{bmatrix} \frac{1}{2}\cos\theta \\ \frac{1}{2}\sin\theta \\ -\frac{1}{2l} \end{bmatrix} V_l \tag{3.3}$$

Equation (3.3) can then be used to calculate the location of a differential drive robot in the world plane at any time.

3.4. Open Loop Control

After the differential drive robot is modeled, a control scheme has to be developed. The first control scheme that we will use to control the robot in the world plane is an open loop controller. Open loop control means that a command to move the robot a set distance is given and we will assume that the robot completes this task. Therefore there is no feedback from position sensors such as GPS or heading sensors like compasses. This type of control can also be called dead reckoning control since simple equations can be used to compute the momentary position of the robot relative to a know starting position [15].

For this control scheme we will use an incremental optical encoder to calculate the robots heading and position in the world. An encoder is a device that is used to measure the

rotational velocity of a motor. An incremental optical encoder is basically an instrumented mechanical light chopper that produces a certain square wave pulses for each shaft rotation [15]. These pulses are then added in order to find the velocity of a motor shaft. For this thesis the encoder that will be used is a 16000 pulse per revolution encoder with a 10 to 1 gear head.

With the two wheels encoder increment values N_L and N_R we can calculate the robots relative position and heading. A scale factor first has to be found for the encoders [15].

$$c_m = \frac{\pi D_n}{nC_e} \tag{3.4}$$

Where,

c_m = A Conversion Factor that translates encoder pulses to linear wheel displacement.

D_n = Wheel diameter in millimeters.

C_e = Encoder resolution in pulses per revolution.

n = Gear head ratio.

After this conversion factor is found, we can now compute the incremental travel distance of the left and right wheels, $\Delta U_{L,i}$ and $\Delta U_{R,i}$.

$$\Delta U_{L,R,i} = c_m N_{L,R,i} \tag{3.5}$$

Now the incremental travel distance of the robots center point, ΔU_i can be calculated.

$$\Delta U_i = \frac{(\Delta U_{R,i} + \Delta U_{L,i})}{2} \tag{3.6}$$

The robots incremental heading change $\Delta\theta_i$ can now be computed.

$$\Delta\theta_i = \frac{(\Delta U_{R,i} - \Delta U_{L,i})}{l} \tag{3.7}$$

The robot's new heading θ_i can be computed from

$$\theta_i = \theta_{i-1} + \Delta\theta_i \tag{3.8}$$

Finally, the robots relative position of the center point can be computed from

$$x_i = x_{i-1} + \Delta U_i \cos\theta_i \tag{3.9}$$

$$y_i = y_{i-1} + \Delta U_i \sin\theta_i \tag{3.10}$$

This type of control can be accurate under certain types of conditions but under our operating conditions it has some limitations. The first limitation is that the error between the real distance traveled and the measured distance can be large. The error can be large because on bumpy ground like grass the wheels can slip thus causing errors in our encoder measured distances. Because of this, a new method of control will have to be developed that can integrate GPS as well as compass readings with the encoder measurements to calculate the most accurate position.

3.5. Closed Loop Control with Encoders

In order to accomplish this problem of target acquisition, a precise controller has to be developed that will be able to control the robot along a path. For this problem the path will be a path between two waypoints. The first type of control will use encoder values and dead reckoning in order to control the robot to a specific waypoint. A diagram for this type of control is shown below in Figure 3.6.

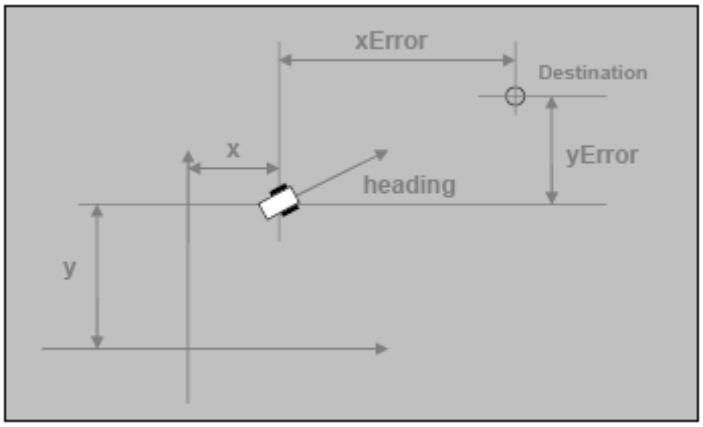

Figure 3.6: Waypoint Navigation Description [16]

In order to drive to the destination we first need to find the heading to the destination. This can be computed from the simple equations shown below.

$$X_{error} = Destination_x - X \tag{3.11}$$

$$Y_{error} = Destination_y - Y \tag{3.12}$$

Then, from equations (3.11) and (3.12).

$$Heading_{waypoint} = ATAN2(X_{error} / Y_{error}) \tag{3.13}$$

This takes into account that the positive x-axis will be the zero degree mark and the angles will range from -180° to 180°. After we calculated the heading to the target we will now need to steer the robot to that desired heading. The type of control that we will use is a proportional controller and it will control the robots heading. This type of controller uses the error between the robots current heading and desired heading as the input. A diagram is shown below in Figure 3.7.

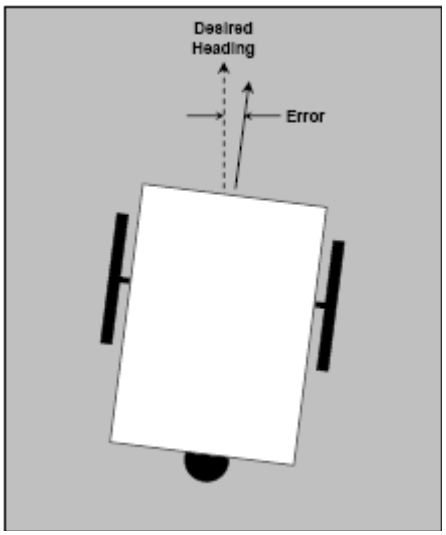

Figure 3.7: Heading Error [16]

Therefore the robot will turn harder towards the target the greater the heading error is. The equations that were used to command the left and right wheel speeds are shown below in equations (3.14),(3.15) and (3.16) [16].

$$Left\,Wheel\,Power = Drive\,Speed - Differential \tag{3.14}$$

$$Right\,Wheel\,Speed = Drive\,Speed + Differential \tag{3.15}$$

$$Differential = Gain * Heading\ \ Error \tag{3.16}$$

One waypoint test run was run in order to check the accuracy of the proposed waypoint navigation strategy. A square test route was set up where each waypoint was six meters away. Shown below in Figure 3.8 is a plot of the test run of the vehicle. The blue circles are the waypoints while the red line is the route Johnny 5 took to complete this task.

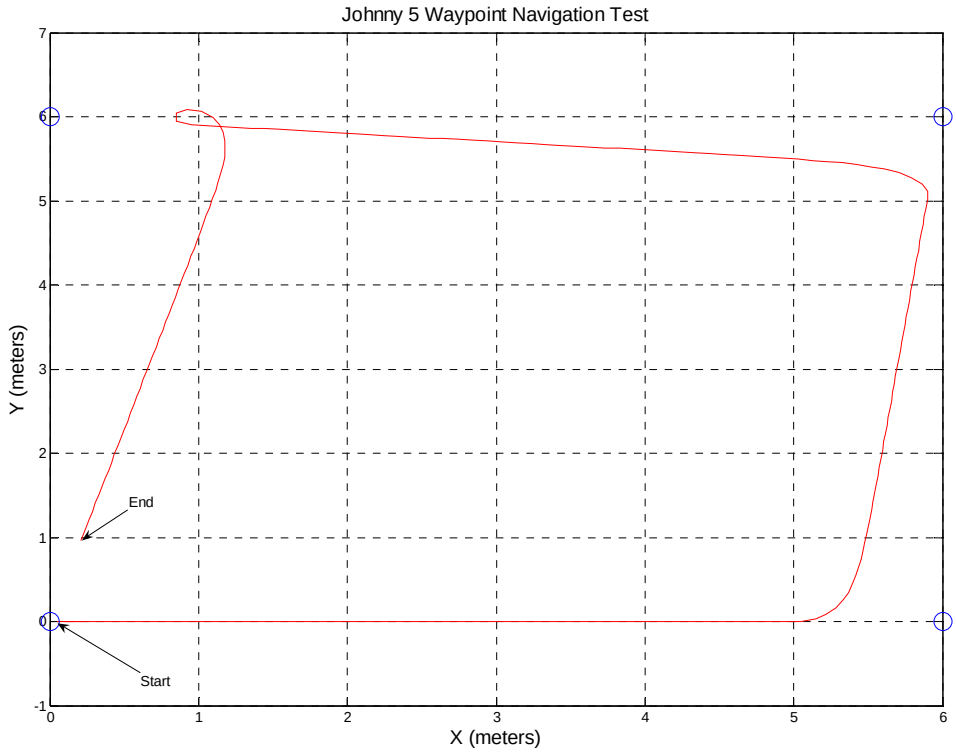

Figure 3.8: Johnny 5 Encoder Waypoint Navigation Test

There are a couple of interesting points to make about this plot. The first point, which is shown on this plot, is that the vehicle does not reach the waypoint. The reason why this happens is because we have to put a threshold distance to the waypoint that the vehicle has to reach in order to start towards the next one. For this specific test the threshold is one meter. As shown on the plot, Johnny 5 does indeed steer and travel to these waypoints fairly accurately and with little steering variation. The next interesting observation is that the vehicle turns in the opposite direction then what is expected when it is traveling between waypoint three and four. The reason why this occurs is because for this specific route the calculation that is used to find the heading tells the vehicle to turn in that direction.

We also plotted the robots heading versus time in order to see the characteristics of this proportional controller with a specific gain. This is shown below in Figure 3.9.

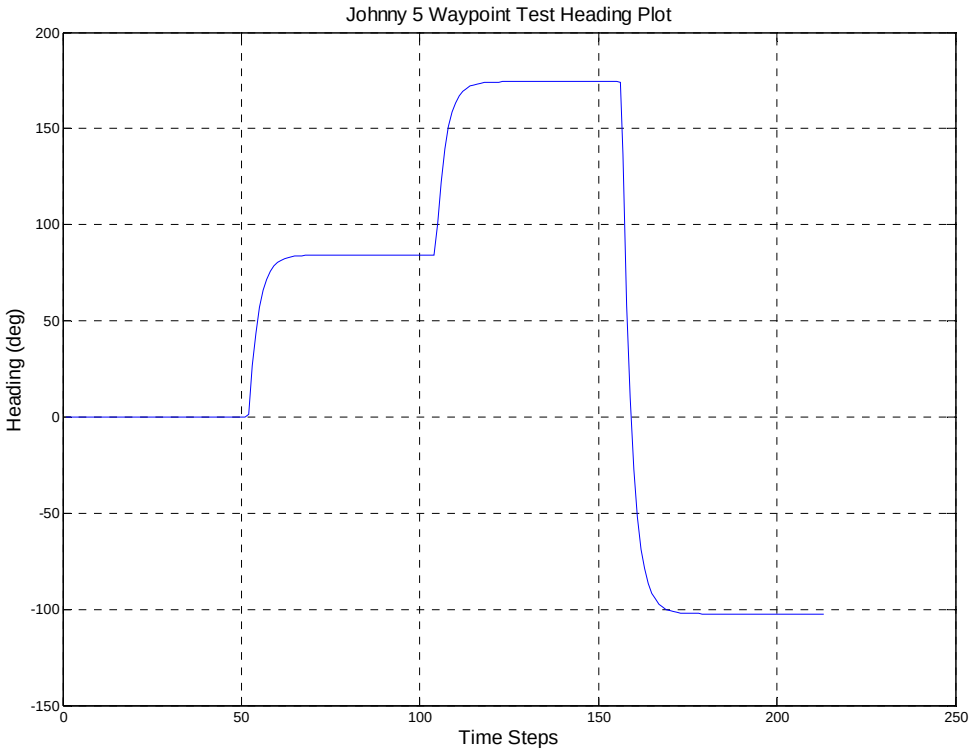

Figure 3.9: Johnny 5 Waypoint Test Heading Plot

As shown in Figure 3.9, this controller does not overshoot the target heading. The rise time of this controller and its tracking ability are all within reasonable tolerances for waypoint navigation.

As shown by this test, waypoint navigation using encoders works fairly well and would suffice under certain circumstances. These circumstances would be for waypoint routes that are not to long. The reason for this is because encoders build up errors from wheel slip and outside factors as the vehicle moves farther along its course.

3.6. Closed Loop Control with GPS

The next type of control that will be used is closed loop control with GPS. This type of controller is the same type as the encoder closed loop controller except that instead of using encoders we will use a Novatel ProPack GPS and a PNI digital compass on Johnny

5 to obtain the vehicles location and heading. All of the other control equations from above were applied to this controller.

3.6.1. GPS System Characteristics

Before we can implement this type of controller we first need to explain the concept of GPS. GPS which is short for Global Positioning System is a satellite system made up of 24 satellites placed into orbit by the department of defense [17].

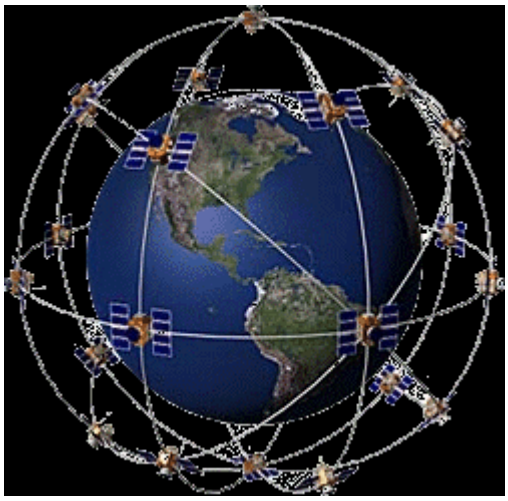

Figure 3.10: GPS Constellation

GPS calculates the position of a receiver based on the theory of triangulation. Each individual satellite has very accurate clocks on them and sends out signals that contain the time the signal was sent out to receivers. The receiver then calculates the distance to the satellite by knowing how long the signal took to reach the receiver. If a receiver has a lock on four or more satellites it can accurately calculate latitude, longitude and altitude position anywhere in the world.
The type of GPS receiver that we will use is called the Novatel ProPack. This GPS receiver is shown below in Figure 3.11.

Figure 3.11: Novatel ProPack GPS Receiver

The ProPack GPS receiver can run in two modes. The first mode is called Single Point mode and the second mode is called OmniSTAR mode. Each type of mode has its own accuracy associated with it. Below in Table 3.1 is a table showing a comparison of the two types of best case mode accuracies.

Mode	Accuracy (m)
Single Point	1.8 m
OmniSTAR	0.10 m

Table 3.1: Comparison of Mode Accuracies

As shown above in the table the OmniSTAR mode is the most accurate mode out of the two. The reason why it is the most accurate is because it is a differential based receiving mode. Differential GPS uses the fact that GPS errors are widespread meaning that an error in one location is about the same as the error in another location. A differential GPS system then uses a base station whose position is accurately known and records that error and sends it out to the receivers who then can account for that error in the measurement of its location. After this simple comparison it is clearly obvious that we will use the OmniSTAR mode for GPS waypoint navigation.

In order to find out if the OmniSTAR accuracy is as good as is stated two simple tests have to be completed. The first one is a data collection test at one location and the other is a simple waypoint navigation test. The single location test will tell us the standard deviation and errors of the received GPS coordinates. Shown below in Figure 3.12 is a scatter plot of received OmniSTAR GPS locations when Johnny 5 is sitting at a single point. It is shown from this plot that most of the data is within a band of about two meters. This error is consistent with the error that was shown in the ProPack GPS datasheets. The standard deviation for the easting is 0.4381 and the standard deviation for the northing is 0.3848.

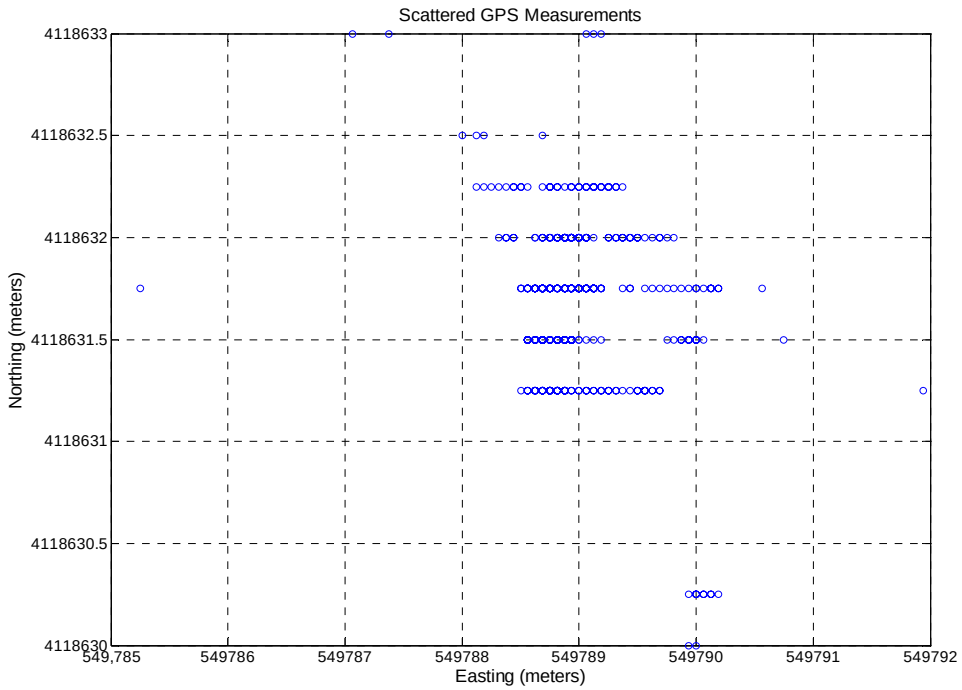

Figure 3.12: Scattered GPS Measurements

For the simple waypoint navigation test a set of waypoints were picked out fairly far away form each other in order to test out the straight line position error from the GPS. Shown below in Figure 3.13 is a plot the path traveled by Johnny 5 in OmniSTAR mode.

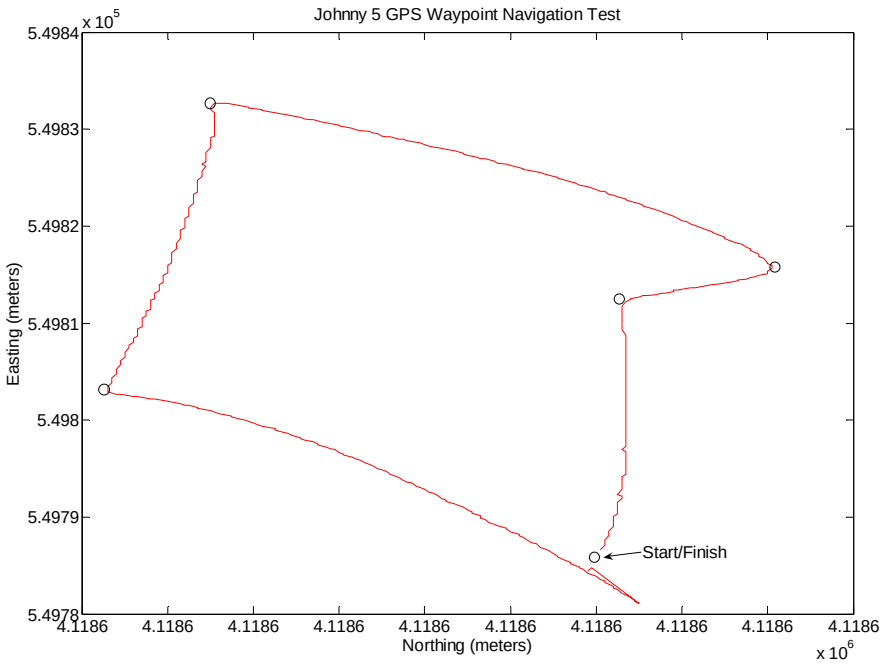

Figure 3.13: Johnny 5 GPS Waypoint Navigation Test

There are a couple of interesting facts about this plot. The first major fact is that the GPS is very accurate. This is shown from the fact that there is small error in the GPS when the vehicle is navigating the waypoints. The next interesting fact is that the vehicle very accurately heads and reaches the waypoints. These two facts should be expected since the OmniSTAR GPS mode has such a low error. Another fact that was observed but not displayed is that when the gain for the proportional controller is picked to be fairly large the robot does not traverse the waypoints this accurately. There is a lot of oscillation in the robots heading when this occurred which can be expected from a proportional controller when the gain is set high. The heading from the compass now has to be plotted for this experiment. This is shown below in Figure 3.14.

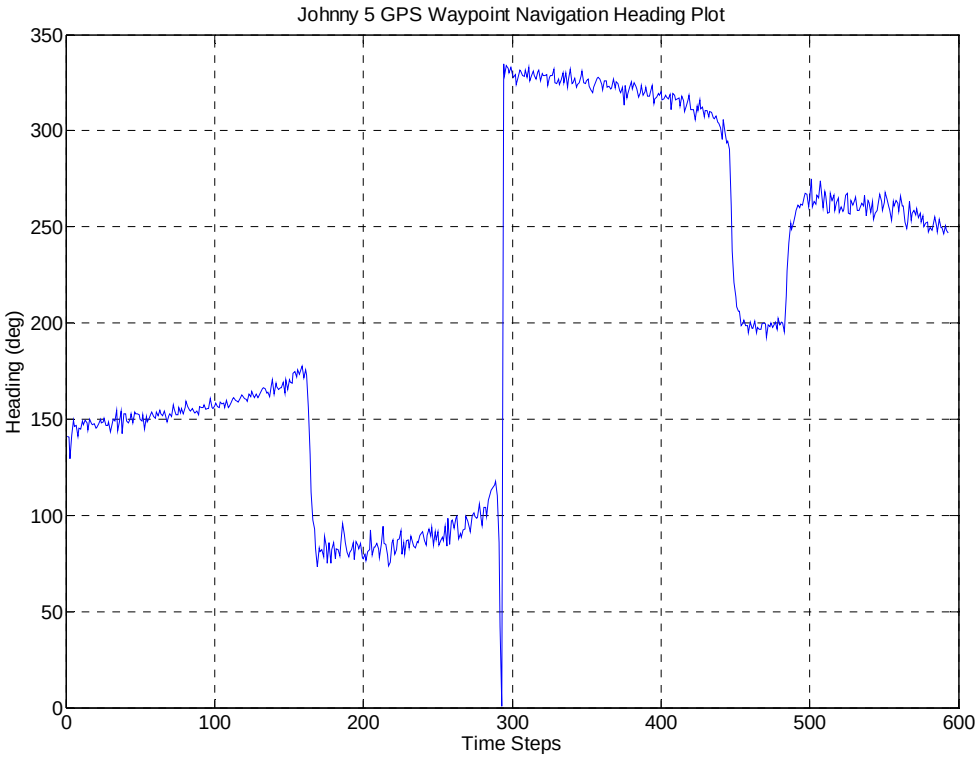

Figure 3.14: Johnny 5 GPS Waypoint Navigation Heading Plot

We can see in this plot that the waypoint navigation with the digital compass performs fairly well. Johnny 5 does not overshoot the waypoints. We can also see that this heading measurement is not as clean as the encoder heading measurements, which can be expected. The reason why it can be expected is because the digital compass is susceptible to noise. We can easily fix this problem with a simple digital filter if needed.

The waypoint navigation algorithm with OnmiSTAR GPS and a digital compass is shown to accurately navigate waypoints very accurately. This mode is an attractive mode since it does not put to much computational stress on the system since the OnmiSTAR processing is done on board of the Novatel ProPack.

3.7. Closed Loop Control with Sensor Fusion

The next way to accurately control the robot to a set of waypoints is closed loop control with sensor fusion. We will use an Extended Kalman filter for this type of fusion to only calculate the exact position of the robot at any given time. We are primary interested in fusing noisy sensors together in order to find a best estimate of the position of the vehicle.

A Extended Kalman Filter is used to produce the optimal state estimate of a system given a set of measurements and relationships between measurements and the state vector [18]. The Extended Kalman filter approach is to apply the standard Kalman filter to nonlinear systems with additive white noise by continually updating a linearization around the previous state estimate. For this specific waypoint navigation problem we will fuse inertial sensor data together with GPS measurements in order to find the best estimate position. Shown below in Figure 3.15 is a simple block diagram for our overall Extended Kalman filtering system.

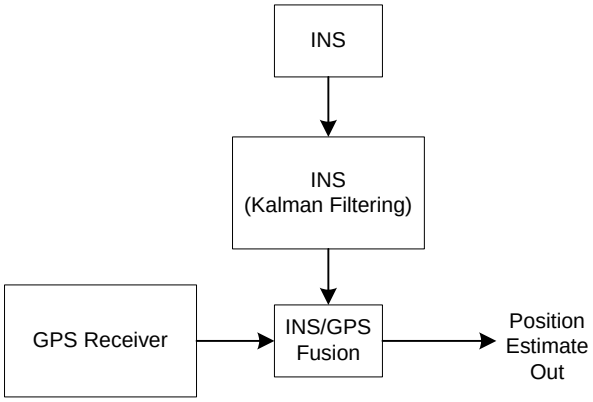

Figure 3.15: Extended Kalman Filtering System

The INS Extended Kalman filtering occurs at 50 Hz while the INS/GPS fusion occurs at 2 Hz.

In order to develop the Extended Kalman filter we first need to mathematically model the nonlinear system with noise.

$$X(k+1) = f\left[k, X(k), u(k)\right] + w(k) \tag{3.17}$$

$$Y(k) = h[k, X(k)] + v(k) \tag{3.18}$$

Where $X(k)$ is the state vector, $u(k)$ is the external input vector and $Y(k)$ is the measurement vector of the system. The noise of the system is modeled as $w(k)$ and $v(k)$ which are assumed to be zero mean white Gaussian noise.

For the Extended Kalman filter of Johnny 5, we will use an eight state Extended Kalman filter where the measurements will be the digital compass data (θ) and the distance traveled (Δu) in meters from the encoders. The state vector and measurement vectors are shown below in equations (3.19) and (3.20).

$$X(k) = \begin{bmatrix} x(meters) \\ y(meters) \\ \theta(\text{deg}) \\ \dot{\theta}(\text{deg/sec}) \\ \ddot{\theta}(\text{deg/sec}^2) \\ \Delta u(meters) \\ \Delta \dot{u}(meters/\text{sec}) \\ \Delta \ddot{u}(meters/\text{sec}^2) \end{bmatrix} \tag{3.19}$$

$$Y(k) = \begin{bmatrix} \theta(\text{deg}) \\ \Delta u(meters) \end{bmatrix} \tag{3.20}$$

We are now ready to define the Extended Kalman filtering algorithm from [19]. We first note that the original Kalman filtering process was designed to estimate the state vector in a linear model. For nonlinear models, a linear Taylor approximation of the system is computed at every time step. The measurement model $h[k, x(k)]$ is linearized about the current predicted state. The plant model $f[k, x(k), u(k)]$ is linearized about the current estimated state vector.

We also note that this Extended Kalman filtering algorithm consists of an estimation phase and a correction phase. The estimation phase will run at about 50 Hz and the correction phase will run at 2 Hz or the GPS update rate. We are not worried about the development and inner working of the Extended Kalman filter we are just worried about

the implementation of it. There are four steps in this algorithm [19]. We first need to predict the covariance matrix of the states.

$$P(k+1|k) = f_x(k)P(k|k)f_x^T(k) + Q \quad (3.21)$$

Where $f_x(k)$ is the Jacobian of $f[k, \hat{x}(k|k), u(k)]$ with respect to the state x.
Next we need to calculate the Extended Kalman gain matrix.

$$K(k+1) = P(k+1|k)h_x^T(k+1)\left[h(k+1)P(k+1|k)h(k+1)^T + R\right]^{-1} \quad (3.22)$$

Third is the calculation of the state estimation.

$$\hat{X}(k+1|k+1) = \hat{x}(k+1|k) + K(k+1)[y(k+1) - h[k, \hat{x}(k+1|k)]] \quad (3.23)$$

The final calculation is the update of the covariance matrix of the states.

$$P(k+1|k+1) = [I - K(k+1)h_x(k+1)]P(k+1|k) \quad (3.24)$$

In the above set of equations $K(k+1)$ is the filter gain, $\hat{X}(k+1|k+1)$ is the updated state estimate and $P(k+1|k+1)$ is the updated state covariance. Finally $h_x(k)$ is the Jacobian of $h[k, x(k)$ with respect to the state x. There are a couple of matrixes that have to be initialized from collected data. These are the R and the Q matrices. The R matrix is the measurement noise that contains the expected covariance of the measurement data while the Q matrix is the state noise matrix which contains the expected covariance of the states [18].
The above process is the estimation part of the Extended Kalman filter, there also has to be a correction part. The correction part happens every time a new GPS point is received. The GPS point is then fused with the estimated location. This is done using equations (3.25), (3.26) and (3.27).

$$Lat_{est} = \psi_{pos} Lat_{GPS} + (1 - \psi_{pos}) Lat_{filt} \quad (3.25)$$

$$Long_{est} = \psi_{pos} Long_{GPS} + (1 - \psi_{pos}) Long_{filt} \quad (3.26)$$

$$Heading_{est} = \psi_{head} Heading_{compass} + (1 - \psi_{head}) Heading_{filt} \quad (3.27)$$

For this thesis the OnmiSTAR GPS waypoint navigation method will be used instead of the Extended Kalman filtering method. The reason being is that the OmniSTAR method can accurately calculate the position of Johnny 5 within an acceptable error range. The reason why the Extended Kalman filtering waypoint navigation will not be used is because the error of the encoders can be large depending on what type of turf Johnny 5 is running on and Johnny 5 will be running on grass with holes and bumps. Through observation it was shown that there was significant slip in the wheels when traveling over this type of terrain.

Chapter 4

Vision Processing Techniques

This chapter will describe the different types of algorithms that can be used to find a target in an image.

4.1. Target Extraction Techniques

The target that will be used throughout this thesis will be the International Aerial Robotics Competition target. The target is shown below in Figure 4.1.

Figure 4.1: IARC Target [20]

This target is circular with a diameter of one meter. It is black with two white lines crossing in the center. Two image processing techniques will be tested and compared to see how well they can find the target in an image. The first technique that will be discussed is the Hough transform algorithm. The second technique is pattern matching

using cross-correlation. After these two techniques are explained they will be compared and the most robust method will be used for the rest of this thesis.

4.1.1. Hough Transform

The first algorithm that will be explained is the Hough transform. If the Hough transform is to be used one assumption has to be made, that the target will be the only complete circular object in the area. With this assumption, we can then choose the Hough transform method to detect this target in the environment. The Hough transform is used to detect circles in a digitized image by choosing the center of a circle and it's radius as parameters in the transform [21] . In order to reduce the computational time of the Hough transform, image edge detection will first be used. Edge detection is a process of detecting a sudden change in image intensity, which is most likely an edge of an object [22]. This can be completed with image masks, but for this thesis the Canny edge detection algorithm will be used. The reason why the Canny edge detection method will be used is because it is an optimal edge detector that works even when there is noise in the image.

The Canny edge detector works by first smoothing the image by a Gaussian convolution. Then an edge detector mask operation is used to give ridges in a gradient magnitude image. Next, the algorithm tracks the top of these ridges and puts every pixel that is not at the top of these ridges to a value of zero, while the pixels at these ridges stay at a value of one. Finally, the algorithm exhibits hysteresis that is controlled by two variables, so it ensures that the edges are not broken up into fragments [23]. The output of this process is a binary array of pixels that can then be run through the Hough transform. Shown below in Figure 4.2 is the output of the IARC target after it is run through the canny edge detector algorithm.

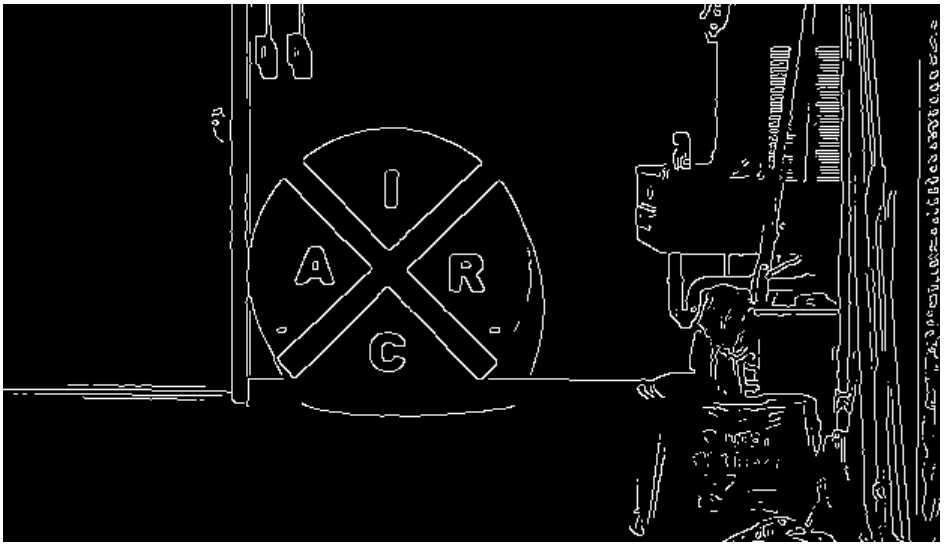

Figure 4.2: Canny edge image of IARC target

A Hough transform can now be performed since the image is in binary form. The Hough transform works by taking in the binary image and converting it to a Hough space where a circles position and radius on the image can be extracted. In order to accomplish this task, the Hough transform space, $H(Sx, Sy, r)$ first has to be defined and set to zero. For the circle detection, this space has to be three dimensions, which is shown below in Figure 4.3.

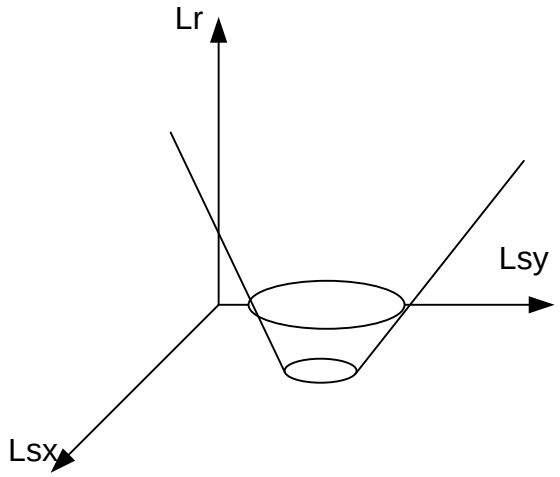

Figure 4.3: Three dimensional Hough Space [24]

The Hough space is bounded by three parameters, Lr which is the search radius bounds of the circles, Lsx which is the length of the column size of the image in pixels, and Lsy is the length of the row size in pixels. This space can also be thought of an accumulator array.

Next, the transformation between the image and the Hough space has to be accomplished. Equations for a circle have to be used for this task. They are shown below in equations (4.1) and (4.2) , where $0^\circ \le \theta \le 360^\circ$.

$$Sx = x_p + r\cos(\theta) \tag{4.1}$$

$$Sy = y_p + r\sin(\theta) \tag{4.2}$$

Equations (4.1) and (4.2) are used to calculate a circular array of binary data for specified ranges of r for each white pixel in the image. This process can be computationally intensive since it has to search for a range of radius. To reduce this computation time the θ and r resolution can be set to a greater value.

With these values a vote of 1 is then entered into the Hough space as shown below in equation [24].

$$H\ (Sx\ ,\ Sy\ ,\ r\ )\ =\ H\ (Sx\ ,\ Sy\ ,\ r\ )\ +\ 1 \tag{4.3}$$

Finally, the Hough space is searched for local maxima, this maxima is the three dimensional values of the circle. They are the column and row position of the circle as well as the detected circles radius. Shown below in Figure 4.4 is the resultant circle plotted over the edge detected image.

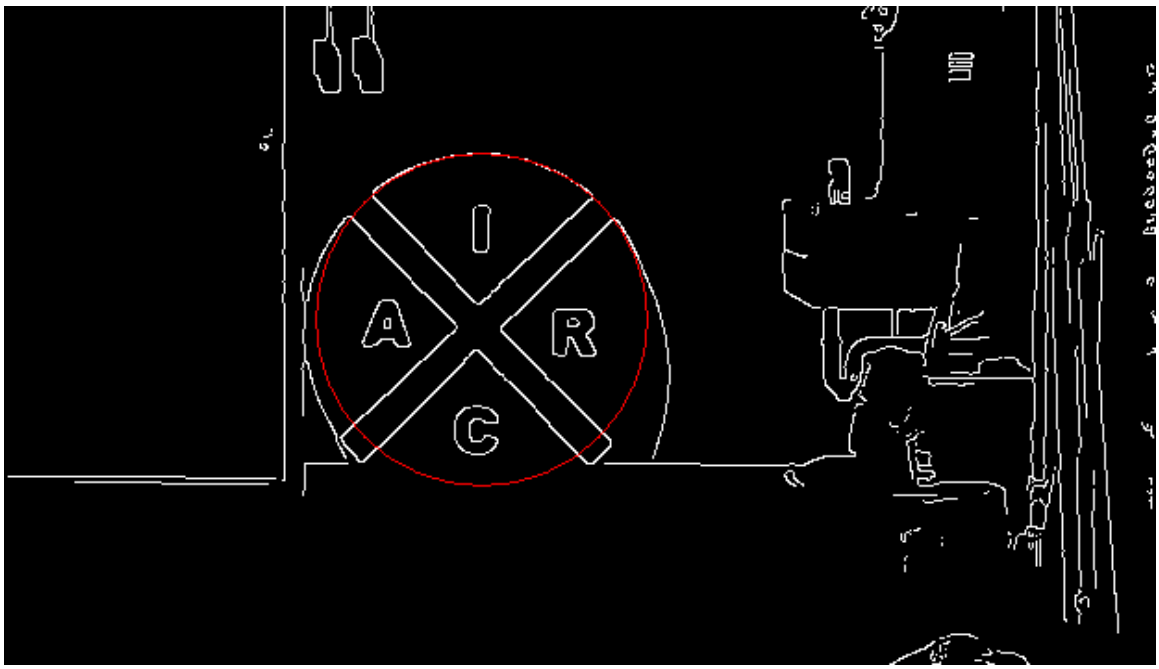

Figure 4.4: Hough Transform Result

As shown above the Hough transform does a good job at detecting where the circle is in the image and its radius, which is shown by the red circle.

4.1.2. Pattern Matching

The next method that can be used for detecting the IARC target is pattern matching with cross-correlation. This algorithm uses a saved template of the target in order to correlate the acquired image with. This algorithm requires many steps in order to complete. Shown below in Figure 4.5 is a block diagram of the pattern matching algorithm.

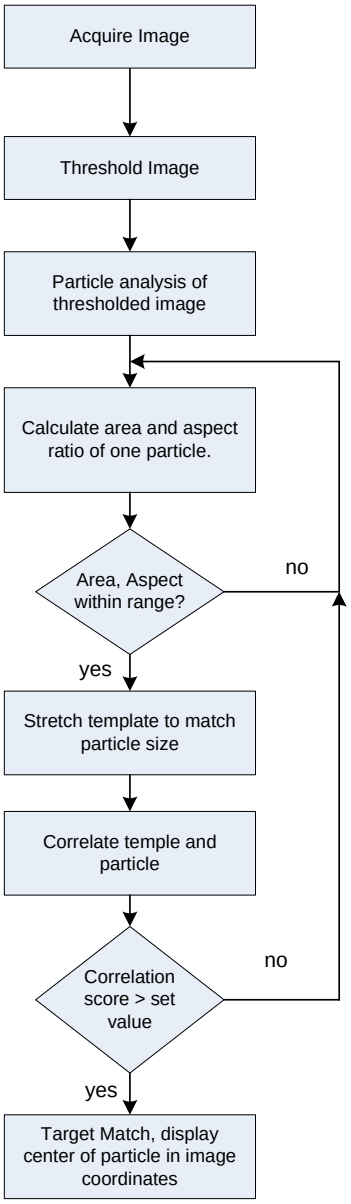

Figure 4.5: Block Diagram of the Pattern Matching Algorithm

The first task that has to be after the image is captured is a task called image thresholding. According to [22] an image threshold is when every pixel in an image is compared to a certain threshold value T. The one assumption that will be made is that the image will be a grayscale image. Equation (4.4), shown below states this rule in mathematical terms.

$$I_{NEW}(row, column) = \begin{cases} 1 & if\ I(r,c) \le T \\ 0 & if\ I(r,c) > T \end{cases} \qquad (4.4)$$

The result of (4.4) is a binary image. A thresholded image of Figure 4.1 is shown below in Figure 4.6.

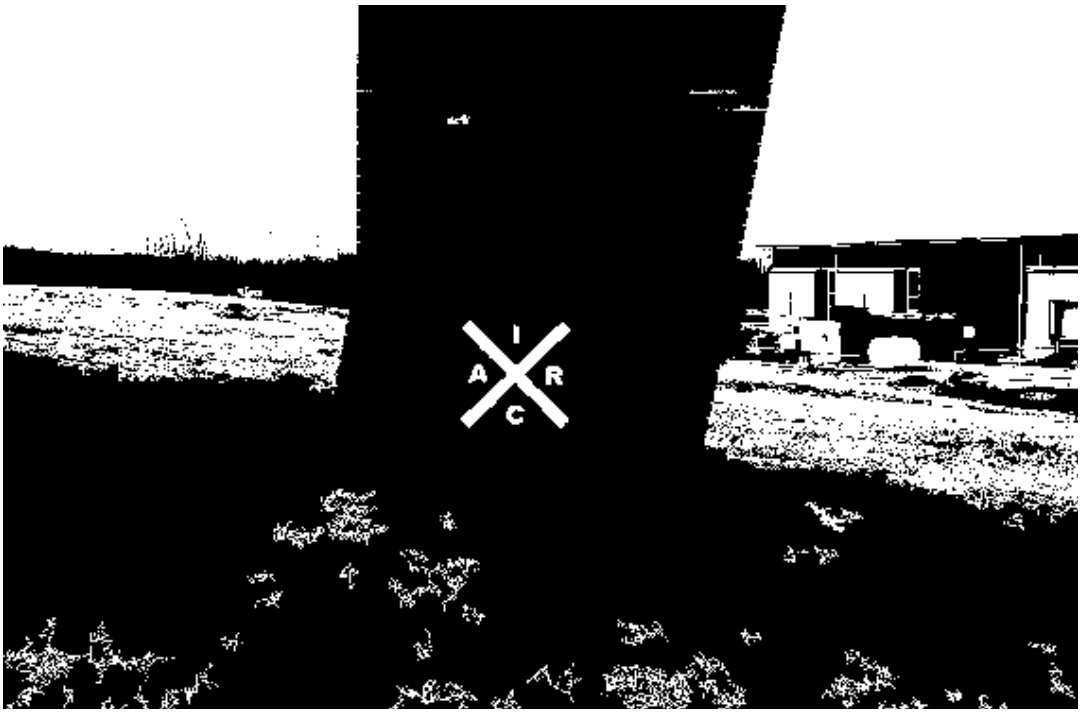

Figure 4.6: Target Image Threshold

The next task that has to be completed is a task that we will call particle analysis. Particle analysis means that the algorithm will search through the thresholded image and pick out all of the individual regions in it. It will also extract features such as width and height of the region. For example, a particle extraction algorithm will search through Figure 4.6, and pick out all of the areas of concentrated white pixels. It will also return the features of the concentrated areas.

In order to understand how particle extraction works we first need to define what a region is. A region is a connected portion of an image [22]. With this information we can find a fast method to find regions with one assumption, there are no holes present in any region. We start by searching through the image and finding the number of internal (I) and external (E) corners. When this information is acquired we can use equation (4.5) to calculate the number of regions in the image.

$$regions = (E - I)/4 \qquad (4.5)$$

After the number of regions is computed, a process called region labeling can then be used to label all of the separate regions in the thresholded image. The length and width of a regions bounded box can be computed and from this the regions bounding box area can be computed. The aspect ratio (AR) can be computed using equation (4.6) where w is the width of the regions bounding box and h is the height of the regions bounding box.

$$AR = \frac{w}{h} \tag{4.6}$$

Finally, the center of the region can be determined. This is simply calculated by taking the halfway point in the length and width.

We can now start stepping through all of the detected regions and proceed with the main part of the algorithm. After the area and aspect ration are computed a filter can be used to filter out a region whose areas and aspect ratios are not within a set range. If they are within a certain range the template will then be used as a comparison. This happens by first taking the template and scaling it to the same size as the selected region. Now that the template and region are the same size a cross correlation can be performed on the two images. Cross correlation for pattern matching is a standard method of estimating the degree to which two images are correlated [25]. The value that will be returned is a number between -1 and 1, where values close to 1 mean that the two images are highly correlated. Equation (4.7) shown below is the equation that will be used to calculate the correlation between the two images.

$$r = \frac{\sum_i [(x(i) - \mu_x)(y(i) - \mu_y)]}{\sqrt{\sum_i (x(i) - \mu_x)^2} \sqrt{\sum_i (y(i) - \mu_y)^2}} \tag{4.7}$$

Where x and y are the input image and template image respectively. The means μ are also used in equation (4.7). After the cross correlation coefficient is calculated it is compared to a set value, if it is over that value we can assume that the two images match and the matched regions bounding box is overlaid onto the original image and its center is displayed. Below in Figure 4.7 is the matched regions bounding box overlaid onto the

original image showing that there was a correct match of the template, thus picking out the target in the image. In this figure the score is the result of correlation and the other values have units in pixels.

Figure 4.7: Matched Region of IARC Target

4.2. Comparison of the Two Techniques

We now need to decide which method will be used for target extraction. In order to decide which technique will be used, an experiment will be set-up. In this experiment, the target will be set-up and pictures of the target will be taken from straight on, to the left and to the right of the target for various distances. The distances that are used are 20, 30, 50, 80, and 100 meters. After the pictures are taken, the Hough Transform and Pattern Matching technique are used on the images and the result is if each of the two techniques can positively identify the object and it's location in the image. Below in Table 4.1 are the results of this experiment. What is shown below are the results of the experiment with the images taken straight on.

Image Distance	Hough Transform	Pattern Matching
20 meters	No	Yes
30 meters	No	Yes
50 meters	No	Yes
80 meters	No	Yes
100 meters	No	Yes

Table 4.1: Results of the Two Different Techniques

As shown in Table 4.1, the pattern matching method is the most robust method to use for this target extraction problem. Another comparison variable that was observed supports the use of the pattern matching method. This variable is the run time for each algorithm. What was observed is that the pattern matching method takes about a twentieth of a second to complete while the Hough transform method takes over a second. Since the Hough transform cannot run quickly enough, it cannot run at near real time while the pattern matching method can.

Chapter 5

Target Distance Acquisition Techniques

This chapter will discuss the different types of methods that can be used for stereo vision.

5.1. Camera Calibration

The specific camera that will be used throughout this thesis is the Axis 213 PTX Network Camera. This is shown below in Figure 5.1.

Figure 5.1: Axis 213 PTZ Network Camera [8]

This camera's pixel size is 5.6 um square. The focal length is adjustable and its range is 3.5-91 mm [8]. Since the focal length is adjusted automatically and there is no exact way to determine this, a calibration procedure has to be performed. The first step in this calibration procedure is to take a circular object of know diameter and position this object a set distance away from the camera. A snapshot of the circular object is then taken and it's diameter in pixel size is measured from the snapshot. Using the objects actual

diameter, object diameter in pixels, camera pixel size, and distance away from the target the focal length, f in meters can then be calculated using equation (5.1).

$$f = \frac{z * y' * (5.6x10^{-6})}{y} \qquad (5.1)$$

Where z is the distance from the camera to the object in meters, y' is the diameter of the object in pixels, and y is the actual diameter of the object in meters.

5.2. Target Distance Measurement Methods

5.2.1. Method 1

The first way to locate a target in the world coordinate frame is to use the actual known size of the object and from this calculate its position. The set-up for this test is shown below in Figure 5.2. The camera is located directly in front of the target circle.

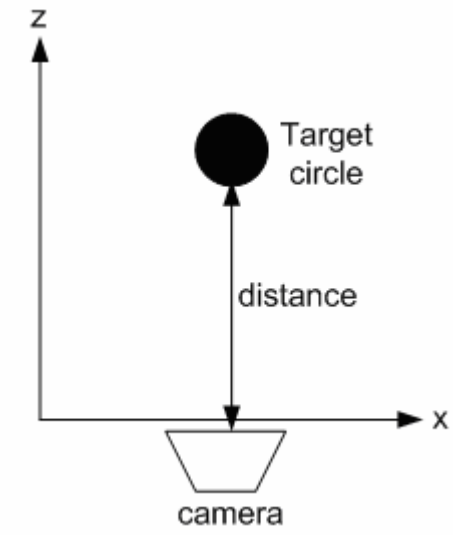

Figure 5.2: Experiment 1 Set-up

Figure 5.3, shown below shows the projection of the circle onto the cameras image plane.

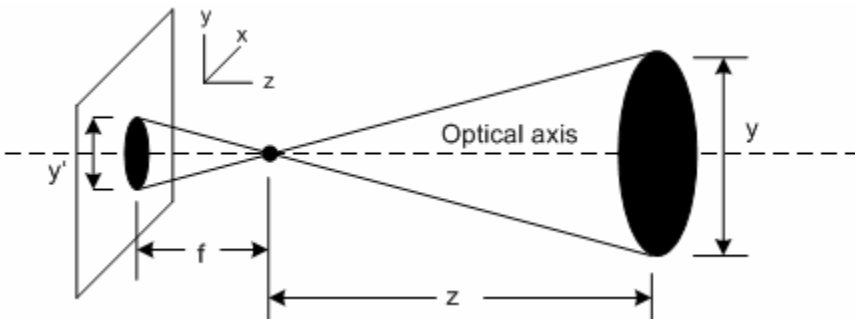

Figure 5.3: Image Plane Projection

From Figure 5.3, f is the focal length of the camera. Z is the distance from the focal point to the object. Y' is the "height" of the object in the image plane and Y is the "height" of the object in the real world "scene." From this figure and using the law of similar triangles equation (5.2) can be found.

$$y' = f\frac{y}{z} \tag{5.2}$$

With equation (5.2) we can now find the distance away from the camera to the object.

$$z = f\frac{y}{y'} \tag{5.3}$$

$$d = f + z \tag{5.4}$$

Where d is the distance from the camera to the object. With this information the target can be located in the world coordinate frame.

5.2.2. Method 2

Parallel image planes

The next method that can be used to find the location of a target in world coordinates is by a process called stereo vision. Stereo vision works by finding the same features in each of the two images, then, by vision processing, measuring the distances to objects containing the features by triangulation. Stereo vision is mostly implemented by using two different cameras that are separated by a known distance. What will be different in this set-up is the fact that one camera will be used. Below, Figure 5.4 is the set-up for this stereo vision problem. Position 1 is the position of the camera when it takes the first image. The camera is then moved to position 2 and another image of the target is taken. F, in this figure, is the focal length of the camera. B is the distance between the two camera positions. Finally, x_l' and x_r' is the position of the target's center point in the camera's image plane with respect to each camera's optical axis.

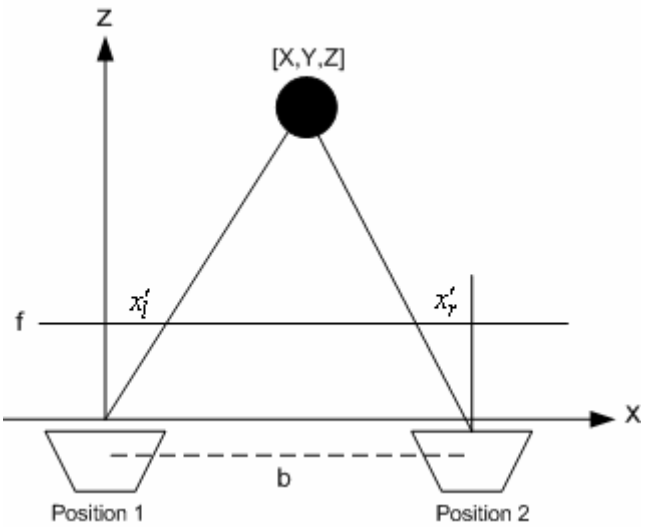

Figure 5.4: Stereo Vision Set-up

Just like in the first experiment, the captured images first have to be run through a threshold algorithm. The next step would be to find the diameter of the target circle. With this diameter, the position of the center point of the target can be found in the image

plane. This information can then be used with equations (5.5), (5.6) and (5.7) shown below to find the targets [X, Y, Z] position in the real world [22].

$$Z = \frac{bf}{(x_l' - x_r')} \tag{5.5}$$

$$X = x_l' \frac{Z}{f} \tag{5.6}$$

$$Y = y_l' \frac{Z}{f} \tag{5.7}$$

With this information the target can then be located in the world coordinate frame.

General Image Planes

The case that was described above is an ideal case where the optical axes of the two camera positions are exactly parallel. In the real world it would be very hard to accomplish this task while moving the camera. Therefore, a new set of equations can be used that takes into account the relation of the second camera position with respect to the first. If the camera's rotation and transition with respect to the first image position is known, the targets distance away from the object can be computed. Figure 5.5, shown below is an example of a general orientation case.

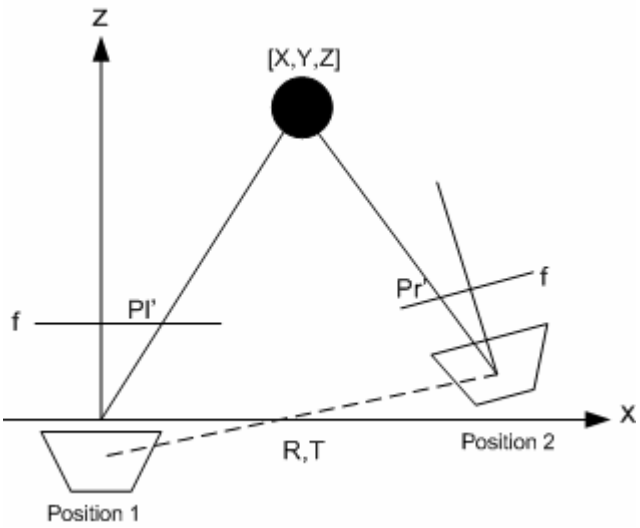

Figure 5.5: General Stereo Vision Setup

In Figure 5.5, position 2 is translated by T, which is the three dimensional translation matrix and R, which is the 3x3 rotation matrix about the y axis. The values of P_l', P_r' are the position of the object in each camera's respective image plane. Now, the depth equations can be found. First the rotation matrix R and translation matrix T has to be defined as

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{5.8}$$

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} r_{14} \\ r_{24} \\ r_{34} \end{bmatrix} \tag{5.9}$$

Now from [26], the position coordinates of the target [X, Y, Z] can now be computed. We can start with equations (5.10), (5.11) and (5.12).

$$(r_{21}\frac{x_l'}{f}+r_{22}\frac{y_l'}{f}+r_{23})z_l+r_{24}=\frac{x_r'}{f}z_r \qquad (5.10)$$

$$(r_{11}\frac{x_l'}{f}+r_{12}\frac{y_l'}{f}+r_{13})z_l+r_{14}=\frac{x_r'}{f}z_r \qquad (5.11)$$

$$(r_{31}\frac{x_l'}{f}+r_{32}\frac{y_l'}{f}+r_{33})z_l+r_{34}=\frac{x_r'}{f}z_r \qquad (5.12)$$

Where x' and y' are the points P' in each corresponding camera position images planes. The variables r are also defined in (5.8) and (5.9). Then, using any two of equations (5.10), (5.11) and (5.12), z_l and z_r can be solved. With z_l and z_r the position of the target point can be computed in each camera position reference plane.

$$r_l=(x_l,y_l,z_l)^T=\left(\frac{x_l'}{f},\frac{y_l'}{f},1\right)^T z_l \qquad (5.13)$$

$$r_r=(x_r,y_r,z_r)^T=\left(\frac{x_r'}{f},\frac{y_r'}{f},1\right)^T z_r \qquad (5.14)$$

5.3. Test Results

After the equations for the three different types of methods are found, the methods can now be tested to see how accurate they are at finding the distance to a target.

5.3.1. Method 1 Test

The first test that will be performed is a test of method one, where a single image is taken of the target straight on. In this test we will know the real world value of the targets diameter and we will obtain the pixel diameter through the pattern matching method described in chapter 4. Below in Figure 5.6 is a diagram of the test scene with the set distance shown.

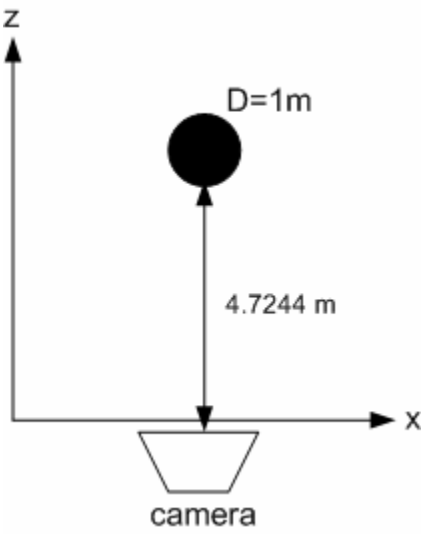

Figure 5.6: Method One Test

The focal length of the camera is found to be 4.76 millimeters. After the diameter of the target in pixels are found then the distance to the target can be found using equations (5.2) and (5.3). The distance to the target is found to be 4.6211 meters. The difference between the real distance and the calculated distance is 0.1033 meters. The percentage error is calculated from equation (5.15) shown below.

$$\%\,Error = \frac{actual - measured}{actual}\ x100 \qquad (5.15)$$

The percentage error from (5.15) is found to be 2% which is acceptable error for this application.

5.3.2. Method 2 Test

Parallel image planes Test

We are now ready to test the parallel image planes method. We first have to set up a controlled test scene. We will first set the target up at a known distance. Next the first image will be captured and then the camera will be moved a set distance to the right,

while keeping the image planes parallel. Then the second image will be taken. Below in Figure 5.7 is a diagram of the test scene with the set distances shown.

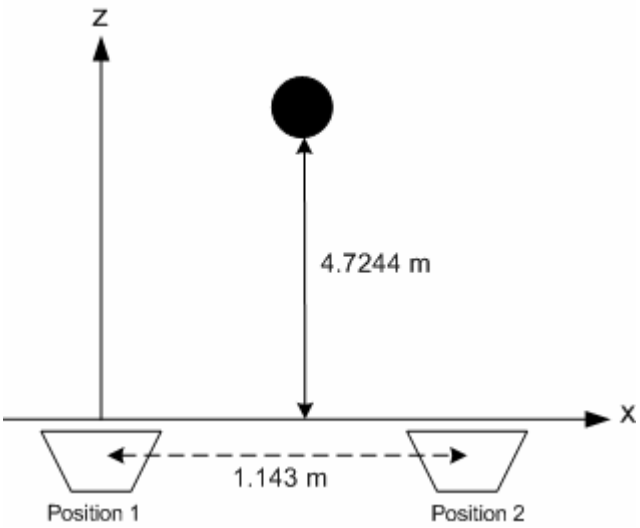

Figure 5.7: Parallel Image Plane Test

The focal length after calibration is found to be 4.76 millimeters. The images are then run though the pattern matching algorithm that is described in chapter 4. With the location of the target in both image planes acquired, the distance from each camera to the target can be calculated with equation (5.5) and is found to be 4.3582 meters. The difference between the real distance and the calculated distance is 0.3662 meters. The percent error is then shown to be around 8%, which is accurate enough for our application.

General image planes Test

The next method to test is the method where the image planes are not parallel with respect to each other. The set-up measurements for this test will be exactly the same as the parallel image plane test except for the rotation of camera two. The rotation will be strictly around the y-axis. A diagram for this test is shown below in Figure 5.8, where $\theta = -10^{\circ}$.

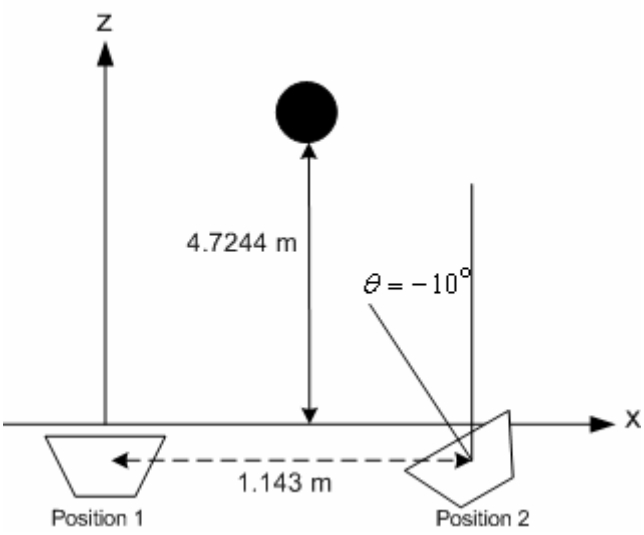

Figure 5.8: General Image Plane Test

After the location of the target is found in the two images by pattern matching, equations (5.8)-(5.12) are then used to calculate the distance to the object with respect to each camera's reference plane. The distance was found to be 4.4305 meters with respect to the left camera position reference plane. The difference between the real distance and the calculated distance is 0.2939 meters and the percent error is 6%, which is accurate enough for out application. The tests of these two methods show that they can be used within a certain degree of accuracy to find the distance to a target and thus the targets location with some error.

Chapter 6

Target Extraction with Robot Control

6.1 Open-Loop Target Acquisition

This first experiment that will be performed is target extraction with open loop control. In this experiment the robot will first take a picture and identify the target and the target's location within the image. The robot will then turn a set angle and travel a set distance then turn back a set angle and take another picture. The distance traveled will be recorded as well as the difference in the second picture angle with respect to the first angle. These measurements will only be recorded by wheel encoders. A diagram of this experiment is shown below in Figure 6.1.

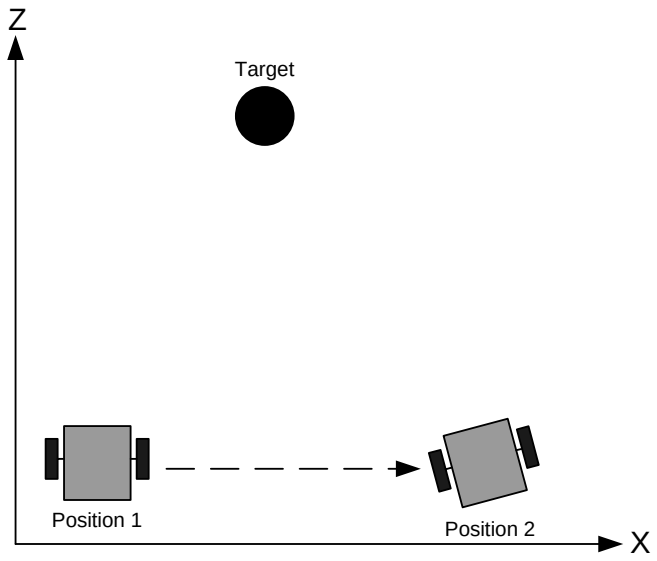

Figure 6.1: Open Loop Control Experiment Set-up

This experiment was run at different target distances ranging from ten meters to thirty meters. Below in Table 6.1 is a list of the results as well as the percent error between the real world distance and the calculated distances.

Actual Target Distance (m)	Calculated Target Distance (m)	% Error Target Distance	Actual Distance Moved (m)	Encoder measured Distance move	% Error Distance Moved
10.05	6.9	31.3	2.31	2.00	13.42
10.05	6.45	35.8	1.16	1.00	13.79
17.67	14.00	20.8	4.64	4.01	13.79
17.67	16.32	7.64	5.00	5.00	0
17.67	16.26	7.97	5.96	6.00	0.67
37.31	21.54	42.3	9.44	10.00	5.93
37.31	16.82	54.9	5.79	5.79	0

Table 6.1: Open Loop Experiment Results

There can be many observations made about these results. The first and most obvious result is the fact that the open look control does not work to well. This can be shown from the percent error values between the real target distance and the one that is calculated with this algorithm. There are many reasons why the open loop control method does not work well. One reason is that the open loop control does not control the robots position accurately enough. Since this control only uses encoders there is some error involved with this. One type of error involved is slip error. This error is when the wheel slips along the ground and increments the encoder value but the robot does not move.

We can see from this table that when the distance moved error is large the target distance error is large also. This makes sense because the distance moved factors into the calculation of the target. What was also observed is that the target distance method is very sensitive to variation. This sensitivity is amplified the farther away the measured target is. This is because when the robot moves a set distance the pixel location of the detected target will move even less compared to if the target is close.

From Table 6.1 above we can see that the open loop method works best when the target is moderately far away from the robot and it doesn’t have to move to far in order calculate

the target distance. This is shown by the fact that when the target was about seventeen meters away and the robot moved about five meters the calculated target distance error was low.

6.2 Closed-Loop Target Acquisition

The next type of experiment that has to be completed is a closed-loop target acquisition experiment. This experiment consists of a robot traveling to a waypoint with GPS and compass feedback. The type of GPS that is used here is the OmniSTAR differential GPS. In this experiment the robot will start at a point and take a picture then travel to a set waypoint and take another picture. It will then calculate the distance to the target in meters with respect to robot at the second waypoint. Below in Figure 6.2 is a diagram of this experiment.

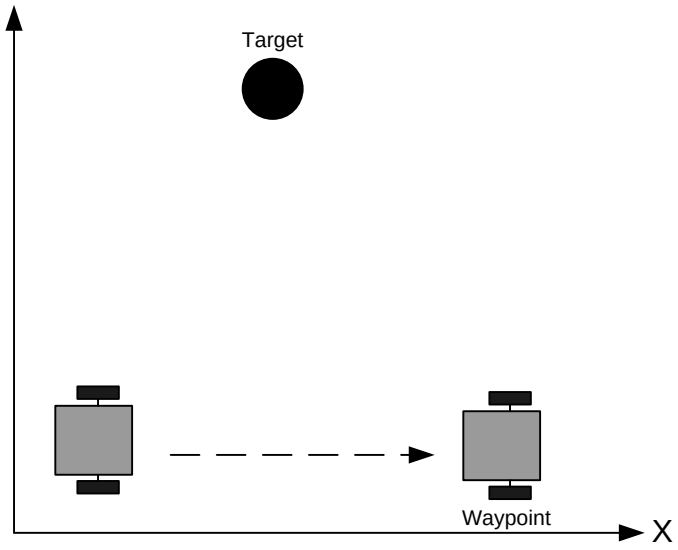

Figure 6.2: Closed Loop Control Experiment Set-up

This experiment was then run at different distances to the target. This distance ranged from eight to thirty meters away from the target. Also the waypoints were set so that when the robot was at the waypoint the target was just inside the field of view. This makes sure that the robot has traveled the maximum distance in order to calculate the

target distance accurately. Table 6.2 shown below is a summary of real world distances and the calculated distances.

Actual Target Distance (m)	Calculated Target Distance (m)	% Error Target Distance
8.4	7.25	13.69
11.7	10.64	9.05
14.9	13.9	6.71
20.07	22.07	9.96
30.51	30.8	0.95

Table 6.2: Closed Loop Experiment Results

What can be shown in Table 6.2 is that this method of closed loop control with GPS is fairly accurate at locating the targets position. We see that this algorithm correctly calculates the distance to the target within a tolerance that is acceptable. The next step that has to be completed is to find out if these errors are within the worst case errors for this system. The worst case scenario is when the GPS and compass both have maximum error in it when a measurement is taken. This can be completed by a process knows as propagation of uncertainty.

6.3 Propagation of Uncertainty

In order to compare the two experiments that are described above, a method of determining if the found distances are within a certain error bound has to be found. This process is called propagation of uncertainty analysis. In order to calculate this uncertainty we first have to assume that the measurements that are used to calculate the distance are independent variables. We can now state the propagation of uncertainty theorem from [27].

We can say that a calculated result y is a function of several independent measured variables $\{x_1, x_2, ..., x_n\}$. Also each measured value has some uncertainty

$\{u_1, u_2, ..., u_n\}$ and these uncertainties lead to an uncertainty in y which is called u_y.

This uncertainty u_y can be calculated from (6.1).

$$u_y = \sqrt{\left(\frac{\delta y}{\delta x_1} u_1\right)^2 + \left(\frac{\delta y}{\delta x_2} u_2\right)^2 + ... + \left(\frac{\delta y}{\delta x_n} u_n\right)^2} \qquad (6.1)$$

Since our equations to calculate distance are over constrained because we have three equations and two unknowns, we have to calculate the uncertainty u_y discretely for different scenarios. We first have to assume that the measurements are independent variables. We also have to assume that the vehicle is not moving when the measurements are being made.

We are also going to only complete an error propagation study for the closed loop target acquisition method. The reason being is that we can easily obtain and accurately model the errors for the digital compass and GPS. In order to complete an error propagation study for the open loop target acquisition method we would need to model the errors associated with wheel slip. This error is not easily found since the terrain that we are traveling on is not uniform and the wheel slip can vary immensely.

In order to find u_y we first need to find the uncertainties associated with the compass and GPS. Since the GPS will be using the OmniSTAR service the uncertainty will be 1 meter. According to the OmniSTAR data sheet this is a two standard deviation uncertainty. The uncertainty for the digital compass is 0.5 degrees. We tried to contact PNI about the standard deviation associated with this compass measurement but we could not find it. Therefore we will assume that the compass uncertainty will be within two standard deviations also. We can convert uncertainty to standard deviation, which we will have to accomplish first, by using equation (6.2).

$$\sigma = \frac{uncer}{2} \qquad (6.2)$$

We will not model the camera errors since we cannot obtain an error model accurately. The camera errors will also be relatively small compared to the GPS error hence it will not have a large effect on the outcome of the system .We can now show that

$$Z_r = F(f, \theta, x_l, x_r, y_l, y_r, t_x, t_y) \tag{6.3}$$

Where equation (6.3) is defined in chapter five of this thesis. We also will simplify our equations by saying that we will only use the distance to the target from the second location, Z_r. We now have to solve for the discrete version of equation (6.1). We first have to choose a nominal value of an independent variable. This nominal value is a real world measured value in the experiment. We can now calculate the discrete partial derivative with respect to the GPS error in one dimension.

$$\frac{\Delta Z_r}{\Delta t_x} = \frac{F(\bar{t}_x + \Delta t_x, ...) - F(t_x, ...)}{\Delta t_x} \tag{6.4}$$

Where Δt_x is the error associated with the GPS. We can now calculate the discrete partial derivatives with respect to the GPS error in the second dimension as well as the compass error with equations that are similar to (6.4). Finally after each discrete partial derivative is found we can calculate u_{Zr}

$$u_{Zr} = \sqrt{\left(\frac{\Delta Z_r}{\Delta t_x}\Delta t_x\right)^2 + \left(\frac{\Delta Z_r}{\Delta t_z}\Delta t_z\right)^2 + \left(\frac{\Delta Z_r}{\Delta \theta}\Delta \theta\right)^2} \tag{6.5}$$

With this analysis we can say what error bounds the distance must be in for the worst case scenario of errors. We are now ready to complete the error propagation calculations for our closed loop control target acquisition experiment. In order to complete this task a MATLAB program was developed in order to simplify the multiple calculations. This MATLAB program is shown in Appendix A.

Below in Table 6.3 is the standard deviation for each distance calculation at different target locations. In the table we also show the actual distance to the target as well as the calculated distance to the target.

Actual Target Distance (m)	Calculated Target Distance (m)	Standard Deviation (m)
11.9	10.83	2.035
6.4	6.1	2.51
17.4	15.9	1.705

Table 6.3: Experimental Data with Two Standard Deviation Error

The first comment that can be made of this table is the fact that this algorithm calculates the actual distance to a target quite accurately. What is meant by quite accurately is the fact that the calculations are within two standard deviations of the mean. The worst case scenario is when the GPS has a standard deviation of 0.5 meters and the compass has a standard deviation of 0.25 degrees. We can now double these uncertainties and see what effect they have on the calculations. Below in Table 6.4 shows the results if we increase the GPS uncertainty to two meters and the compass uncertainty to one degree.

Actual Target Distance (m)	Calculated Target Distance (m)	Standard Deviation (m)
11.9	10.83	4.075
6.4	6.1	5.025
17.4	15.9	3.405

Table 6.4: Experimental Data with Greater Error Bounds

We can see from this table above that when the uncertainties of the sensors are increased the uncertainties in the target distance calculations are increased. There is almost a linear relationship between the sensor uncertainties to the calculation uncertainties. What this

also shows is that the distance calculation algorithm is very sensitive to perturbations in measurements. This means that if the robot does not know its position very accurately then it will not be able to calculate the location of the target very accurately.

Chapter 7

Conclusions

7.1. Concluding Remarks

This thesis describes in detail a target acquisition algorithm using just a single camera. We also develop a simple proportional controller that will be able to control the robot very accurately during waypoint navigation. We first start out with an open loop controller for the vehicle using just position in order to find out where we are. We then develop a closed loop dead reckoning system that would drive waypoints using only encoder feedback. We then developed a closed loop controller using a very accurate GPS and compass system. Finally, we laid the foundation that would allow us to use Extended Kalman filtering of the INS system and encoder dead reckoning systems in order to accurately locate where the robot is in real world. This system would only be useful if the GPS errors were large.

We then developed an algorithm that will be able to pick out the IARC target in an image. The first algorithm that was tested was the Hough Transform algorithm. This proved to not be very robust. Therefore we developed a pattern matching method that proved to be very robust. After we picked out the target in the picture we then had to develop an algorithm that would be able to calculate the distance to the target. The calculations that we developed were based off of general stereo vision algorithms.

After combining all of these steps we were then able to use the robots inertial data as well as the targets position in the image plane in order to accurately acquire the location of the target within our tolerances.

7.2. Future Work

7.2.1. Waypoint Controller

While the waypoint controller proved to be very accurate with the current GPS and compass on it there can also be some improvement. For this experiment the environment was very controlled. Meaning the GPS has the most accurate signal since it was operating in a field with clear line of sight. Now if the GPS was to operate in an environment where it would get cut out there would have to be some type of Extended Kalman filtering of GPS and encoder measurements. We would need to develop an Extended Kalman filter if we were to use a cheaper GPS with no corrections in it.

7.2.2. Target Extraction with Robotic Control

The integration of the target extraction algorithm has many future work options associated with it. One upgrade to the system would be a way to stabilize the camera on the mast when the vehicle is moving so the target acquisition algorithm can be run when the vehicle is moving continuously. We can move the camera down the mast to reduce this backlash of the mast and thus the error in the target extraction measurements but we would loose the greater field of view. Therefore the only option would be to add some type of stabilizing gimbal where the camera will be stabilized even if the robot is rocking around. This would greatly reduce the error in the distance calculations while the robot is moving.

References

1. *Darpa Grand Challenge*. 2006 [cited April 2006]; Available from: http://www.darpa.mil/grandchallenge/.
2. Sights, B., et al., *Integrated Control Strategies Supporting Autonomous Functionalities in Mobile Robots.* SSC Robotics Update, 2005. **5**.
3. Pachidis, T. and Lygouras, J. *A Pseudo Stereo Vision System as a Sensor for Real Time Path Control of a Robot*. in *IEEE Instrumentation and Measurement Technology Conference*. 2002. Anchorage, AK.
4. Ishiguro, H., Yamamoto, M., and Tsuji, S. *Onmi-Directional Stereo for Making Global Map*. in *IEEE Int. Conf. Computer Vision*. 1990.
5. *How Digital Cameras Work*. How Stuff Works 2005 [cited 2005 September]; Available from: http://electronics.howstuffworks.com/digital-camera4.htm.
6. *Fire-I Board Digital Camera Specification Sheet.* 2005 [cited 2005 September]; Available from: http://www.unibrain.com/1394_products/fire-i_board_cam/fire-i_board_camera.htm.
7. *Basler A100 Series Specification Sheet*. 2005 [cited 2005 September]; Available from: www.basler-vc.com.
8. *Axis PTZ 213 Network Camera Specification Sheet*. 2005 [cited 2005 September]; Available from: www.axis.com.
9. *3-D Imaging Using Stereo Vision*. CSIRO Mathematical and Information Sciences. 2005 [cited 2005 September 2005].
10. *Point Grey Research Bumblebee^{TM} Specification Sheet*. 2005 [cited 2005 October]; Available from: www.ptgrey.com.
11. *TYZX DeepSea V2 Specification Sheet.* 2005 [cited 2005 October]; Available from: www.tyzx.com.
12. *SICK PLS Product Information*. 2005 [cited 2005 October]; Available from: www.sick.de.
13. *Opti-Logic RS100 Specification Sheet*. 2005 [cited 2005 October]; Available from: http://www.opti-logic.com/industrial_rangefinders.htm.
14. Dudek, G. and Jenkin, M., *Computational Principles of Mobile Robotics*. 2000, Cambridge: Cambridge University Press.
15. Borenstein, J., Everett, H.R., and Feng, L., *Where am I? Sensors and Methods for Mobile Robot Positioning*. 1996: University of Michigan.
16. Ridgesoft. *Programming Your Robot to Navigate*. 2005 [cited; Available from: http://www.ridgesoft.com/articles/navigation/ProgrammingYourRobotToNavigate.pdf.
17. *GPS Explained*. 2006 [cited 2006 April]; Available from: http://www.gpspassion.com/Hardware/explained.htm.
18. Mayhey, D.M., *Multi-rate Sensor Fusion for GPS Navigation Using Kalman Filtering*, in *Electrical Engineering*. 1999, Virginia Tech.
19. Grabowski, B., Navarro-Serment, L., and Bererton, C. *Description of the Extended Kalman Filter*. [cited 2006 April]; Available from: http://www.contrib.andrew.cmu.edu/~rjg/millibots/millibot_kalman.html.

20. *Rules for the Current Internation Aerial Robotics Competition Mission*. 2005 [cited 2006 Febuary]; Available from: http://avdil.gtri.gatech.edu/AUVS/CurrentIARC/200xCollegiateRules.html.
21. Pei, S.C. and Horng, J.H., *Determination of Circular Arc Length and Midpoint by Hough Transform.* IEEE Internation Symposium on Circuits and Systems, 1994. **Vol. 3**: p. 1-4.
22. Kachroo, P. *Computer Vision Systems Notes*. 2005 [cited 2006 January]; Available from: http://www.ee.vt.edu/%7epushkin/ece5554/ece5554.html.
23. R. Fisher, S.P., A. Walker, E. Wolfarrt. *Canny Edge Detector*. 2003 [cited 2006 Febuary 4]; Available from: http://homepages.inf.ed.ac.uk/rbf/HIPR2/canny.htm.
24. Luo, D., Smart, P., and Macleod, J., *Circular Hough Transform Roundness measurement of Objects.* Patern Recognition, 1995. **Vol 28**: p. 1745-1749.
25. Bourke, P. *Cross Correlation*. 1996 [cited 2006 Febuary]; Available from: http://astronomy.swin.edu.au/~pbourke/other/correlate/.
26. Horn, B.K.P., *Robot Vision*. 1986, Cambridge, Massachusetts: McGraw-Hill.
27. Beckwith, T., Marangoni, R., and Lienhard, J., *Mechanical Measurements*. fith ed. 1995: Addison Wesley.

Appendix A.

MATLAB Source Code

A.1. Experiment2.m

```
%Matthew Simone
%1/5/2006

%This is a program that will simulate stereo vision equations.
%Everything will be in meters

rx = 640    ;       %Middle point of pixel resolution x in pixels
b = 0.4318;         %Distance between cameras
f = 4.41*10^-3;     %Focal Length of camera in meters
ps = 5.6*10^-6;     %Pixel size

p1 = 399;           %Distance of Point 1 in pixels
p2 = 237;           %Distance of Point 2 in pixels

beta = 0*(pi/180);  %Rotation about y axis
tx = b;             %Translation in x direction from left camera to right camera
ty = 0;             %Translation in y direction from left camera to right camera
tz = 0;             %Translation in z direction from left camera to right camera

% This part uses the fact that the two cameras are
% parallel to each other.
xl = (p1-(rx/2))*ps;
xr = (p2-(rx/2))*ps;

Z = b*f/(xl-xr)    %Position Z in coordinate frame
X = (xl*Z)/f;      %Position X in coordinate frame

%*****************************************************************
```

```
yl = 0;
yr = 0;
% This part uses the fact that the two cameras are
% are not parallel to each other and uses a rotation
% and translation matrix

R = [cos(beta) 0 -sin(beta);0 1 0; sin(beta) 0 cos(beta)];       %Rotation Matrix
T = [tx;ty;tz];                  %Translation Matrix

A = [R*[xl/f; yl/f; 1] -[xr/f; yr/f; 1]];
B = -T;
distance = A\B;

%Display the distances from the cameras
zl = -distance(1)
zr = -distance(2)

%Point of object with respect to left and right camera positions.
rl = [xl/f; yl/f; 1]*zl;
rr = [xr/f; yr/f; 1]*zr;
```

A.2. Error_Propagation.m

```
%Matthew Simone
%4/18/2006

close all
clear all

%This is a program that will simulate stereo vision equations.
%This program will calculate the error propagation for the different types
%of error.

rx = 640;          %Middle point of pixel resolution x in pixels
ry = 480;          %Middle point of pixel resolution y in pixels

f = 0.0054208;     %Focal Length of camera in meters
ps = 5.6*10^-6;     %Pixel size

Easting1 = 549793.625;
Northing1 = 4118687.75;
Heading1 = 140.1000061;
px1 = 486.5;          %Distance of x for Point 1 in pixels
py1 = 356;
```

```
Easting2 = 549794.8125;
Northing2 = 4118687.25;
Heading2 = 144.1000061;
px2 = 226.5;          %Distance of x for Point 2 in pixels
py2 = 374.5;

Ry = Heading2 - Heading1;   %Rotation about y axis
tx = Easting2 - Easting1;   %Translation in x direction from left camera to right camera
ty = 0;                     %Translation in y direction from left camera to right camera
tz = Northing2 - Northing1; %Translation in z direction from left camera to right camera

beta = Ry*(pi/180);  %Rotation about y axis in radians

% This part uses the fact that the two cameras are
% parallel to each other.
x1 = (px1-(rx/2))*ps;
x2 = (px2-(rx/2))*ps;
y1 = (py1-(ry/2))*ps;
y2 = (py2-(ry/2))*ps;

delta_tx = 2;               %Error in the GPS signal in meters in one direction   originally 1
delta_tz = 2;               %Error in the GPS signal in meters in second direction  originally
1
delta_beta = 1*(pi/180);    %Error in the compass measurements in degrees originally 0.5

tx_error = tx + delta_tx;       %This is where we introduce the GPS error
tz_error = tz + delta_tz;
beta_error = beta + delta_beta;  %This is where we introduce the compass error

%*************************************************************************
%We will now go and calculate the error propagation

%First vary the GPS position

%*************************************************************************
%Calculate delta_Z/delta_tx
%F(tx+tx_error,...)
R = [cos(beta) 0 -sin(beta);0 1 0; sin(beta) 0 cos(beta)];       %Rotation Matrix
T = [tx_error;ty;tz];                                    %Translation Matrix

A = [R*[x1/f; y1/f; 1] -[x2/f; y2/f; 1]];
B = -T;
distance_error = A\B;
Zr_error = -distance_error(2);

%F(tx,...)
```

```
R = [cos(beta) 0 -sin(beta);0 1 0; sin(beta) 0 cos(beta)];       %Rotation Matrix
T = [tx;ty;tz];                                  %Translation Matrix

A = [R*[x1/f; y1/f; 1] -[x2/f; y2/f; 1]];
B = -T;
distance = A\B;
Zr = -distance(2);

delta_ZrTx = (Zr_error - Zr)/delta_tx;


%**************************************************************************
%Calculate delta_Z/delta_tz
%F(tz+tz_error,...)
R = [cos(beta) 0 -sin(beta);0 1 0; sin(beta) 0 cos(beta)];       %Rotation Matrix
T = [tx;ty;tz_error];                                %Translation Matrix

A = [R*[x1/f; y1/f; 1] -[x2/f; y2/f; 1]];
B = -T;
distance_error = A\B;
Zr_error2 = -distance_error(2);

%F(tz,...)
R = [cos(beta) 0 -sin(beta);0 1 0; sin(beta) 0 cos(beta)];       %Rotation Matrix
T = [tx;ty;tz];                                  %Translation Matrix

A = [R*[x1/f; y1/f; 1] -[x2/f; y2/f; 1]];
B = -T;
distance = A\B;
Zr2 = -distance(2);

delta_ZrTz = (Zr_error2 - Zr2)/delta_tz;


%**************************************************************************
%Calculate delta_Z/delta_beta
%F(beta+beta_error,...)
R = [cos(beta_error) 0 -sin(beta_error);0 1 0; sin(beta_error) 0 cos(beta_error)];
%Rotation Matrix
T = [tx;ty;tz];                                  %Translation Matrix

A = [R*[x1/f; y1/f; 1] -[x2/f; y2/f; 1]];
B = -T;
distance_error = A\B;
Zr_error3 = -distance_error(2);
```

```
%F(beta,...)
R = [cos(beta) 0 -sin(beta);0 1 0; sin(beta) 0 cos(beta)];       %Rotation Matrix
T = [tx;ty;tz];                                    %Translation Matrix

A = [R*[x1/f; y1/f; 1] -[x2/f; y2/f; 1]];
B = -T;
distance = A\B;
Zr3 = -distance(2);

Calculated_Distance = Zr3

delta_Zrbeta = (Zr_error3 - Zr3)/delta_tx;

%**************************************************************************
%Now calculate the overall error propagation
U_zr = sqrt((delta_ZrTx*delta_tx)^2 + (delta_ZrTz*delta_tz)^2 +
(delta_Zrbeta*delta_beta)^2)
```

Vita

Matthew Simone is a native of Bridgewater, NJ and graduated from Bridgewater High School in 1999. In 2004 he received his Bachelor's degree in Electrical engineering from Virginia Tech in Blacksburg, VA. In the fall of 2004 he started graduate school at Virginia Tech where he started to work for the Joint Unmanned Systems Test, Experimentation, and Research Group at Virginia Tech. During this time he conducted research in the field of unmanned systems for JOUSTER. He will graduate with his masters degree in electrical engineering in May 2006 where he will then move to the Washington D.C. to start work in the unmanned systems field.