

REGISTRATION AND TRACKING OF OBJECTS WITH COMPUTER VISION FOR
AUTONOMOUS VEHICLES

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

Andrew Nevin

Certificate of Approval:

David M. Bevly, Co-Chair
Associate Professor
Mechanical Engineering

Thaddeus A. Roppel, Co-Chair
Associate Professor
Electrical and Computer Engineering

Thomas Denney
Professor
Electrical and Computer Engineering

Alan Scottedward Hodel, Co-Chair
Associate Professor (deceased)
Electrical and Computer Engineering

George T. Flowers
Dean
Graduate School

REGISTRATION AND TRACKING OF OBJECTS WITH COMPUTER VISION FOR
AUTONOMOUS VEHICLES

Andrew Nevin

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

Auburn, Alabama
May 9 ,2009

REGISTRATION AND TRACKING OF OBJECTS WITH COMPUTER VISION FOR
AUTONOMOUS VEHICLES

Andrew Nevin

Permission is granted to Auburn University to make copies of this thesis at its discretion, upon the request of individuals or institutions and at their expense. The author reserves all publication rights.

Signature of Author

Date of Graduation

VITA

Andrew Nevin, son of Ronald and Brenda Nevin, was born June 8, 1983, in Nashville, Tennessee. He graduated from Montgomery Bell Academy in 2002. He attended Auburn University in September 2002 and graduated cum laude with a Bachelor of Electrical Engineering degree in December 2006. He enrolled in the Graduate School, Auburn University, in January 2006, working as a graduate teaching assistant for Electrical Engineering Lab II and a graduate research assistant. He graduated with a Master's of Science in Electrical Engineering degree in May 2009.

THESIS ABSTRACT
REGISTRATION AND TRACKING OF OBJECTS WITH COMPUTER VISION FOR
AUTONOMOUS VEHICLES

Andrew Nevin

Master of Science, May 9 ,2009
(B.S., Auburn University, 2006)

70 Typed Pages

Directed by Scotte Hodel and David Bevly

For this research, an autonomous leader-follower vehicle caravan is being designed using computer vision. The focus of this research is to investigate computer vision methods of identifying objects detected by the leader vehicle and determining the follower vehicle's position relative to the objects. Many object registration and tracking methods are presented in this paper such as Hough transforms, optical flow, and normalized cross correlation. A novel statistical-gradient method is presented to combine feature based methods and area based methods for object registration and tracking. These methods are evaluated against both synthetic and captured video data. This registration method is shown to be effective with both synthetic data and acquired data.

ACKNOWLEDGMENTS

The author would like to thank his advisors Dr. Alan Scottedward Hodel and Dr. David Bevly, as well as his committee members Dr. Thaddeus Roppel and Dr. Thomas Denney for their guidance and patience through this research. Additionally, he would like to thank Dr. Stanley Reeves for instruction in Digital Image Processing and Advanced Digital Image Processing. Without the course instruction, advice, and motivation given to the author, this research would not have been possible.

The author wishes to express his appreciation for his grandparents Robert and Louise Howell, his parents Ronald and Brenda Nevin, his brother Ryan, and two sisters Ashley and Caroline for all of their continuous love, support, and encouragement throughout his entire educational career.

Style manual or journal used Journal of Approximation Theory (together with the style known as “aums”). Bibliography follows van Leunen’s *A Handbook for Scholars*.

Computer software used The document preparation package T_EX (specifically L^AT_EX) together with the departmental style-file `aums.sty`.

TABLE OF CONTENTS

LIST OF FIGURES	x
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Previous Work	3
1.3 Approach	4
2 DIGITAL IMAGE PROCESSING BACKGROUND	6
2.1 Processing Pixels	6
2.2 Implementation of filters and edge detection	7
2.3 Image Filters	8
2.4 Multi-resolution processing	9
2.5 Edge Detection Methods	11
2.6 Hough Transform	13
2.7 Object Registration	14
2.7.1 Definition	14
2.7.2 Classifications	16
2.7.3 Methods of Object Registration	16
2.7.4 Normalized Cross Correlation	18
2.8 Optical Flow	18
3 EXPERIMENTS AND RESULTS	24
3.1 Synthetic Data	24
3.2 Acquired Data	26
3.2.1 Edge Detection	27
3.2.2 Hough Transform	27
3.2.3 Normalized Cross Correlation	28
3.3 Comments	28
4 STATISTICAL-GRADIENT OBJECT REGISTRATION	30
4.0.1 Area-Based	30
4.0.2 Feature-Based	31
4.0.3 Code Outline	32
4.0.4 Validation using Synthetic Images	33
4.0.5 Validating on Acquired Images	34
4.0.6 Motion Prediction	37
4.0.7 Stop Sign Registration	39
4.0.8 Multiple Green Signs Registration	40

5	CONCLUSIONS	44
6	FUTURE WORK	46
	BIBLIOGRAPHY	47
	APPENDICES	49
A	NOMENCLATURE	50
B	ORIENTATION RESULTS OF RED BLOCK	51
C	ORIENTATION RESULTS OF THE BLUE BLOCK	55
D	TOLERANCE VALUES FOR STATISTICAL ANALYSIS	59

LIST OF FIGURES

2.1	Gaussian noise is present in the image in a). A 5 x 5 pixel mask is used as the low-pass filter. The result is in b). Notice the blurring and removing of some the noise as well as the letters in the image. A 2 x 2 pixel mask is used for the median filter. The result is in c). All of the noise has been removed, but there is a little deformation of the letters.	9
2.2	Sobel Edge detection	13
2.3	Points on the Cartesian coordinate system	15
2.4	Points from the Cartesian system in Fig 2.3 transformed into parameter space.	15
2.5	The Hough transform from the image in a) is shown as an image in b). The lines are extracted with the inverse Hough transform and plotted on top of the original image in c).	15
2.6	Normalized cross correlation output	18
2.7	Barber pole	20
2.8	All three partial derivatives at the center are calculated from the average of the first four differences of adjacent pixels. The column index is denoted as j , the row index is denoted as i , and the time index is denoted as k	21
2.9	Optical flow calculated between the two input images. The block in the top right corner is moving diagonally down from left to right, and the block in the bottom right corner is moving from right to left.	23
3.1	Synthetic image for object registration	24
3.2	(a) The magnitude of the x and y gradients using the Sobel kernels. (b) Lines extracted from the Hough transform are plotted on top of the grayscale input image.	25
3.3	This is a surface plot of the magnitude of the normalized correlation coefficients. The red block is on the left, and the red straight line is on the right. Notice that the maximum coefficient is in the middle of the actual red block.	26

3.4	The output from the digital camera. Included in the image are the four blocks to be detected. There is a black background on the rink, there is a blue and white surface, and there are some pebbles.	27
3.5	Edge detection on the acquired data	27
3.6	Hough transform on the acquired data	28
3.7	The output is a surface plot of the normalized cross correlation coefficients γ . The red surface indicates a high correlation coefficient and blue represents a low correlation coefficient.	29
4.1	The plot on the left contains 10 randomly generated line segments. The plot on the right are the line segments that are perpendicular to each other. . .	32
4.2	Synthetic image for object registration	34
4.3	The output from the algorithm used on the input from Fig 4.2.	35
4.4	The output from the line segment algorithm on the lines extracted from the Hough transform on the output from the object identification method. . . .	35
4.5	Four outputs of the registered red block	36
4.6	The center of the registered object red block in ten selected frames from the image sequence is shown above.	37
4.7	Four outputs of the registered blue block	38
4.8	Plots of the X estimation, Y estimation, and Z estimation of the red block.	38
4.9	Plots of the X estimation, Y estimation, and Z estimation of the blue block.	39
4.10	Four outputs from the object orientation method.	41
4.11	A frame of the video sequence of the green signs.	42
4.12	Four outputs from the video sequence of the green signs	43

CHAPTER 1
INTRODUCTION

1.1 Motivation

The desire and necessity to build autonomous vehicles is increasing every year. Autonomous vehicles can be designed to replace humans on jobs that are dangerous, time-consuming, or difficult. Recent events have shown the value of using highly automated technology in the battlefield. Autonomous aircraft, terrestrial, and marine vehicles have accomplished important missions while reducing risk to personnel [6]. They can be used in rescue operations, such as collapsed mines or building fires, where it is dangerous to send workers. Other tasks include avoiding obstacles, keeping a safe distance behind other vehicles, regulation of speed according to traffic conditions and road characteristics, and driving and parking within urban environments [1]. Autonomous vehicles can be sent along a dangerous path to send supplies to soldiers in need without risking injuries or fatalities to others. Research in vehicle following has attracted the attention of several researchers in the past decades, particularly in the USA and Europe where safety, energy consumption, and traffic congestions are the primary motivators [2], [3]. In a recent study by the Virginia Department of Motor Vehicles and Virginia Commonwealth University, researchers conclude that between 25 to 50 percent of all motor vehicle crashes in the U.S. are caused by driver distraction. Rubbernecking (slowing down to look at another accident) accounts for 16% of accidents caused by distractions. Driver fatigue (12 percent), looking at scenery (10 percent), other passengers or children (9 percent), adjusting the radio (7 percent), reading maps or other documents (2 percent) account for the majority of accidents. Cell-phone use is another distraction that is increasing the number of accidents. Driver fatigue accounts for 100,000 crashes each year. Aggressive driving is another frequent cause of accidents. It

is evident that there is a need to prevent fewer accidents each year. Creating autonomous or semi-autonomous vehicles is an important step toward reducing accidents.

The first step to creating an autonomous vehicle is self-localization. Self-localization is the process by which an autonomous vehicle calculates its position relative to the environment. GPS is typically used to estimate the vehicle's current position, however, due to indoor environments, dense trees or urban areas, the line of sight between the satellites and vehicle can be blocked. Other sensors, such as IMU, LIDAR, and vision can be used to estimate the position of the vehicle when GPS is not available.

A desire to create an autonomous leader-follower caravan for military convoys is growing. In [4], the authors discuss the current state of research in these convoys. The leader-follower capability allows one man-driven vehicle to be followed by one or more unmanned vehicles in a convoy-like operation. The man-driven lead vehicle establishes the route for the unmanned follower vehicles by sending the waypoints or GPS breadcrumb coordinates to the followers. Additionally, the follower vehicles are instructed to trail the leader at a specified distance [4]. The motivation behind the leader-follower convoy is to reduce the risk of soldiers in dangerous situations.

For this research, an autonomous leader-follower vehicle caravan is being designed using computer vision. As the leader vehicle navigates through a landscape, it will need to recognize and identify features it may "see", which includes obstacles that it may have to detour around. The leader vehicle identifies these features using digital image processing techniques such as gradient detection, Hough transforms, and local averages and variances of pixel values. In this research it is assumed that the leader estimates the orientation and distance of an object, and transmits the data to the follower vehicle. The follower vehicle searches the landscape for the object identified by the leader vehicle, and calculates the orientation and distance from the object. The follower will then make attempts to correct its position based on the leader vehicle's data. The focus of this research is to investigate computer vision methods of identifying objects detected by the leader vehicle and determining the follower vehicle's position relative to the objects.

1.2 Previous Work

In [5], the authors use a leader vehicle, which is not equipped with GPS or inter-vehicle communication networks, and a follower vehicle equipped with an on-board laser scanner and odometry sensors. The leader position is estimated using the measurement data from the laser scanner. In [6], the authors present a leader-follower design for multiple autonomous underwater vehicles operating in formation to perform tedious and dangerous tasks, such as minesweeping. The leader vehicle broadcasts its position to the follower vehicles through underwater acoustic communication. The follower vehicles move into formation relative to the leader to scan for mines. This work requires that the leader vehicle be close to the follower vehicles. In [7], the authors use cameras for an autonomous vehicle to follow an astronaut in outer space. They tag the astronaut with a green color blob. A green color filter is used to convert the data from the camera on the follower to a binary image. The orientation and size of the detected color blob is used to control the distance between the astronaut and the follower. This follower robot helps the astronaut carry moon rocks and other heavy items that would otherwise burden the astronaut. However in many operations it may be impractical to attach a highly visible signal to the leader vehicle. Additionally in some leader follower operations, the follower may be far enough back that the lead vehicle is not visible.

In [8], the authors present the design of a different algorithm using visual sensors. The follower vehicle is equipped with only a camera. The tracking objective requires information related to the leader's motion, which is extracted from the video data from the follower vehicle. A reference image is captured and stored in the memory of the leader vehicle while it is in the desired position and orientation relative to the follower. The control algorithm attempts to match this reference image as closely as possible. For this research, a leader-follower caravan is designed that does not require the follower to be close to the leader. For example, if the leader vehicle travels down a path and somehow determines this is the

optimal path for other vehicles, then the leader vehicle will transmit data describing the path to the follower vehicle.

Many different approaches have been implemented for identifying and tracking objects through vision sensors. The most common algorithms include normalized cross correlation, optical flow, edge detection, or the Hough transform. In [17], the authors use a fast normalized cross correlation algorithm for self-localization of a robot. In [9], the authors use an optical flow algorithm that can be used for road navigation. However, after the optical flow is calculated, no other information is extracted for navigation of a following vehicle. In [19], the authors use an area-based algorithm to statistically measure a small window of points between two images for registering Synthetic Aperture Radar (SAR) images. A feature-based method, the Hough transform, can be used to detect the shapes of objects. In [11], the authors use the Hough transform to identify different shaped objects. In [12], the authors use the Hough transform to detect ellipses.

Simultaneous Localization and Mapping (SLAM) is another active area of research. In SLAM, it is important to continuously localize and estimate 3D positions of new landmarks in an unknown environment. In [14], a high speed CMOS camera is used to implement SLAM in real-time on a robotic arm.

1.3 Approach

First, the algorithms for the follower vehicle will be developed. This development includes computer vision algorithms. This research will start with the underlying assumptions that the leader vehicle has previously passed down a path, has identified good features to track through vision sensors, and has transmitted the data to the follower vehicle. The follower vehicle will then try to locate these specific objects with vision sensors and navigate along the same path as the leader vehicle. The follower vehicle could travel down this path in seconds, hours, or days after the leader vehicle has traversed it. Before experimenting with images from landscape and the actual vehicle, a smaller scale experiment is performed. A digital camera is placed on a cart and pushed across a rink. There are three surfaces:

a smooth blue one, a white sandy one, and pebbles. There are also four blocks scattered around the surfaces. There is a red block, blue block, black block, and a tan colored block. It is assumed that the leader vehicle detected and located the red block as a good feature to track. After demonstrating successful object registration, the algorithms are applied to video data taken from a vehicle on the road. The video data obtained includes stop signs and green signs periodically placed 200 feet apart. In this thesis, different methods of object registration in a sequence of images using digital image processing techniques are reviewed, and a proposal for a fusion of two methods to optimize the identification of objects for tracking is tested.

CHAPTER 2

DIGITAL IMAGE PROCESSING BACKGROUND

An image may be defined as a two-dimensional function, $f(x, y)$, where x and y are spatial (plane coordinates), and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point [15]. Digital image processing is the use of computer algorithms to perform image processing on digital images. The purpose of computer vision is to recognize specific events and perform actions through visual inputs instead of humans. This chapter provides the background on basic digital image processing algorithms, such as implementation of filters, multi-resolution, edge detection, Hough transform, normalized cross correlation, and optical flow.

2.1 Processing Pixels

Any given pixel at coordinates (x, y) has four horizontal and vertical neighbors described by the following notation:

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1) \quad (2.1)$$

There are also four diagonal pixel neighbors defined as:

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1) \quad (2.2)$$

These equations for a pixel only describe binary or gray scale images. It is a little more computationally expensive to process colored images. A colored image is composed of three

channels: red, green, and blue. The value of a pixel can be represented as a vector:

$$p(x, y) = \begin{bmatrix} R(x, y) \\ G(x, y) \\ B(x, y) \end{bmatrix} \quad (2.3)$$

For example, a color image can have dimensions 256 x 256 x 3 pixels in a memory array in a computer, but have 256 x 256 pixels for a binary or grayscale images. To convert a colored image to grayscale, the magnitude of the vector in (2.3) is computed:

$$p_{grayscale} = \sqrt{R(x, y)^2 + G(x, y)^2 + B(x, y)^2} \quad (2.4)$$

Transforming a colored image into grayscale has many advantages. Eliminating out two-thirds of the pixel data, allows for faster processing time on certain operations. But in some cases, it is desirable to use all color information obtained from a scene. Basic arithmetic operations can be performed on the pixels and neighboring pixels, such as expected values, edge detection, low-pass or median filtering.

2.2 Implementation of filters and edge detection

More advanced digital image processing techniques can be implemented. Image filters can be used for several purposes: color extraction, noise subtraction, or smoothing an image. Edge detection can be used for image restoration, calculating optical flow, or shape detection. Implementing these two operations can be done in the spatial domain or the frequency domain. This section provides the background on implementing filtering and edge detection operations using two-dimensional convolution in both domains.

Two-dimensional convolution between an image $f(x, y)$ and template $h(x, y)$ is defined as:

$$f(x, y) * h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)h(x - m, y - n) \quad (2.5)$$

This operation can also be performed in the frequency domain. The two-dimensional Fourier transform of an image $f(x, y)$ is given by:

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M+vy/N)} \quad (2.6)$$

The inverse Fourier transform of $F(u, v)$ is given by:

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M+vy/N)} \quad (2.7)$$

Now two-dimensional convolution can be implemented by multiplying the Fourier transforms of the image $f(x, y)$ and the template $h(x, y)$, and then implementing the inverse Fourier transform of the product, which results in an output image $g(x, y)$:

$$g(x, y) = F^{-1}[H(u, v)F(u, v)] \quad (2.8)$$

Computationally, it is easier and faster to implement two dimensional convolution in the frequency domain rather than in the spatial domain.

2.3 Image Filters

In dealing with acquired data from a camera, the majority of images are sensitive to additive noise. Noise is a high frequency occurrence in images. There are two common methods for removing noise: low- pass or median filtering. A low-pass filter uses a mask with dimensions $M \times N$. When a low-pass filter is applied to an image, all pixel values p change to the value of the average value of the $M \times N$ surrounding pixels. A 3×3 low-pass filter would look like:

$$p(x, y) = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (2.9)$$

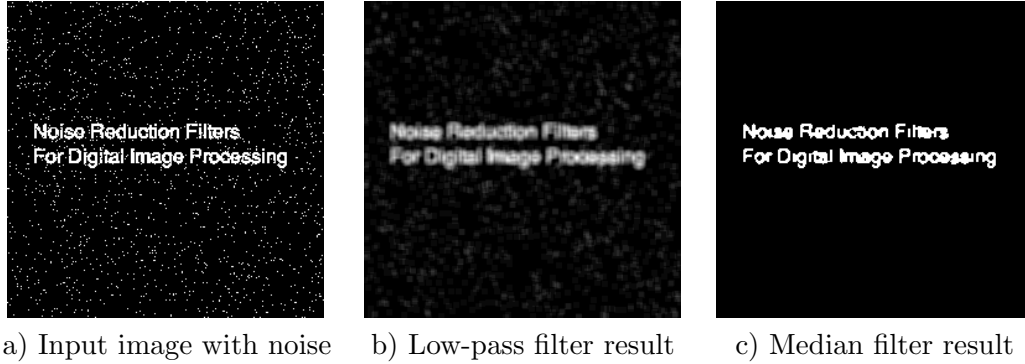


Figure 2.1: Gaussian noise is present in the image in a). A 5 x 5 pixel mask is used as the low-pass filter. The result is in b). Notice the blurring and removing of some the noise as well as the letters in the image. A 2 x 2 pixel mask is used for the median filter. The result is in c). All of the noise has been removed, but there is a little deformation of the letters.

The resulting image would be more smooth and have less sharp transitions of intensities. However, the noise has a blurring effect on the image once the filter has been applied. The negative side of using a low-pass filter, is that other important high frequency content i.e. edges are blurred as well. The larger the filter mask, the more noise will be smoothed out, but the blurring of edges will increase as well. A median filter is another method to remove noise from an image, but without blurring edges. The median of a set of values describes the number separating the higher half of a sample from the lower half. A median filter has $M \times N$ dimensions, and the pixel value assigned is the median value of the pixels in the surrounding pixels within the $M \times N$ neighborhood. This filter is very useful for removing sharp impulse noise, but does not remove or blur edges. In Fig 2.1, there is a noisy image in (a), a result from the low-pass averaging filter in (b), and a result from the median filter in (c). The low-pass averaging filter result does reduce the influence of noise, but blurs the entire image. The median filter removes all noise, but some of the edges around the letters are removed as well.

2.4 Multi-resolution processing

A common problem in computer vision is identifying objects that are small relative to the surroundings. Large distance from the observer to the object is the main cause of this.

In these cases, it can be difficult to extract valuable information, such as the gradients. For example, if a vehicle is approaching a stop sign, it is desirable to recognize the sign as soon as possible. From far away, the stop sign looks like a red blob, and as the distance decreases between the vehicle and sign, the size of the stop sign increases, which makes it easier to calculate the gradients. The gradients are important in this case, because the shape of the stop sign is unique, which serves as a method to identify it. Multi-resolution processing takes an image $f(x, y)$ as an input, performs interpolation on it, and outputs an image $g(x, y)$. Interpolation is the process of using known data to estimate values at unknown locations. The three most commonly used methods of interpolation are *nearest-neighbor*, *bilinear*, and *bicubic* interpolation.

Nearest-neighbor interpolation is computationally easy to implement compared with the other methods. The intensity value at $p(x, y)$ to be estimated, takes on the value of the intensity of its nearest-neighbor.

Bilinear interpolation uses the four nearest-neighbors of $p(x, y)$ to estimate the intensity value at location (x, y) on the new resolution grid. It is described by:

$$v(x, y) = ax + by + cxy + d \tag{2.10}$$

where the four coefficients are determined from four equations (one for each neighboring pixel) with four unknowns.

The intensity value at location (x, y) using *bicubic* interpolation is determined by:

$$v(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j \tag{2.11}$$

where the sixteen coefficients are solved for from the sixteen equations with sixteen unknowns obtained from the nearest sixteen neighbors. This method is the most computationally expensive of all of the interpolation methods.

2.5 Edge Detection Methods

Edges in physical objects often result in large local gradients in an image. Edge detection is the process of extracting points where there is a sharp change or gradient of image brightness in an image. The first order gradient of a digital image $f(x, y)$ is defined as the vector:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (2.12)$$

The magnitude of the gradient is defined as:

$$mag(\nabla f) = \sqrt{G_x^2 + G_y^2} \quad (2.13)$$

The edge detection result is the magnitude of the gradients in the x and y direction of an image. Detecting edges is useful for feature extraction or object recognition. There are many methods to detect edges: Sobel, Canny, Prewitt, and Roberts method as discussed in [10]. The main difference in these methods are the assigned weights of the kernels used to compute the gradients.

The Sobel operator calculates the gradient of an image intensity at every point. The Sobel kernels used to calculate the gradients in the x and y direction are:

$$\nabla_x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2.14)$$

$$\nabla_y = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (2.15)$$

If there is no change in a 3 x 3 pixel area, the kernel sums to 0, which means that there is no change in intensity over that particular area in the image. Performing two-dimensional convolution with the kernel in (2.14) and an input image will yield the gradient in the x-direction. Also, performing two-dimensional convolution with the kernel in (2.15) and an input image will yield the gradient in the y-direction. The Prewitt's kernels are

$$\nabla_x = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad (2.16)$$

$$\nabla_y = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad (2.17)$$

The weight value 2 on the Sobel kernels results in a smoother output. A Sobel edge detection result is displayed in Fig 2.2. The input is a synthetic image of rectangles. G_x and G_y are shown as images, as well as the magnitude of the gradients.

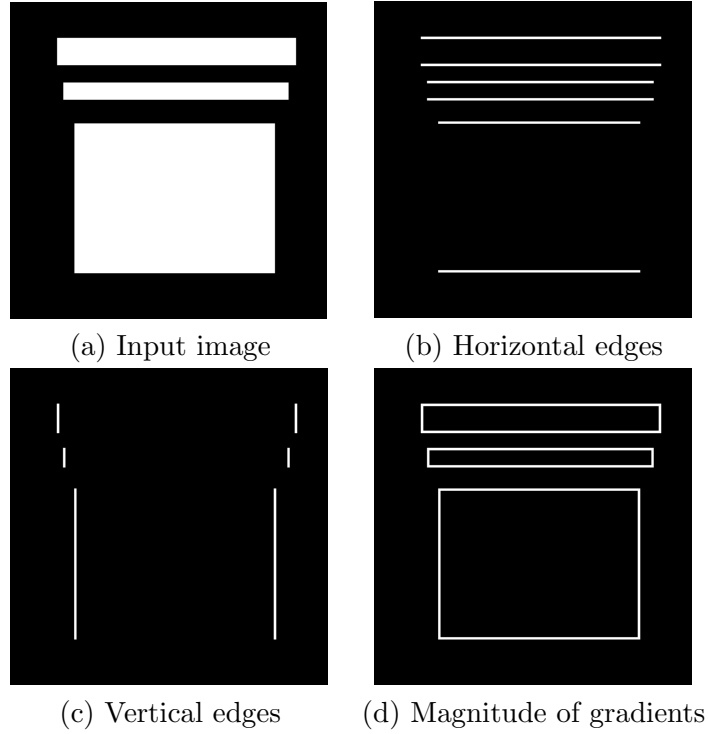


Figure 2.2: Sobel Edge detection

2.6 Hough Transform

Background

The Hough Transform is used to detect straight lines in an image. Consider the slope-intercept form $y = mx + b$, where m is the slope, and b is the y-intercept. It can describe a straight line between a pair of points on the Cartesian space. Computers have a problem in calculating m for vertical lines, which have a slope that does not exist (division by zero). Instead, a point is described in parameter space by the variables ρ and θ , where ρ is the distance between the line and the origin, and θ is the angle from the origin to the closest point on the line. This method for describing a line in parameter space is known as the Hough transform. The equation for a straight line can be written as:

$$y = -\frac{\cos\theta}{\sin\theta}x + \frac{\rho}{\sin\theta} \quad (2.18)$$

which is simplified to:

$$\rho = x\cos\theta + y\sin\theta \quad (2.19)$$

The parameter space of the image is divided into accumulator cells. The accumulator value $A(i, j)$ in a cell represented the number of points in the Cartesian coordinate system that lie on a straight line. As the number of accumulator cells increase, so does the accuracy of the colinearity of these points on the Cartesian coordinate system. Below is the procedure for implementing the Hough transform:

1. Compute the gradients of the image and choose a threshold to create a binary image.
2. Create sub-divisions for the accumulator cells in the parameter space
3. Evaluate the value of each accumulator cells. A higher value indicates a strong edge in the Cartesian coordinate system.

A plot of some points are plotted on the Cartesian coordinate system in Fig 2.3. There is a line connecting three points on the plot. The same points that are represented in the Cartesian coordinate system are represented in parameter space via the Hough transform in Fig 2.4. Each point from the Cartesian plot is represented as a cosine wave in parameter space. Wherever the cosine wave intersects another cosine wave, indicates a line in Cartesian space. The more cosine wave intersections at one location indicate a stronger line in cartesian space, or an edge in an image. An example of implementing the Hough transform on an image is in Fig 2.4.

2.7 Object Registration

2.7.1 Definition

Sets of data, acquired by sampling the same scene or object at different times or from different perspectives, will be in different coordinate systems. Object registration is the process of transforming the different sets of data into one coordinate system. The original image is known as the reference image. The image to be mapped is known as the target

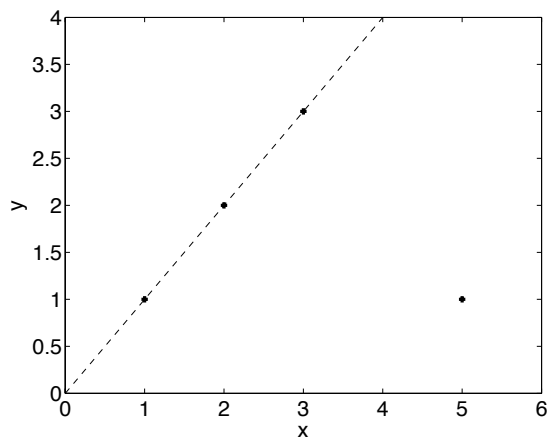


Figure 2.3: Points on the Cartesian coordinate system

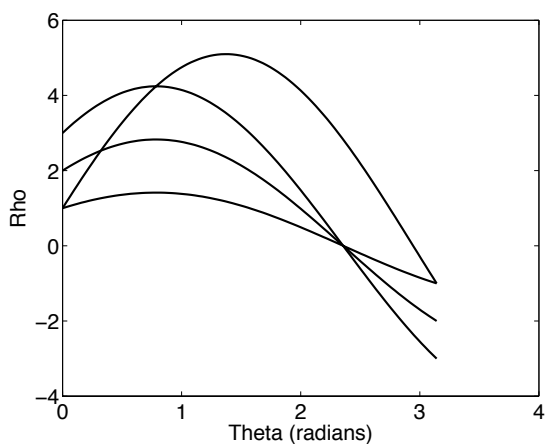


Figure 2.4: Points from the Cartesian system in Fig 2.3 transformed into parameter space.

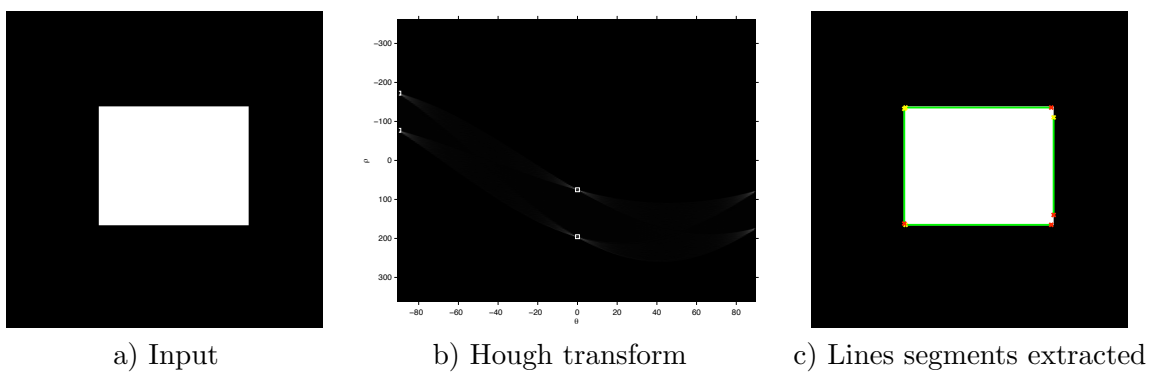


Figure 2.5: The Hough transform from the image in a) is shown as an image in b). The lines are extracted with the inverse Hough transform and plotted on top of the original image in c).

image. There are four main groups into which image registration is divided: different viewpoints, different times, different sensors, and scene to model registration. Different viewpoints (multiview analysis) is the process when images of the same scene are acquired from different viewpoints. The goal is to gain larger 2D or 3D representations of the scene. Different times (temporal analysis) is the process in which images of the same scene are acquired at different times. Different sensors (multimodal analysis) is when images of the same scene are acquired by different sensors. The goal is to integrate the information obtained from different sensors to gain a detailed scene representation. Scene-to-model registration compares the images of a scene to a computer model image of the same scene. The model can be a computer representation of the scene (for example digital elevation maps). The goal is to localize the acquired image in the scene/model to compare them.

2.7.2 Classifications

Area based algorithms look at the structure of the image with correlation metrics, Fourier properties, and other means of structural analysis. Instead of looking at the overall structure of images, feature-based algorithms concentrate on image features such as lines, curves, points, line intersections, and boundaries. Standard edge detection methods such as the Canny or the Laplacian detector can be implemented to extract these features.

2.7.3 Methods of Object Registration

Object registration can be implemented in either the spatial or frequency domain. Traditionally, image registration is used between a set of images of the same scene at different angles. The transformation of one image to the next is estimated. However in this research, the camera is moving, which introduces more complexities. The objects are changing sizes, color intensities can be brighter or darker due to shadows, and additive noise can disrupt image registration.

Spatial-domain

Image registration in the spatial domain chooses matching sets of control points between images. Features, structures, and textures are matching criteria. After the control points have been selected, the transformation from one to the other is estimated. Consider the four pairs of control points to be (v, w) and (x, y) for the input image and the reference image respectively. Bilinear approximation is given by:

$$x = c_1v + c_2w + c_3vw + c_4 \quad (2.20)$$

$$y = c_5v + c_6w + c_7vw + c_8 \quad (2.21)$$

The coefficients describe the transformation from the reference image to the input image.

In practice, obtaining a reference image with only stationary elements is not always possible, and building a reference from a set of images containing one or more moving objects becomes necessary. This applies particularly to situations describing busy scenes or in cases where frequent updating is required [15].

Frequency

Some algorithms use properties in the frequency domain to directly determine shifts between two images. The phase correlation method is used on a pair of images which produces a third image with a single peak. The location of this peak corresponds to the relative translation between the two images. The phase correlation algorithm is described by the following steps:

- Compute the 2D Fourier Transform
- Calculate the cross-power spectrum by taking the complex conjugate of the second image's Fourier transform. Then multiply the Fourier transforms together element wise, and normalize the product element-wise.
- Obtain the normalized cross correlation by applying the inverse Fourier transform

- Determine the location of the peak

2.7.4 Normalized Cross Correlation

Image cross correlation is defined as:

$$\gamma(x, y) = \frac{\sum_{s,t} [(f(s, t) - \bar{f}(s, t)) [h(x + s, y + t) - \bar{h}]]}{\sqrt{\{\sum_{s,t} [f(s, t) - \bar{f}(s, t)]^2 \sum_{s,t} [h(x + s, y + t) - \bar{h}]^2\}}} \quad (2.22)$$

where h (template) is the image of the object to be located within f . The primary reason to use image correlation is for matching. The cross correlation coefficient γ will be maximum where the template and image are co-located or displaced on top of each other. A generic output for normalized cross correlation is shown in Fig 2.6. The sharp peak represents the maximum cross correlation coefficient γ (see [15] for details). There is a disadvantage of using image correlation in that there is high probability that a window containing a smooth area without any prominent details will be matched incorrectly with other smooth areas in the reference image due to its non-saliency [10]. It is clear that normalized cross-correlation is not the ideal approach to feature tracking since it is not invariant with respect to imaging scale, rotation, and perspective distortions [16].

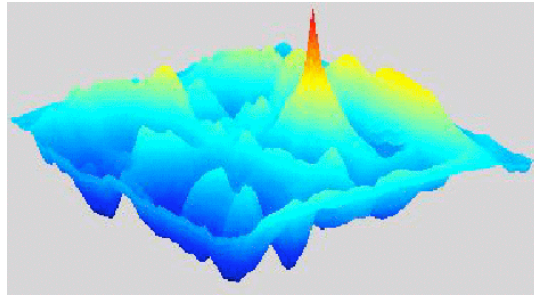


Figure 2.6: Normalized cross correlation output

2.8 Optical Flow

Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image. By calculating optical flow in a sequence of images, one can detect the motion

of objects or people. This detection could be very useful in many situations, such as for automatic cruise control in a vehicle, slow motion of geographical structures (i.e. glaciers), and other desirable situations where someone wants to track moving objects. There are many methods to calculate optical flow, such as the Horn-Schunck method or the Lucas-Kanade method. The main difference between these two methods is the constraints added to solve the problem of one equation with two unknowns. For this research, the Horn-Schunck method is arbitrarily chosen to be studied.

In an ideal environments a point on any object in motion does not change intensity over time. Therefore the derivative of intensity with respect to time must be zero:

$$\frac{dE}{dt} = 0 \tag{2.23}$$

Intensity of a pixel is defined as $E(x, y, t)$. The variables x and y are the position coordinates and t is time. Taking the partial derivative of $E(x, y, t)$ yields the equation:

$$\frac{\partial E}{\partial x} \frac{dx}{dt} + \frac{\partial E}{\partial y} \frac{dy}{dt} + \frac{\partial E}{\partial t} = 0 \tag{2.24}$$

To simplify the variables, denote the partial derivative of intensity of x with respect to time as E_x , and E_y and E_t respectively, and to simplify the equation change $\frac{dx}{dt}$ to u and $\frac{dy}{dt}$ to v . The variable u denotes the velocity in the x-direction, and the variable v denotes the velocity in the y-direction. The resulting equation is shown below.

$$E_x u + E_y v + E_t = 0 \tag{2.25}$$

In vector form:

$$\begin{bmatrix} E_x & E_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -E_t \tag{2.26}$$

The movement component is in the direction of the brightness gradient:

$$-\frac{E_t}{\sqrt{E_x^2 + E_y^2}} \quad (2.27)$$

This constraint is known as the aperture effect. The velocity components computed lie in the direction of the brightness gradient. This effect is evident on a barber pole as in Fig 2.7. The pole is actually rotating from left to right, but the gradients are moving up. This effect leads to inaccurate estimates of the direction of velocity of the object.



Figure 2.7: Barber pole

There is a small problem with equation (2.26). E_x , E_y , and E_t can be estimated. However, u and v must be calculated and are unknown. Therefore, there is one equation and two unknowns. An additional constraint needs to be derived to solve this equation.

Horn and Schunck introduce a smoothness constraint, which is the assumption that if an object is in motion, the neighboring pixels are moving with similar velocities, rather than every point on the brightness pattern moving independently. For example, if there is an image sequence containing a blue ball moving, then it is assumed that all of the pixels that comprise the ball are moving in the same direction. Horn and Schunck introduce the smoothness constraint by minimizing the square of the gradient magnitudes of the optical flow velocity:

$$\frac{\partial u^2}{\partial x} + \frac{\partial u^2}{\partial y} \quad \text{and} \quad \frac{\partial v^2}{\partial x} + \frac{\partial v^2}{\partial y} \quad (2.28)$$

The first step in calculating optical flow is estimating the partial derivatives of E_x , E_y , and E_t . Each estimate is the average of the first four differences taken over neighboring pixels. The partial derivatives are described by:

$$E_x = \frac{1}{4}(E_{i,j+1,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i+1,j,k} + E_{i,j+1,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i+1,j,k+1}) \quad (2.29)$$

$$E_y = \frac{1}{4}(E_{i+1,j,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i,j+1,k} + E_{i+1,j,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i,j+1,k+1}) \quad (2.30)$$

$$E_t = \frac{1}{4}(E_{i,j,k+1} - E_{i,j,k} + E_{i+1,j,k+1} - E_{i+1,j,k} + E_{i,j+1,k+1} - E_{i,j+1,k} + E_{i+1,j+1,k+1} - E_{i+1,j+1,k}) \quad (2.31)$$

The box in Fig 2.8 is a visualization of the pixels used to estimate the partial derivatives.

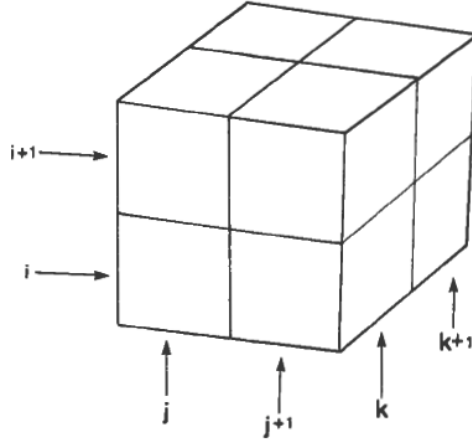


Figure 2.8: All three partial derivatives at the center are calculated from the average of the first four differences of adjacent pixels. The column index is denoted as j , the row index is denoted as i , and the time index is denoted as k .

The Laplacians of the flow velocities u and v need to be estimated as well. The estimations can be written as

$$\nabla^2 u = k(\bar{u}_{i,j,k} - u_{i,j,k}) \quad \text{and} \quad \nabla^2 v = k(\bar{v}_{i,j,k} - v_{i,j,k}) \quad (2.32)$$

where the local averages of \bar{u} and \bar{v} are:

$$\bar{u}_{i,j,k} = \frac{1}{6}[u_{i-1,j,k}+u_{i,j+1,k}+u_{i+1,j,k}+u_{i,j-1,k}] + \frac{1}{12}[u_{i-1,j-1,k}+u_{i-1,j+1,k}+u_{i+1,j+1,k}+u_{i+1,j-1,k}] \quad (2.33)$$

$$\bar{v}_{i,j,k} = \frac{1}{6}[v_{i-1,j,k}+v_{i,j+1,k}+v_{i+1,j,k}+v_{i,j-1,k}] + \frac{1}{12}[v_{i-1,j-1,k}+v_{i-1,j+1,k}+v_{i+1,j+1,k}+v_{i+1,j-1,k}] \quad (2.34)$$

After the estimations have been made for the partial derivatives and the Laplacians, the problem becomes a minimization of the error in the constant brightness equation in (2.25) and the departure from smoothness in (2.28).

The total error to be minimized becomes:

$$\xi^2 = \int \int (\alpha^2 \xi_c^2 + \xi_b^2) dx dy \quad (2.35)$$

where ξ_c is the error from the departure of smoothness, ξ_b is the error from the brightness equation, and α is a weighting value. The value α plays a significant role only for areas where the brightness gradient is small, preventing haphazard adjustments to the estimated flow velocity occasioned by noise in the estimated derivatives [13]. Using a least squares minimization on (2.35), and substituting the estimates for the partial derivatives and Laplacians yields the iterative solution:

$$u^{n+1} = \bar{u}^n - E_x \left(\frac{E_x \bar{u}^n + E_y \bar{v}^n + E_t}{\alpha^2 + E_x^2 + E_y^2} \right) \quad (2.36)$$

$$v^{n+1} = \bar{v}^n - E_y \left(\frac{E_x \bar{u}^n + E_y \bar{v}^n + E_t}{\alpha^2 + E_x^2 + E_y^2} \right) \quad (2.37)$$

Optical flow is calculated using synthetic images as inputs. The inputs and results are shown in Fig 2.9.

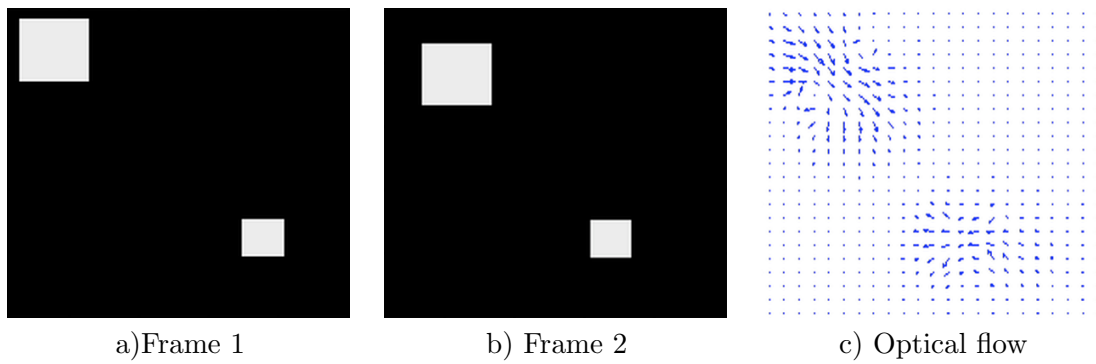


Figure 2.9: Optical flow calculated between the two input images. The block in the top right corner is moving diagonally down from left to right, and the block in the bottom right corner is moving from right to left.

CHAPTER 3
EXPERIMENTS AND RESULTS

This chapter presents an overview of validating the digital image processing techniques discussed in the previous chapter. Synthetic data and acquired data are both used in this validation. Synthetic data is useful to test digital image processing algorithms on, due to the fact that they can be free from noise, artifacts from video coding, and background clutter. After the object registration methods are validated on the synthetic data, they are used on the acquired data from the digital camera. Results are discussed.

3.1 Synthetic Data

Sobel Edge Detection

The synthetic image in Fig 3.1 is used as the input to the Sobel edge detection algorithm. The magnitude of the gradients in the x and y direction is displayed in Fig 3.2(a).

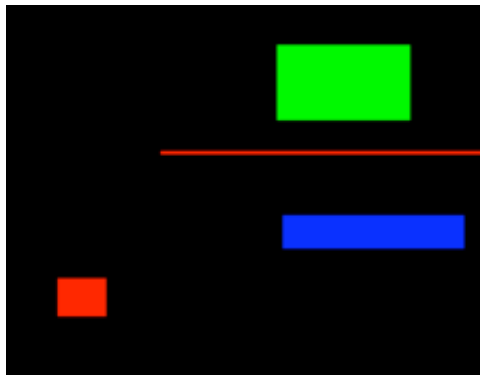


Figure 3.1: Synthetic image for object registration

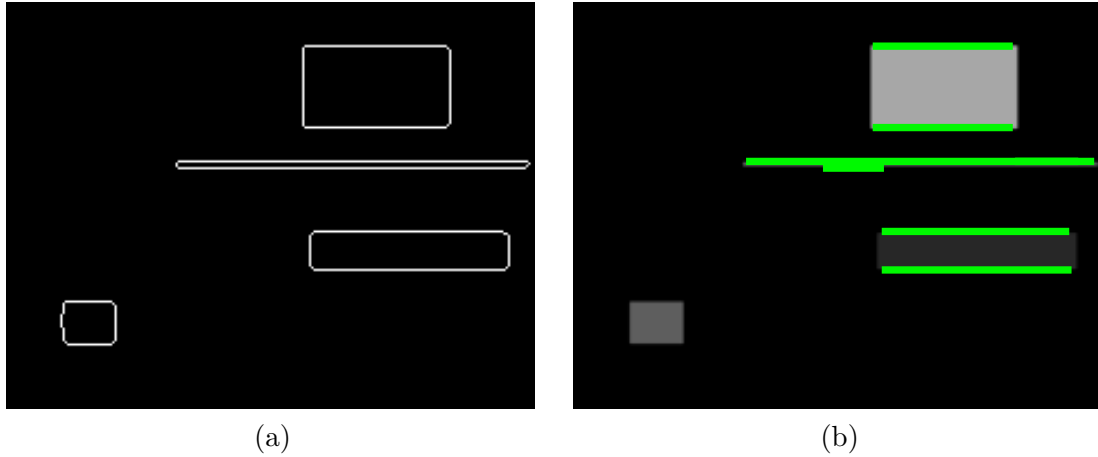


Figure 3.2: (a) The magnitude of the x and y gradients using the Sobel kernels. (b) Lines extracted from the Hough transform are plotted on top of the grayscale input image.

Hough Transform

The result from the Sobel edge detection is used as the input to the Hough transform. The lines are extracted from the Hough transform and plotted on top of the original input image in grayscale in Fig 3.2(b). The Hough transform detects the strongest edges or gradients in the image. Not all edges were detected by the Hough transform in this experiment. The Hough transform extracted the longest lines in the image, which compose of the larger blocks. None of the edges of the block in the lower left-hand corner were extracted. Even if the Hough transform detects all of the edges, there is no method to determine which block is which from this information alone.

Normalized Cross Correlation

The red block is cropped and stored in memory. This red block will be assigned as the function $h(x, y)$ in the normalized cross correlation equation. The image in Fig 3.1 will be stored as the function $f(x, y)$. After the cross-correlation coefficients γ are calculated for all possible shifts of $h(x, y)$ onto $f(x, y)$, the maximum coefficient is where there is a match of $h(x, y)$ and $f(x, y)$. The output is shown as a surface plot of all of the normalized cross correlation coefficients in Fig 3.3. The red block is the group of coefficients to the left. The

group to the right is from the red straight line. The highest correlation value is 1, which is near the middle of the red block. This method successfully identified the template $h(x, y)$ within the image $f(x, y)$.

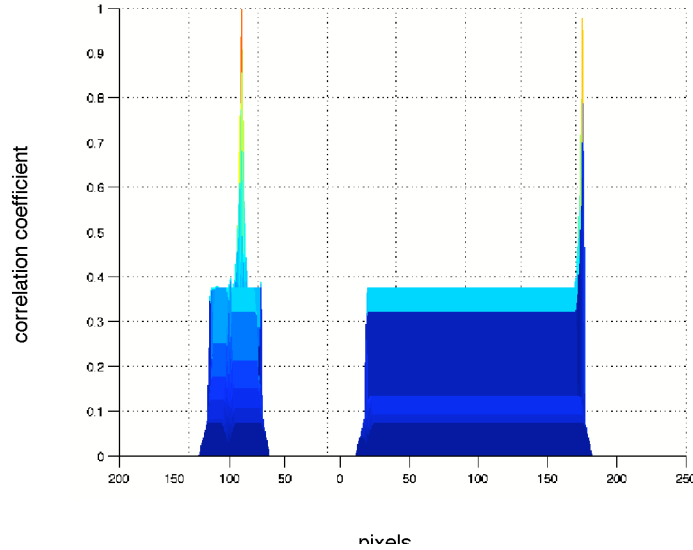


Figure 3.3: This is a surface plot of the magnitude of the normalized correlation coefficients. The red block is on the left, and the red straight line is on the right. Notice that the maximum coefficient is in the middle of the actual red block.

3.2 Acquired Data

Digital video was obtained in an arena designed for a robotic competition. Four different colored blocks were placed in front of a robot. The background and rink introduce additional difficulties such as changing colors on parts of the floor, different textured surfaces such as pebbles, and a window. A digital camera was placed on top of a robot in the rink, and was pushed forward for a couple of seconds. A total of 299 images were used as data to process. A frame taken from this video is in Fig 3.4. In this section, the edge detection, Hough transform, and the normalized cross correlation operations are applied to this image.

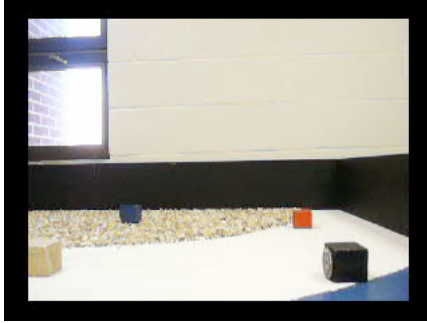


Figure 3.4: The output from the digital camera. Included in the image are the four blocks to be detected. There is a black background on the rink, there is a blue and white surface, and there are some pebbles.

3.2.1 Edge Detection

The Sobel edge detection discussed in the previous chapter is used on the image in Fig 3.4. The magnitude of the gradients are in Fig 3.5. As one can see, the presence of the pebbles introduces some noise. The edges of the window, rink, and the change of color on the floor are detected, as well as the blocks.

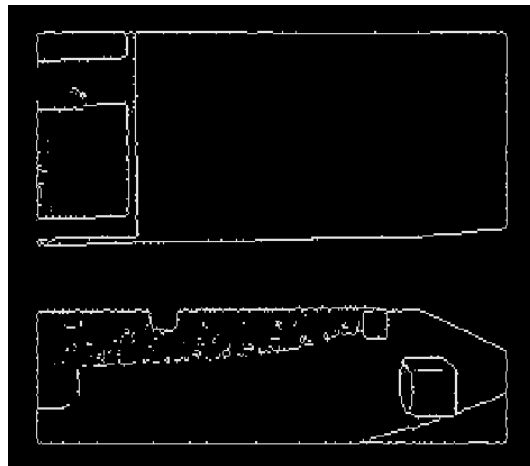


Figure 3.5: Edge detection on the acquired data

3.2.2 Hough Transform

The Hough transform as discussed in the previous chapter is applied to the edge detection result in Fig 3.5. The results of the Hough transform are in Fig 3.6. As expected, the

strongest lines are extracted, which are the lines comprising of the rink and the window. It is apparent at this point that the edge detection and Hough transform operations alone are not very useful in registering objects.



Figure 3.6: Hough transform on the acquired data

3.2.3 Normalized Cross Correlation

Finally the normalized cross correlation operation is performed on the acquired data. The red-block is stored as the template $h(x, y)$ for normalized cross correlation in equation (2.22). The input image from Fig 3.4 is stored as the function $f(x, y)$. The output is a matrix of correlation values displayed as a surface plot in Fig 3.7. The red color on the surface plot coincides with a high correlation value, and the blue color represents a low correlation value. There is strong correlation across the lower borders of the rink, the edge of the pebble surface, and the some bricks in the background. The red-block has a medium correlation coefficient (light blue color).

3.3 Comments

The edge detection and Hough transform are useful methods to extract the gradients of the images. Normalized cross correlation works well with the synthetic data, but has issues with the acquired data. An issue in using normalized cross correlation is that $h(x, y)$ and

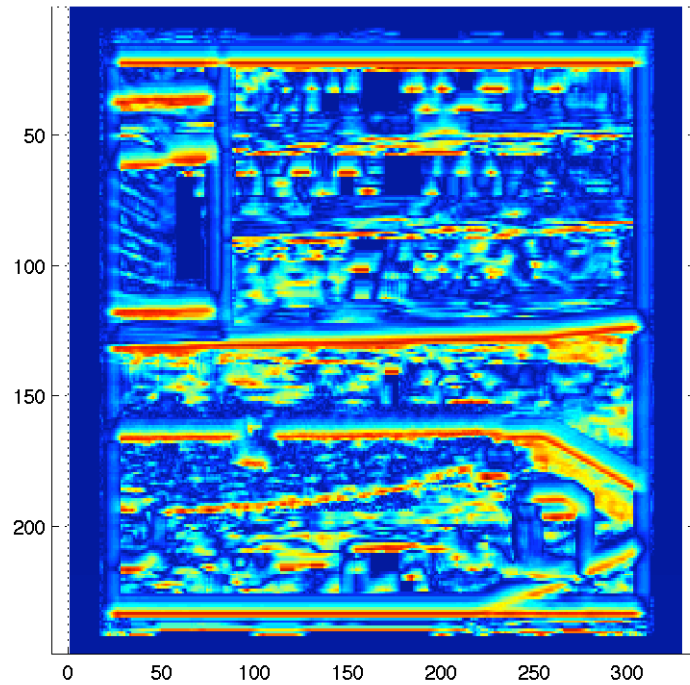


Figure 3.7: The output is a surface plot of the normalized cross correlation coefficients γ . The red surface indicates a high correlation coefficient and blue represents a low correlation coefficient.

$f(x, y)$ need to be in grayscale. Intensity information in RGB space is lost in this process. As a result, there are many areas away from the block with high correlation coefficients. It is evident that there needs to be an algorithm that uses these digital image processing techniques in combination to form a robust computer vision algorithm.

CHAPTER 4

STATISTICAL-GRADIENT OBJECT REGISTRATION

This chapter presents a new method for object registration. This method detects objects by combining a series of two identification methods. There are two common types of methods that are used to identify objects: area-based and feature-based, as discussed in [10]. Area-based classifications look at the structure of the image with correlation metrics or Fourier analysis. Feature-based methods concentrate on the overall structure of the image, such as edges, line intersections, and curves. The first identification method used is based on local averages and variance of surrounding pixels. The result is a binary image, which is used with the second method which uses the Hough transform to extract the gradients.

4.0.1 Area-Based

This area-based method uses six parameters to classify an object: the local averages and variance for the red, green, and blue channels in a given pixel window that compose a digital image. A template $h(x, y)$ contains the object that is desired to be located in an image $f(x, y)$. The object in $h(x, y)$ is divided in the computer as a three dimensional array ($M \times N \times 3$). The three dimensions are composed of pixel values for the red, green, and blue space. Statistical analysis is used to characterize each color matrix. First, the average values for each color matrix is calculated. Then the variance for each pixel in the template $h(x, y)$ is calculated. After the template $h(x, y)$ has been classified with this method, the image $f(x, y)$ is searched for areas that match the parameters for the template $h(x, y)$. If a pixel in $f(x, y)$ matches the parameters given by $h(x, y)$, then the location of the pixel is mapped into a separate output 2-dimensional matrix with the same row-column size of the image $f(x, y)$. A tolerance value is utilized that will allow values that have small deviation from the calculated parameters to be accepted as a match. Refer to the appendix

for specific tolerance values. Note that this method fails to incorporate any information regarding the shape of the object. This fault could have minimal effect, however if there are many objects of similar color. then it would be difficult to determine with confidence that the object detected is the desired one. Therefore, the result from this area-based method, is tested with the Hough transform to match the shape of the object.

4.0.2 Feature-Based

The output matrix from the proposed area-based method is then used as the input to the Hough transform. After the Hough transform is performed, the output lines are sent to an algorithm that checks the angle of intersection between the line segments. From the synthetic and acquired data discussed in Section 3, it is evident that the objects are cubes. Therefore, the intersection of the gradients of the image are perpendicular. To calculate the angle of intersection, the slope of each line is calculated. After all of the slopes are calculated, the algorithm compares the negative inverse slope to that of all the other slopes. If they are equal within a controllable window of tolerance, then the lines are perpendicular. If there is no match, the line segment is thrown out. If the lines intersections are perpendicular, then the probability of the detected region being the template $h(x, y)$ is high. If the angles are not perpendicular, then there is some error.

An experiment of randomly distributed line segments are used as inputs to test gradient method for determining if line segments are perpendicular within 3 degrees of tolerance. The plot in Fig 4.1 contains the randomly created line segments and the output, which is a plot of the lines that are perpendicular. It appears that other sets of lines are perpendicular to each other, which is the reason that the tolerance of error (in degrees) can be adjusted. This procedure works on these randomly created line segments, so the next step is to use this method on the real line segments extracted from the Hough transform.

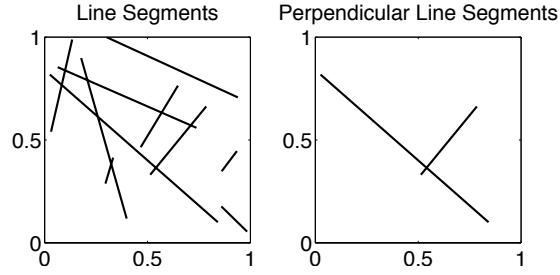


Figure 4.1: The plot on the left contains 10 randomly generated line segments. The plot on the right are the line segments that are perpendicular to each other.

4.0.3 Code Outline

This section provides an outline of the software used to implement the statistical-gradient object registration method.

Inputs- the template $h(x, y)$ and the image $f(x, y)$

Output- an image of the detected object with lines drawn around it

1. calculate $N_h \times M_h$ of the template $h(x, y)$ and $N_f \times M_f$ input image $f(x, y)$
2. calculate the average value and variance for the pixel values in the template $h(x, y)$ for each RGB channel.
3. create an averaging mask with the row-column dimensions $N_h \times M_h$
4. increase the resolution of $f(x, y)$ to $2N_f \times 2M_f$ with nearest-neighbor interpolation
5. perform two-dimensional convolution with the averaging mask and the input image $f(x, y)$ for each RGB channel separately.
6. calculate the variance for each pixel in the input image $f(x, y)$ within the two-dimensional space spanned by the size of the template $h(x, y)$ and the variance of the output from the two-dimensional convolution with the input image $f(x, y)$ and the averaging mask for each RGB channel separately.
7. set the window of tolerance for accepting values as a match with the template

8. search the average values and variance from the input image $f(x,y)$ in each RGB channel and compare with the six parameters calculated from the template $h(x,y)$.
9. if the values satisfy all six parameters store the pixel locations in a separate output matrix $out(x,y)$ with row-column dimensions $2N_f \times 2N_f$
10. use the output matrix $out(x,y)$ as the input to the Hough transform
11. calculate the slope in degrees relative to the horizontal for all line segments
12. verify the line segments are perpendicular
13. calculate the center and area of the object
14. plot all perpendicular line segments onto the input image $f(x,y)$ with row-column dimensions $2N_f \times 2M_f$.

To experimentally validate this statistical-gradient object registration method, synthetic data (Fig 4.2) and acquired data (Fig 3.4) are used as the input images. Synthetic data is useful to test digital image processing algorithms on, because there is no noise, no artifacts from the video coding, and a less busy background. After the object registration method work on the synthetic data, it is used on the acquired data from the digital camera.

4.0.4 Validation using Synthetic Images

A synthetic red-block is created and placed in the bottom left-hand corner. There is also a line with the same characteristics along with blue and green blocks added to the image. The RGB values for the red-block were set to the values in Table 4.1. A 6 x 6 pixel block was created to resemble the real red block on the real surface from the acquired data. The dimensions of the block were chosen because it was desired to have a block that was smaller than other blocks in the image. Having a smaller block introduces a higher level of difficulty for digital image processing algorithms. Table 4.1 describes the synthetic block according to its expected value and variance of each red, green, and blue parameters. An

image is created with the red block in Table 4.1 along with a green block, blue rectangle, and a red line with the same characteristics as the synthetic block described by Table 4.1. This image is displayed in Fig 4.2.

Table 4.1: Synthetic Block Characteristics

	Expected Value	Variance
Red	219	0
Green	60	0
Blue	54	0

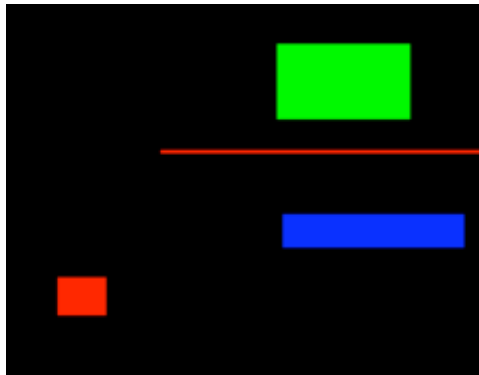


Figure 4.2: Synthetic image for object registration

Statistical-gradient Object Registration

This method is tested on the synthetic image in Fig 4.2, and the output image is shown in Fig 4.3. As seen in Fig 4.3, the red block was detected.

4.0.5 Validating on Acquired Images

The next step is to test this method on the acquired data, which has more background noise. To demonstrate the algorithm on more realistic data, digital video was obtained in an arena designed for a robotic competition. A total of 299 images were used as data to process. The thirty-seventh frame is shown in Fig 3.4. The statistical-gradient method is applied to the image sequence.

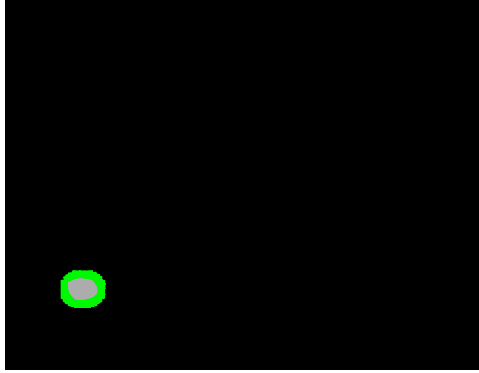


Figure 4.3: The output from the algorithm used on the input from Fig 4.2.

Registering the Red Block

The red-block was copied from the input image in Fig 3.4 and saved as the template $h(x, y)$. The pixel values of each RGB channel of $h(x, y)$ were processed with digital image processing techniques. The statistical values from this operation is given in Table 4.2. The statistical-gradient registration method is performed on the entire image sequence (299 frames) taken from the digital camera. The gradient result can be seen in Fig 4.4. All of the the lines are perpendicular within 9 degrees of tolerance. The same template is used for the entire image sequence. This registration method located the template $h(x, y)$ in all of the 299 of the input images $f(x, y)$. If $h(x, y)$ is smaller than the object in $f(x, y)$, it appears to have minimal effect on the outputs.

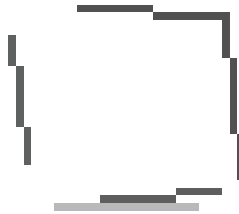


Figure 4.4: The output from the line segment algorithm on the lines extracted from the Hough transform on the output from the object identification method.

Table 4.2: Red Block Characteristics

	Expected Value	Variance
Red	202.30	85.08
Green	71.41	40.07
Blue	32.90	58.00

Four output frames are displayed in Fig 4.5. A vector from the origin to the centroid of the block is plotted on top of the image. The location of the center of the registered object is an output, as well as the area of the object. The area of the object is used to interpret if the object is moving closer or farther away. The area of the block generally increases as the camera moves closer.

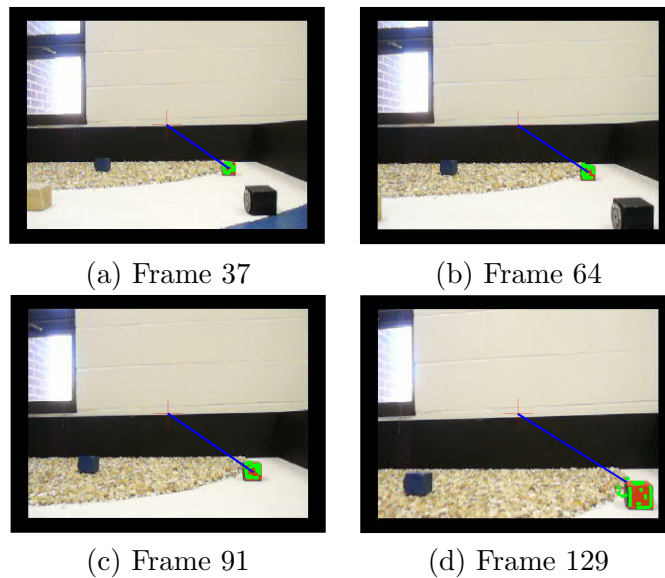


Figure 4.5: Four outputs of the registered red block

Registering the Blue Block

Additionally, the blue block in the image is registered as well. The template $h(x, y)$ is the blue block. The six statistical parameters used to classify the object are in Table 4.3. Four output frames from the video sequence are in Fig 4.7

Frame	X position	Y position	Area (<i>pixels</i> ²)
5	326	441	113.12
10	321	442	305.25
28	331	447	242.25
37	327	446	414.62
50	343	445	477.62
64	326	452	520.00
75	344	468	469.00
91	358	496	691.50
110	378	512	1387.38
129	392	548	2071.25

Figure 4.6: The center of the registered object red block in ten selected frames from the image sequence is shown above.

Table 4.3: Blue Block Characteristics

	Expected Value	Variance
Red	51.08	40.28
Green	59.94	16.62
Blue	82.26	38.85

4.0.6 Motion Prediction

It is assumed the objects are moving at a constant velocity, therefore it reasonable to predict where it will be in the next frame. If the location of the object can be accurately estimated, then the search area of the input frame $f(x, y)$, can be significantly reduced. Knowledge of the location of the object would increase the processing time, as well as eliminate false detections. The open loop estimation of a moving object in an image sequence can be described by the following equation:

$$Obj_{k+1}(x, y, z) = \begin{bmatrix} x_k + (x_{k-2} - x_{k-1}) \\ y_k + (y_{k-2} - y_{k-1}) \\ z_k + (z_{k-2} - z_{k-1}) \end{bmatrix} \quad (4.1)$$

where the variables x, y, z refer the the x and y coordinate of the center of the registered object, and z refers to the size of the registered object. Fig 4.8 and Fig 4.9 contain plots

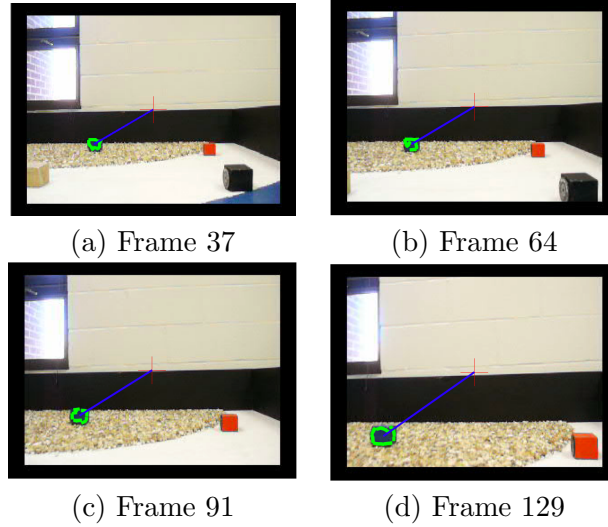


Figure 4.7: Four outputs of the registered blue block

of the estimations of the centers for the blue and red blocks using equation in (4.1). The largest prediction error for the location of the center of the object for the x-coordinate is about 20 pixels. This error is the same for the prediction of the center of the object for the y-coordinate. The prediction of the area of the block error is significantly higher, due to the nonlinear growth of the area of the block as it gets closer to the camera. The mean, variance, and standard deviation of the motion prediction of the red and blue blocks are in Table 4.4 and Table 4.5, respectively. To further reduce these error predictions, a Kalman filter could be implemented.

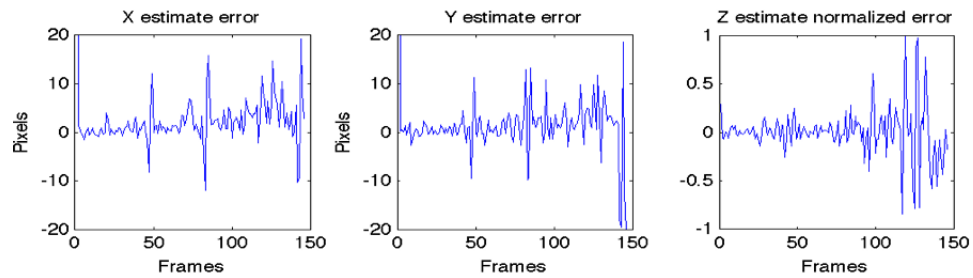


Figure 4.8: Plots of the X estimation, Y estimation, and Z estimation of the red block.

Table 4.4: Motion Prediction Error for the Red Block

	Expected Value (<i>pixels</i>)	Variance (<i>pixels</i> ²)	Standard Deviation (<i>pixels</i>)
X	2.04	30.79	5.54
Y	1.39	58.87	7.67
Z	11.61	42,825	206.94

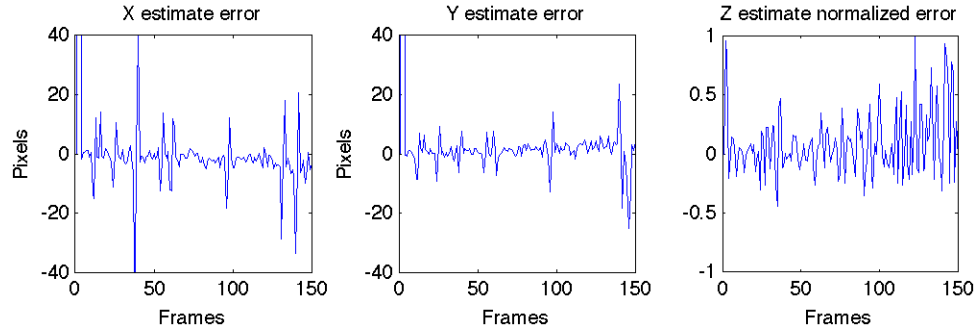


Figure 4.9: Plots of the X estimation, Y estimation, and Z estimation of the blue block.

4.0.7 Stop Sign Registration

To further validate the statistical-gradient registration method, video data was recorded with a web cam placed on the top of a car. The video data was divided into 55 images. The video displays the car driving toward a stop sign. There is a railroad crossing sign behind the stop sign, as well as lots of trees and dirt to add clutter to the background. The stop sign is the desired object to track in this video sequence. Two shades of red are stored as the template $h(x, y)$ to search for within the input images, because as the distance decreases between the vehicle and the sign, the color intensity gets brighter. The results from the statistical-gradient registration method are shown in Fig 4.10. The white lettering on the stop sign causes the algorithm to detect the stop sign in two sections or just finds blocks of red on the sign. When the stop sign is far away in Frame 11, the sign is so small that the gradients (green lines) do not compose an octagon.

Table 4.5: Motion Prediction Error for the Blue Block

	Mean (<i>pixels</i>)	Variance (<i>pixels</i> ²)	Standard Deviation (<i>pixels</i>)
X	.9052	78.02	8.8
Y	.9118	23.10	4.81
Z	32.73	11,930	109.22

4.0.8 Multiple Green Signs Registration

To further validate the statistical-gradient registration method, video data taken from the dashboard inside a vehicle contains multiple green signs on the side of the road. Each sign is located a distance of 200 feet apart. Fig 4.11 contains a frame from the video. The green sign is used as the template $h(x, y)$. The output images are in Fig 4.12

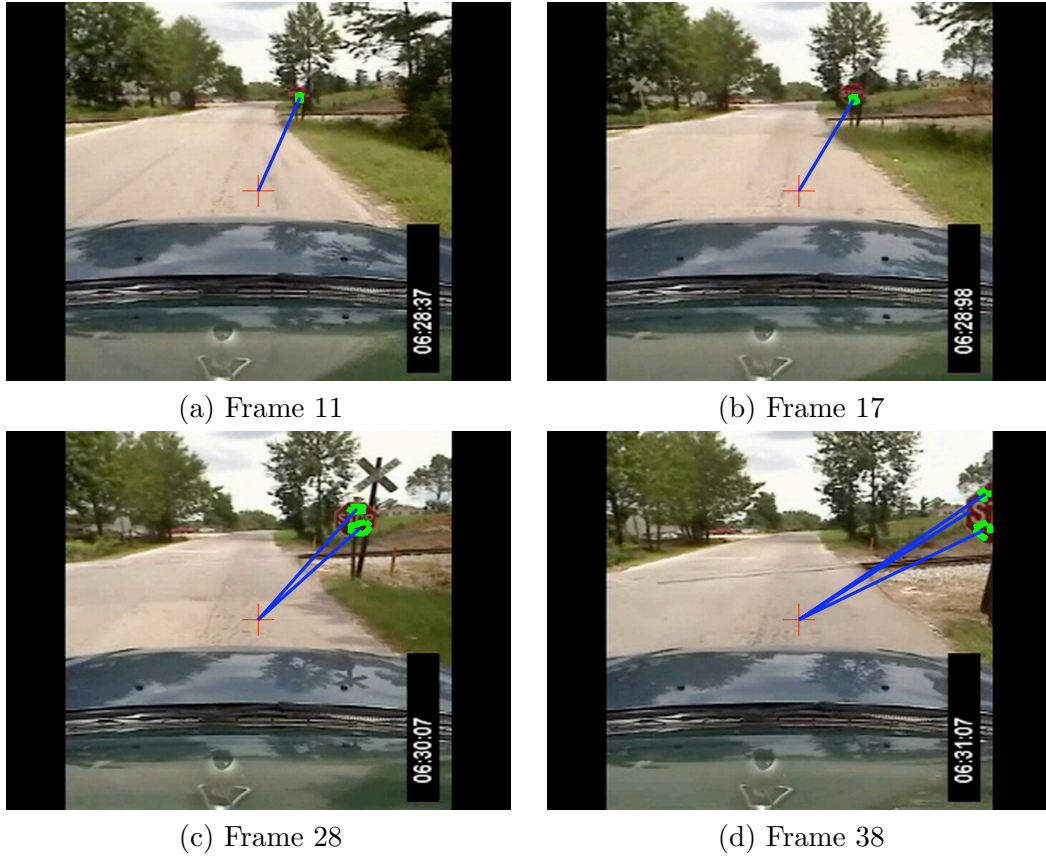
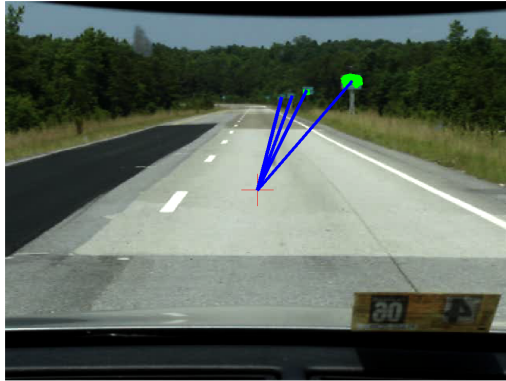


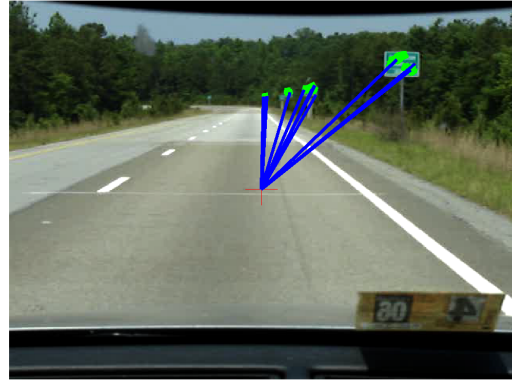
Figure 4.10: Four outputs from the object orientation method.



Figure 4.11: A frame of the video sequence of the green signs.



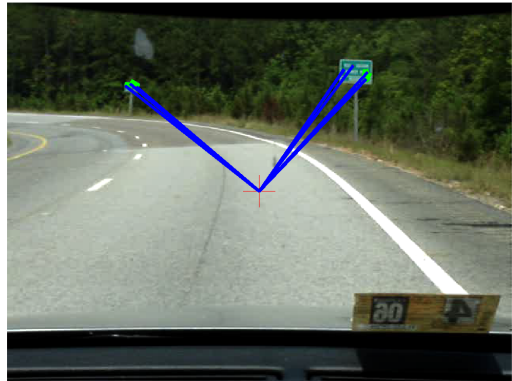
(a) Frame 1



(b) Frame 153



(c) Frame 229



(d) Frame 278

Figure 4.12: Four outputs from the video sequence of the green signs

CHAPTER 5

CONCLUSIONS

This thesis has presented multiple object registration methods to address detection of objects in a different settings, such as indoor and outdoor environments. These registration methods work well on noiseless synthetic data, but prove to be troublesome in noise corrupted data obtained from a camera. Quantization noise was seen to be a major source of corrupt data in digital image processing, as well as blurring from motion of the camera, or an object.

Optical flow was seen to be difficult to implement in object registration and tracking for a number of reasons. For the image sequences used in this research, the camera was in motion. All objects and the background are moving. This motion makes it difficult to register objects that are still. If other objects were in motion, the relative velocities could then be segmented.

Feature based methods such as edge detection and the Hough transform provided useful results to begin classifying objects. However, there needs to be a higher level of computer processing to register the object. Statistics of areas of pixels can provide useful information to be extracted for object registration as well. By combining these two methods, it ensures that the template $h(x, y)$ has been successfully matched to itself in the input image $f(x, y)$.

The red and blue blocks from the first experimental study were registered in every frame of the video sequence. The resolution is changed via nearest-neighbor interpolation to enlarge the gradients when the block is distant from the camera. As the block is closer to the camera, the gradients become a little distorted due to the resolution increase, as was seen in Fig 4.5(d).

The algorithm for the stop sign and green sign images also exhibited some limitations. Because the template $h(x, y)$ is small relative to the objects as the camera moves closer

to them. The gradients also change for the objects. When the stop sign is far away, the shape is not an octagon, but a red blur. As the stop sign gets closer, the octagon shape is recognizable.

CHAPTER 6

FUTURE WORK

Future work entails more extensive testing of the algorithm, as well as implementing it in real-time. Experimental validation through a leader-follower demonstration is another goal for the future. An experiment set-up could be to visually tag a robot with a specific mark that the follower robot would be searching for. After the demonstration is successful, a larger scale experiment could be used in fusion with other sensors such as IMU or GPS to follow the leader vehicle.

Another goal is to write algorithms for the leader vehicle. At this point in the research, an algorithm has been written to register objects. However, an algorithm for the leader vehicle to identify good features to track is important. The leader vehicle must survey the environment, and decide what objects are other features of the image are good features to track. After the leader vehicle has done this, it will transmit this data to the follower vehicle.

BIBLIOGRAPHY

- [1] A. Broggi, P.Cerri, and P.C. Antonello, "Multi-resolution vehicle detection using artificial vision", in Proc. Intelligent Vehicle Symp., 2004, pp 310-314
- [2] Committee on Army Unmanned Ground Vehicle Technology, "Technology Development for Army Unmanned Ground Vehicles", The National Academies Press, Washington, D.C. 2002.
- [3] Sheikholeslam , and CA. Desoer, "Longitudinal control of a platoon of vehicles with no communication of the lead vehicle information", Proc. of American control conference 3, pp 3102-3106, 1991.
- [4] Michael Price and Steven Munkeby. "Future Combat Systems Autonomous Navigation System Technology Revolutionize Warfare." 2008
- [5] Teck Chew Ng, Javier Guzman, and Martin Adams. "Autonomous Vehicle-Following Systems: A Virtual Trailer Link Model"
- [6] D.B. Edwards, T.A. Bean, D.L. Odell, and M.J. Anderson." A Leader-Follower Algorithm for Multiple AUV Formations."
- [7] E. M. Atkins, J. A. Lennon. "Color-based Vision Tracking for an Astronaut EVA Vehicle" 2001
- [8] Hariprasad Kannan, Vilas K. Chitrakaran, Darren M. Dawson, and Timothy Burg. " Vision-Based Leader/Follower Tracking for Nonholonomic Mobile Robots." In Proc. American Control Conference. pp 2159-2164, 2007
- [9] Andrea Giachetti, Marco Campani, and Vincent Torre. "The Use of Optical Flow for Road Navigation." IEEE Robotics and Automation, Vol 14, No.1, February 1998.
- [10] Flusser, Jan. Barbara Zitova. "Image registration methods: a survey," Image and Vision Computing 21 (2003) 9771000
- [11] Strzodka, R.; Ihrke, I.; Magnor, M., "A graphics hardware implementation of the generalized Hough transform for fast object recognition, scale, and 3D pose detection," Image Analysis and Processing, 2003.Proceedings. 12th International Conference on , vol., no., pp. 188-193, 17-19 Sept. 2003
- [12] Chia, A.Y.S.; Leung, M.K.H.; How-Lung Eng; Rahardja, S., "Ellipse Detection with Hough Transform in One Dimensional Parametric Space," Image Processing, 2007. ICIP 2007. IEEE International Conference on , vol.5, no., pp.V -333-V -336, Sept. 16 2007-Oct. 19 2007

- [13] B.K.P. Horn and B.G. Schunk. Determining Optical Flow. *Artificial Intelligence*, 17:185-203, 1981
- [14] Gemeiner, P.; Ponweiser, W.; Einramhof, P.; Vincze, M., "Real-Time SLAM with a High-Speed CMOS Camera," Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on , vol., no., pp.297-302, 10-14 Sept. 2007
- [15] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 2nd ed. Prentice Hall, 2002.
- [16] J.P. Lewis. "Fast Normalized Cross-Correlation", Industrial Light and Magic
- [17] Briechele, K.; Hanebeck, U.D., "Self-localization of a mobile robot using fast normalized cross correlation," Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on , vol.4, no., pp.720-725 vol.4, 1999
- [18] Richard O. Duda, Peter E. Hart. "Use Of The Hough Transformation To Detect Lines and Curves In Pictures" Comm.ACM, vol 15, No. 1, pp. 11-15 (January 1972).
- [19] Kumar, Suvesh; Rishiwal, Vinay; Joglekar, P.N., "Area Based Approach to Registration of SAR Images with Sub-pixel Accuracy," TENCON 2006. 2006 IEEE Region 10 Conference , vol., no., pp.1-4, Nov. 2006

APPENDICES

APPENDIX A
NOMENCLATURE

$f(x, y)$ - image as a two dimensional function

$h(x, y)$ - template

$p(x, y)$ - pixel value at location (x, y) in an image

G_x - gradient with respect to x of an image

G_y - gradient with respect to y of an image

$F(u, v)$ image represented in the frequency domain

$A(i, j)$ - accumulator value of Hough transform at location (i, j) in parameter space

ρ - the distance from the line to the origin for the Hough transform

θ - the angle of the vector from the origin to the closest point

\bar{u} - average velocity in the x -direction

\bar{v} - average velocity in the v -direction

γ - two-dimensional normalized cross correlation coefficient

APPENDIX B
ORIENTATION RESULTS OF RED BLOCK

Frame= 1	X_pos= 0.00	Y_pos= 0.00	Area= 0.00
Frame= 2	X_pos= 441.94	Y_pos= 322.30	Area= 203.50
Frame= 3	X_pos= 441.77	Y_pos= 322.44	Area= 197.25
Frame= 4	X_pos= 441.96	Y_pos= 322.55	Area= 183.62
Frame= 5	X_pos= 441.85	Y_pos= 325.72	Area= 113.12
Frame= 6	X_pos= 441.24	Y_pos= 317.53	Area= 114.62
Frame= 7	X_pos= 441.07	Y_pos= 325.91	Area= 173.38
Frame= 8	X_pos= 440.78	Y_pos= 323.83	Area= 143.50
Frame= 9	X_pos= 440.98	Y_pos= 320.44	Area= 227.75
Frame= 10	X_pos= 441.54	Y_pos= 321.03	Area= 305.25
Frame= 11	X_pos= 441.89	Y_pos= 321.87	Area= 237.75
Frame= 12	X_pos= 440.77	Y_pos= 324.30	Area= 211.38
Frame= 13	X_pos= 441.57	Y_pos= 323.02	Area= 213.12
Frame= 14	X_pos= 440.03	Y_pos= 320.38	Area= 227.25
Frame= 15	X_pos= 440.85	Y_pos= 321.07	Area= 220.00
Frame= 16	X_pos= 441.22	Y_pos= 323.05	Area= 173.75
Frame= 17	X_pos= 441.65	Y_pos= 324.06	Area= 206.88
Frame= 18	X_pos= 440.60	Y_pos= 324.03	Area= 241.00
Frame= 19	X_pos= 441.43	Y_pos= 326.66	Area= 197.75
Frame= 20	X_pos= 444.69	Y_pos= 323.95	Area= 241.00
Frame= 21	X_pos= 443.67	Y_pos= 325.23	Area= 261.00
Frame= 22	X_pos= 443.80	Y_pos= 324.42	Area= 314.38
Frame= 23	X_pos= 444.27	Y_pos= 326.11	Area= 256.88
Frame= 24	X_pos= 443.76	Y_pos= 324.82	Area= 295.12
Frame= 25	X_pos= 443.86	Y_pos= 326.31	Area= 242.25
Frame= 26	X_pos= 445.19	Y_pos= 327.04	Area= 239.50
Frame= 27	X_pos= 444.27	Y_pos= 326.18	Area= 248.12
Frame= 28	X_pos= 79.00	Y_pos= 316.00	Area= 272.12
Frame= 29	X_pos= 444.61	Y_pos= 325.80	Area= 281.88
Frame= 30	X_pos= 444.54	Y_pos= 326.47	Area= 317.38
Frame= 31	X_pos= 437.00	Y_pos= 319.00	Area= 259.12
Frame= 32	X_pos= 445.43	Y_pos= 326.48	Area= 339.50
Frame= 33	X_pos= 445.42	Y_pos= 325.94	Area= 344.25
Frame= 34	X_pos= 447.83	Y_pos= 326.71	Area= 309.50
Frame= 35	X_pos= 447.06	Y_pos= 327.72	Area= 345.12
Frame= 36	X_pos= 61.00	Y_pos= 410.00	Area= 312.88
Frame= 37	X_pos= 72.00	Y_pos= 318.00	Area= 414.62
Frame= 38	X_pos= 61.00	Y_pos= 412.00	Area= 337.62
Frame= 39	X_pos= 45.90	Y_pos= 413.60	Area= 300.88
Frame= 40	X_pos= 46.33	Y_pos= 415.33	Area= 451.50
Frame= 41	X_pos= 47.97	Y_pos= 416.25	Area= 377.50
Frame= 42	X_pos= 45.00	Y_pos= 420.30	Area= 350.50
Frame= 43	X_pos= 42.78	Y_pos= 420.56	Area= 332.62
Frame= 44	X_pos= 42.00	Y_pos= 420.92	Area= 376.50
Frame= 45	X_pos= 41.00	Y_pos= 424.50	Area= 316.00
Frame= 46	X_pos= 41.33	Y_pos= 425.67	Area= 361.62
Frame= 47	X_pos= 41.00	Y_pos= 425.50	Area= 398.25
Frame= 48	X_pos= 49.18	Y_pos= 396.37	Area= 475.50
Frame= 49	X_pos= 48.13	Y_pos= 406.13	Area= 418.75

Frame= 50	X_pos= 44.65	Y_pos= 405.78	Area= 477.62
Frame= 51	X_pos= 45.36	Y_pos= 404.60	Area= 559.50
Frame= 52	X_pos= 43.86	Y_pos= 409.41	Area= 442.12
Frame= 53	X_pos= 41.00	Y_pos= 401.00	Area= 480.88
Frame= 54	X_pos= 41.20	Y_pos= 418.60	Area= 454.50
Frame= 55	X_pos= 41.88	Y_pos= 399.12	Area= 475.00
Frame= 56	X_pos= 41.29	Y_pos= 416.94	Area= 338.88
Frame= 57	X_pos= 41.00	Y_pos= 417.50	Area= 408.75
Frame= 58	X_pos= 41.14	Y_pos= 414.07	Area= 327.62
Frame= 59	X_pos= 41.50	Y_pos= 398.50	Area= 360.75
Frame= 60	X_pos= 41.54	Y_pos= 399.23	Area= 368.75
Frame= 61	X_pos= 42.20	Y_pos= 427.50	Area= 440.00
Frame= 62	X_pos= 42.35	Y_pos= 427.05	Area= 518.38
Frame= 63	X_pos= 60.00	Y_pos= 323.00	Area= 415.00
Frame= 64	X_pos= 59.00	Y_pos= 321.22	Area= 520.00
Frame= 65	X_pos= 56.50	Y_pos= 322.33	Area= 512.25
Frame= 66	X_pos= 462.21	Y_pos= 334.80	Area= 512.80
Frame= 67	X_pos= 463.07	Y_pos= 337.62	Area= 519.38
Frame= 68	X_pos= 57.00	Y_pos= 323.25	Area= 518.62
Frame= 69	X_pos= 52.33	Y_pos= 325.67	Area= 575.88
Frame= 70	X_pos= 62.00	Y_pos= 348.00	Area= 517.25
Frame= 71	X_pos= 52.40	Y_pos= 325.40	Area= 683.38
Frame= 72	X_pos= 52.29	Y_pos= 325.43	Area= 526.62
Frame= 73	X_pos= 76.50	Y_pos= 379.00	Area= 594.00
Frame= 74	X_pos= 52.00	Y_pos= 328.00	Area= 555.25
Frame= 75	X_pos= 53.00	Y_pos= 326.67	Area= 468.38
Frame= 76	X_pos= 51.50	Y_pos= 346.00	Area= 549.00
Frame= 77	X_pos= 154.00	Y_pos= 345.80	Area= 557.75
Frame= 78	X_pos= 51.93	Y_pos= 327.64	Area= 538.50
Frame= 79	X_pos= 52.92	Y_pos= 327.62	Area= 537.38
Frame= 80	X_pos= 47.33	Y_pos= 331.33	Area= 550.38
Frame= 81	X_pos= 47.00	Y_pos= 330.00	Area= 655.25
Frame= 82	X_pos= 45.00	Y_pos= 330.00	Area= 463.88
Frame= 83	X_pos= 46.86	Y_pos= 331.29	Area= 640.50
Frame= 84	X_pos= 41.00	Y_pos= 333.00	Area= 731.62
Frame= 85	X_pos= 41.00	Y_pos= 397.00	Area= 766.75
Frame= 86	X_pos= 41.00	Y_pos= 332.00	Area= 687.62
Frame= 87	X_pos= 40.00	Y_pos= 332.00	Area= 702.25
Frame= 88	X_pos= 43.09	Y_pos= 333.00	Area= 862.00
Frame= 89	X_pos= 41.80	Y_pos= 335.40	Area= 891.00
Frame= 90	X_pos= 71.00	Y_pos= 377.33	Area= 848.38
Frame= 91	X_pos= 42.00	Y_pos= 397.50	Area= 691.50
Frame= 92	X_pos= 54.00	Y_pos= 334.00	Area= 765.12
Frame= 93	X_pos= 53.00	Y_pos= 424.00	Area= 802.25
Frame= 94	X_pos= 60.92	Y_pos= 382.72	Area= 803.25
Frame= 95	X_pos= 58.50	Y_pos= 375.50	Area= 767.50
Frame= 96	X_pos= 45.20	Y_pos= 428.60	Area= 740.75
Frame= 97	X_pos= 45.50	Y_pos= 336.00	Area= 1147.00
Frame= 98	X_pos= 51.06	Y_pos= 381.70	Area= 1016.62

Frame= 99	X_pos= 44.29	Y_pos= 437.57	Area= 1177.25
Frame= 100	X_pos= 43.54	Y_pos= 388.38	Area= 990.38
Frame= 101	X_pos= 44.00	Y_pos= 388.60	Area= 961.75
Frame= 102	X_pos= 40.00	Y_pos= 443.50	Area= 1099.38
Frame= 103	X_pos= 41.88	Y_pos= 388.94	Area= 1284.25
Frame= 104	X_pos= 41.29	Y_pos= 393.00	Area= 1231.12
Frame= 105	X_pos= 45.00	Y_pos= 377.50	Area= 1407.88
Frame= 106	X_pos= 45.44	Y_pos= 379.52	Area= 1204.88
Frame= 107	X_pos= 44.00	Y_pos= 440.00	Area= 1147.50
Frame= 108	X_pos= 41.00	Y_pos= 439.00	Area= 1388.88
Frame= 109	X_pos= 40.00	Y_pos= 400.00	Area= 1190.88
Frame= 110	X_pos= 40.00	Y_pos= 399.00	Area= 1387.38
Frame= 111	X_pos= 42.00	Y_pos= 383.00	Area= 1076.88
Frame= 112	X_pos= 53.00	Y_pos= 437.50	Area= 1008.50
Frame= 113	X_pos= 42.00	Y_pos= 379.62	Area= 1354.25
Frame= 114	X_pos= 41.00	Y_pos= 380.00	Area= 1172.38
Frame= 115	X_pos= 40.00	Y_pos= 384.00	Area= 1411.25
Frame= 116	X_pos= 56.83	Y_pos= 408.08	Area= 1285.62
Frame= 117	X_pos= 41.00	Y_pos= 446.00	Area= 1590.12
Frame= 118	X_pos= 54.31	Y_pos= 411.54	Area= 1305.25
Frame= 119	X_pos= 41.00	Y_pos= 362.50	Area= 1533.75
Frame= 120	X_pos= 97.20	Y_pos= 347.60	Area= 1375.12
Frame= 121	X_pos= 102.00	Y_pos= 388.00	Area= 1651.62
Frame= 122	X_pos= 143.50	Y_pos= 430.00	Area= 1730.00
Frame= 123	X_pos= 100.50	Y_pos= 390.00	Area= 1847.88
Frame= 124	X_pos= 209.00	Y_pos= 439.00	Area= 1949.75
Frame= 125	X_pos= 203.44	Y_pos= 432.22	Area= 1724.75
Frame= 126	X_pos= 45.20	Y_pos= 405.00	Area= 1736.50
Frame= 127	X_pos= 134.71	Y_pos= 442.00	Area= 1900.75
Frame= 128	X_pos= 89.00	Y_pos= 399.50	Area= 2023.00
Frame= 129	X_pos= 44.00	Y_pos= 373.00	Area= 2071.25
Frame= 130	X_pos= 55.00	Y_pos= 409.50	Area= 2131.75
Frame= 131	X_pos= 41.00	Y_pos= 443.50	Area= 2203.62
Frame= 132	X_pos= 49.00	Y_pos= 434.00	Area= 2373.75
Frame= 133	X_pos= 43.46	Y_pos= 438.92	Area= 2721.12
Frame= 134	X_pos= 86.83	Y_pos= 422.17	Area= 2195.00
Frame= 135	X_pos= 41.67	Y_pos= 441.50	Area= 2121.00
Frame= 136	X_pos= 133.41	Y_pos= 422.29	Area= 1858.62
Frame= 137	X_pos= 42.89	Y_pos= 409.56	Area= 1685.50
Frame= 138	X_pos= 50.00	Y_pos= 446.00	Area= 1783.00
Frame= 139	X_pos= 40.00	Y_pos= 439.00	Area= 1464.62
Frame= 140	X_pos= 56.00	Y_pos= 420.00	Area= 1368.25
Frame= 141	X_pos= 67.15	Y_pos= 435.38	Area= 1189.75
Frame= 142	X_pos= 45.00	Y_pos= 421.50	Area= 1209.38
Frame= 143	X_pos= 41.00	Y_pos= 399.50	Area= 1427.62
Frame= 144	X_pos= 46.27	Y_pos= 442.18	Area= 943.00
Frame= 145	X_pos= 42.60	Y_pos= 414.20	Area= 1652.50
Frame= 146	X_pos= 40.00	Y_pos= 439.50	Area= 1346.88
Frame= 147	X_pos= 41.00	Y_pos= 442.50	Area= 263.00

APPENDIX C
ORIENTATION RESULTS OF THE BLUE BLOCK

Frame= 1	X_pos= 0.00	Y_pos= 0.00	Area= 0.00
Frame= 2	X_pos= 200.70	Y_pos= 312.42	Area= 401.25
Frame= 3	X_pos= 199.51	Y_pos= 312.62	Area= 346.00
Frame= 4	X_pos= 198.97	Y_pos= 311.88	Area= 313.00
Frame= 5	X_pos= 200.04	Y_pos= 311.68	Area= 344.88
Frame= 6	X_pos= 199.46	Y_pos= 312.54	Area= 373.50
Frame= 7	X_pos= 201.32	Y_pos= 312.77	Area= 397.00
Frame= 8	X_pos= 200.69	Y_pos= 313.03	Area= 353.75
Frame= 9	X_pos= 200.19	Y_pos= 312.14	Area= 317.00
Frame= 10	X_pos= 201.07	Y_pos= 311.61	Area= 353.88
Frame= 11	X_pos= 187.00	Y_pos= 303.50	Area= 339.12
Frame= 12	X_pos= 186.00	Y_pos= 303.00	Area= 370.12
Frame= 13	X_pos= 199.33	Y_pos= 310.56	Area= 330.38
Frame= 14	X_pos= 185.00	Y_pos= 305.00	Area= 302.88
Frame= 15	X_pos= 197.99	Y_pos= 311.18	Area= 324.88
Frame= 16	X_pos= 199.30	Y_pos= 311.42	Area= 317.88
Frame= 17	X_pos= 197.14	Y_pos= 312.68	Area= 348.38
Frame= 18	X_pos= 197.69	Y_pos= 312.19	Area= 352.25
Frame= 19	X_pos= 198.84	Y_pos= 312.42	Area= 351.62
Frame= 20	X_pos= 199.14	Y_pos= 313.94	Area= 384.38
Frame= 21	X_pos= 198.82	Y_pos= 312.62	Area= 287.38
Frame= 22	X_pos= 197.60	Y_pos= 314.14	Area= 362.62
Frame= 23	X_pos= 196.05	Y_pos= 312.79	Area= 294.88
Frame= 24	X_pos= 186.50	Y_pos= 305.00	Area= 233.12
Frame= 25	X_pos= 196.63	Y_pos= 316.00	Area= 374.75
Frame= 26	X_pos= 197.00	Y_pos= 314.22	Area= 292.00
Frame= 27	X_pos= 197.13	Y_pos= 315.88	Area= 265.12
Frame= 28	X_pos= 195.90	Y_pos= 315.03	Area= 386.00
Frame= 29	X_pos= 195.46	Y_pos= 314.80	Area= 359.50
Frame= 30	X_pos= 197.40	Y_pos= 315.46	Area= 333.75
Frame= 31	X_pos= 193.31	Y_pos= 313.96	Area= 358.25
Frame= 32	X_pos= 194.01	Y_pos= 315.56	Area= 435.38
Frame= 33	X_pos= 191.28	Y_pos= 316.39	Area= 457.00
Frame= 34	X_pos= 194.35	Y_pos= 315.69	Area= 323.00
Frame= 35	X_pos= 194.11	Y_pos= 314.11	Area= 272.25
Frame= 36	X_pos= 189.91	Y_pos= 317.21	Area= 486.88
Frame= 37	X_pos= 188.05	Y_pos= 317.03	Area= 468.75
Frame= 38	X_pos= 135.00	Y_pos= 311.00	Area= 438.75
Frame= 39	X_pos= 187.85	Y_pos= 318.23	Area= 422.50
Frame= 40	X_pos= 186.87	Y_pos= 318.52	Area= 441.12
Frame= 41	X_pos= 185.18	Y_pos= 318.33	Area= 403.50
Frame= 42	X_pos= 186.12	Y_pos= 320.32	Area= 433.75
Frame= 43	X_pos= 184.07	Y_pos= 319.44	Area= 405.12
Frame= 44	X_pos= 182.90	Y_pos= 321.31	Area= 406.75
Frame= 45	X_pos= 182.96	Y_pos= 321.23	Area= 471.38
Frame= 46	X_pos= 181.95	Y_pos= 321.12	Area= 465.50
Frame= 47	X_pos= 184.17	Y_pos= 321.04	Area= 535.12
Frame= 48	X_pos= 181.60	Y_pos= 323.00	Area= 448.75
Frame= 49	X_pos= 182.31	Y_pos= 322.69	Area= 477.50

Frame= 50	X_pos= 181.69	Y_pos= 323.34	Area= 419.50
Frame= 51	X_pos= 180.64	Y_pos= 322.82	Area= 470.88
Frame= 52	X_pos= 181.43	Y_pos= 323.42	Area= 454.75
Frame= 53	X_pos= 182.80	Y_pos= 322.73	Area= 449.12
Frame= 54	X_pos= 169.00	Y_pos= 317.00	Area= 430.50
Frame= 55	X_pos= 180.29	Y_pos= 323.21	Area= 458.38
Frame= 56	X_pos= 182.80	Y_pos= 324.18	Area= 470.00
Frame= 57	X_pos= 180.57	Y_pos= 323.55	Area= 515.25
Frame= 58	X_pos= 181.33	Y_pos= 323.42	Area= 427.25
Frame= 59	X_pos= 181.38	Y_pos= 323.79	Area= 405.62
Frame= 60	X_pos= 169.00	Y_pos= 331.00	Area= 436.38
Frame= 61	X_pos= 168.75	Y_pos= 325.00	Area= 433.50
Frame= 62	X_pos= 180.96	Y_pos= 323.78	Area= 472.62
Frame= 63	X_pos= 178.08	Y_pos= 323.77	Area= 577.25
Frame= 64	X_pos= 179.71	Y_pos= 323.37	Area= 517.62
Frame= 65	X_pos= 175.99	Y_pos= 323.80	Area= 526.50
Frame= 66	X_pos= 175.43	Y_pos= 324.47	Area= 589.25
Frame= 67	X_pos= 175.72	Y_pos= 325.84	Area= 621.62
Frame= 68	X_pos= 175.17	Y_pos= 325.71	Area= 601.88
Frame= 69	X_pos= 175.28	Y_pos= 326.61	Area= 594.25
Frame= 70	X_pos= 173.87	Y_pos= 327.51	Area= 606.88
Frame= 71	X_pos= 173.75	Y_pos= 328.09	Area= 601.88
Frame= 72	X_pos= 175.06	Y_pos= 328.71	Area= 663.75
Frame= 73	X_pos= 175.48	Y_pos= 328.22	Area= 616.12
Frame= 74	X_pos= 176.36	Y_pos= 328.01	Area= 565.00
Frame= 75	X_pos= 175.72	Y_pos= 328.78	Area= 541.88
Frame= 76	X_pos= 175.44	Y_pos= 330.58	Area= 728.12
Frame= 77	X_pos= 175.81	Y_pos= 330.86	Area= 631.50
Frame= 78	X_pos= 175.52	Y_pos= 329.92	Area= 623.75
Frame= 79	X_pos= 173.13	Y_pos= 333.47	Area= 674.38
Frame= 80	X_pos= 172.78	Y_pos= 332.96	Area= 687.75
Frame= 81	X_pos= 172.06	Y_pos= 334.48	Area= 719.88
Frame= 82	X_pos= 169.17	Y_pos= 336.27	Area= 740.75
Frame= 83	X_pos= 166.92	Y_pos= 336.56	Area= 727.25
Frame= 84	X_pos= 166.78	Y_pos= 337.31	Area= 688.38
Frame= 85	X_pos= 165.51	Y_pos= 338.16	Area= 648.38
Frame= 86	X_pos= 164.53	Y_pos= 338.66	Area= 848.75
Frame= 87	X_pos= 163.06	Y_pos= 340.56	Area= 766.12
Frame= 88	X_pos= 162.56	Y_pos= 340.02	Area= 897.50
Frame= 89	X_pos= 160.04	Y_pos= 341.69	Area= 844.38
Frame= 90	X_pos= 160.35	Y_pos= 342.14	Area= 747.88
Frame= 91	X_pos= 158.93	Y_pos= 340.69	Area= 765.25
Frame= 92	X_pos= 155.72	Y_pos= 345.04	Area= 724.38
Frame= 93	X_pos= 157.38	Y_pos= 345.32	Area= 826.50
Frame= 94	X_pos= 155.24	Y_pos= 346.15	Area= 899.88
Frame= 95	X_pos= 155.82	Y_pos= 344.36	Area= 753.62
Frame= 96	X_pos= 136.80	Y_pos= 333.40	Area= 778.88
Frame= 97	X_pos= 150.16	Y_pos= 347.48	Area= 789.50
Frame= 98	X_pos= 148.94	Y_pos= 347.56	Area= 778.25

Frame= 99	X_pos= 148.78	Y_pos= 347.90	Area= 883.75
Frame= 100	X_pos= 144.26	Y_pos= 350.89	Area= 1024.62
Frame= 101	X_pos= 143.90	Y_pos= 350.65	Area= 1026.88
Frame= 102	X_pos= 142.84	Y_pos= 351.14	Area= 1020.25
Frame= 103	X_pos= 142.85	Y_pos= 353.02	Area= 981.50
Frame= 104	X_pos= 141.61	Y_pos= 352.35	Area= 1030.75
Frame= 105	X_pos= 140.92	Y_pos= 354.36	Area= 949.38
Frame= 106	X_pos= 139.71	Y_pos= 356.19	Area= 982.25
Frame= 107	X_pos= 139.76	Y_pos= 357.53	Area= 933.75
Frame= 108	X_pos= 137.10	Y_pos= 358.52	Area= 1016.75
Frame= 109	X_pos= 137.39	Y_pos= 357.00	Area= 861.12
Frame= 110	X_pos= 136.72	Y_pos= 359.09	Area= 1091.50
Frame= 111	X_pos= 137.15	Y_pos= 358.00	Area= 1062.00
Frame= 112	X_pos= 134.78	Y_pos= 357.39	Area= 988.88
Frame= 113	X_pos= 134.16	Y_pos= 359.82	Area= 1162.38
Frame= 114	X_pos= 133.62	Y_pos= 360.10	Area= 1210.50
Frame= 115	X_pos= 131.78	Y_pos= 363.35	Area= 1052.88
Frame= 116	X_pos= 132.74	Y_pos= 363.13	Area= 1189.25
Frame= 117	X_pos= 132.01	Y_pos= 366.30	Area= 1224.50
Frame= 118	X_pos= 129.89	Y_pos= 367.97	Area= 1156.00
Frame= 119	X_pos= 127.26	Y_pos= 368.88	Area= 1135.38
Frame= 120	X_pos= 131.97	Y_pos= 368.00	Area= 1270.75
Frame= 121	X_pos= 127.15	Y_pos= 370.86	Area= 1063.00
Frame= 122	X_pos= 127.52	Y_pos= 372.23	Area= 1375.62
Frame= 123	X_pos= 126.50	Y_pos= 372.40	Area= 1481.38
Frame= 124	X_pos= 126.80	Y_pos= 374.42	Area= 1326.88
Frame= 125	X_pos= 124.35	Y_pos= 376.83	Area= 1415.62
Frame= 126	X_pos= 121.66	Y_pos= 376.88	Area= 1503.75
Frame= 127	X_pos= 121.37	Y_pos= 377.50	Area= 1590.50
Frame= 128	X_pos= 117.51	Y_pos= 382.84	Area= 1442.50
Frame= 129	X_pos= 117.65	Y_pos= 382.00	Area= 1542.38
Frame= 130	X_pos= 113.03	Y_pos= 384.76	Area= 1579.88
Frame= 131	X_pos= 89.00	Y_pos= 384.00	Area= 1590.00
Frame= 132	X_pos= 111.11	Y_pos= 387.30	Area= 1646.75
Frame= 133	X_pos= 107.07	Y_pos= 389.97	Area= 1895.38
Frame= 134	X_pos= 106.62	Y_pos= 389.27	Area= 1820.50
Frame= 135	X_pos= 105.74	Y_pos= 389.76	Area= 1803.62
Frame= 136	X_pos= 102.43	Y_pos= 393.13	Area= 2000.50
Frame= 137	X_pos= 99.19	Y_pos= 393.89	Area= 2044.50
Frame= 138	X_pos= 97.16	Y_pos= 396.34	Area= 1992.50
Frame= 139	X_pos= 91.80	Y_pos= 397.65	Area= 2007.38
Frame= 140	X_pos= 63.50	Y_pos= 420.00	Area= 1859.38
Frame= 141	X_pos= 87.42	Y_pos= 399.22	Area= 2116.00
Frame= 142	X_pos= 84.29	Y_pos= 401.58	Area= 2249.00
Frame= 143	X_pos= 81.00	Y_pos= 400.17	Area= 2454.50
Frame= 144	X_pos= 79.98	Y_pos= 397.92	Area= 2469.12
Frame= 145	X_pos= 78.81	Y_pos= 389.66	Area= 2351.88
Frame= 146	X_pos= 70.02	Y_pos= 372.68	Area= 2795.62
Frame= 147	X_pos= 73.95	Y_pos= 376.78	Area= 2640.75

APPENDIX D

TOLERANCE VALUES FOR STATISTICAL ANALYSIS

Table D.1: Tolerances from the experiments

	Expected Value Tolerance	Variance Tolerance
Red block	15	50
Blue block	40	70
Stop sign	15	40
Green signs	40	150