

**Stereoscopic Imaging for Obstacle Detection**  
**Onboard Low-Flying Unmanned Aerial**  
**Vehicles**

Submitted by  
**Emad Hanna, B.Eng.**

A thesis submitted to  
the Faculty of Graduate Studies and Research  
In partial fulfillment of  
the requirements for the degree of

**Master of Applied Science**

Ottawa-Carleton Institute for Electrical and Computer Engineering  
Faculty of Engineering  
Department of Systems and Computer Engineering

Carleton University  
Ottawa, Ontario, Canada K1S 5B6  
April 2010

(C) Copyright 2010, Emad Hanna



Library and Archives  
Canada

Published Heritage  
Branch

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

Bibliothèque et  
Archives Canada

Direction du  
Patrimoine de l'édition

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*  
ISBN: 978-0-494-68629-4  
*Our file* *Notre référence*  
ISBN: 978-0-494-68629-4

#### NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

#### AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

The undersigned recommend to  
the Faculty of Graduate Studies and Research  
acceptance of the thesis

# **Stereoscopic Imaging for Obstacle Detection Onboard Low-Flying Unmanned Aerial Vehicles**

submitted by

**Emad Hanna, B.Eng.**

in partial fulfillment of the requirements for

the degree of

**Master of Applied Science in Electrical Engineering**

---

Chair, Howard M. Schwartz, Department of Systems and Computer Engineering

---

Thesis Supervisor, Rafik A. Goubran

---

Thesis Supervisor, Paul V. Straznicky

Carleton University  
April 2010

[i]

# ABSTRACT

Obstacle detection for low-flying unmanned aerial vehicles (UAVs) poses unique engineering challenges in terms of real-time implementation as well as system accuracy. Of all the available techniques for carrying out this task, optical sensors stand out as an inexpensive, lightweight and reliable solution. Image processing methods are used to analyze the images captured by the UAV camera(s) and to generate information pertaining to the location and motion of the obstacles in the field of view. These methods, however, can be computationally intensive and must be optimized if they are to be implemented in a moving vehicle. Additionally, the measurement of distance usually requires a high level of calibration before the results are useful.

This thesis proposes a calibration method rooted in image data analysis and shows how this can be used to accurately predict the distance to obstacles. An algorithm tailored specifically to low-flying UAVs (Sparse Edge Reconstruction) is presented. Both the calibration method and the algorithm were used to analyze video gathered on a low-altitude test flight.

# **ACKNOWLEDGMENTS**

I would like to sincerely thank my supervisors Professor Rafik A. Goubran and Professor Paul V. Straznicky for the tremendous level of guidance and support they have provided me over the course of my research and studies. This work would not have been possible without their encouragement and direction. I would also like to thank Sander Geophysics Limited for carrying out the test flight whose raw data was invaluable to this research. Additionally, I wish to thank the Ontario Centers of Excellence and NSERC (Collaborative Research and Development Grant and Discovery Grant) for their financial support over the course of this research.

Finally I wish to thank my parents Adel and Eva Hanna for their limitless support and patience over the course of my studies.

# Table of Contents

<b>ABSTRACT .....</b>	<b>ii</b>
<b>ACKNOWLEDGMENTS .....</b>	<b>iii</b>
<b>TABLE OF CONTENTS .....</b>	<b>iv</b>
<b>LIST OF TABLES .....</b>	<b>vii</b>
<b>LIST OF FIGURES .....</b>	<b>viii</b>
<b>LIST OF ABBREVIATIONS .....</b>	<b>xi</b>
<b>CHAPTER 1      INTRODUCTION .....</b>	<b>1</b>
1.1      Overview .....	1
1.2      Problem Statement .....	2
1.3      Objective .....	4
1.4      Thesis Contributions .....	5
1.5      Thesis Outline .....	6
<b>CHAPTER 2      BACKGROUND REVIEW OF OBSTACLE DETECTION AND                      COMPUTER VISION TECHNOLOGY .....</b>	<b>7</b>
2.1      Low-flying Unmanned Aerial Vehicles .....	7
2.2      Autonomous Obstacle Detection and Avoidance .....	9
2.2.1      Remote Sensors .....	9
2.2.2      Optical Sensors .....	9
2.3      Stereoscopic Imaging .....	11

2.3.1	The Correspondence Problem .....	12
2.3.2	Simplification of Stereo Imaging .....	15
2.3.3	The Disparity Interpretation Problem .....	19
2.3.4	Limitations of Stereoscopic Imaging: Baseline Separation, Camera Resolution and Camera Motion .....	20
2.4	Applications of Optical Sensors for Obstacle Detection .....	24
<b>CHAPTER 3</b>	<b>EXPERIMENTAL SETUP .....</b>	<b>28</b>
3.1	Statistical Analysis Setup .....	28
3.2	Test Pattern Setup .....	34
3.3	Test Flight .....	35
3.4	MatLab Toolboxes and Functions .....	39
<b>CHAPTER 4</b>	<b>EXPERIMENTAL RESULTS OF DISPARITY ANALYSIS AND DISTANCE MEASUREMENTS .....</b>	<b>40</b>
4.1	Baseline Separation .....	41
4.1.1	Effect of Baseline Separation on Disparity Values .....	42
4.1.2	Dynamic Range of Disparity Values .....	50
4.1.3	Distance Prediction Test Using Disparity Values .....	54
4.2	Effect of Camera Motion on Disparity Measurements .....	56

<b>CHAPTER 5</b>	<b>REAL-TIME IMPLEMENTATION OF A STEREOSCOPIC OBSTACLE DETECTION SYSTEM .....</b>	<b>60</b>
5.1	Sparse Edge Reconstruction Algorithm .....	61
5.2	Testing and Performance of Algorithm .....	72
5.3	Analysis of Video Captured During Test Flight .....	81
5.3.1	Application of Sparse Edge Reconstruction to Video .....	83
5.3.2	Application of Distance Prediction to Video .....	85
<b>CHAPTER 6</b>	<b>CONCLUSIONS .....</b>	<b>89</b>
6.1	Summary of Results .....	89
6.2	Recommendations for Future Research .....	91
<b>REFERENCES</b> .....		<b>93</b>



# LIST OF TABLES

TABLE 4.1	Training Objects and their Respective Distances .....	45
TABLE 5.1	SNR for Sparse Edge Reconstruction Algorithm .....	77
TABLE 5.2	SNR for Reconstruction with a Single Edge Detector .....	77
TABLE 5.3	Computation Time Required for Different Algorithms .....	79

# LIST OF FIGURES

FIGURE 2.1	Epipolar Geometry .....	11
FIGURE 2.2	Occlusion in Stereo Images .....	13
FIGURE 2.3	Parallel Epipolar Geometry .....	16
FIGURE 2.4	Simplified Epipolar Geometry .....	18
FIGURE 2.5	Minimum Object Distance Required for Detection .....	21
FIGURE 2.6	Degrees of Freedom for Optical Sensor .....	23
FIGURE 3.1	Combinations of Camera Stereo Positions .....	30
FIGURE 3.2	Camera Position Setup .....	31
FIGURE 3.3	Sample Point Location at Different Camera Positions .....	32
FIGURE 3.4	Visual Test Pattern .....	34
FIGURE 3.5	Visual Test Pattern in Grayscale .....	35
FIGURE 3.6	UAV Camera Setup (Stereo Cameras in Detail B) .....	36
FIGURE 3.7	Stereo Setup (Detail B in Figure 3.6) .....	37
FIGURE 3.8	Stereo Setup (Detail A in Figure 3.6) .....	37
FIGURE 3.9	Sample Frame from UAV-captured Stereoscopic Video ...	38
FIGURE 4.1	Number of Stereo Pairs at Different Camera Separations .....	43
FIGURE 4.2	Training Environment and Selected Points of Interest .....	44
FIGURE 4.3	Position of Training Objects at Different Camera Locations .....	46

FIGURE 4.4	Position of Two Training Objects at Different Camera Positions .....	47
FIGURE 4.5	Disparity vs. Camera Separation .....	48
FIGURE 4.6	Adjusted Disparity vs. Camera Separation .....	49
FIGURE 4.7	Dynamic Range of Disparity Values .....	50
FIGURE 4.8	Disparity vs. Distance at Different Camera Separations ...	52
FIGURE 4.9	Behaviour of the Slope of the Trend Line .....	53
FIGURE 4.10	Behaviour of the Intercept of the Trend Line .....	53
FIGURE 4.11	New Data plotted over Training Set .....	55
FIGURE 4.12	Minimum Camera Separation due to Translation .....	57
FIGURE 4.13	Maximum Camera Separation due to Translation .....	58
FIGURE 4.14	Potential Error in the Dynamic Range due to Translational Motion .....	59
FIGURE 5.1	Sparse Edge Reconstruction Algorithm .....	62
FIGURE 5.2	Original Image before Processing (300x300 pixels) .....	65
FIGURE 5.3	Image after Sparse Edge Detection .....	65
FIGURE 5.4	Correlation Windows (20x20 pixels) .....	66
FIGURE 5.5	Intensity Image of the Number of Pixels in each Window .	66
FIGURE 5.6	Binary Image after the Application of a Threshold of 5 pixels .....	67
FIGURE 5.7	Original Disparity Map .....	69
FIGURE 5.8	Edges Overlaid on Original Disparity Map .....	70
FIGURE 5.9	Disparity Map after the Optional Step is Applied .....	70

FIGURE 5.10	Test Pattern Stereo Images .....	72
FIGURE 5.11	Test Pattern Image after the Block A in FIGURE 5.1 .....	73
FIGURE 5.12	Disparity map after the Sparse Edge Detection Algorithm .....	74
FIGURE 5.13	Test Pattern with the Application of a Single <i>Prewitt</i> Edge Detector .....	75
FIGURE 5.14	Disparity map after a Stereoscopic Reconstruction with a Single <i>Prewitt</i> Edge Detector .....	76
FIGURE 5.15	Behaviour of SNR as the Image Sequence Increases .....	78
FIGURE 5.16	Sparse Edge Reconstruction without the Optional Step ...	80
FIGURE 5.17	Sparse Edge Reconstruction with the Optional Step .....	80
FIGURE 5.18	Sample Frame from the UAV-captured video .....	82
FIGURE 5.19	Sample Stereo Pair from the UAV-captured video .....	83
FIGURE 5.20	Sparse Edge Reconstruction of the Stereo Pair in FIGURE 5.19 .....	84
FIGURE 5.21	Single Edge Detector Reconstruction of the images in FIGURE 5.19 .....	84
FIGURE 5.22	Disparity vs. Distance in Test Flight Video .....	86
FIGURE 5.23	Low altitude Test Flight Image .....	87
FIGURE 5.24	Distance from Aircraft to Obstacles .....	88

# LIST OF ABBREVIATIONS

FPGA	Field Programmable Gate Array
GA	Genetic Algorithm
LIDAR	Light Detection and Ranging
MER	Mars Explorer Rover
SAD	Sum of Absolute Differences
SER	Sparse Edge Reconstruction
SGL	Sander Geophysics Limited
SSD	Sum of Squared Differences
SVM	Support Vector Machine
UAV	Unmanned Aerial Vehicle

# **Chapter 1**

## **Introduction**

### **1.1 Overview**

Obstacle detection for autonomous vehicles is a complicated task that poses unique design challenges. Various methods can be used to carry out this function depending on the type of vehicle, terrain, environmental condition and obstacle. Some of the technologies that have been used include radar, sonar and LIDAR (light detection and ranging). Notably, optical sensors have been successfully used to carry out obstacle detection and offer the advantage of being much lighter as well as less expensive than the more traditional radio and laser ranging methods. They are also not limited to direct line of sight ranging as is the case for LIDAR and they can detect much farther ranges than sonar sensors [1].

Although optical sensors offer a potentially robust solution to obstacle detection given such methods as optical flow analysis and stereoscopic imaging but they require intensive computation as well as a high level of technical precision. In the case of vehicles travelling at high speeds, the intensive computation must be carried out in real-time and this remains one of the significant challenges of these methods [2-8].

The theory behind the obstacle detection techniques that make use of optical sensors is well defined and understood. However, the technical requirements of such systems often require a significant amount of calibration to render the predictions made by the system useful. The level of accuracy and precision that is required of each system depends on how far potential objects are and how much the vehicle can afford to misjudge an object's distance.

## **1.2 Problem Statement**

The two main methods for obstacle detection using optical sensors are optical flow analysis and stereoscopic imaging. The former can deduce the motion of objects from consecutive images that are taken over a time interval while the latter involves more than one image sensor and can measure the proximity of objects using simultaneous images taken from different spatial locations. Both of these methods involve repeatedly correlating pairs of images. Although this task is fairly easy to accomplish by a human observer, it is notoriously difficult to perform with a computer. This is generally the

computational bottleneck of such systems and is referred to as the 'correspondence problem'. This problem is aggravated when the correlation must be carried out under a time restriction using limited computational resources.

This thesis deals with the special case of low-flying, terrain-following unmanned aerial vehicles (UAVs) which impose additional restrictions on the obstacle detection system. The low altitudes imply more obstacles to be avoided and therefore require that the system be generally more robust than for a vehicle flying at a higher altitude. Additionally, the horizon usually is hidden or at the very top of the field of view. This means that the analysis cannot be simplified by segmenting the environment into 'sky' segments and 'terrain' or 'non-sky' segments.

Obstacle detection systems that use optical sensors often make assumptions and simplifications to the optical theories and the geometry that are involved. This is especially true for real-time systems which do not have the computational resources necessary to carry out detailed correlations and reconstructions. Because of this, the predictions that are made by the obstacle detection systems can be quite coarse and require a large amount of calibration before they are useful. Since all systems are designed differently and with varying assumptions, the calibration is usually specific only to the system involved. Additionally, any system, even after calibration, must be able to deal with the various sources of optical noise that can be introduced by the environment such as glare, haze, precipitation, etc.



The research presented in this thesis is geared to specifically contribute to the work carried out at Carleton University in collaboration with Sander Geophysics Limited (SGL), a company based in Ottawa, Canada that specializes in airborne geophysical surveys. The goal is to develop a fully autonomous low-level flying unmanned aerial vehicle to carry out high resolution geomagnetic surveys. This aircraft is designed to fly at speeds between 60 and 100 knots at altitudes of 7 to 30 m (20 to 100 ft). A typical mission for the aircraft involves flying predetermined flight lines while following the contours of the terrain. In the case of an obstacle, the aircraft must avoid any danger and then return to the flight pattern to continue the mission. The full specifications of this system are described in [9]

### **1.3 Objectives**

The main objectives of this thesis are:

- To investigate some of the computational limitations of a real-time obstacle detection system for a low-flying UAV and to design an algorithm tailored to addressing these constraints,
- To carry out a statistical analysis on a large amount of optical sensor data in an attempt to evaluate the accuracy of the distance predictions made in stereoscopic imaging,
- To demonstrate a general calibration procedure based on the above-mentioned analysis,

- To evaluate the effect of camera baseline separation and camera motion on the performance of stereoscopic imaging systems.

Meeting the above objectives will contribute significantly to the implementation of an effective obstacle detection system for low-flying, terrain-following UAVs.

## **1.4 Thesis Contributions**

The main contributions of this thesis are:

- Determination of the effects of camera baseline separation and camera motion on the performance of stereoscopic imaging systems through image data analysis as well as definition of the limitations imposed by these factors,
- Proposition of a general method for calibrating a stereoscopic imaging system. The method is based on the relationships that are deduced from image data analysis and is designed to allow accurate prediction of the distance of obstacles. This method of calibration does not require a detailed knowledge of the camera models,
- Proposition of a real-time stereoscopic imaging algorithm designed specifically for low-flying UAVs. This algorithm (Sparse Edge Reconstruction) was tested with sample images and evaluated for performance. The results were published in [10],

- Testing of the proposed algorithm as well as the proposed method of calibration on video data gathered during a low-altitude test flight carried out on November 23, 2006 by SGL.

## **1.5 Thesis Outline**

The following chapters of this thesis are arranged as follows:

- CHAPTER 2: Provides a background review of low-flying UAVs, obstacle detection methods, a detailed description of stereoscopic imaging as well as a literature survey of the applications of optical sensors for obstacle detection,
- CHAPTER 3: Describes the experiment design as well as the methods of data acquisition that were used,
- CHAPTER 4: Presents the statistical tests that were carried out as well as the description and testing of the proposed calibration method,
- CHAPTER 5: Presents the proposed algorithm (Sparse Edge Reconstruction) for real-time obstacle detection tailored for low-flying UAVs,
- CHAPTER 6: Provides conclusions as well as recommendations for future research and a summary of the thesis contributions.

## **Chapter 2**

# **Background Review of Obstacle Detection and Computer Vision Technology**

### **2.1 Low-flying Unmanned Aerial Vehicles**

Obstacle detection for autonomous aircraft usually attempts to separate the field of view into sky and non-sky segments. This can be accomplished with visible spectrum sensors during the day or infrared sensors at night. This distinction is convenient because in the sky segment, all non-sky objects can be labelled as potential obstacles. The task then becomes one of determining whether the potential obstacle poses a threat to the aircraft (ex: a tower, bird,

another aircraft,...) or not (ex: clouds, glare, debris on camera lens,...). The division of the field of view into two general areas can be carried out using standard pattern classification methods such as a support vector machine (SVM) [11]. It is also helpful to find a horizon line to make this separation more evident [11-13]. However, for an aircraft flying close to the ground, the horizon line is usually very high up in the field of view and perhaps not visible [12]. This means that the task of finding obstacles is more complex since all objects must now be distinguished from the clutter of the terrain.

Additionally, since the aircraft that are of interest in this thesis follow the contours of the terrain, the horizon line will vary drastically over time. It may occasionally become visible and may even appear low in the field of view. Because the horizon line will fluctuate in this manner, calculating the horizon line for each image may be computationally counterproductive.

Autonomous aircraft can be designed to avoid obstacles by using the perceived obstacles as constraints in their path. In this way the aircraft can freely navigate the path that maximizes the distance to potential hazards [14]. Since the aircraft that are considered in this thesis are designed to fly specific lines as part of their missions, it is not possible to adopt such a technique. Instead, an aircraft should only deviate from its predetermined path if this path will intersect with an obstacle. Moreover, the aircraft must return to its original path once the obstacle is avoided [9].

## **2.2 Autonomous Obstacle Detection and Avoidance**

### **2.2.1 Remote Sensors**

Any autonomous obstacle detection algorithm can only be as good as the input it receives from the sensors. It is therefore very important to choose reliable and informative sensors for any robust system [15]. Many sensors have been proposed and used in various vehicle obstacle detection systems but the type of application usually has a strong bearing on which sensors can and should be engaged [15,16].

A reliable way to measure the location of points in three dimensions is LIDAR which is a form of laser telemetry. Although this type of sensor can be very effective for obstacle detection, it is generally large and heavy and may pose a problem for small, payload restricted aircraft [11], [14-17]. Another sensor that is often used in obstacle detection is sonar but this type of sensor is not effective for the large distances that this application requires. Recently, advances in miniature radar technology have paved the way for radio ranging onboard smaller vehicles. These types of systems are becoming more prominent as viable obstacle detection and ranging systems [11], [16].

### **2.2.2 Optical Sensors**

Optical sensors are a logical choice as a remote sensing technology for small unmanned aircraft because they are inexpensive, small, light, passive and

power efficient [6], [7], [11], [12], [15], [16], [18], [19]. There are two ways in which obstacle detection can be achieved with optical sensors: optical flow analysis and stereoscopic imaging. Optical flow involves using a single sensor to record multiple images as the aircraft is moving. By observing the way in which objects in the image appear to shift over time, the motion and distance of various objects in the field of view can be inferred [12], [15], [20], [21]. In order to do this, objects in one image must be correlated to objects in the following image of the sequence. This correspondence is relatively easy to accomplish because the consecutive images are very similar to each other. One of the limits of this technique is that it does not allow the disparity measurements to be translated directly to distance measurements since the consecutive views of the camera do not have any baseline separation [21].

Stereoscopic imaging makes use of two optical sensors. By analyzing the differences between the two views, a three dimensional reconstruction can be generated. Although stereoscopic imaging does not give any information about the motion of the objects in the way that optical flow does, it is a more accurate way of measuring distances [2], [5], [12], [19], [21], [22], [23].

This thesis only addresses the method of stereoscopic imaging. In practical unmanned aerial vehicle applications, neither method alone is completely robust. It has been shown that the best results are obtained when both optical flow analysis and stereoscopic imaging are used [12].

## 2.3 Stereoscopic Imaging

By using two optical sensors instead of one, the distance between objects and the sensors (or 'depth' of the objects) can be measured directly instead of using indirect statistical methods [18]. In order to measure the distance between two optical sensors and a given point,  $P$ , epipolar geometry must be used. This is essentially the geometry of stereoscopic images. The plane created by the optical centres of the two sensors  $O_l$  and  $O_r$  and the point  $P$  is the *epipolar plane* [Figure 2.1].

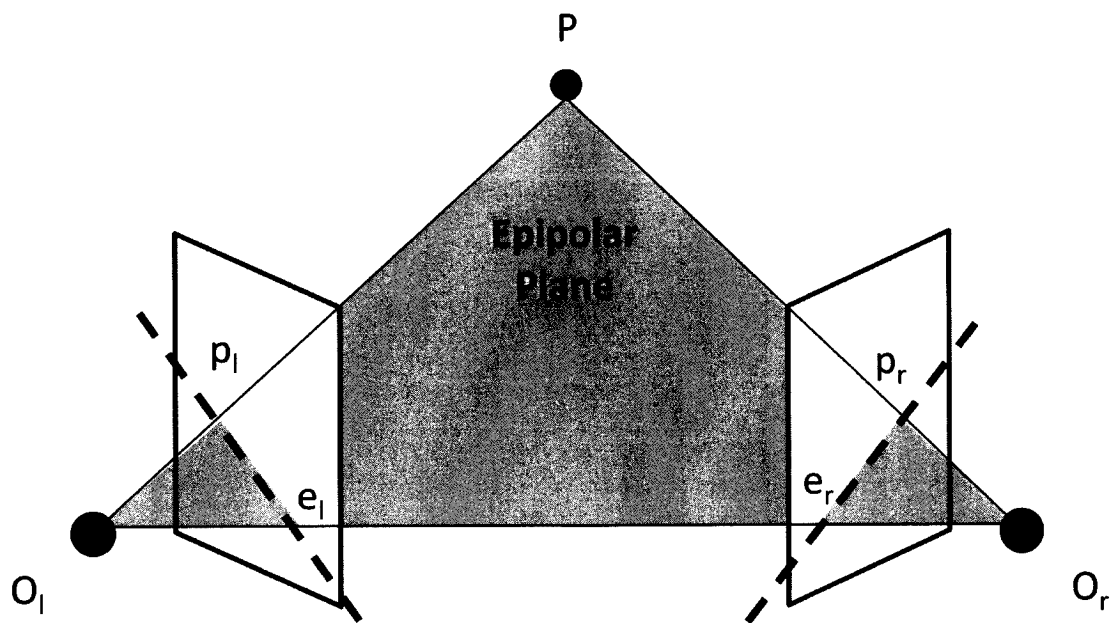


FIGURE 2.1 Epipolar Geometry

The intersection of the epipolar plane with the two image planes of the sensors results in two corresponding *epipolar lines* defined by the projection of the point  $P$



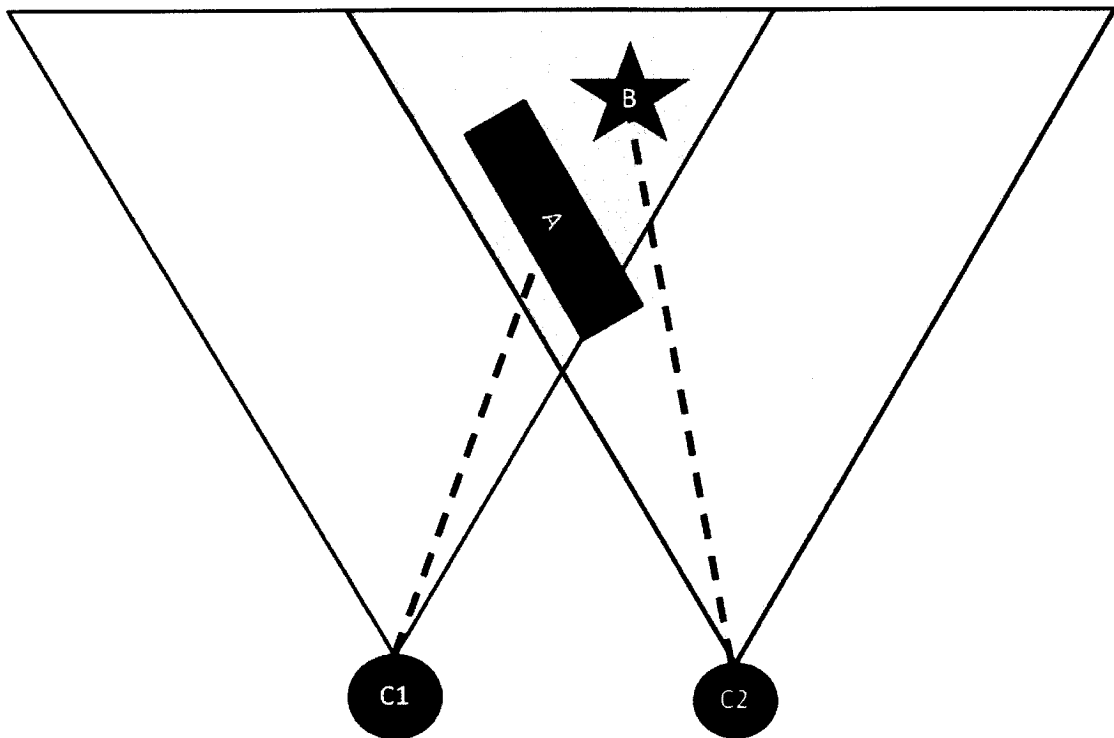
( $p_l$  and  $p_r$ ) and the *epipoles* ( $e_l$  and  $e_r$ ). Every point on the epipolar line of one image is located in the second image somewhere along the corresponding epipolar line. The difference in position of the projection of point P along the two epipolar lines is its *disparity*. The distance between P and the optical sensors is directly related to this disparity and can be calculated using triangulation [18], [22-27]. Two optical sensors with arbitrary orientations are stereoscopically related by a *fundamental matrix*. This matrix dictates where corresponding epipolar lines are located [24], [28].

An algorithm that uses stereoscopic imaging for obstacle detection must solve the two fundamental problems of stereo correspondence and disparity interpretation [29], [21], [30]. Correspondence is generally seen as the more difficult problem and much research has been dedicated to finding useful algorithms that are computationally economical [21], [18].

### 2.3.1 The Correspondence Problem

The biggest challenge in accurate stereoscopic imaging for obstacle detection is to be able to effectively correlate corresponding points between images [18]. There are many factors that can make this a very difficult task. Differences in the two images can be caused by occlusion, for instance. In this case the features of an object identified in one image cannot be correlated to the second image because they are hidden by another object on account of perspective. For example a bird that is identified in one image (B in Figure 2.2)

may be hidden in the second image due to a building occluding the second camera's perspective (A in Figure 2.2).



**FIGURE 2.2 Occlusion: B is only visible to C2. From C1's perspective, it is occluded by A**

As well, there are inherent optical differences that arise when sensors are separated by a significant baseline distance that may pose difficulties in the correlation process [21]. The correlation procedure is also very computationally intensive and is considered to be the main 'bottleneck' of any optical sensor

obstacle detection system in both stereoscopic analysis as well as optical flow analysis [21], [24], [31-33].

Most stereoscopic obstacle detection systems make use of 'low-level' features to perform correlation between images. One of the most often used methods is comparing image intensity values in the two images. An implicit assumption is made that the intensity properties of an image represent significant structural points that can be useful in obstacle detection [18]. The actual correlation function can be evaluated as a simple sum of absolute differences (SAD) [34], sum of squared differences (SSD) [24], or using more complicated evaluations such as a normalized cross correlation function [35]

$$\gamma(u, v) = \frac{\sum_{x,y} [f(x,y) - \bar{f}_{u,v}] * [t(x-u, y-v) - \bar{t}]}{\left\{ \sum_{x,y} [f(x,y) - \bar{f}_{u,v}]^2 * \sum_{x,y} [t(x-u, y-v) - \bar{t}]^2 \right\}^{0.5}} \quad [\text{EQN 2.1}]$$

where  $f(x, y)$  is the image to which the segment  $t$  is correlated at the position  $(u, v)$ .  $\bar{f}_{u,v}$ , is the mean of the image in the region under the window segment and  $\bar{t}$  is the mean of the segment itself. Other area correlation techniques can also be used for this purpose [31].

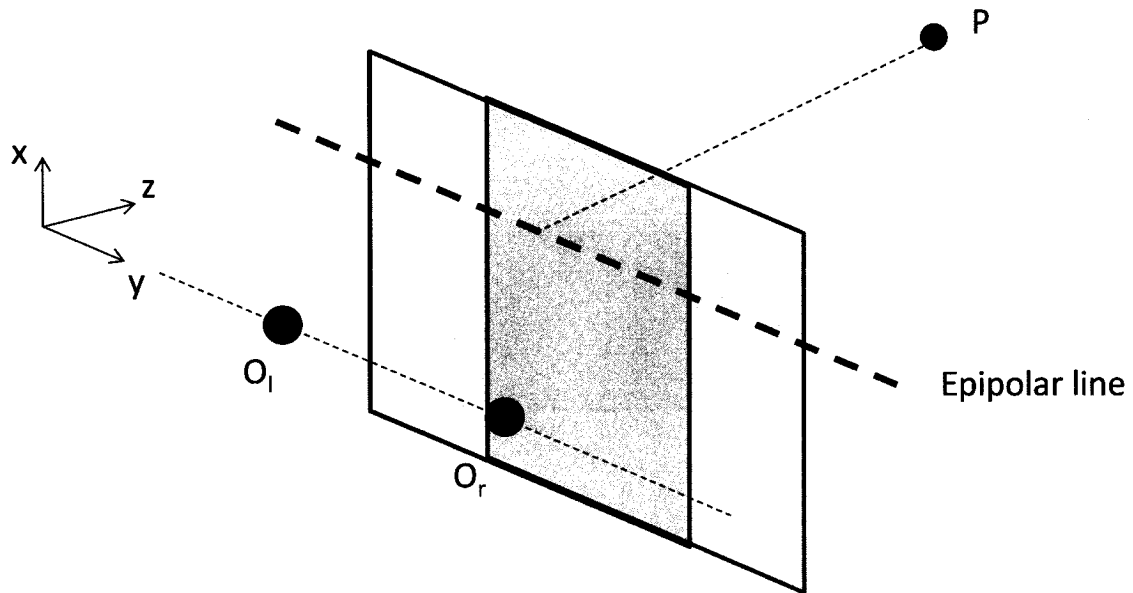
For a real-time obstacle detection system, especially one onboard a moving vehicle, it is imperative that the correlation between stereo images be carried out quickly for any effective analysis [5], [19], [16]. Because of this important limitation, almost all autonomous stereoscopic systems attempt to simplify the correlation process in various ways. One of the easiest ways to do this is to lower the resolution of the images [32]. Additionally, instead of carrying

out a pixel to pixel correlation, the images can be segmented and different correlation windows can be used [24], [32], [34]. The size of the correlation windows can vary according to the system used but reasonable sizes are from 0.005 to 0.1 percent of the total image area [34]. The disadvantage to using these correlation windows is that instead of a point by point disparity (and depth) map, a coarser map is generated where a single disparity value is calculated at each window. This means that a three-dimensional reconstruction of the environment carried out by a computationally limited UAV in real-time will necessarily be an approximation and may potentially miss objects whose features are very fine.

In order to simplify the task of stereoscopic correlation, most stereoscopic imaging systems make several simplifications to the epipolar geometry. Chapter 5 presents a real-time stereo correspondence algorithm designed for use in low-flying unmanned aerial vehicles

### **2.3.2 Simplification of Stereo Imaging**

The first simplification involves the orientation of the axes of the optical sensors. If the axes of the sensors are kept parallel to each other, then the corresponding epipolar lines will coincide. Additionally, if the cameras are aligned such that their image planes are vertical and parallel, then the epipolar lines will be horizontal and will coincide with the pixel rows of the images [Figure 2.3].

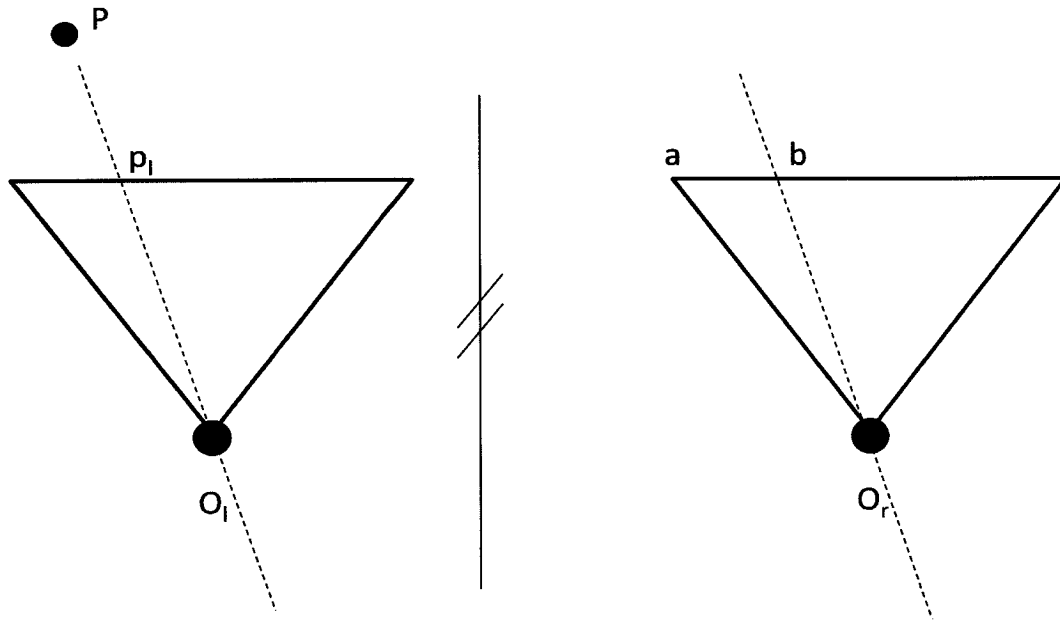


**FIGURE 2.3 Parallel Epipolar Geometry**

This simplification represents a large reduction in the search space of the correlation because the corresponding location for any given point in an image will be located on the same row in the second image. The correlation search is, therefore, reduced by a factor of the vertical resolution of the image [18], [25], [32-34], [36]. In order to take advantage of this *epipolar constraint*, it is imperative that the cameras be well aligned. It has been found that a vertical misalignment of the sensors of as little as  $\pm 0.05\%$  of the field of view (i.e. 0.5 pixels out of 1024) yields a 13% decrease in correlation performance while a  $\pm 0.1\%$  vertical misalignment (1 pixel out of 1024) results in a 50% reduction in correlation performance in terms of number of valid disparity pixels [34]. It is important to distinguish this type of misalignment from the movement of the two sensors as an attached pair. For instance on a UAV experiencing roll, the orientation of the cameras might change but their relative alignment remains stable. The problem,

in this case, might occur when the cameras begin to move with respect to each other due to distortions of the aircraft, for instance. If distortions or vibration of this degree are anticipated, it may be necessary to carry out real-time alignment of the cameras by keeping track of a marker on the aircraft (fiducial points). Chapter 4 presents the effects of such motion as well as camera separation on the accuracy of the distance predictions.

Another simplification that helps in the correspondence problem is dictated by the position of an object's projection along the epipolar lines. Figure 2.4 shows the geometry of parallel stereoscopic sensors observing a point  $P$ . If  $P$  is projected onto the position  $p_l$  in the left image plane, then the corresponding projection point  $p_r$  must be found between  $a$  and  $b$  in the right image plane. The point  $b$  is at the intersection of the right image plane and a line parallel to  $PO_l$  running through  $O_r$ . This simplification is due to the fact that camera views only overlap partially [25].



**FIGURE 2.4 Simplified Epipolar Geometry**

Another way to reduce the computational complexity of the matching algorithm is to impose a maximum disparity for the correlation. This means that for any point on an epipolar line in one image, the corresponding epipolar line in the second image is searched with the assumption that the corresponding point will be found within a maximum distance of the original projection point. This assumption can greatly reduce computational expenditure [29]. One potentially negative result of this simplification is that it puts an upper limit on the minimum distance that an object can be for the system to recognize it [32]. This minimum distance can be calculated from the equations in section 2.3.3 once the camera constant has been determined through calibration.

Finally, a higher level algorithm may be implemented to extract features from the stereo images and evaluate the correlation between lines, shapes,

edges and other structural primitives. Although these extra steps introduce more computation, they reduce the number of correlations necessary which may reduce the overall complexity of the system [37], [25], [29].

### 2.3.3 The Disparity Interpretation Problem

The value of the disparity is determined by finding the difference between the original location of a point and the location of the maximum value in the correlation function. The accuracy of the disparity measurement depends, therefore, on how well the correlation function is defined. If the correlation function yields no obvious maximum peak then the disparity value is not reliable [31].

The disparity values that are calculated for the desired points or windows in the stereoscopic images are used to predict the distance between the objects and the optical sensors at those locations. In general, a large disparity indicates that an object is closer to the sensor while objects that are very far away do not exhibit any disparity due to a finite pixel size. It is not difficult to determine the relative distances of the objects but it is quite challenging to translate the disparity values into absolute distance values [18]. In theory, the distance from the midpoint of the two optical sensors to the point being observed is defined by the equation:

$$Z = \frac{fB}{qd} \quad [\text{EQN 2.2}]$$



Where  $f$  is the focal length of the sensors,  $B$  is the baseline separation between them,  $q$  is the pixel size and  $d$  is the disparity [34], [31], [36]. In practice this equation is seldom applied in this form due to varying camera and lens models and instead a calibration process is used to determine the parameters. Since the value of  $B$  and  $d$  can be accurately measured, the equation is often reduced to

$$Z = \frac{c(B)}{d} \quad \text{EQN 2.3}$$

where  $c(B)$  is a ‘camera constant’ at a given camera baseline separation [38]. Chapter 4 presents a statistical approach to estimating Equation 2.3 as a way to deal with the disparity interpretation problem.

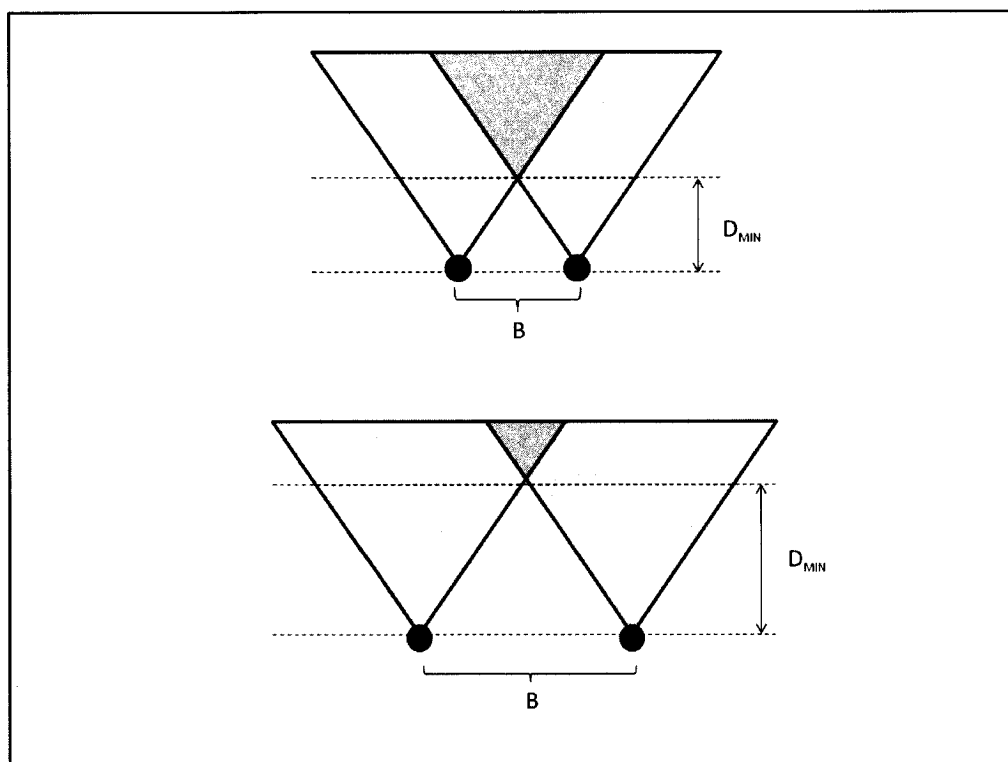
### **2.3.4 Limitations of Stereoscopic Imaging: Baseline Separation, Camera Resolution and Camera Motion**

There are various factors that influence the performance of stereoscopic systems both in terms of the correlation and the disparity interpretation. Some of the most important factors include the separation of the optical sensors, the resolution of the images being processed, and potential motion of the sensors. These limitations are discussed here and Chapter 4 presents an analysis of some of their effects on stereoscopic imaging.

## Baseline Separation

The accurate measurement of distance from stereoscopic images requires a significant separation between the sensors since real disparity measurements are needed for the prediction. However a large separation between the sensors can have significant repercussions on a performance of the stereoscopic reconstruction system [39].

The first direct result of increasing the baseline separation is that it increases the minimum distance that an object can be in order to be detected.



**FIGURE 2.5 Minimum Object Distance Required for Detection**

Along with this increase in minimum distance, there is an accompanying decrease in the overlap between the two fields of view which means that less

objects can be detected by the system [34] as can be seen in Figure 2.5. However, if the angles of the field of view are wide enough, and if the anticipated objects are far enough from the sensors then these may not be significant problems.

The second, and more important, effect of increasing the baseline separation is that the images become dissimilar due to the sensors' different 'views' of the environment [34], [18]. This can make it extremely challenging to correlate points and objects in the two images as described in section 2.3.1.

### Camera Resolution

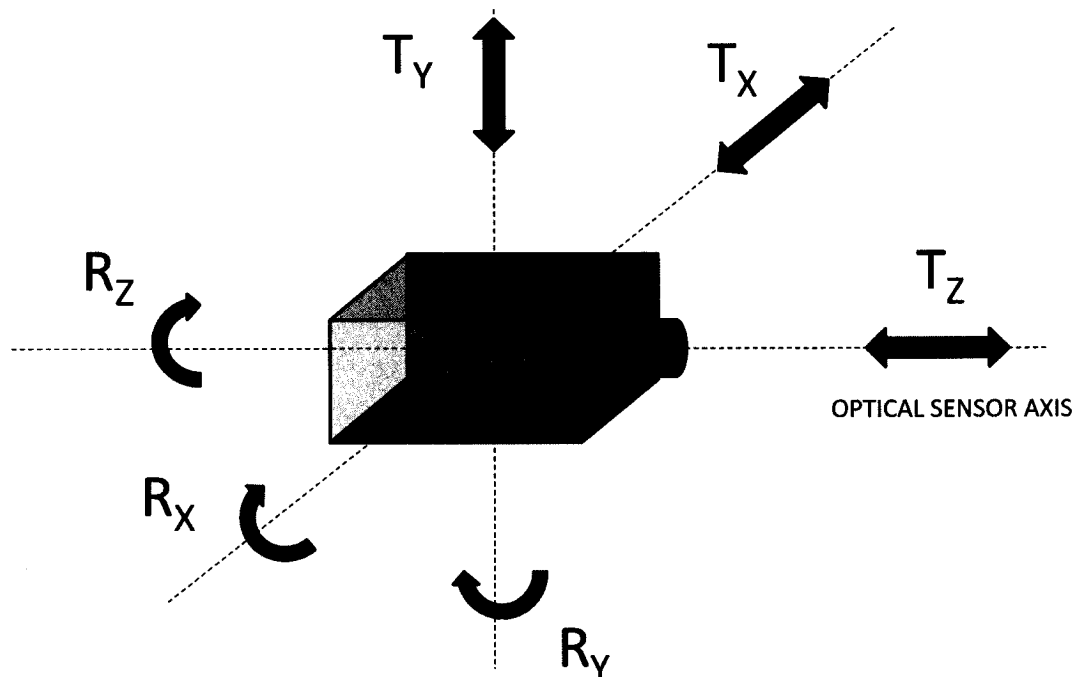
The resolutions of the cameras that are used in stereoscopic imaging have a direct effect on the quality of the measurements. Firstly, the resolution of the disparity measurements is a function of the resolution of the cameras. A higher resolution will yield a wider range of possible disparity values which will be used to predict the distance measurements. Additionally, since the smallest possible disparity is one pixel wide, a low resolution will translate into coarse distance measurements.

Camera resolution is also an important factor in the correlation procedure. On one hand, two high resolution images can be confidently correlated because there are more well-defined features in the images that will result in more accurate matching. However, this also means that the procedure is more

computationally intensive because there are more pixels to consider. As described in section 2.3.1, the computational complexity of stereoscopic correlation can paralyze an obstacle detection system. Because of this, many systems chose to lower the resolution of the images before correlation [15], [32]. A real time application of stereoscopic imaging must, therefore, find a compromise between accurate matching and fast execution.

### Camera Motion

Stereoscopic imaging is very sensitive to camera motion. In three-dimensional space optical sensors are subject to six degrees of freedom which can pose challenges to any stereo reconstruction (Figure 2.6).



**FIGURE 2.6 Degrees of Freedom for Optical Sensor**

Because of the epipolar simplification described in section 2.3.2, motion which would cause the horizontal rows of the images to deviate from each other can be devastating to any reconstruction [40]. This is the case for direct translational motion in the vertical direction ( $T_Y$ ), rotation of the optical sensor with respect to the horizontal axis ( $R_X$ ), as well as rotation with respect to the sensor axis ( $R_Z$ ). In the case of  $T_Y$ , the cameras can easily be adjusted by locating common points in the image before any analysis is carried out. In the case of flying aircraft, this would optimally be done by locating fixed features on the aircraft itself (fiducial points). In the case of  $R_X$  and  $R_Y$ , the epipolar assumption must be dropped and either the system must deal with a very large correlation search space or the real epipolar lines in the image must be found before stereoscopic correlation. The three remaining types of motion do not alter the horizontal epipolar lines but they do pose challenges to the reconstruction. In the case of translational motion in the direction of the sensor axis ( $T_Z$ ), the effect is minimal if objects are relatively far from the cameras, which is the case in UAV obstacle detection. The other two types of motion, however, significantly alter the results of the reconstruction. The effect of translation along the horizontal axis ( $T_X$ ), is analyzed in section 4.2.

## 2.4 Applications of Optical Sensors for Obstacle Detection

There are many examples in literature of obstacle detection systems that use stereoscopic imaging. A catalogue of the many parameters and performance metrics that are involved in such systems can be found in [41]. There also exists

a number of techniques that attempt to deal with the correspondence problem for real time application. For instance, although using a fine-to-coarse approach such as pixel to pixel correlation can minimize error propagation, a coarse-to-fine method is often preferred due to a large decrease in computation [36],[20]. Another way to reduce the complexity of the problem is to find structures in the images before correlation [5], [20]. For instance in UAV applications, objects like the horizon, roads, and the sky can be extracted and used to reduce the correlation search space [11].

Some systems have attempted to control the stereoscopic complexity by implementing the correlation in hardware such as a field-programmable gate array (FPGA). Using this method makes the computational time required by the correlation 'short' and 'deterministic' [31].

There are also many examples in the literature of attempts at increasing the accuracy of stereoscopic correlation. These systems are difficult to implement in real-time obstacle detection but they may be necessary in certain scenarios. For instance the stereoscopic system onboard the Mars Exploration Rover (MER) slows it down such that the rover requires 43 seconds to navigate a 10mx10m area [32]. With increasing computational power, it can be expected that some of the more accurate correlation algorithms can be implemented in real time. One example of such an algorithm includes terrain slope and surface recognition [30], [33], [36]. Some systems have also attempted to deal with stereo correspondence as an optimization problem and have designed genetic algorithms (GAs) to search for the best solutions [21]. Another interesting technique involves

extracting linear edges from the images and creating a relationship tree describing the orientations of the different lines. The correspondence problem, therefore, becomes one of finding the nodes in the graph that are the most similar between the two images [25].

Some techniques do not adhere to the standard assumptions of stereoscopic imaging. For instance, [37] foregoes the epipolar constraint and uses cameras that can verge inwards (point towards each other) or outwards giving the system a wider field of view. Another system navigates an unmanned aircraft through a canyon by using optical flow to keep the vehicle centred in the canyon and stereoscopic imaging to navigate. The performance of this system is better than either technique alone. [12].

This thesis addresses the problem of real-time obstacle detection and avoidance for unmanned aerial vehicles (UAVs). The research involves 'low-flying' UAVs which are here defined as having an altitude in the range of about 7 to 30 m (20 to 100 ft) above ground level. Additionally the vehicles are expected to follow the contours of the terrain at these low altitudes. A final constraint is that the UAVs are following specific flight lines, which is an essential condition in surveying and mapping missions. These properties of the vehicles and the mission profiles dictate the types of obstacle detection systems that will be feasible and effective. One important limitation is that any system must be lightweight given the limited payload of most UAVs [11], [15], [17].

The thesis presents a stereoscopic algorithm (Chapter 4) which stresses the need for reducing the computational complexity of the correspondence procedure and is generally designed for low-flying unmanned aircraft. At the same time, the algorithm does not propagate errors to the same extent that coarse-to-fine approaches usually do. The thesis also investigates the statistical relationship between stereoscopic measurements and the various parameters that have been described (Chapter 5). The next chapter describes the various experimental setups that were used to perform both of these tasks.



## **Chapter 3**

# **Experimental Setup**

For the purpose of this thesis, several experiments were designed to test and evaluate the proposed algorithms and analysis tools. The programming was carried out in MatLab (see section 3.4) in order to take advantage of the available toolboxes as well as the ease with which the software handles matrix operations. In addition to these experiments, a flight test was carried out to collect raw data for the system to evaluate.

### **3.1 Statistical Analysis Setup**

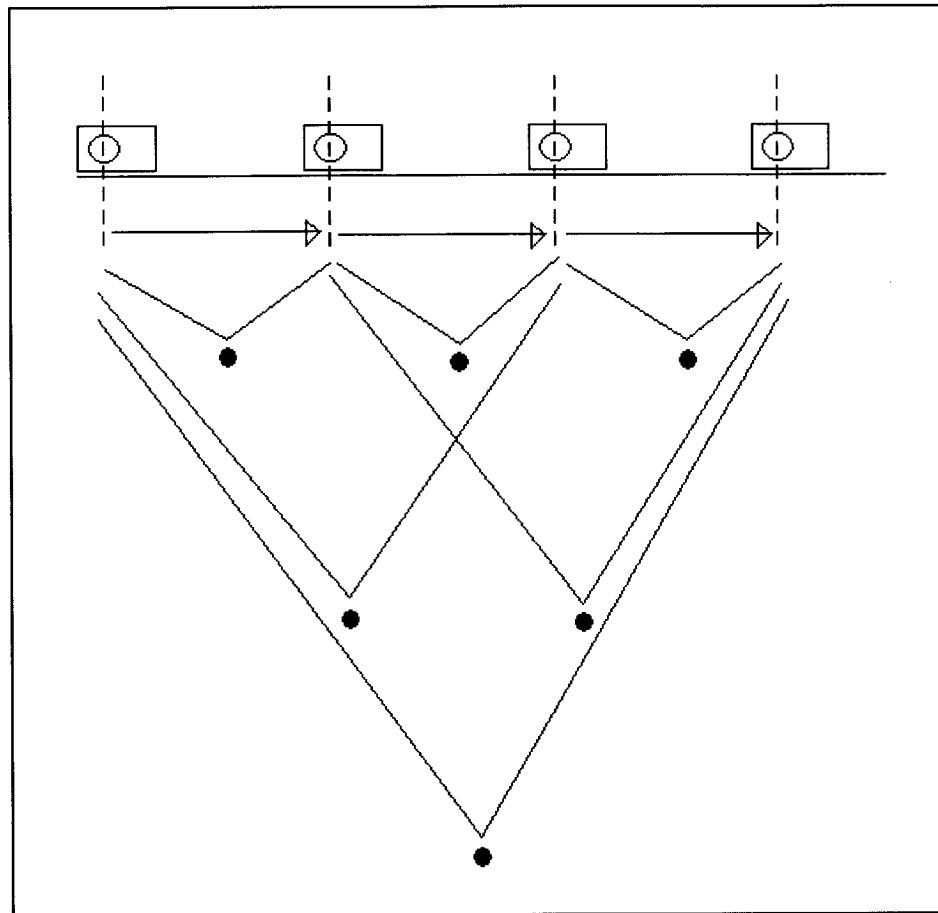
In order to perform a statistical analysis of various parameters involved in stereoscopic imaging, a large number of stereo pairs was required. In order to acquire the images, a single Fujifilm FinePix A820 camera was used. The camera

was placed in a starting position and then shifted incrementally to capture images at increasing separations.

There are two advantages to this method. Firstly, since the same camera is used for all the images, the same camera model can be assumed for any given stereo pair. Secondly, any two images can be selected to generate a unique stereo pair. This means that for  $n$  camera positions, the number of unique stereo pairs that can be generated,  $k$ , can be defined as:

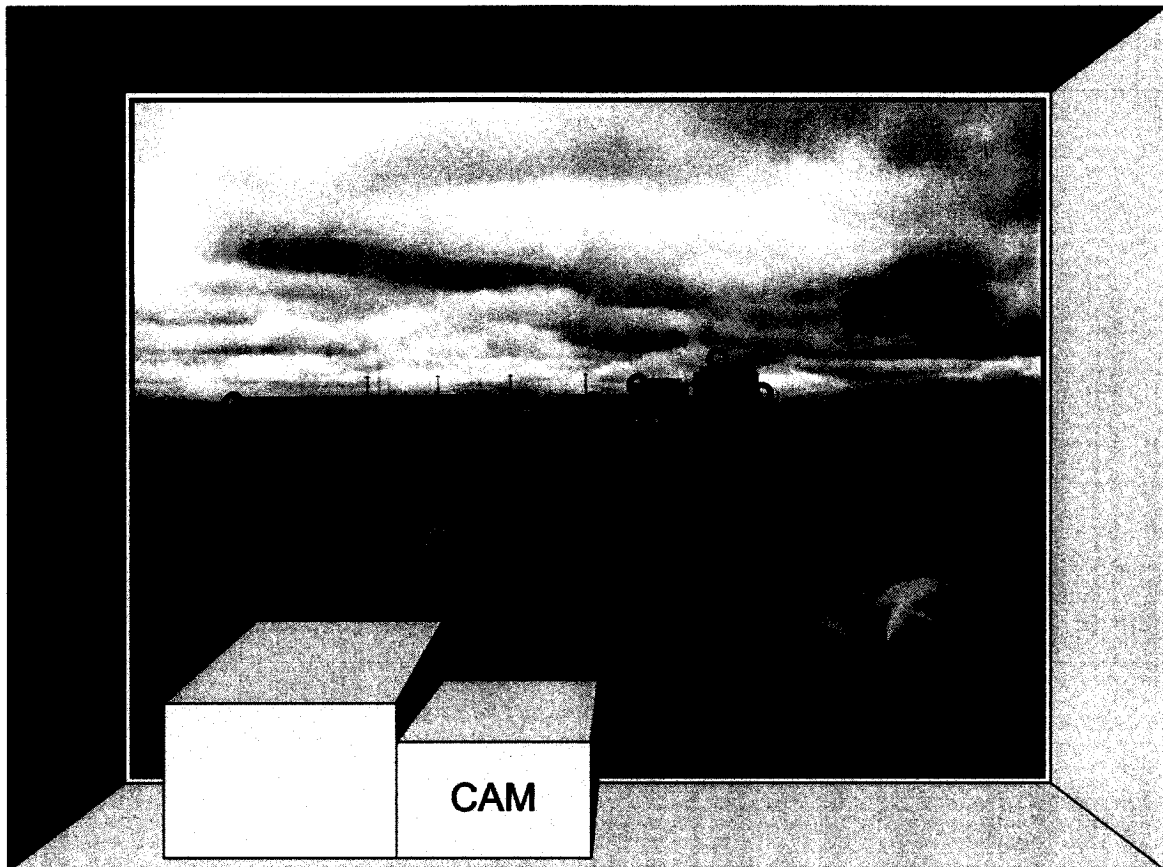
$$k = \sum_{i=1}^{n-1} n - i \quad \text{EQN 3.1}$$

For instance, in Figure 3.1, four camera positions ( $n = 4$ ) are used to generate six unique stereo pairs (red dots) as defined by this equation.



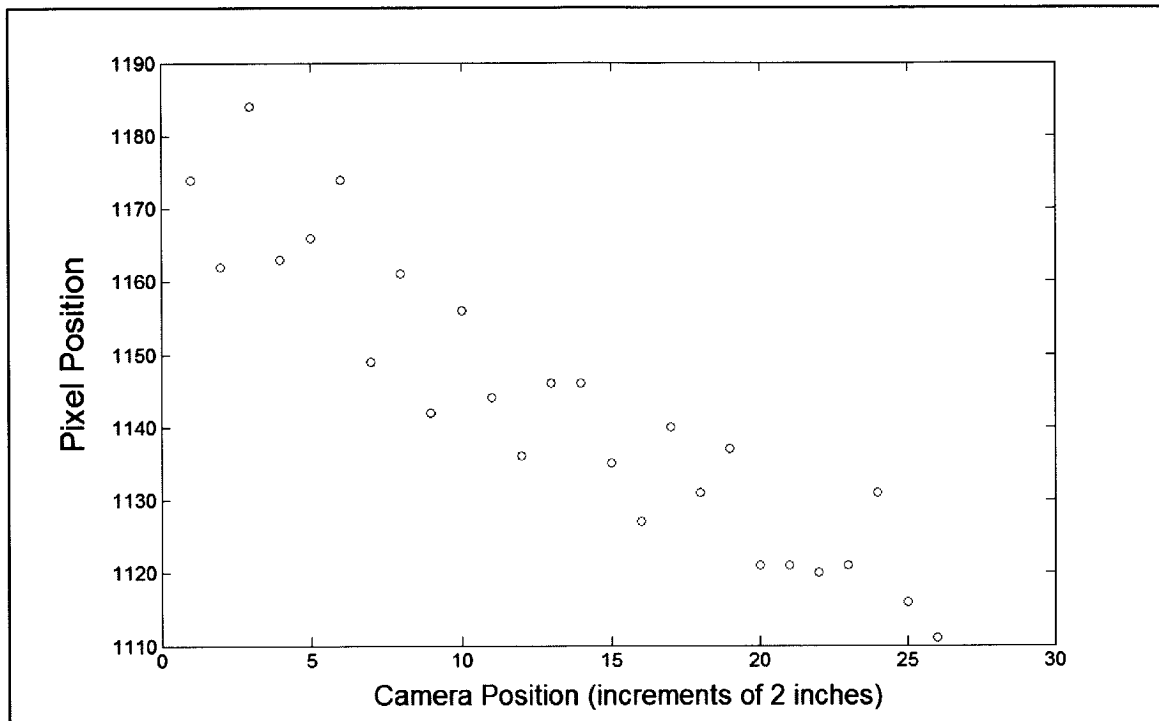
**FIGURE 3.1 Combinations of Camera Stereo Positions**

As described in section 2.3.2, stereoscopic analysis is extremely sensitive to camera motion. Therefore, extreme precaution must be taken to ensure that as the camera is moved from one position to the next, it does not change orientation. Since the camera was moved manually, this condition could only be satisfied by supporting the sides of the camera so that it could not move. Figure 3.2 shows how three of the six surfaces of the camera were aligned and a sample test image. In these images, points of interest were marked and measured as described in section 4.1



**FIGURE 3.2 Camera Position Setup**

As can be seen in Figure 3.2, the front of the flat camera lens casing was placed against a window. The left surface was aligned with a rectangular support and both the camera and the support were aligned with the flat windowsill. At each position the rectangular support was translated by the desired amount and aligned with the window and the windowsill. The result of this setup is a minimization of the camera motion. Figure 3.3 shows a sample from the raw data gathered using this setup.



**FIGURE 3.3 Sample Point Location at Different Camera Positions**

The deviations from the line of best fit (i.e. from a linear regression of the data) are due to the inherent errors introduced by this setup. Over the gathered data, the error falls within  $\pm 10$  pixels of the line of best fit. At a horizontal resolution of 2034 pixels, this translates to approximately 0.5% of the field of view of the camera.

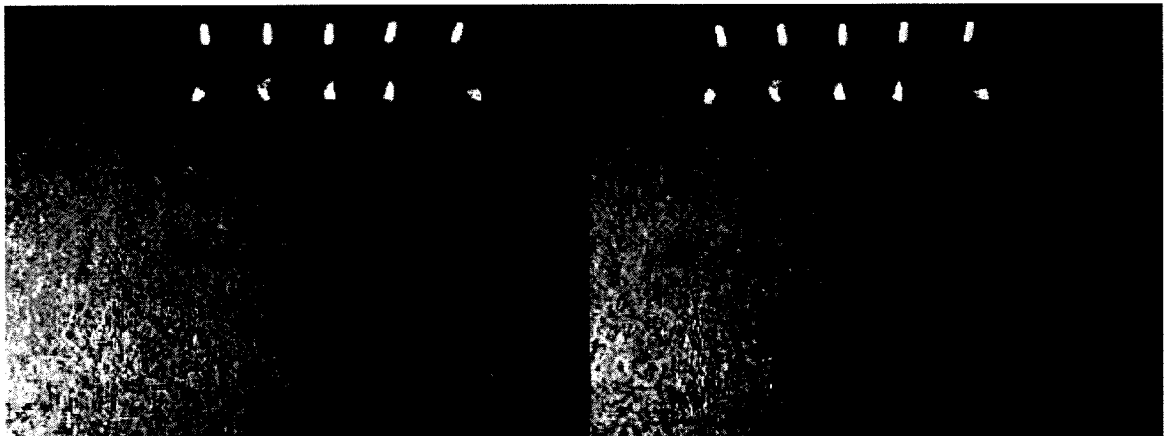
The distance between the camera and various points in the field of view was estimated using the 'Ruler' function in Google Earth (TM). This introduces three main sources of error in the measurements. First the points being observed from one projection must be mapped onto another projection and it is not always clear where the exact point is located. Secondly, most of the points are found on

tall buildings because they are visible at far distances. Since the satellite images are not usually taken from a position directly above the region of interest, the projections of the buildings often appear to be 'leaning' in the horizontal plane. The magnitude of the error induced depends on the relative angle of the axis of the satellite and that of the building. In the buildings being observed, this inaccuracy was never more than a dozen meters. Finally, the third source of error is the accuracy of the measuring tool itself that is supplied with the software. Although Google does not provide any values for the inaccuracy in its measuring tool, comparisons between the best estimates of the length of the equator as well as the meridional circumference both yield an error of approximately -0.1%.

The data analysis carried out for this thesis (Chapter 4) is based on the disparity values calculated in stereo pairs. In order to calculate disparity, the image matrices were plotted in MatLab and the positions of the points of interest were recorded manually.

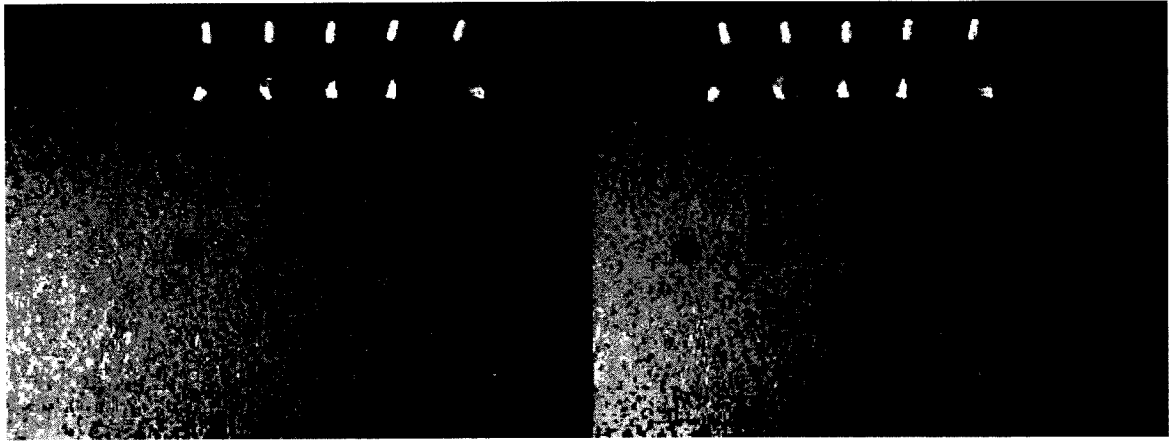
### 3.2 Test Pattern Setup

In Chapter 5 an algorithm is presented to carry out stereoscopic obstacle detections for low-flying UAVs. A series of test patterns was created to test this algorithm. The two stereo cameras that were used are Logitech Quickcam Fusion.. In these tests the goal was to determine how well the algorithm was able to distinguish between objects and their environment. All objects were kept at the same distance from the camera (a few feet). Figure 3.4 shows a typical test pattern that was used. The sizes of the objects are all roughly the same (about 1 to 2 cm). Each image in the stereo pair has a resolution of 240x320 pixels.



**FIGURE 3.4 Visual Test Pattern**

The objects seen in Figure 3.4 were placed on a textured carpet such that it is not trivial to isolate them from their environments. In the stereo reconstructions described in section 5.1, the correlation function that is used is normalized cross correlation. This function only works for greyscale images and so the original RGB images are converted to greyscale. This makes the objects in the images even more difficult to separate from their environments as seen in Figure 3.5



**FIGURE 3.5 Visual Test Pattern in Grayscale**

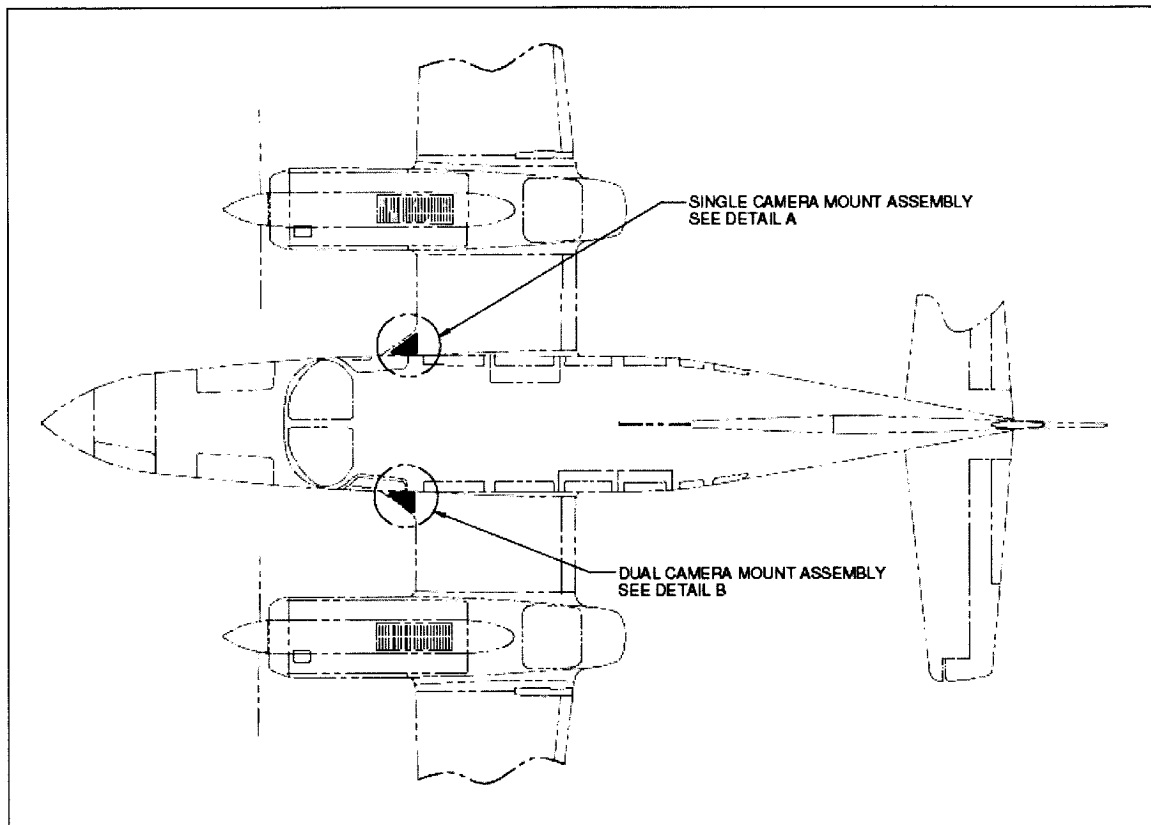
In order to generate a data set of similar test patterns, it was necessary to alter the position and lighting of the objects. In order to do this, the cameras were placed on top of a running desktop computer which induced slight vibrations and corresponding differences in the results.

### **3.3 Test Flight**

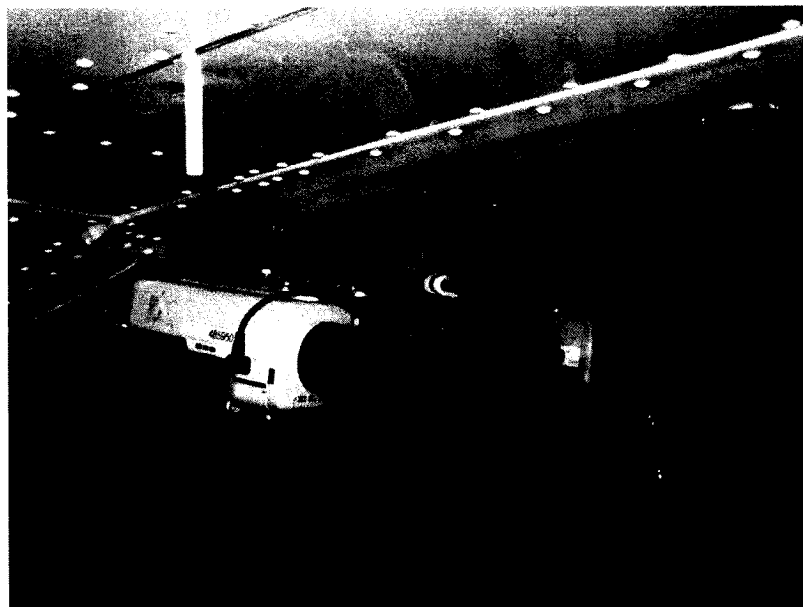
In addition to the tests described by section 3.2 and in order to ascertain the type of environment that might be encountered by a low-flying Unmanned Aerial Vehicle (UAV), a test flight was carried out on November 23, 2006. The aircraft used for this test was a Cessna 404 and three cameras were attached to the aircraft. A stereo pair consisting of two ELMO model TSN411D colour digital CCD cameras was attached to the left inboard wing panel of the aircraft (Figure 3.6 and Figure 3.7). The camera mounts were bolted to the aircraft to ensure that



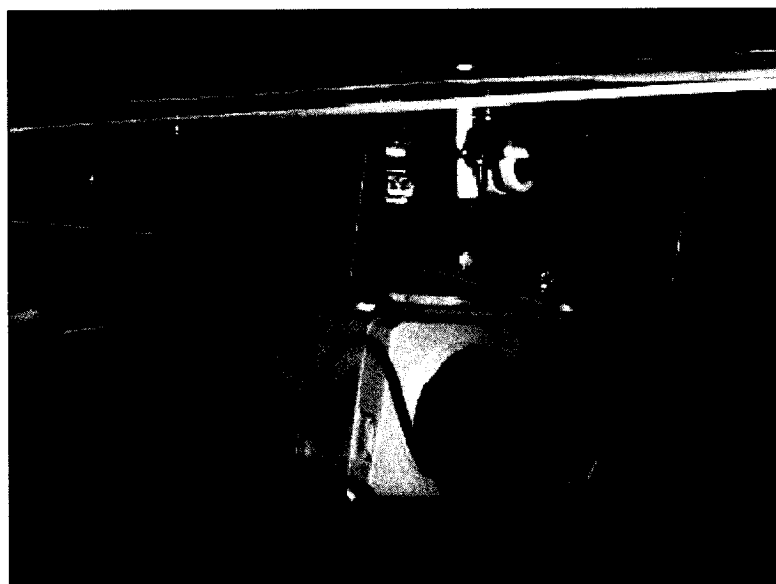
they did not move significantly with respect to the aircraft, although no measurement of stiffness was recorded.



**FIGURE 3.6 UAV Camera Setup (Stereo Cameras in Detail B)**



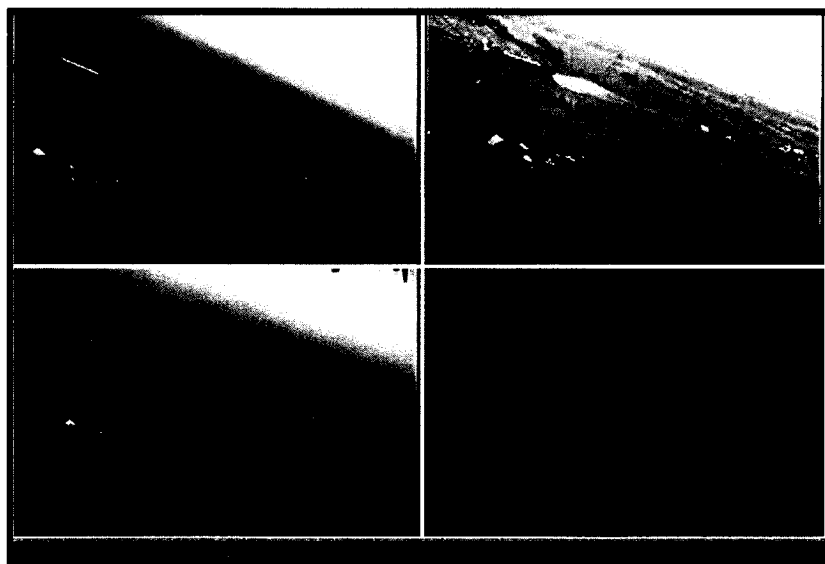
**FIGURE 3.7** Camera Setup (Detail B in Figure 3.6)



**FIGURE 3.8** Camera Setup (Detail A in Figure 3.6)

On the right wing panel a single ELMO model TNC4604 camera was attached. All three cameras were equipped with a Honeywell lens: HLD5V5OF13L, 5-50MMIF1.3, 1/3" CS Mount and had a downward angle of 15 degrees.

About a third of the flight was conducted at relatively low altitudes of 200 to 300ft. For safety reasons, the aircraft did not fly at any lower altitudes. Although this altitude range is significantly higher than what was defined for low-flying UAVs in section 2.1, it was very useful in predicting what might be expected at even lower altitudes. Approximately 90 minutes of video (1.2GB AVI file) was captured during this flight as the aircraft surveyed various locations in the Ottawa-Gatineau region. To interpret this data, the file was loaded to MatLab. Figure 3.7 shows a typical frame from the video, where the stereo pairs are located on the bottom left and top right of the screen. These images were extracted from each frame and processed. The results will be discussed in Chapter 5.



**FIGURE 3.9 Sample Frame from UAV-captured Stereoscopic Video**

### **3.4 MatLab Toolboxes and Functions**

The algorithms and techniques described in chapters 4 and 5 were implemented in MatLab (R) Version 7.0.1.24704 (R14) Service Pack 1. The stereo images that were processed were treated as matrices of pixels and all image transformations were applied as matrix operations. The 'Image Acquisition Toolbox' (TM) was used to extract images from video sources as well as to pre-process the images and the 'Image Processing Toolbox' (TM) was used for various image transforms, edge detections as well as correlation between stereo pairs. The algorithms were evaluated within the development environment and were not compiled as standalone executable files.

## **Chapter 4**

### **Experimental Results of Disparity**

### **Analysis and Distance Measurements**

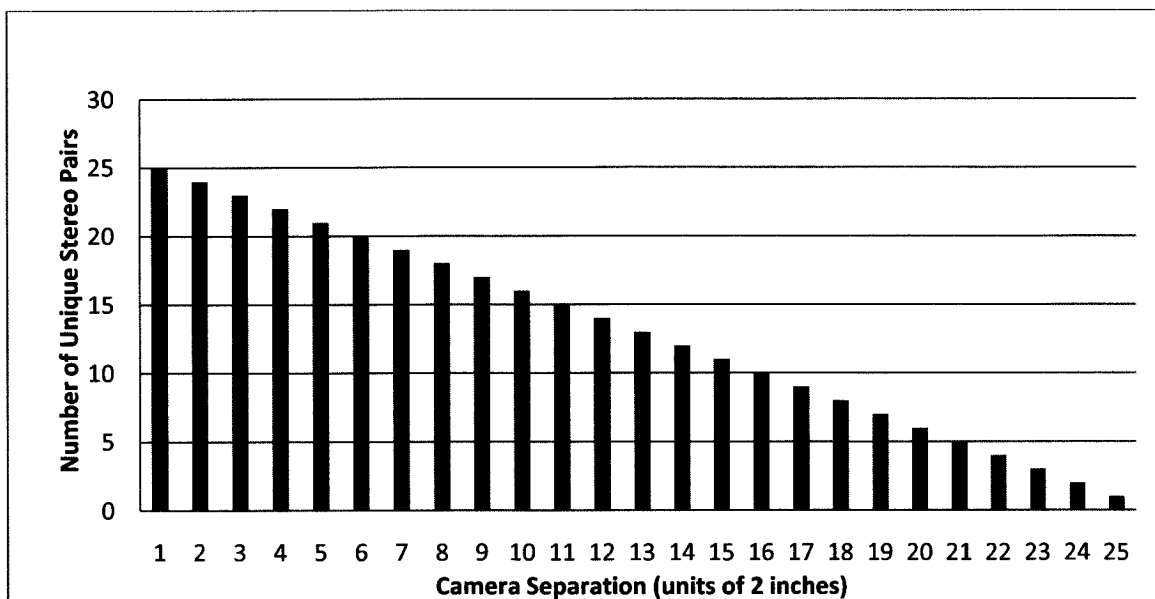
In stereoscopic imaging, the distance from the cameras to objects in the field of view can be determined by measuring the disparity that the objects exhibit. As described in section 2.3.3, it is necessary to calibrate the system in order to make any useful predictions about distance. This chapter presents a data analysis of the correlation between the baseline separation of the cameras and the stereoscopic reconstruction performance as well as an analysis of the effects of camera motion on system performance. A general calibration procedure is then presented which does not require a detailed knowledge of the camera or lens model.

## 4.1 Baseline Separation

In the epipolar geometry that is characteristic of stereoscopic imaging, the separation between the cameras determines the disparity between the two images. A large camera separation means that an object will appear significantly shifted in the different fields of view. As this separation approaches zero, the images become more and more alike and the shift ultimately disappears. In this thesis, the dynamic range of the disparity values is defined to be the difference between the largest and smallest observable disparity values for a given distance range. Therefore, for a small camera separation, the disparity values have a low dynamic range. When the separation between the cameras is large, the dynamic range of the disparity values will also be large and this will translate to a higher resolution when predicting the distances to the objects being observed. In addition, the larger camera separation makes it difficult to ensure that the cameras do not move significantly with respect to each other. The task of choosing an appropriate camera separation becomes one of striking a balance between a high dynamic range in the disparity measurements and an acceptable level of correlation quality.

### 4.1.1 Effect of Baseline Separation on Disparity Values

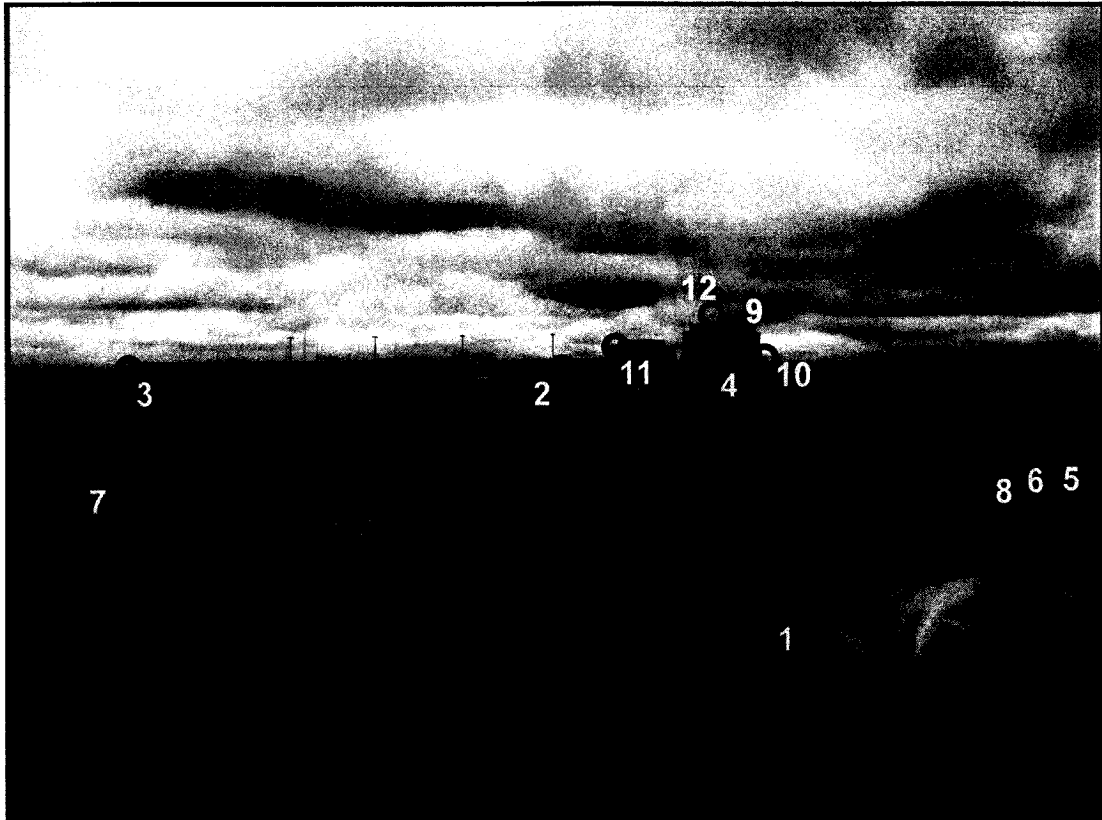
In order to quantify the effect of camera separation on the measured disparity values, a series of tests was carried out using the setup described in section 3.1. In each test, twenty-six images were taken by the camera. Each image was taken at a different camera position such that there is a difference of 2 inches separating each consecutive position in the direction perpendicular to the camera axis (Figure 3.1). Since this is a manual procedure, there is an inherent error in the positioning of the camera. Section 4.2 examines the repercussions of translational motion of the camera. Each test set, which was composed of twenty-six images, resulted in 325 unique stereo pairs per test according to Equation 3.1. It is important to note that because of the way in which the images were combined (Figure 3.1), the distribution of the camera separations in this data set is not uniform. Although the separations range from two inches to fifty inches, there are more stereo pairs with smaller separations. The number of stereo pairs for a camera separation  $x$  is given by:  $\frac{1}{d}x + n$  ; where  $d$  is the incremental separation and  $n$  is the number of images in the set. Figure 4.1 shows a histogram of this skewed data set.



**FIGURE 4.1 Number of Stereo Pairs at Different Camera Separations**

One of the basic assumptions for simplified stereoscopic imaging is that the camera axes are always parallel to each other (section 2.3.2). In this experiment, the camera was moved from one location to the next manually and this introduced uncertainty in the alignment of the camera. The extent of this problem is discussed in detail in section 4.2. Section 3.1 describes how the cameras were stabilized in order to ensure that the stereoscopic reconstruction is accurate.





**FIGURE 4.2 Training Environment and Selected Points of Interest**

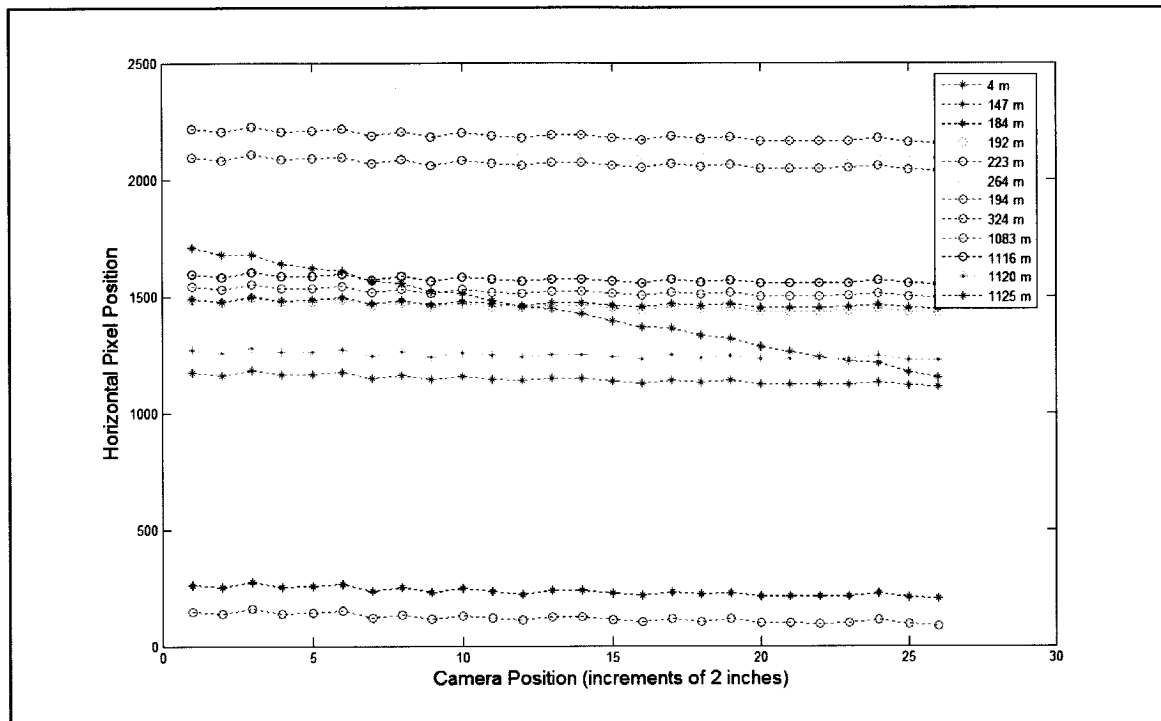
In this set of images (pointing in a south-eastern direction), twelve specific points were chosen to provide the data (Figure 4.2). These points are easily identifiable in all of the images and they are static. They are mostly located on straight edges in the vertical direction (ex: side of a building) which allows for more confident manual measurements of the disparities. The distance from the camera to these points was estimated using Google Earth. The position of these points ranged from approximately 147m to 1125m away from the camera. Table 4.1 shows their distribution. Additionally, a control point at 4m was used. Since objects very close to the camera exhibit clear and substantial motion as the camera is shifted, these points can be used to ensure that the positions of the

cameras are relatively equidistant and incremental. It should be noted that there are no data points between 327m and 1086m due to the lack of clearly defined points in this range.

Object Number	Object Description	Approximate distance (m)
1	control point on roof (MC Carleton University[CU])	4
2	edge on CU garage roof	147
3	CU garage NE side edge	184
4	CU garage SW side edge	192
5	1 <sup>st</sup> street light on Campus Ave	223
6	2 <sup>nd</sup> street light on Campus Ave.	264
7	SW corner edge of maintenance building (CU)	194
8	Sign between University Rd and the Rideau River	324
9	Canada Post building (NW corner)	1083
10	Canada Post building (SW edge)	1116
11	Building next to Canada Post NE edge	1120
12	Canada Post building antenna	1125

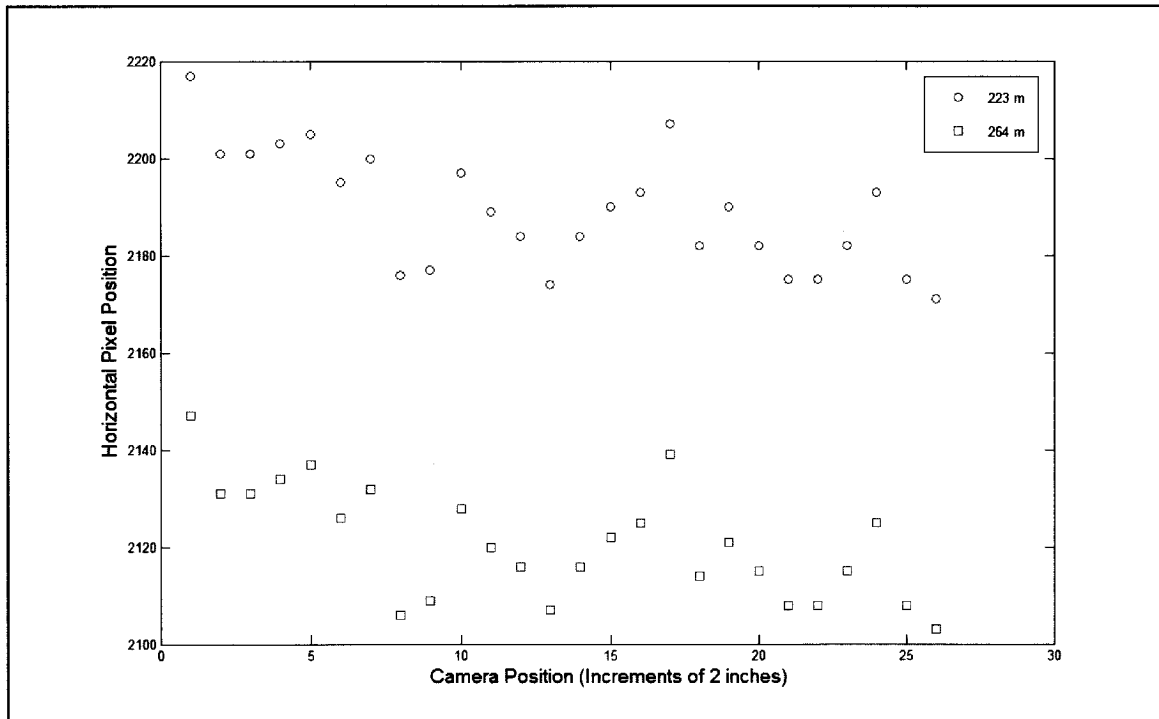
**TABLE 4.1 Training Objects and their Respective Distances**

Once the images were acquired, they were transferred to MatLab and the pixel coordinates of each of the twelve points were measured for all the images. Figure 4.3 shows the data for one of the tests.



**FIGURE 4.3 Position of Training Objects at Different Camera Locations**

These lines represent the apparent motion of the points in the field of view as the camera position changes. Although the point at 4m has the most exaggerated apparent motion because of its proximity to the camera, all the points exhibit a similar motion to varying degrees. Every image in the data set can be imagined as a vertical cross-section of Figure 4.3.

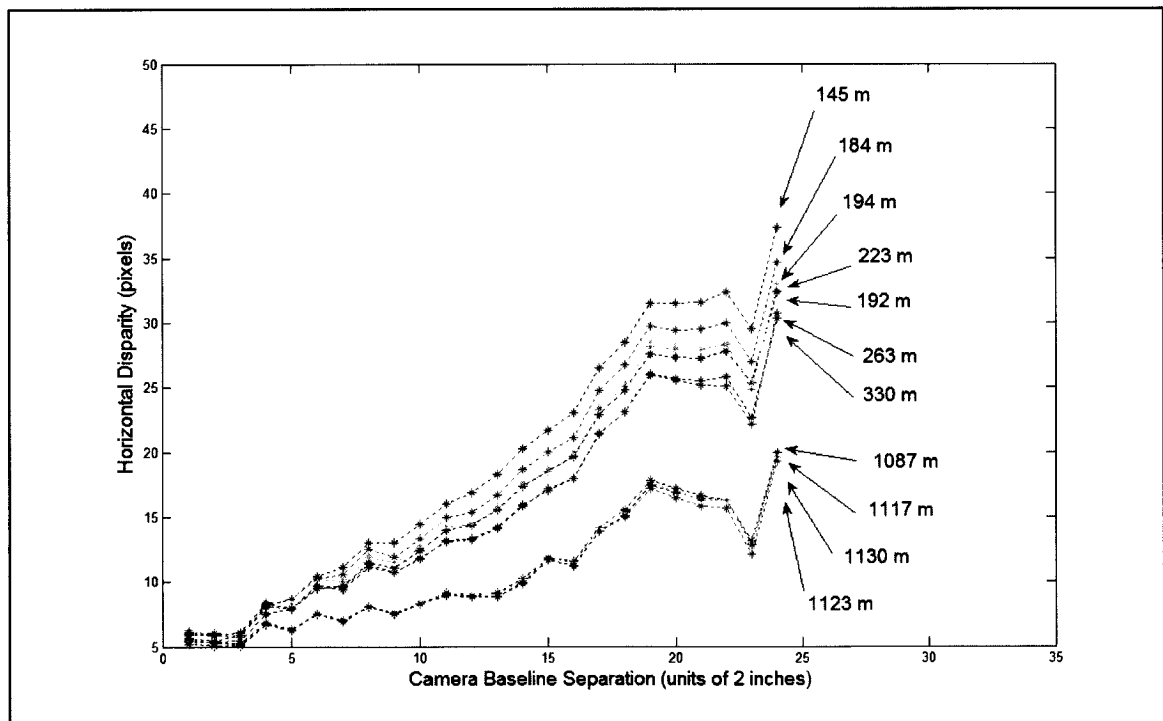


**FIGURE 4.4 Position of Two Training Objects at Different Camera Positions**

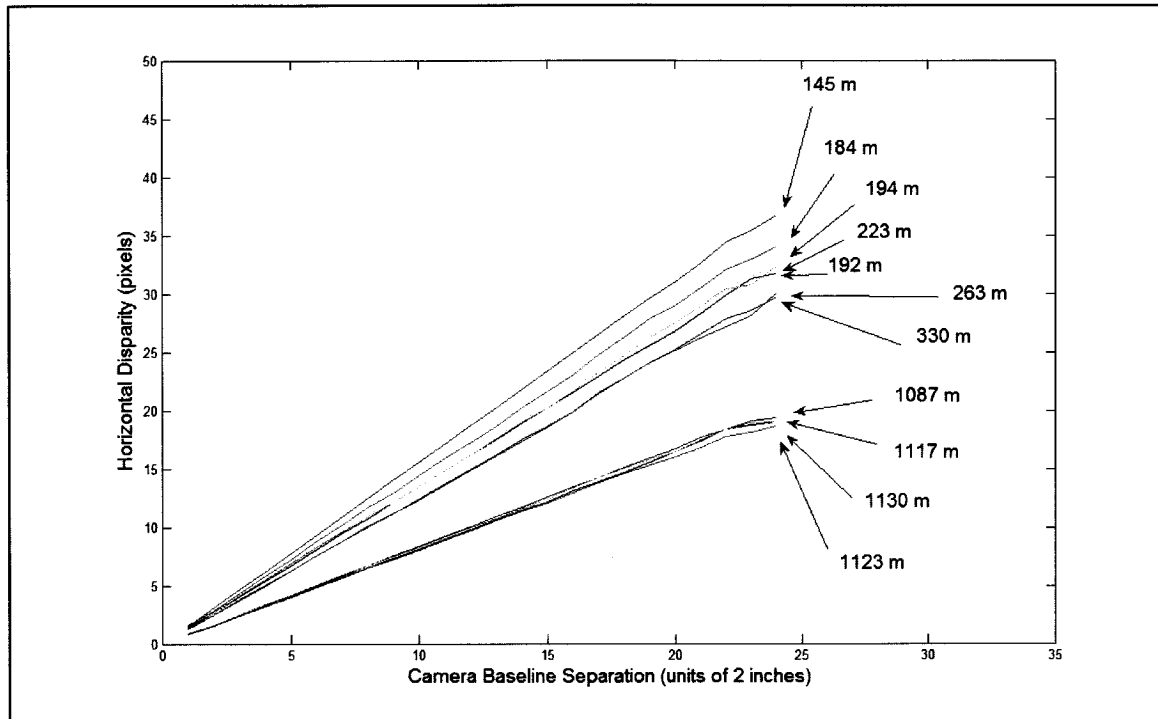
Figure 4.4 shows a close up of two of the objects. There are two obvious properties to be noted. First, all of the objects in the data set exhibit a general downward slope as the camera is shifted. Secondly, all of the objects exhibit similar patterns of movement. It is obvious from this that each image (i.e. cross-section) introduces a specific motion of the camera that influences all the points being observed uniformly. These deviations are misleading when calculating disparities because a deviation due to camera motion is easily interpreted as an actual shift in the apparent position of the point. To counteract this source of error, an adjusted data set was constructed. To calculate these adjusted values, linear regression was carried out on the control point (4m from the camera) and the specific deviation at each camera position was recorded. All the other points

were then adjusted according to these values. After this adjustment, it was observed that the curves corresponding to all the objects 'collapsed' to their respective trendlines as predicted by linear regression.

All the combinations of stereo pairs were extracted from the data set as described in section 3.1. The data points were grouped by camera separation and an average disparity value for each object point was calculated in each group. The disparity value is calculated simply as the difference between the horizontal pixel positions of a given point. No other deviation is assumed, in accordance with the epipolar constraint described in section 2.3.2. Figure 4.5 shows the compiled results with the original values and Figure 4.6 shows the results using the values adjusted for camera motion. These results are compiled from three separate tests each containing 325 unique stereo pairs.



**FIGURE 4.5 Disparity vs. Camera Separation**



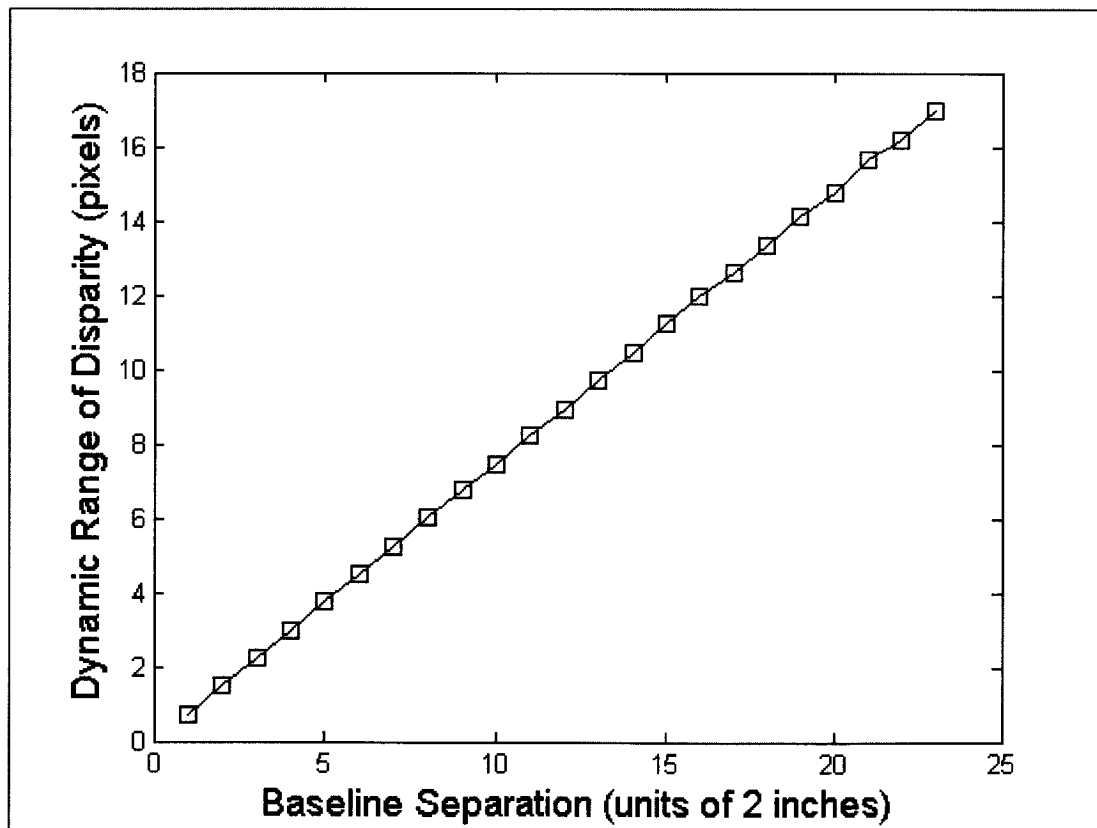
**FIGURE 4.6 Adjusted Disparity vs. Camera Separation**

Note that in Figure 4.5, the camera motion in the original measurements results in large disturbances in the disparity measurements. This is especially true at the larger camera separations because there are fewer data points at those separations (Figure 4.1). The adjusted data results in less noisy disparity measurements and results clearly indicate a positive linear relationship between disparity and camera separation.

It can also be seen from Figure 4.6 that there is a strong relationship between slope and object distance. The distribution of the values is observed in more detail in the next section.

### 4.1.2 Dynamic Range of Disparity Values

In Figure 4.5 and Figure 4.6, it is clear that as the camera separation increases, the disparity measurements for the observed points become more separated and distinguishable. This property of the measurements can be quantified by the dynamic range of the disparity values as defined in section 4.1. To calculate this value, vertical cross-sections of Figure 4.6 are used and the difference between the minimum and maximum disparity value for each cross section is calculated. Therefore, in this case, the metric represents the dynamic range of the disparity for objects that are between 147m and 1125m. Figure 4.7 shows the relationship between the dynamic range and the camera separation.



**FIGURE 4.7 Dynamic Range of Disparity Values**

Clearly, there is a positive linear relationship between the dynamic range of the disparity values and the camera separation. If a simple linear regression is used with the intercept set at the origin then the dynamic range of the disparity values for this camera setup can be described as

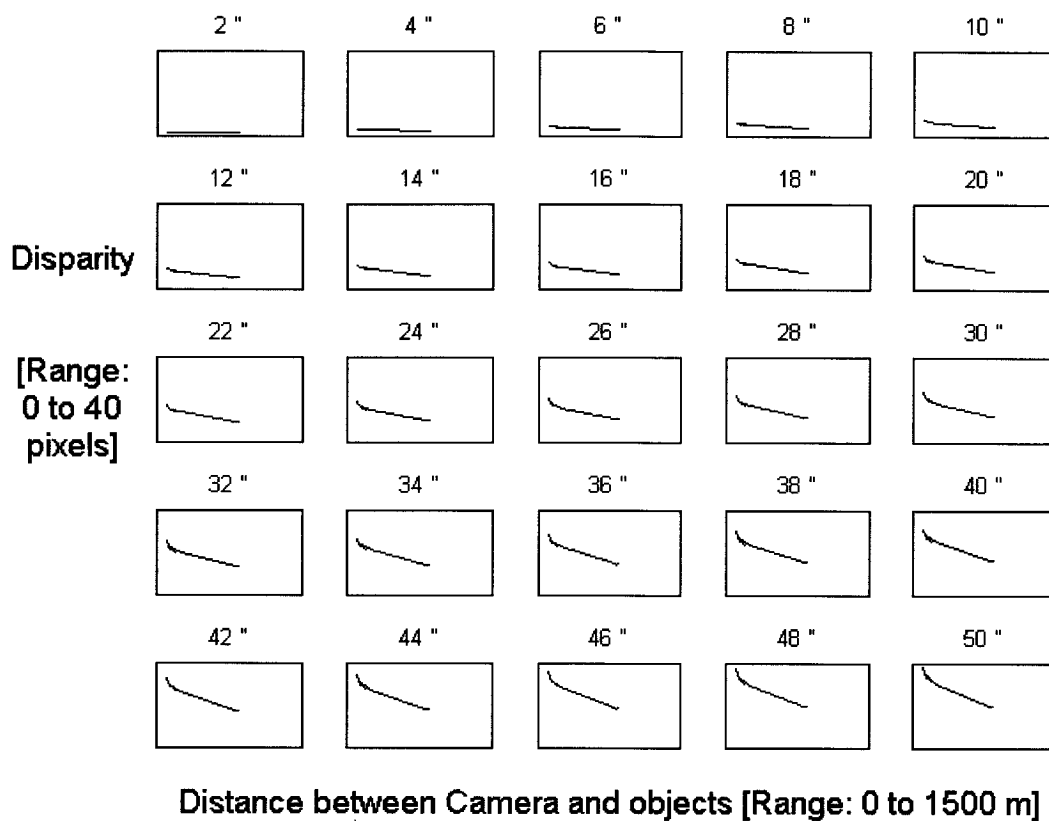
$$DR = (0.37) \times CS \quad [\text{EQN 4.1}]$$

Where CS is the camera separation (in unites of inches/2). Equation 4.1 also confirms that deriving the distance of a point from the disparity measurement is fundamentally related to the camera separation as suggested by the relationship

$$Z = \frac{cB}{d} \quad [\text{EQN 2.3}]$$

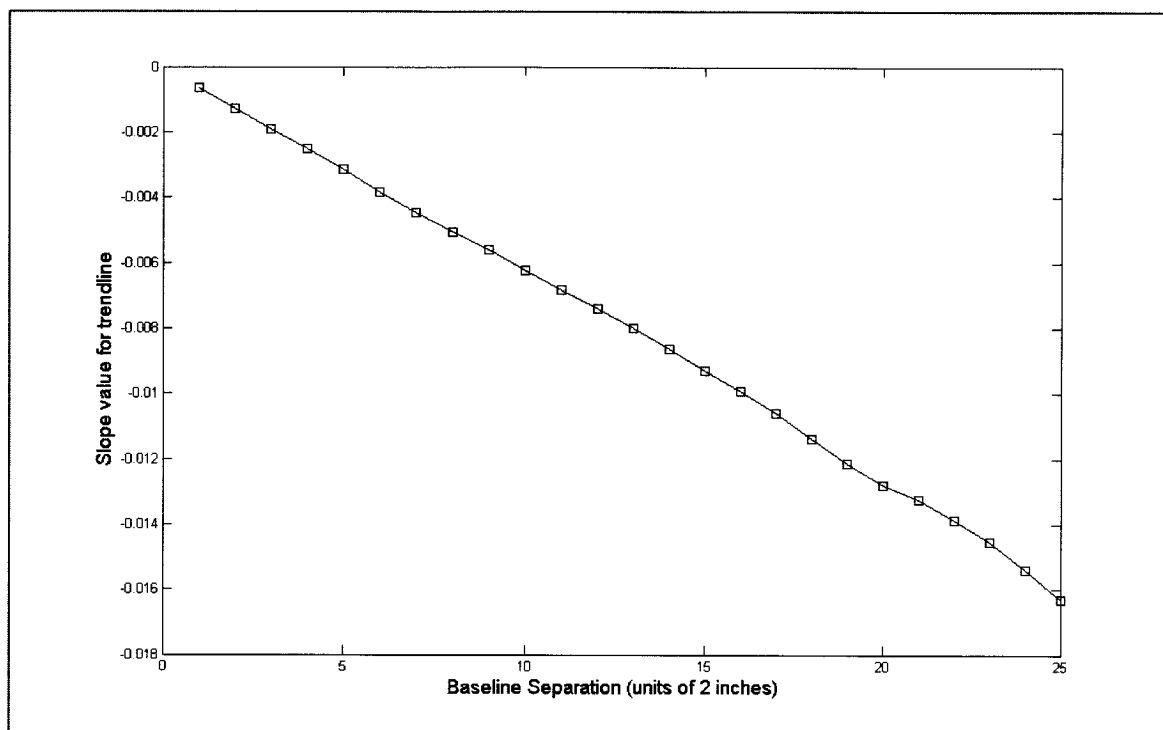
To investigate this relationship further, the correlation between distance and disparity was observed at each camera separation (Figure 4.8). Each subplot in Figure 4.8 represents a different camera separation and includes all of the objects measured in the experiments.



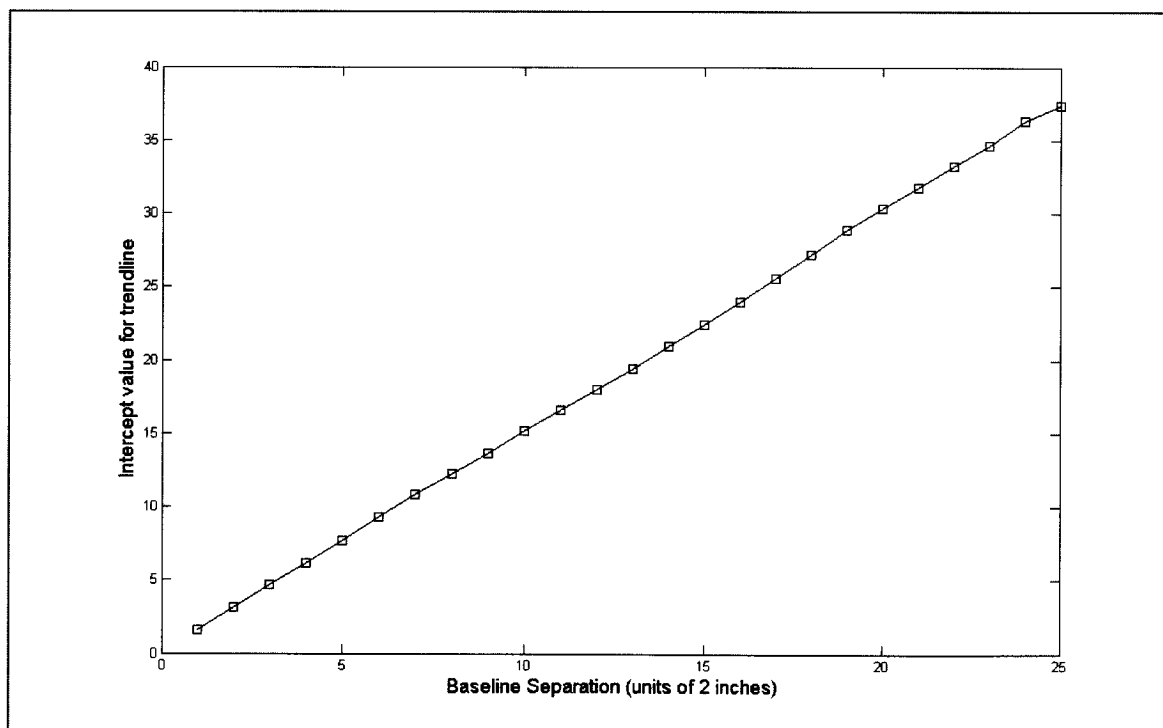


**FIGURE 4.8 Disparity vs. Distance at Different Camera Separations (2 - 50 inches)**

At each camera separation, there appears to be a negative relationship between the distance from the camera to the object and the disparity value. To simplify the analysis, it is assumed that this relationship is linear and regression is carried out at each camera separation. It is clear that there is a correlation between the slope and intercept of these lines and the camera separation. The slopes and the intercepts from each of the subplots in Figure 4.8 are then plotted to give Figure 4.9 and Figure 4.10.



**FIGURE 4.9 Behaviour of the Slope of the Trend Line**



**FIGURE 4.10 Behaviour of the Intercept of the Trend Line**

The well-defined linear relationships in Figure 4.9 and Figure 4.10 can be used directly to find the distance of an object given its disparity at a given camera separation. Using a linear regression intersecting with the origin, the equations for the trend line parameters are:

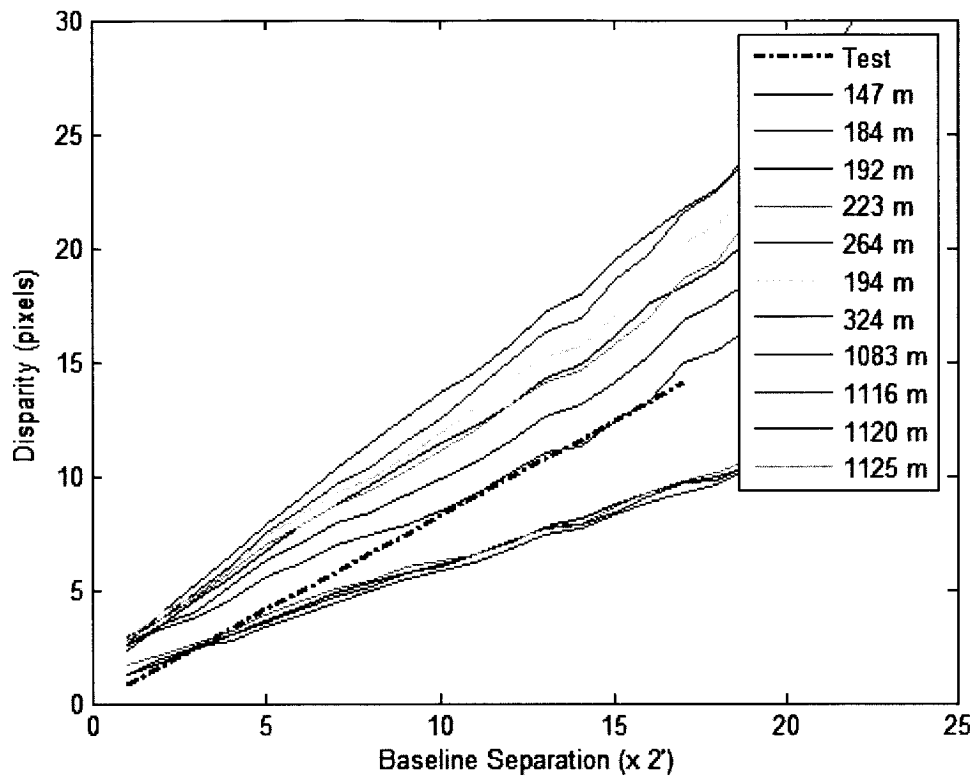
$$m = (-3 \times 10^{-4}) \times CS \quad [\text{EQN 4.2}]$$

$$b = (0.75) \times CS \quad [\text{EQN 4.3}]$$

Where  $m$  is the slope parameter,  $b$  is the intercept parameters and  $CS$  is the camera separation (in unites of inches/2). By applying this above analysis, a given baseline separation can be used to predict the dynamic range of the disparity values that are to be expected for a certain distance range (Equation 4.1). Additionally, the linear equation that relates disparity to distance has a slope defined by (Equation 4.2) and intercept defined by (Equation 4.3).

### 4.1.3 Distance Prediction Test Using Disparity Values

In order to verify the usability of the results presented in this section, the same setup was used to predict the distance from the camera setup to a new point in a different environment. The distance of the object to the cameras is approximately 310m. Three sets of stereoscopic measurements were made and adjusted as described in section 4.1.1. The resulting curves are averaged and overlaid onto the previous results (Figure 4.11).



**FIGURE 4.11 New Data plotted over Training Set**

The new data is clearly consistent with the training set since the new curve lies in the expected region near the 310m range. The curve that defines the relationship between the distance ( $R$ ) and the disparity ( $D$ ) can be expressed by the linear equation:

$$D = mR + b \quad [\text{EQN 4.4}]$$

Where the slope ( $m$ ) and the intercept ( $b$ ) are determined from Equations 4.2 and 4.3 respectively. The distance was calculated using this method and was determined to be 274m. Since the distance of the object is approximately 310m,

this yields an absolute error of about 12%. Additionally, because the objects in this experiment spanned from about 147m to 1125m, this prediction error represents about 3.7% of the dynamic range. In comparison, a “high accuracy” ground stereo vision system with a much more limited maximum range of 90m (less than 10% of the maximum range presented here) can make predictions within a 2% error margin [42]. The error range generated in this experiment is also comparable with range error predictions made for the Mars Rovers where the expected error after applying the disparity to range equations is around 2% but fluctuates up to 10%. The maximum range of that system is also much lower than what is presented here and reaches these error levels at a distance of 14m. [34]

## **4.2 Effect of Camera Motion on Disparity Measurements**

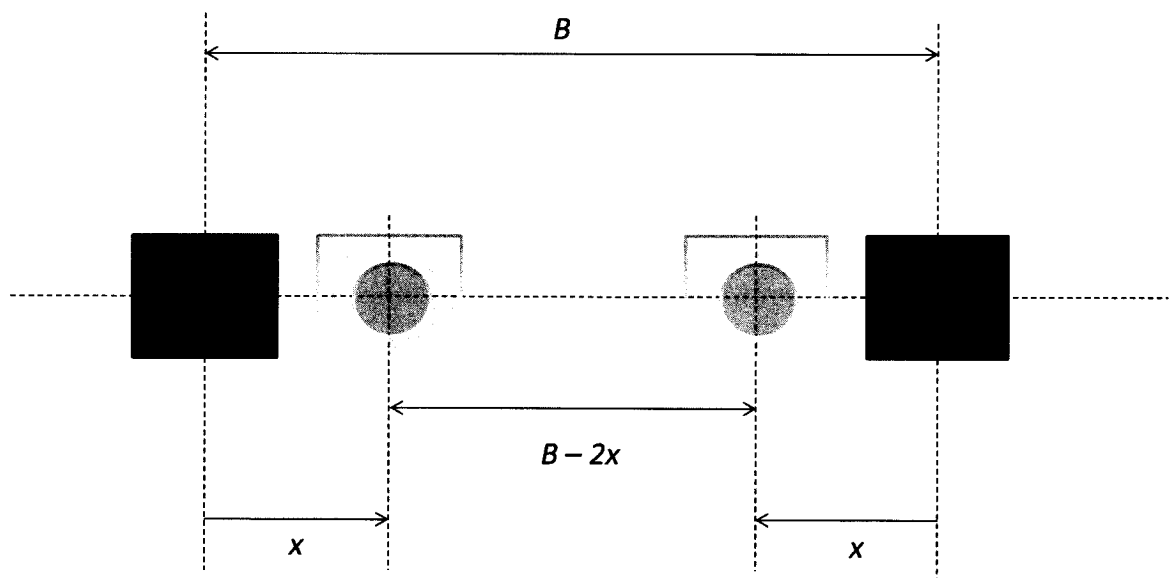
The measurements that are described in section 4.1 are very sensitive to any camera motion or misalignment. Several sources of such motion have been identified in section 2.3.4. This section focuses on a simple translational motion in which the separation between the parallel axes of the cameras becomes either smaller or larger than what is intended. Since the two cameras in a stereoscopic vision system are usually attached to each other such that there is no room for large deviations, translational motion is usually due to the vibration. In order to model this type of camera motion it is assumed that the axes of the cameras are always parallel to each other and that the cameras can move only in the

horizontal direction. A similar motion in the vertical direction introduces the additional problem that the epipolar lines in the images are no longer coincident. This means that the images must first be vertically shifted before correspondence can occur (motion in the other directions is discussed in section 2.3.4). The relation between a horizontal translation of the camera axis and the resulting pixel shift is easily obtained from geometry (Figure 4.15 and Figure 4.16). If the nominal camera separation is  $B$  and if an arbitrary unit of translational motion,  $x$ , is assumed, then the smallest possible camera separation will be

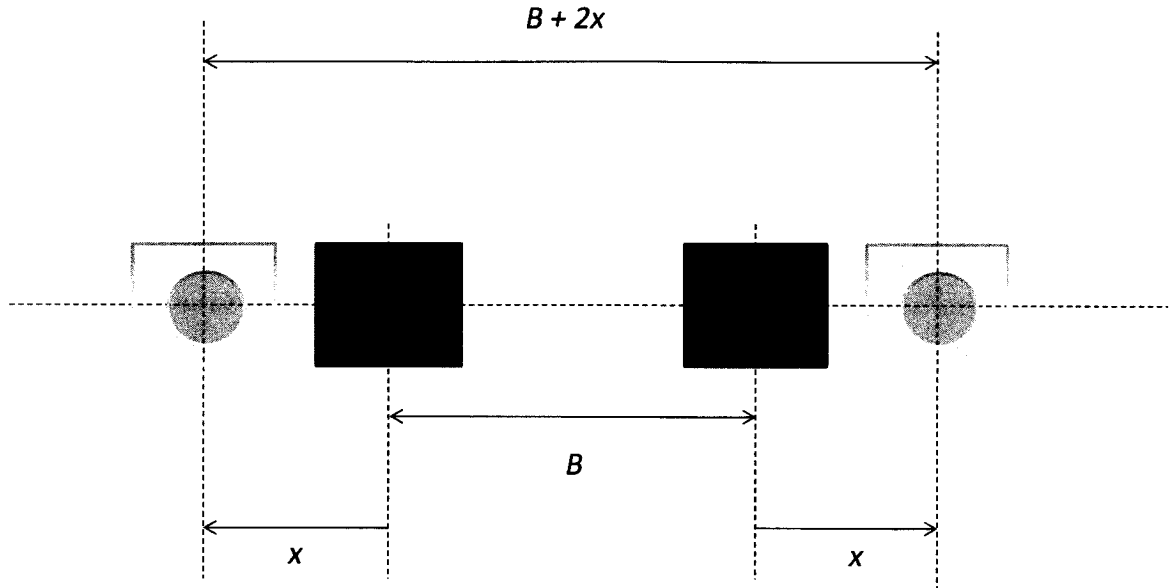
$$B - 2x \quad [\text{EQN 4.5}]$$

Similarly, the largest camera separation will be

$$B + 2x \quad [\text{EQN 4.6}]$$



**FIGURE 4.12 Minimum Camera Separation due to Translation**



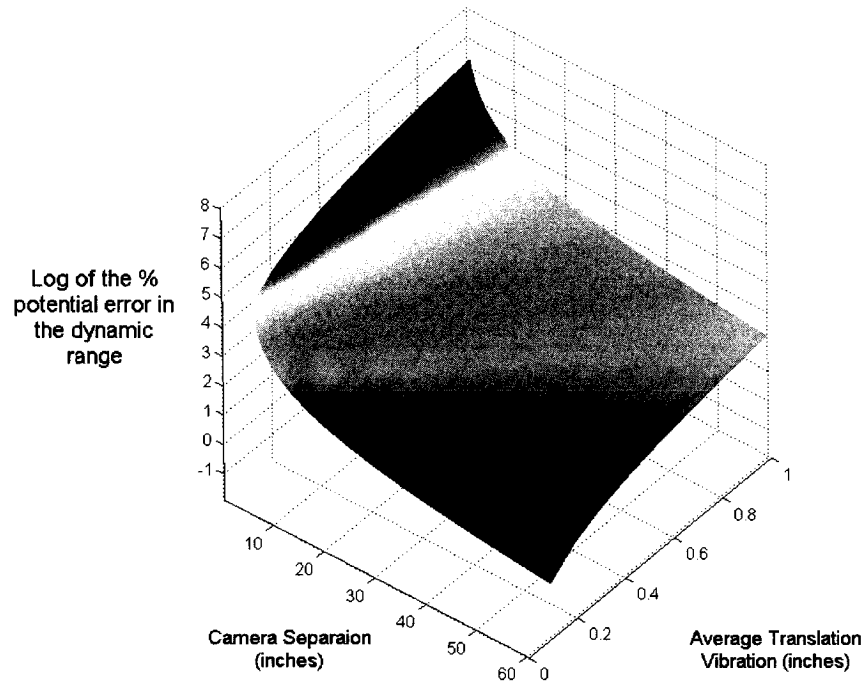
**FIGURE 4.13 Maximum Camera Separation due to Translation**

If it is assumed that vibration will induce a maximum displacement of value  $x$ , then the translational component of the vibration can be modelled in this way and for a nominal camera separation, an error range due to translational motion can be estimated. Using this error range, the relationships described in section 4.1 can be updated. Equation 4.1 gives the relationship between camera separation and dynamic range for the give camera setup. The error bounds on the dynamic range, given an amount of translational motion, can be calculated by determining the minimum and maximum separation for a nominal separation and re-evaluating the dynamic range. The variability of the dynamic range for this camera setup is therefore given by:

$$U_{DR} = (0.37)(B + 2x) \quad [\text{EQN 4.7}]$$

$$L_{DR} = (0.37)(B - 2x) \quad [\text{EQN 4.8}]$$

where  $U_{DR}$  is the upper limit of the dynamic range,  $L_{DR}$  is the lower limit of the dynamic range, and  $x$  is a unit of translational motion representative of the expected motion.



**FIGURE 4.14 Potential Error in the Dynamic Range due to Translational Motion**

As expected, the effect of translation motion is most prominent at small camera separations. For instance, at a baseline separation of 25.4 cm (10 inches), and an average translational motion of 0.64 cm (0.25 inches), the dynamic range of the disparity values can be expected to oscillate within 5 percent of its nominal value. Similar bounds can be obtained for the intercept and the slope of the trend line that describes the relationship between disparity and distance.



## **Chapter 5**

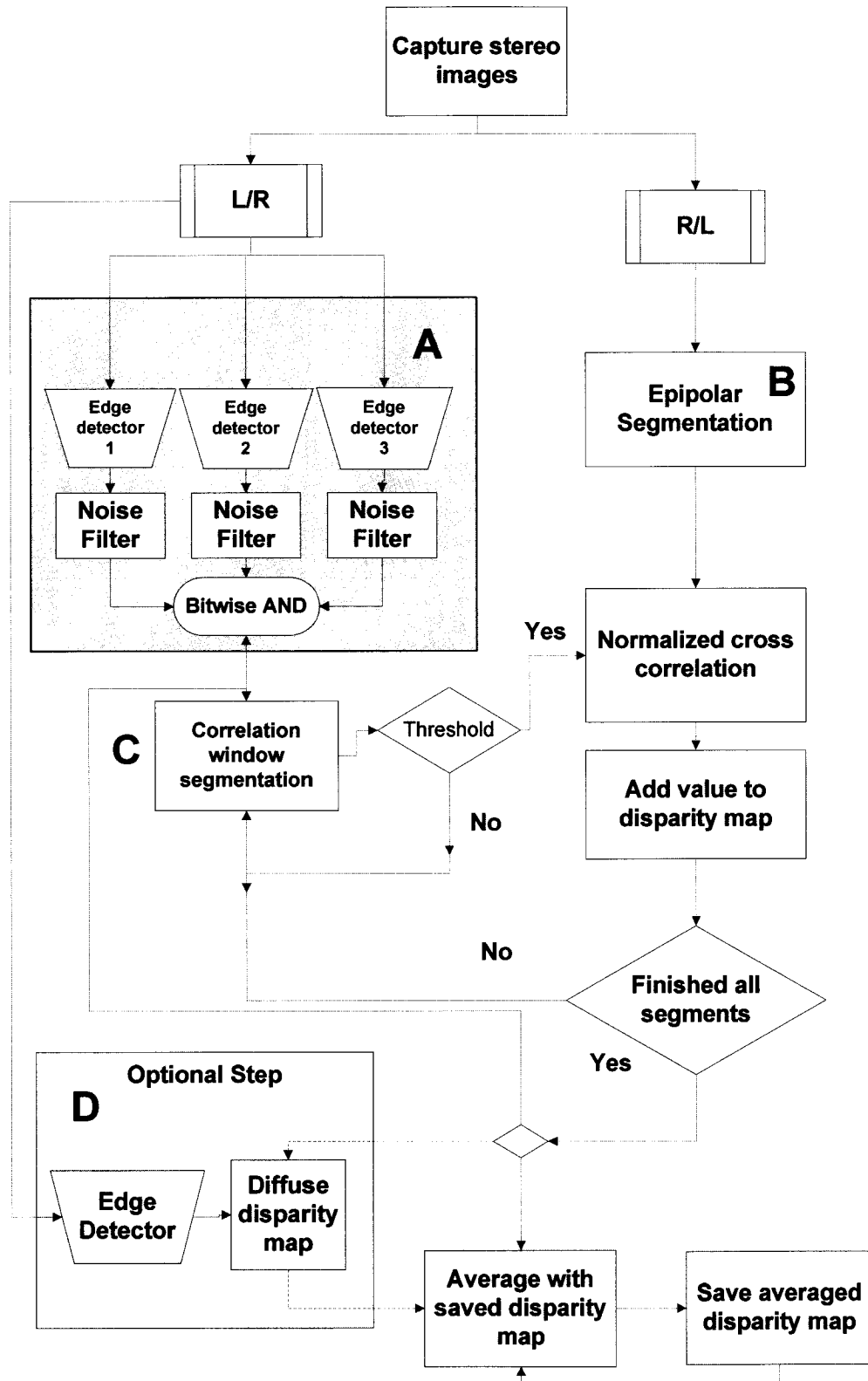
# **Real-time Implementation of a Stereoscopic Obstacle Detection System**

In the previous chapter, an image data analysis of stereoscopic imaging for obstacle detection in unmanned aerial vehicles (UAVs) was presented. This chapter presents a specific algorithm for carrying out the necessary computations to make such a system possible. When attempting to carry out obstacle detection for low-flying, terrain-following UAVs, there are two central requirements that cannot be compromised. The first requirement is that the obstacle detection algorithm must be effective at determining the position of any object that might be in the path, or on a collision course with the aircraft. The second equally important restriction is that the algorithm must be able to detect obstacles in real time. Because the aircraft may be moving at high speeds (30 to 50 m/s) and

since its proximity to the terrain predisposes it to a larger number of obstacles, the real time requirement is essential for any viable system. Stereoscopic image reconstruction can be very computationally intensive, mainly due to the correlation procedure which often becomes a restrictive bottleneck in real-time systems. Without an extremely powerful processor, it would be impossible to carry out complete pixel-by-pixel stereo analysis on real-time data [21], [24], [31], [32], [33]. This chapter presents an algorithm (Sparse Edge Reconstruction) that is tailored for this type of situation. The algorithm implements very coarse and sparse stereoscopic imaging techniques which over time are used to generate a refined three-dimensional map. This algorithm was published in [10].

## **5.1 Sparse Edge Reconstruction Algorithm**

The purpose of Sparse Edge Reconstruction (SER) is to generate an accurate, real-time model of the environment of a low-flying unmanned aerial vehicle without intensive computation. The basic mechanism of this algorithm is a coarse-to-fine approach ([36], [20]) which only analyzes a relatively small subset of data in real-time while updating the complete model in the system memory. Figure 5.1 represents the Sparse Edge Reconstruction algorithm.



**FIGURE 5.1 Sparse Edge Reconstruction Algorithm**

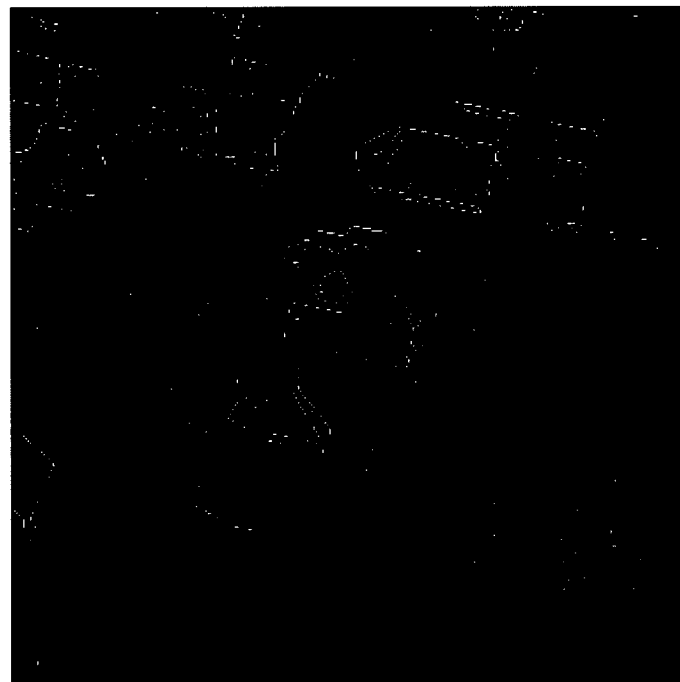
The stereo images are the inputs to the system and they are updated in real time. Instead of analyzing the images thoroughly, a preliminary step (Block A in Figure 5.1) is used to identify the areas of interest in the images. This is done based on the intensity contrast of objects with their environment as determined by linear edge detectors. Of course, contrast alone is not enough to identify an area of an image as being a potential obstacle since an object may appear to be blended with its environment. However, an assumption is made that because this process is repeated with every new pair of images, and since the vehicle (and possibly the objects) is constantly in motion, any object will, over time, become defined by its contrast even if at any one instant it does not appear visually distinct. This assumption is shown to be valid in sections 5.2 and 5.3. One of the images in the stereo pair is first put through a number of different edge detectors in parallel such that the set of edges from each detector is unique. The binary image from each edge detector is then passed through a noise filter which removes all isolated edge pixels (i.e. edge pixels with no direct neighbours). The purpose of these filters is to remove features that have high contrast with their surroundings but are not part of some larger object that might pose a threat to the aircraft. This step is very helpful in separating objects from the terrain. The filtered binary images are then combined together with a bitwise AND operator. This results in a single binary image with a higher sparsity than any of the component images. The number of edge detectors in the algorithm is tuneable and the sparsity of the image (and consequently the speed of the system) depends on the number of unique edge representations that can be combined at the AND gate.

Once the original image has been reduced to a manageable set of edges, the correlation procedure can be carried out between the two images at the points that have been identified by block A in Figure 5.1.

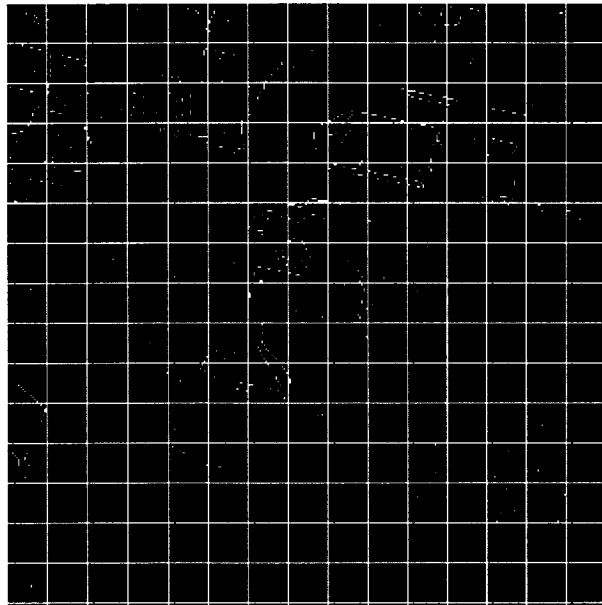
Before the correlation procedure, however, the two stereo images must be segmented according to the assumed epipolar geometry (blocks B and C in Figure 5.1). As described in section 2.3.2, an assumption is made that the axes of the cameras are parallel to each other and lie on the same horizontal plane. The reason for this simplification is that points in one image are now constrained to a single horizontal line in the other image. This in turn results in a drastic reduction in the computations involved in the correlation procedure. As described in 2.3.1, the images are segmented into correlation windows. The size of these windows will determine the speed and the efficiency of the correlation. In [34], it is suggested that for a 1024x768 image resolution, correlation windows of size 7x7 to 9x9 pixels ( approximately 0.01% of image size) be used for a detailed correlation while windows of size 15x15 to 29x29 pixels (0.03% - 0.1% of image size) are better for fast analysis. To further increase the speed of the system, a segment in the image that has been processed by the edge detectors will only be correlated if the number of detected edge pixels in that segment is above a certain threshold. Figures 5.2, 5.3, 5.4, 5.5 and 5.6 show the total effect of applying the edge detectors, segmentation and threshold to an arbitrary image.



**FIGURE 5.2** Original Image before Processing (300x300 pixels)



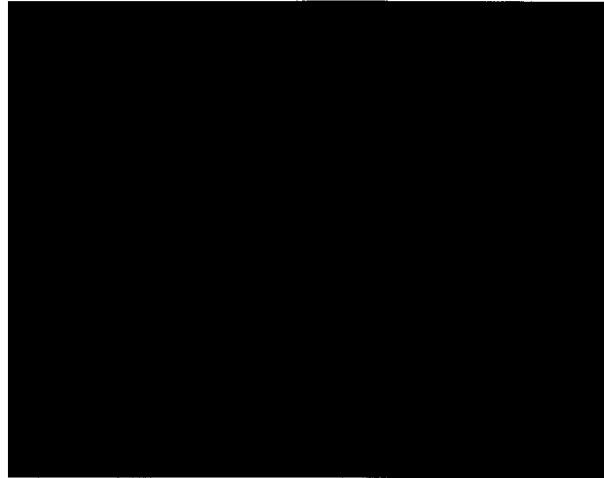
**FIGURE 5.3** Image after Sparse Edge Detection



**FIGURE 5.4 Correlation Windows (20x20 pixels)**



**FIGURE 5.5 Intensity Image of the Number of Pixels in each Window. Brighter Windows Correspond to higher numbers of Pixels Detected**



**FIGURE 5.6 Binary Image after the Application of a Threshold of 5 pixels**

Once the window segments to be correlated are identified in the first image, they are correlated to the second image along the same horizontal line. The correspondence between the images is carried out using the normalized cross correlation [35], [section 2.3.1]. Using this equation gives a correlation coefficient for the different positions in the second image where the correlation window from the first image might be. The higher the coefficient, the more likely it is that the position in question is the true counterpart of the window segment from the first image. In order to find a correspondence between the images, the position of the maximum value of the correlation coefficient is determined for each window segment. The difference between the horizontal position of the window segment and the location of the coefficient maximum is the disparity between the two images for that object. This procedure is carried out for each window segment and the resulting values are used to generate a disparity map of the image [16]. This map contains the values of the disparities located at their respective positions. Given that disparity can be used to directly infer distance

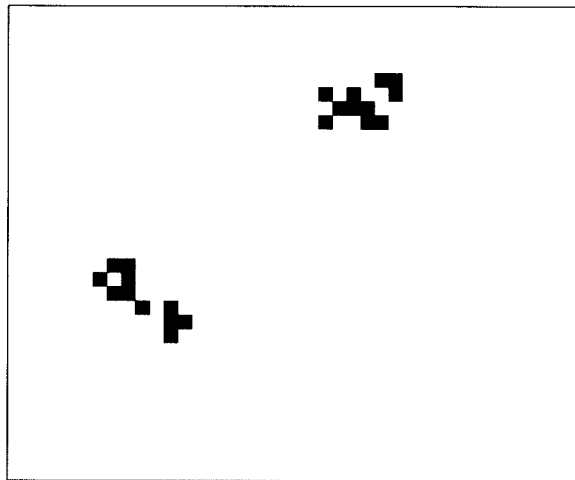


[section 2.3.3], a disparity map is essentially a model of the distances between the cameras and the objects in the images. It is important to note that the resolution of the disparity map is entirely dependent on the size of the window segments since a single disparity value is calculated for each segment. This should not be confused with the resolution of the disparity measurements since the correlation function provides a measurement in terms of pixels. Therefore, the resolution of the disparity measurements is the same as the horizontal resolution of the images.

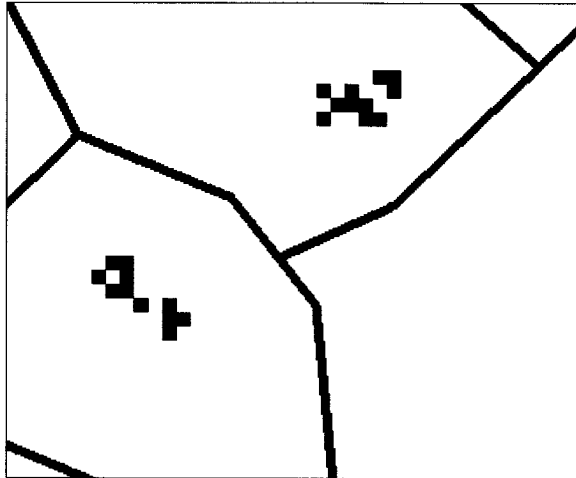
A fundamental component of the SER algorithm is the environment model which is a saved disparity map that is constantly being updated. Every time a sparse disparity map is generated from a new pair of images, it is averaged with the saved map. The rationale behind this system is that although individual disparity maps may be very incomplete or may unduly emphasize different aspects of the images, as new disparity maps are generated with different lighting conditions and camera positions (given a moving vehicle), the model will come to emphasize real objects and deemphasize everything else. It is to be expected that the system model will be incomplete for a short initialization period and then stabilize to an accurate assessment of the environment.

There is an optional step included in this algorithm (Block D in Figure 5.1) where the disparity map is 'diffused' or 'blurred' in a controlled way before combining it with the saved disparity map. This option should only be used when the sparsity of the image after the AND gate is still too low given a certain threshold. This might occur, for instance, in urban environments with many

visually distinct obstacles that are spatially separated. Since the low sparsity will translate to more image segments that meet the correlation threshold condition, a higher computation time will be required for the stereo pair. An extra step is, therefore, added to extract more information and attempt to make up the loss in efficiency. The original image is put through an edge detector that is more detailed than those in block A in Fig 5.1 such as a *Canny* detector, for instance. The purpose of this detector is to find the edge boundaries in the image. The disparity values that have already been measured are then allowed to 'bleed' into the image until they encounter an edge (Figures 5.7, 5.8 and 5.9)



**FIGURE 5.7 Original Disparity Map**



**FIGURE 5.8 Edges Overlaid on Original Disparity Map**



**FIGURE 5.9 Disparity Map after the Optional Step is Applied**

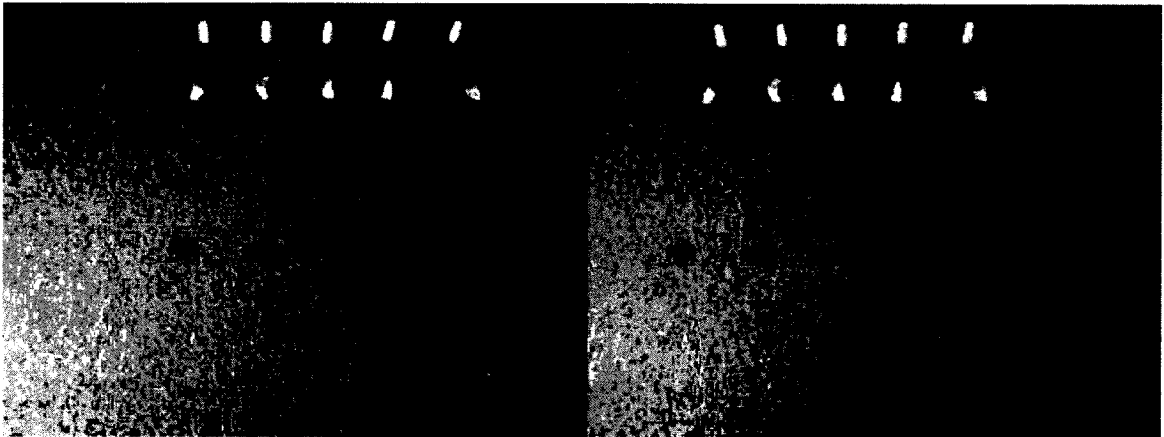
The resulting effect is that instead of supplying disparity points to the system model, entire disparity planes are generated. Of course these planes are assumed to be perpendicular to the camera axes. Note that a detailed edge

detector is itself computationally intensive and unless the resulting increase in efficiency offsets the original computational expenditure, this optional step should not be used.

Finally, it should be noted that this algorithm is asymmetric in that only one of the images is put through the edge detectors and the other image is only used for correlation. This unevenness is undesirable for two reasons. First, the position of the cameras with respect to the aircraft or other objects as well as the varying lighting conditions may cause the cameras to have different perspectives of the environment and if one camera is always used to extract the important information about a scene, there is a potential loss of information. Second, one of the cameras' view may be obstructed by dirt, insect splatter, etc... Then, in addition to a systematic error in the reconstruction, there may be an occlusion of important objects in the field of view that should be examined. The simple solution to these problems is to alternate between the two cameras such that they are both used to extract information and for correlation. In the case of camera obstruction, the resulting periodic oscillation will allow those objects that might be hidden to be represented, and the oscillation may, in fact, be used as a warning sign to take the error into account. Additionally, using both cameras is an excellent way to verify disparity measurements.

## 5.2 Testing and Performance of Algorithm

Figure 5.10 shows a test pattern that was used to evaluate the SER algorithm as described in section 3.2. The pair of images represents a set of objects with various contrasts with their surroundings. The top two rows are easily distinguishable from their surroundings while the third row is much more hidden.

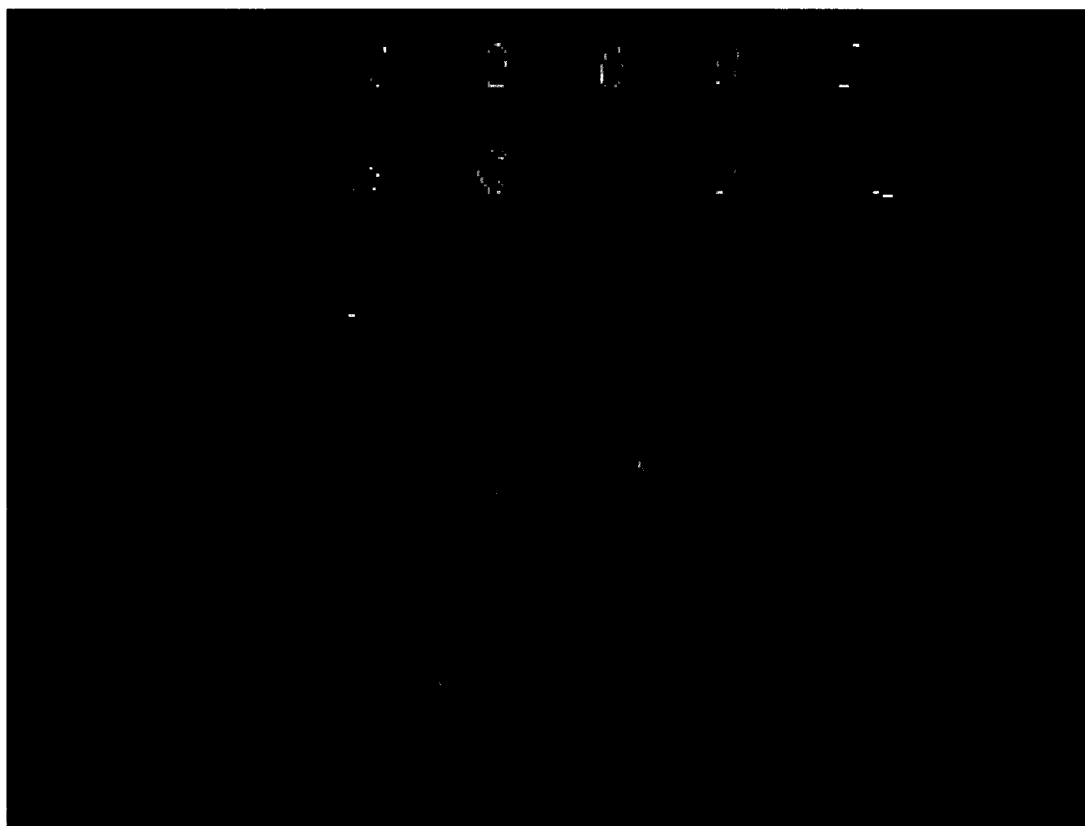


**FIGURE 5.10 Test Pattern Stereo Images**

Additionally, the textured background is very grainy and acts as a source of noise and error. As described in section 5.3 this is similar to the type of terrain that can be expected for low-flying UAVs. These are greyscale images with a resolution of 240x320 pixels.

The purpose of this test was not to carry out stereoscopic reconstruction but rather to determine how much of the information the algorithm could capture in its partial reconstructions and how effective the SER algorithm is at merging the data. In order to do this, a set of stereo pairs was collected and subjected to two different tests.

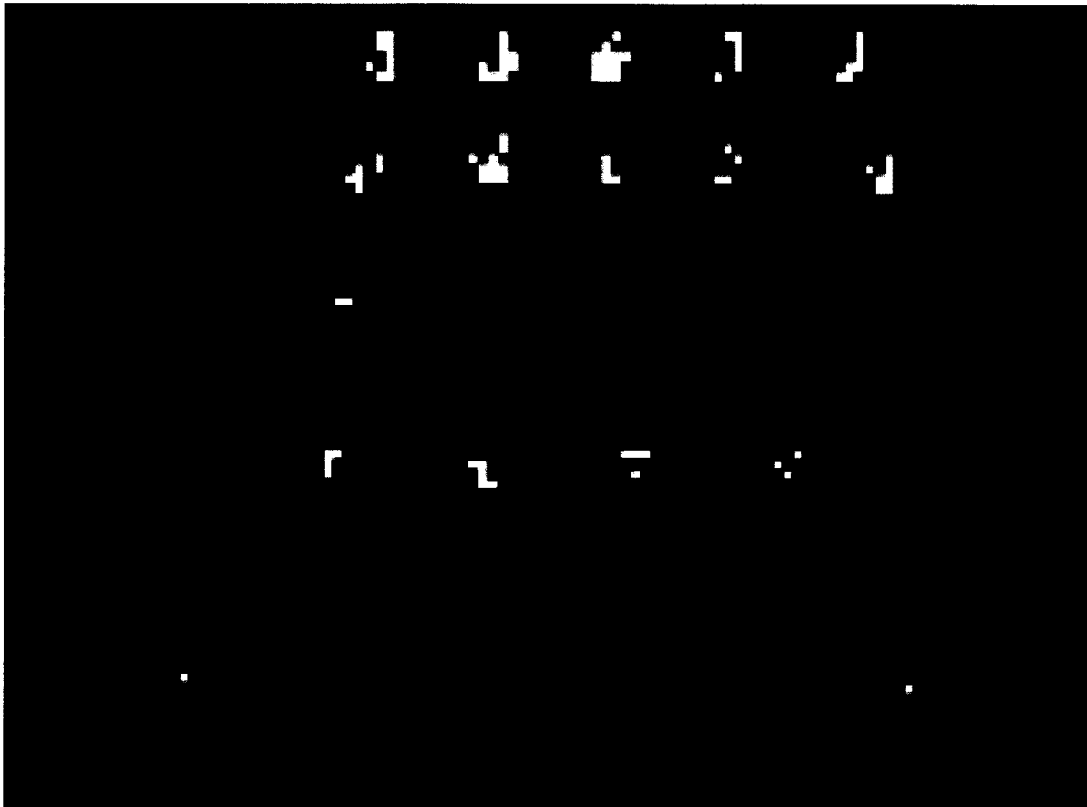
The first test was to use the Sparse Edge Reconstruction algorithm on the set of images as a sequence. Each pair of images was subjected to the partial analysis described above and then added to a system model. Figure 5.11 shows the level of sparsity of a representative image in the sequence after applying block A in Fig 5.1.



**FIGURE 5.11 Test Pattern Image after the Block A in FIGURE 5.1**

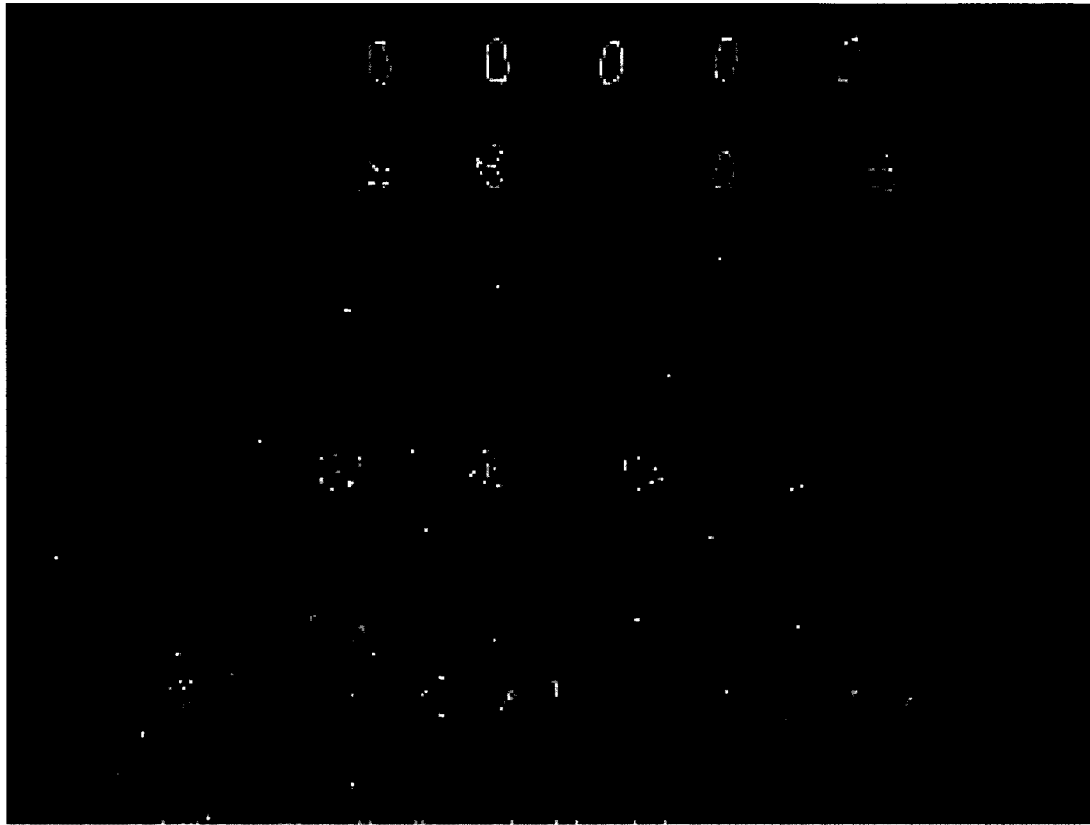
Figure 5.12 shows the overall system model after an arbitrary sequence of 10 images. In this image, the green circles are included to highlight real objects that are present in the test pattern (Figure 5.10) that have been recognized by the

SER algorithm. It is clear that the objects with the higher contrast are more easily recognized. The red squares highlight objects that were recognized by the algorithm but that were not 'real' objects in the test images (in this case these objects are due to the texture of the background).



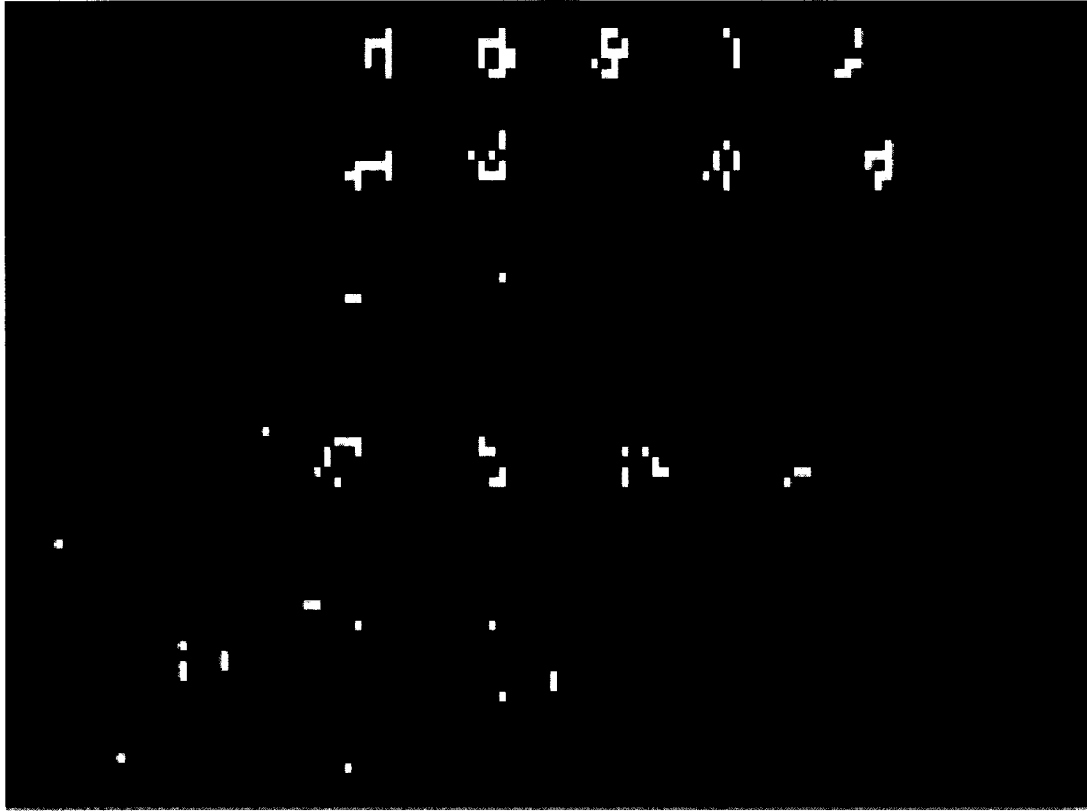
**FIGURE 5.12** Disparity map after the Sparse Edge Detection Algorithm. Green circles indicate properly identified objects and red rectangles indicate false positives

The second test was to carry out an edge detection and stereoscopic analysis on a single pair of images and repeat the process for each pair in the set. Figure 5.13 shows the representative sparsity after using a single *Prewitt* [43] edge detector and Figure 5.14 shows the reconstruction based on the edge detector.



**FIGURE 5.13** Test Pattern with the Application of a Single *Prewitt* Edge Detector





**FIGURE 5.14** Disparity map after a Stereoscopic Reconstruction with a Single *Prewitt* Edge Detector. Green circles indicate properly identified objects and red rectangles indicate false positives

There are several things that are obvious from these simulations. Firstly it is clear that the sparsity for a single pair of images is, in fact, much higher when using the SER algorithm. Additionally, it can be seen that a single detector picks up much more noise than the combination of detectors used in SER. In order to quantify this improvement the SNR value for each reconstruction was measured according to Equation 5.1

$$SNR = 20 \log \left( \frac{s}{N_1 + N_2} \right) \quad [\text{EQN 5.1}]$$

where  $S$  is the number of correctly identified objects in the image (true positives),  $N_1$  is the number of existing objects that have not been identified (false positives) and  $N_2$  is the number of nonexistent objects that have been identified (false negatives). For example Figure 5.12 has  $S = 15$ ,  $N_1 = 5$ , and  $N_2 = 2$ , which gives an SNR of 6.62. The SNR values were then calculated (Table 5.1 and Table 5.2) for various sequences in the first test as well as for all the stereo pairs in the second test.

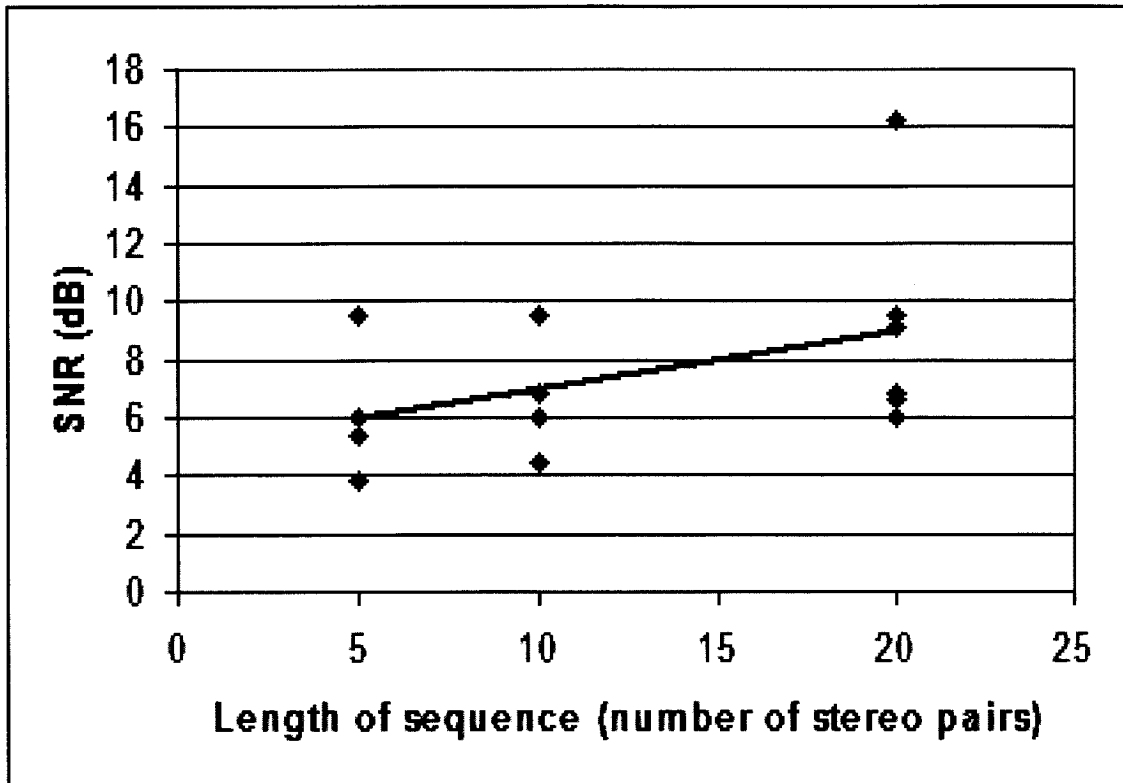
Length of Sequence	Average SNR over 5 sequences (dB)
5	6.16
10	6.74
20	9.06

**TABLE 5.1 SNR for Sparse Edge Reconstruction Algorithm**

Edge Detector	Average SNR over 20 trials (dB)
Prewitt	4.06
Sobel	1.87
Roberts	4.27

**TABLE 5.2 SNR for Reconstruction with a Single Edge Detector**

It is obvious that the SNR that results from using the SER algorithm on a sequence of images is higher than if a single edge detector is used on a pair of images. Figure 5.15 shows the positive linear relationship between SNR and the length of the sequence used in the SER algorithm.



**FIGURE 5.15 Behaviour of SNR as the Image Sequence Increases**

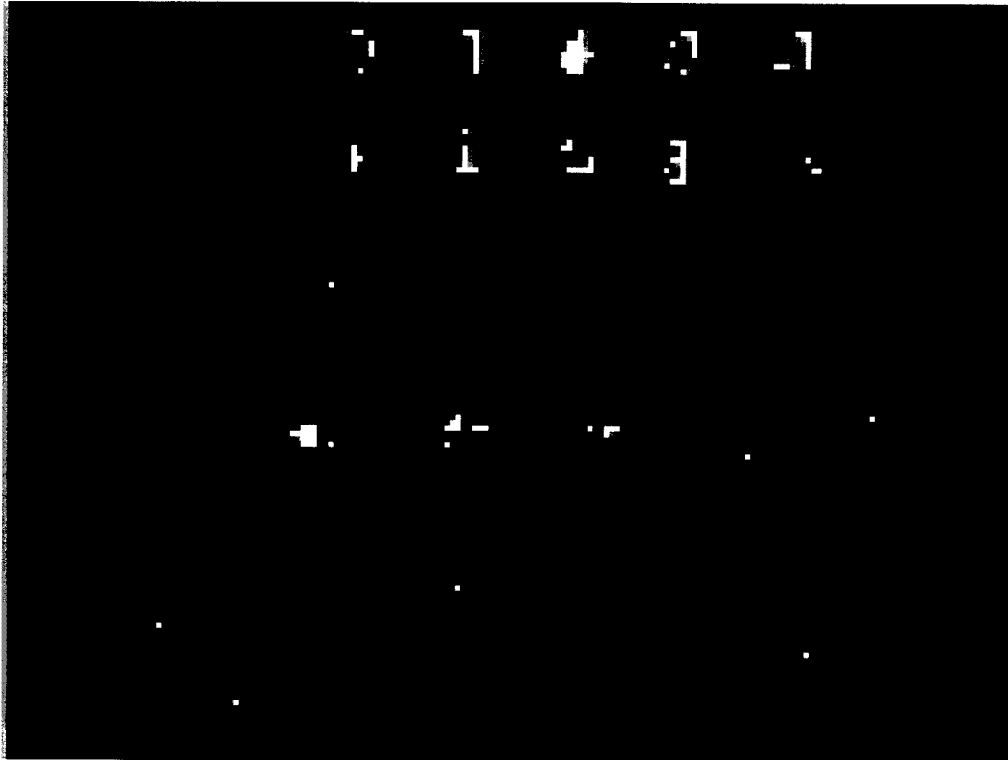
This is not surprising because the accuracy of the reconstruction is increased with each new pair of images in the sequence. However, in a realistic UAV scenario, the objects and the vehicle would be moving and the value of the SNR would not increase indefinitely with sequence length. Table 5.3 shows the computation time for the SER algorithm (without the optional step described in section 5.1) as well as for reconstructions using single detailed edge detectors. These computations were performed using a 3x3 pixels correlation window and a maximum disparity of 36 pixels. This includes the time required for the generation of any edge detectors as well as the normalized cross-correlation.

Type of reconstruction	Time required per pair of frames (sec)	Speed increase with sparse edge reconstruction (%)
Sparse Edge Reconstruction	0.659	N/A
Prewitt Edge Detector	1.1093	40.59
Sobel Edge Detector	1.076	38.75
Roberts Edge Detector	0.8938	26.27

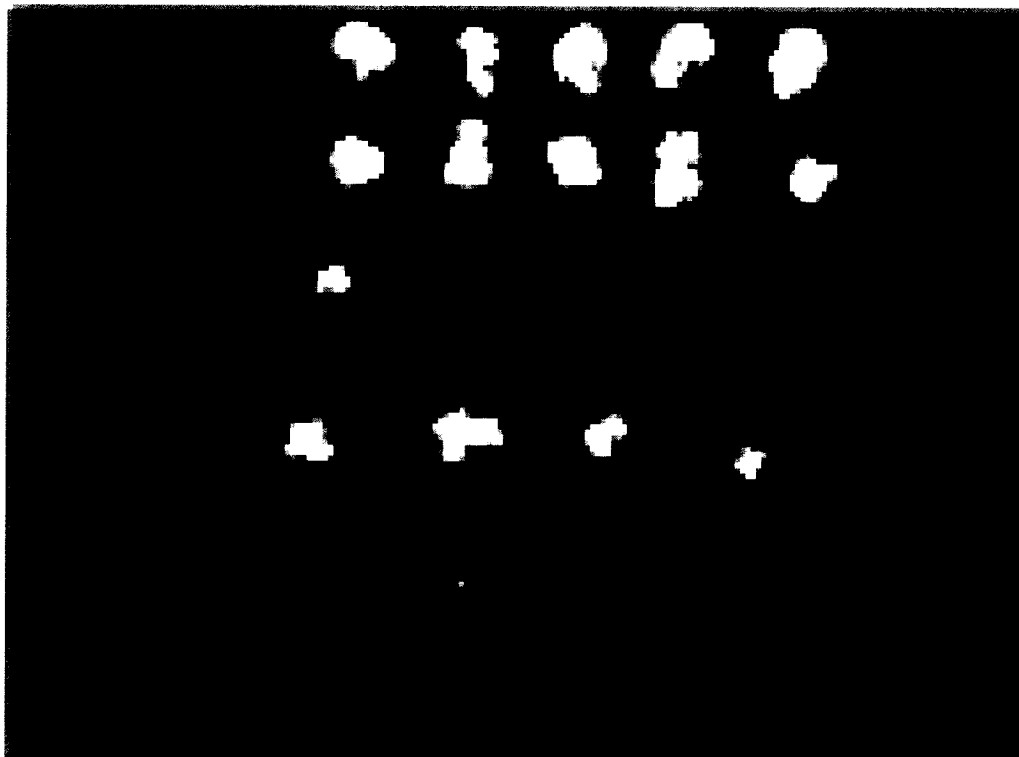
**TABLE 5.3 Computation Time Required for Different Algorithms**

It is important to note that in Table 5.3, the time requirement of the algorithms on *pairs* of images are compared while in Table 5.1, the SNR for a *sequence* of pairs (SER algorithm) is compared with the SNR for single pairs. The reason for this inconsistency is that the SER algorithm creates a reconstruction that is in a constant state of flux. After a short initialization period, the system model becomes robust as it continues to be updated with each incomplete stereo pair. The SNR of this system model (which is derived from a sequence of pairs) is what really matters in the system performance.

Section 5.1 described how an optional step can be added to the SER algorithm. In this step, the disparity map that is generated after the normalized cross-correlation is allowed to 'diffuse' to the nearest edges as determined by a dense edge detector. Figure 5.16 shows a disparity map created from a sequence of images using SER and Figure 5.17 shows the same reconstruction after the optional step is applied.



**FIGURE 5.16** Sparse Edge Reconstruction without the Optional Step



**FIGURE 5.17** Sparse Edge Reconstruction with the Optional Step

In these images, the optional step was generated by using a Canny edge detector as the border of the diffusion. Note that the new disparity map shows a more accurate representation of the environment. The objects in the image that are correct are amplified and the noise elements remain the same. The purpose of this step is to make up for the time spent analyzing the relatively dense image by having the reconstruction generate additional useful information to be saved in the model.

### **5.3 Analysis of Video Captured During Test Flight**

A test flight was carried out (section 3.3) to provide realistic data for evaluation of the SER algorithm. It was assumed that because the UAV is flying relatively close to the ground, the horizon line in the images should appear high up in the field of view or should not be visible at all. Even though the aircraft in this case was flying higher than is to be expected in low-flying UAVs, the horizon line was still high up in the field of view for most of the flight (Figure 5.18).



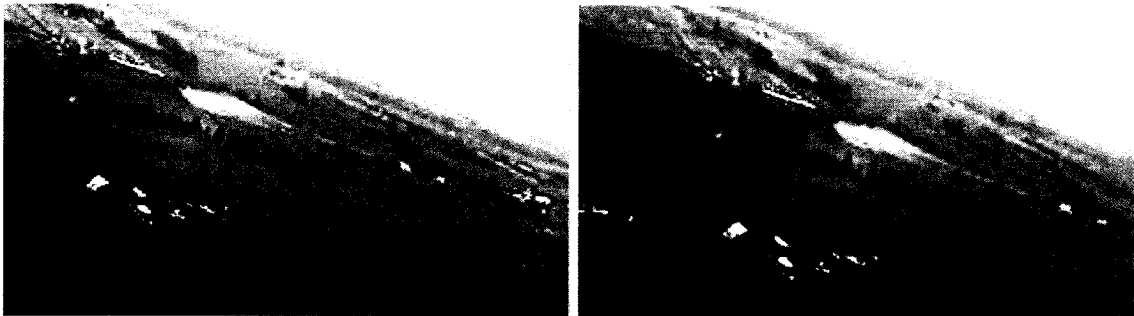
**FIGURE 5.18 Sample Frame from the UAV-captured video**

This result appears to confirm that for a low-flying UAV at level flight, most of the images captured will be the terrain over which the aircraft is flying. This is important because it is more difficult to find an obstacle against the clutter of the terrain than against a sky background (especially a clear sky). It can also be seen that there is a fairly large amount of haze in the area around the horizon line which can make it difficult to see objects in this region of the field of view. Finally, it was noted that at several intervals during the video, the glare from the sun can be misidentified as an object or may 'blind' the system momentarily causing it to miss an obstacle.

It was determined after the test that the right camera was out of focus during the flight. This is problematic for stereoscopic imaging and it is certain that any attempt at reconstruction by stereo correlation will be sub-optimal [E].

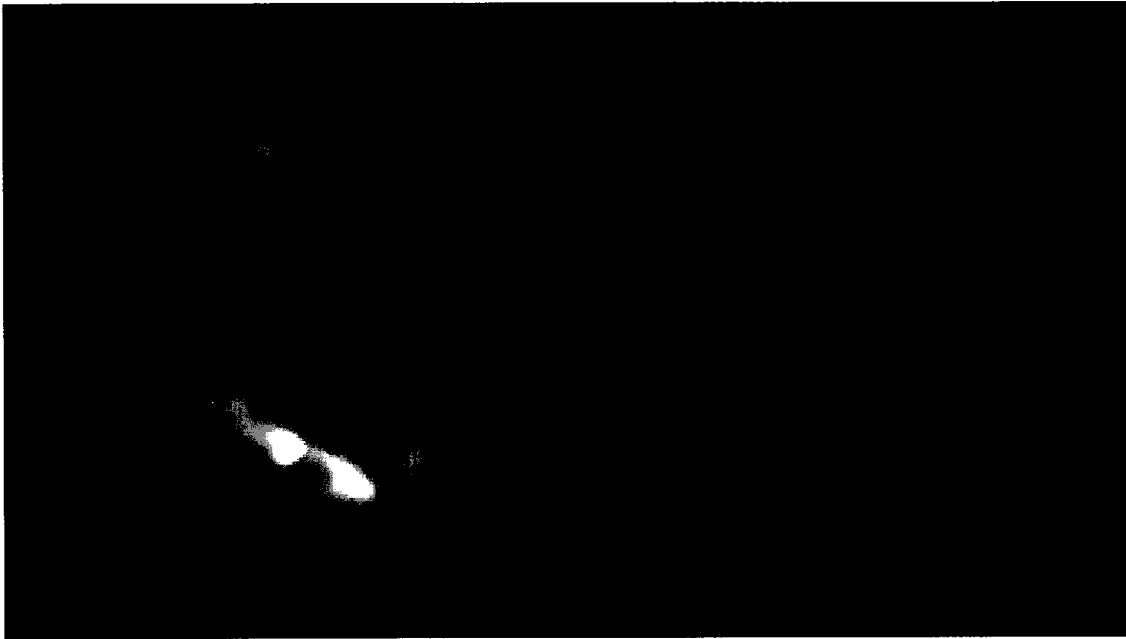
### 5.3.1 Application of Sparse Edge Reconstruction to Video

[Figure 5.20] shows the system model from the SER analysis over a sequence of images [Figure 5.19] from the test flight compared with a reconstruction using a single detailed edge detector [Figure 5.21].



**FIGURE 5.19 Sample Stereo Pair from the UAV-captured video**





**FIGURE 5.20 Sparse Edge Reconstruction of the Stereo Pair in FIGURE 5.19**



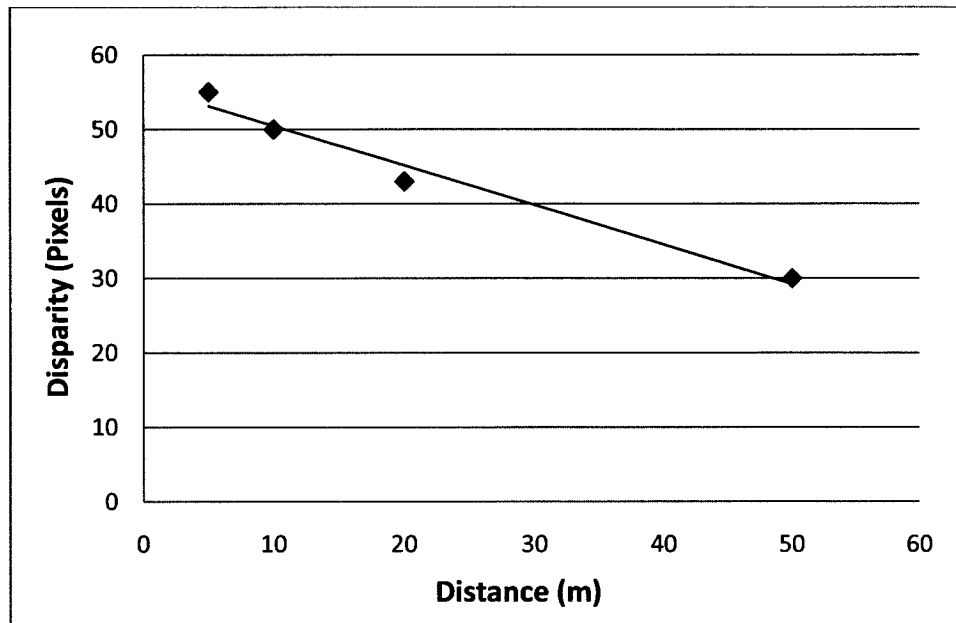
**FIGURE 5.21 Single Edge Detector Reconstruction of the images in FIGURE 5.19**

These are intensity images where the brighter areas represent higher disparity values (i.e. closer distances). It can be seen that the reconstruction generated by

the SER algorithm (Figure 5.20) has more structural nuance and accuracy than any individual stereo pair reconstruction (Figure 5.21) and results in a more comprehensive assessment of the environment that is being observed. The high contrast objects, like the houses in the bottom-left corner of Figure 5.19 for instance, are easily captured in an individual run, but it can take a series of runs to model more subtle, low-contrast objects such as roads or tree clusters that happen to blend with their surroundings. Because the SER algorithm is an adaptive system in continuous flux, data that is incongruent with an actual representation of the environment is effectively weeded from the system and only representations that have meaningful counterparts in the environment remain. A low-flying, terrain-following UAV can benefit greatly from having an obstacle detection system that uses the SER algorithm since the system rapidly updates its conception of the environment but does not do this at the expensive of being incomplete or error-ridden.

### **5.3.2 Application of Distance Prediction to Video**

In chapter 4, a method was described to calculate distance values from disparity measurements. Although a full camera calibration was not carried out prior to the flight test, distance measurements can still be estimated based on the linear relationships that were described in the previous chapter. The distance from the aircraft to various points on the runway were estimated and the horizontal disparity between the left and right camera images was recorded (Figure 5.22)

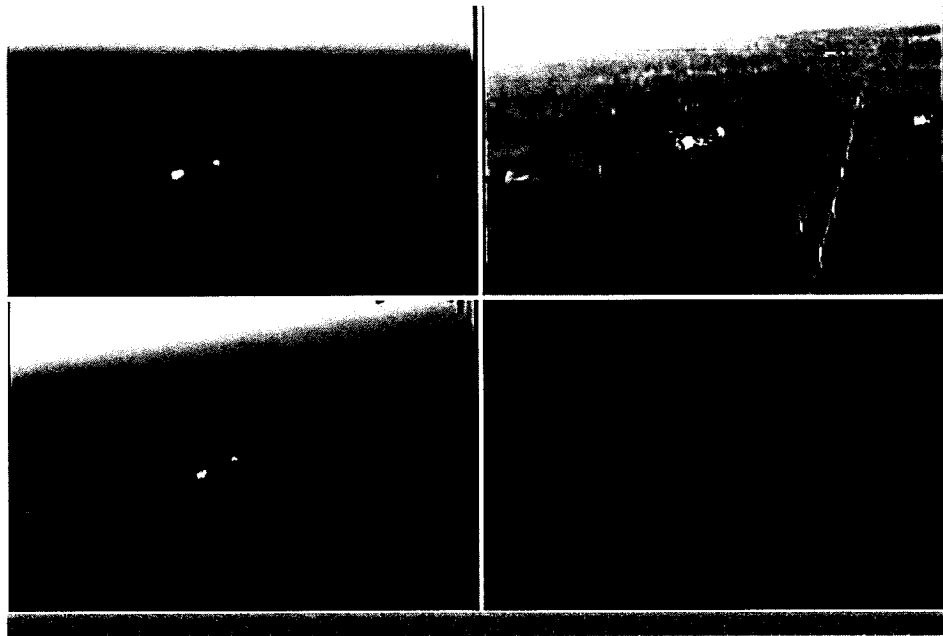


**FIGURE 5.22 Disparity vs. Distance in Test Flight Video**

A linear regression of these test points yields the equation:

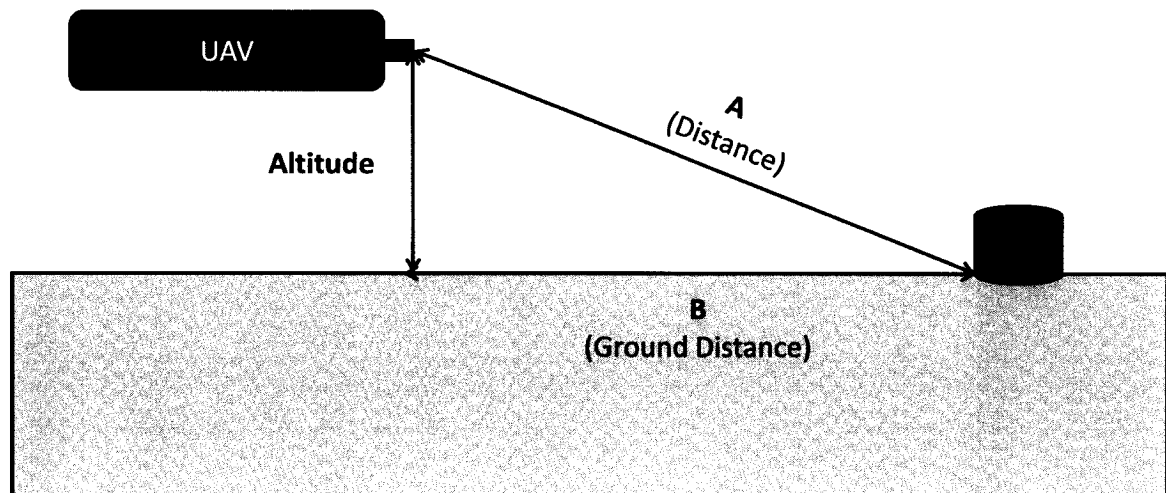
$$D = (-0.53 * R) + 56 \quad [\text{EQN. 5.2}]$$

Where  $D$  is the disparity value and  $R$  is the range (distance) to the object. Using this relationship, some predictions can be made about the rest of the video. The most important sections of the test flight were several 'flight lines' at low altitudes of 61 to 76 m (200 to 250 ft). Figure 5.23 shows a typical image at these altitudes.



**FIGURE 5.23 Low altitude Test Flight Image (Actual Video Frame)**

Several points in the terrain (ex: roads, buildings...) were selected for testing. Although the true distances to these objects are difficult to measure precisely, it can be assumed that they are at least equivalent to the altitude of the aircraft and that they are probably several dozen meters ahead of the it (Figure 5.24). The horizontal disparity was calculated for the points and it was found that the values ranged between 10 and 17 pixels. According to Equation 5.2, this corresponds to distances between 74 and 87 m ('A' in Figure 5.24). These predictions are reasonable given aircraft altitudes between 61 and 76 m. Using simple geometry, the ground distances between the aircraft and these objects (B in Figure 5.23) is between 40 and 60 m.



**FIGURE 5.24 Distance from Aircraft to Obstacles**

## **Chapter 6**

# **Conclusions**

The work in this thesis addresses the problem of obstacle detection for UAVs using optical sensors. The specific requirements and limitations that are dictated by this type of application were discussed and an algorithm targeting these specifications was proposed. A set of images was used to carry out data analysis and this yielded several useful relationships that can be used to calibrate a stereoscopic obstacle detection system.

### **6.1 Summary of Results**

The statistical tests in Chapter 4 yielded several important results. It was confirmed that the horizontal disparity exhibited by an object being observed

stereoscopically is directly proportional to the camera separation. Furthermore, it was found that the dynamic range of the disparity values is very predictable in practice and is also linearly dependent on the camera separation and can be defined by a single slope value with an intercept at the origin. One of the main tasks of the stereoscopic system is to predict distances accurately and it was found that the relationship between the distance of an object and its disparity can also be predicted once a camera separation is selected. It was shown that the values of the slope and the intercept that define the linear relationships between distance and disparity are also linearly dependent on camera separation. The predictability suggested by these results offers a fairly accurate way of calibrating the stereoscopic camera system by taking a large number of image pairs and generating similar curves as the ones produced in chapter 4.

The effect of the horizontal translation of the cameras was also tested and the relationships that were established in the preceding sections were modified to accommodate the anticipated levels of camera motions.

Chapter 5 presented Sparse Edge Reconstruction (SER), an algorithm for carrying out stereoscopic imaging for a low-flying UAV. This algorithm is based on a 'coarse-to-fine' approach that generates a three-dimensional model of the environment over time. By using very sparse edge detection methods, the system can be updated in real-time and yields overall results that are more accurate than using less sparse edge detectors which also require more computation. It was shown that the SER algorithm has a higher SNR value than a stereoscopic reconstruction based on a *Prewitt*, *Sobel* or *Roberts* edge detector. It was also

discussed that the SNR of the SER algorithm increased with image sequence length but that this would not be indefinitely sustainable given a moving vehicle. Finally, it was shown that the SER algorithm is significantly faster to execute than a stereo correspondence based on either of the aforementioned edge detectors.

The SER algorithm and the proposed calibration method were applied to data gathered on a low-altitude test flight carried out by SGL on November 23, 2006.

## **6.2 Recommendations for Future Research**

This research has only addressed a few of the issues that stand in the way of a fully reliable obstacle detection system for low-flying UAVs. Since the methods described fully rely on stereoscopic imaging, any malfunction or obstruction of one of the camera lenses will incapacitate the system. As such, it is important that optical flow analysis methods be used in conjunction with the stereo system not only as a backup but also to enhance the object detection, tracking and avoiding capabilities of the aircraft.

The discussion of camera motion in Chapter 4 was limited to translational motion and did not address the potentially devastating effects of rotation about the horizontal axis or about the axes of the camera themselves. In both of these cases, the epipolar simplifications in Chapter 2 are no longer reliable and unless



the precise motion of the cameras is known a full image correlation search must be carried out. This is especially problematic in the case of real-time systems because this level of correlation can take a significant amount of computation. More research needs to be invested in analyzing the effect of camera rotation on the stereoscopic reconstruction process.

The algorithm described in Chapter 5 has not been implemented on hardware that can be included in a UAV's avionics system. It has yet to be determined what type of processor and implementation is required to make such a system feasible. As well, the descriptions given here describe methods of low-level computer vision and do not describe what is to be done with an object once it has been spotted or how the aircraft is to avoid such an obstacle. These issues are currently being researched as part of Carleton University's Geosurv II project. Additionally, possible integration techniques between the algorithm presented in this thesis and other avionics components onboard the aircraft have been presented [44]. This will allow the algorithm to communicate its analysis to a larger obstacle detection system which will integrate the information in an efficient way and useful manner.

## REFERENCES

- [1] A. Discant, A. Rogozan, C. Rusu, and A. Bensrhair, "Sensors for Obstacle Detection A Survey", *International Spring Seminar on Electronics Technology*, pp 100-105, May 2007
- [2] H. Yamaguchi, Y. Tatehira, K. Akiyama, and Y. Kobayashi, "Stereoscopic Images Disparity for Predictive Coding" In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1976–1979, 1989
- [3] V. Simhadri and P. Chandramani, and Y. Ozturk, "RASCor: Realtime Associative Stereo Correspondence" in *Proc. IEEE International Conference on Image Processing*, Vol. 6, pp. 197-200, 2007
- [4] Y. Kim, S. Park, C. Chen, and H. Jeong, "Real-time Architecture of Stereo Vision for Robot Eye" in *Proc. IEEE International Conference on Signal Processing*, 2006
- [5] I. Giosan, S. Nedevschi, and Silviu Bota, "Real Time Stereo Vision Based Pedestrian Detection Using Full Body Contours" in *Proc. International Conference on Communication and Processing*, pp. 79 – 86, Aug. 2009
- [6] I. A. K. Saeed and N. V. Afzulpurkar, "Real Time, Dynamic Target Tracking using Image Motion", in *IEEE International Conference on Mechatronics*, pp. 241-246, Jul. 2005
- [7] L. Ben, J. Eric, and V. George, "Augmenting UAV Autonomy: Vision-Based Navigation and Target Tracking for Unmanned Aerial Vehicles", in *IEEE Robotics Automation Magazine*, Vol 13, No.3, pp. 63-71, Sep. 2006
- [8] B. Bon and H. Seraji, "Real-Time Model-Based Obstacle Detection for the NASA Ranger Telerobot", Nasa Jet Propulsion Lab document (JPL TRS 1992+), Apr. 1996

- [9] J. Jakym, E. Yost, "GeoSurv II: System Requirements Document; Revision D", Carleton University and Sander Geophysics, Feb. 2006
- [10] E. Hanna, P. Straznicky, R. Goubran, "Obstacle Detection for Low Flying Unmanned Aerial Vehicles Using Stereoscopic Imaging", in *Proc. IEEE Instrumentation and Measurement Technology Conference*, pp.113-118, May 2008
- [11] T. G. McGee, R. Sengupta, and K. Hedrick, "Obstacle Detection for Small Autonomous Aircraft Using Sky Segmentation", in *Proc. IEEE Conference on Robotics and Automation*, pp. 4679 – 4684, Apr. 2005
- [12] S. Hrabar, G.S. Sukhatme, P. Corke, K. Usher, and J. Roberts, "Combined Optic-Flow and Stereo-Based Navigation of Urban canyons for a UAV", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005
- [13] G.L. Barrows, J.S. Chahl, and M.V. Srinivasan, "Biomimetic visual sensing and flight control", in *Proc. International Conference on Unmanned Air Vehicle Systems*, pp.1-15, Apr. 2002
- [14] R. Prazenica, A. Kurdila, R. Sharpley, and J. Evers , "Vision-Based Geometry Estimation and Receding Horizon Path Planning for UAVs Operating in Urban Environments", in *Proc. American Control Conference*, pp.2874-2879, June 2006
- [15] A. E. Ortiz and N. Neogi , "Color Optic Flow: A Computer Vision Approach for Obstacle Detection on UAVs", in *Proc. Digital Avionics Systems Conference*, 4C3-1 – 4C312 ,Oct. 2006
- [16] M. Perrollaz, R. Labayrade, C. Royere , N. Hautiere, and D. Aubert, "Long Range Obstacle Detection Using Laser Scanner and Stereovision" in *Proc. IEEE Intelligent Vehicles Symposium*, pp. 182-187, Jun. 2006
- [17] E. Pastor, J. Lopez, and P. Royo, "UAV payload and mission control hardware/software architecture", in *IEEE Aerospace and Electronic Systems Magazine*, vol. 22, pp. 3, Jun. 2007.

- [18] S. T. Barnard and M. A. Fischler, "Computational Stereo", in *Computing Surveys*, vol 14, No. 4, pp. 553 - 570, Dec. 1982
- [19] C. J. Taylor, J. Malik, and J. Weber , "A Real-Time Approach to Stereopsis and Lane-Finding" in *Proc. IEEE Intelligent Vehicles Symposium*, 1996.
- [20] Z. He, R. V. Iyer, and P.R. Chandler "Vision-Based UAV Flight Control and Obstacle Avoidance", in *Proc. American Control Conference*, pp.2166-2170, June 2006
- [21] F. Dornaika and R. Chung "Stereo Correspondence from Motion Correspondence", In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, vol 1, pp. 70 – 75, Jun. 1999
- [22] W. Ham and S. Kim, "A New Algorithm for Depth Perception in 3D Screen and Its Implementation" in *Proc. International Symposium on Information Technology Convergence*, pp. 373-377 , Nov. 2007
- [23] W. van der Mark, J.C. van den Heuvel, and F. C. A. Groen, "Stereo based Obstacle Detection with Uncertainty in Rough Terrain" in *IEEE Intelligent Vehicles Symposium*, pp. 1005-1012, Aug. 2007
- [24] Y. Kim, H. Kim, "Dense 3D Map Building for Autonomous Mobile Robots", in *Proc. IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 169 – 174, Jul. 2003
- [25] R. Horaud and T. Skordas, "Stereo Correspondence Through Feature Grouping and Maximal Cliques", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 1168 – 1180, Aug. 1989
- [26] H. Yamanoue "The Differences between Toed-in Camera Configurations and Parallel Camera Configurations in Shooting Stereoscopic Images" in *Proc. IEEE International Conference on Multimedia and Expo 2006*, pp. 1701-1704, Jul. 2006
- [27] Y. Wang, D. Li, and C Liu , "A Stereoscopic Imaging Model and Its Calibration of Micro Stereovision for 3D Measurement" in *Proc. IEEE International Conference on Information Acquisition*, pp. 6, May 2006

- [28] N. B. Touzene and S. Larabi, "Obstacle Detection from Uncalibrated Cameras" in *Proc. of Panhellenic Conference on Informatics*, pp. 152-156, 2008
- [29] P. Moallem, K. Faez and J. Haddadnia "Reduction of the Search Space Region in the Edge based Stereo Correspondence", in *Proc. IEEE ICIP*, pp. 149-152, 2001
- [30] R. P. Wilson, "Direct Recovery of Three-Dimensional Scene Geometry from Binocular Stereo Disparity", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 761 – 774, Aug. 1991
- [31] J.Kaszubiak, M. Toornow, R. W. Kuhn, and B. Michaelis "Real-time, 3-D-Multi Object Position Estimation and Tracking", in *Proc. IEEE ICPR*, vol. 1, pp. 785 – 788, 2004
- [32] S. B. Golberg, M. W. Maimone, and L. Matthies "Stereo Vision Rover Navigation Software for Planetary Exploration", in *Proc. IEEEAC*, pp. 2025 – 2036 ,2002
- [33] S. I. Olsen, "Correspondence", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 309 – 315, Mar. 1990
- [34] W. S. Kim, A. I. Ansar, R.D. Steele, and R.C. Steinke, "Performance Analysis and Validation of a Stereo Vision System", in *Proc. IEEE Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1409 -1416, 2005
- [35] D. I. Barnea, H. F. Silverman, "A class of algorithms for fast digital image registration", *IEEE Transactions on Computers* C-21.2, pp. 179-186, Feb. 1972
- [36] G. Dubelman, W. Van der Mark, J. C. Van den Heuvel, and F.C.A. Groen, "Obstacle detection during day and night conditions using stereo vision", in *Proc. IEEE/RSJ*, pp.109-116, Nov. 2007
- [37] E. Krotkov, K. Henriksen, and R. Kories, "Stereo Ranging with Verging Cameras", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 1200 – 1205, Dec. 1990
- [38] J.Kaszubiak, M. Toornow, R. W. Kuhn, B. Michaelis, and C. Knoepfel "Real-Time Vehicle and Lane Detection with Embedded Hardware", *Proc. of the IEEE Intelligent Vehicles Symposium*, pp. 619 – 624, 2005

- [39] W. A. IJsselsteijn, H. de Ridder, J. Vliegen, "Subjective Evaluation of Stereoscopic Images: Effects of Camera Parameters and Display Duration" in *Transactions on Circuits and Systems for Video Technology*, vol. 10.Nr.2, pp.225-233, Mar. 2000
- [40] W. Kang and S. Lee , "Horizontal Parallax Distortion Correction Method in Toed-In Camera with Wide-Angle Lens" in *Proc. 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pp. 1-4, 2009
- [41] D. Scharstein and R. Szeliski "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms", in *International Journal Computer Vision*, vol. 47, no. 1, pp. 7-42, 2002.
- [42] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, R. Schmidt, T. Graf, "High Accuracy Stereo Vision System for Far Distance Obstacle Detection", in *Proc. IEEE Intelligent Vehicles Symposium*, pp. 292-297, Jun. 2004
- [43] S. E. Umbaugh, *Computer Imaging: Digital Image Analysis and Processing*, Boca Raton, Florida, CRC Press, 2005.
- [44] S. Basset, "A Component-Based Three-Layer Autonomy Architecture for Unmanned Aerial Vehicles", Ottawa, ON, Carleton University Sep. 2008