Navigation in GPS Challenged Environments

Based Upon Ranging Imagery

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy

in the Graduate School of The Ohio State University

By

J.N. (Nikki) Markiel, A.S., B.S., MBA, M.S.

Graduate Program in Geodetic Science

*****

The Ohio State University

2012

Dissertation Committee:

Dr. Dorota Grejner-Brzezinska, Advisor

Dr. Alper Yılmaz

Dr. Ralph von-Frese

Dr. Charles Toth
ABSTRACT

The ability of living creatures to navigate their environment is one of the great mysteries of life. Humans, even from an early age, can acquire data about their surroundings, determine whether objects are movable or fixed, and identify open space, separate static and non-static objects, and move towards another location with minimal effort, in infinitesimal time spans. Over extended time periods humans can recall the location of objects and duplicate navigation tasks based purely on relative positioning of landmarks. Our ability to emulate this complex process in autonomous vehicles remains incomplete, despite significant research efforts over the past half century.

Autonomous vehicles rely on a variety of electronic sensors to acquire data about their environment; the challenge is to transform that data into information supporting the objective of navigation. Historically, much of the sensor data was limited to the two dimensional (2D) instance; recent technological developments such as Laser Ranging and 3D Sonar are extending data collection to full three dimensional (3D) acquisition.

The objective of this dissertation is the development of an algorithm to support the transformation of 3D ranging data into a navigation solution within unknown environments, and in the presence of dynamically moving objects. The algorithm reflects
one of the very first attempts to leverage the 3D ranging technology for the purpose of autonomous navigation, and provides a system which enables the ability to complete the following objectives:

- Separation of static and non-static elements in the environment
- Navigation based upon the range measurements of static elements

This research extends the body of knowledge in three primary topics.

1) The first is the development of a general method to identify \( n \) features in an initial data set from \( m \) features in a subsequent data set, given that both data sets are acquired via 3D ranging sensors. Accomplishing this objective, particularly with respect to 2D datasets, has long been a difficult proposition when attempting to link overlapping data sets.

2) Secondly, an innovative methodology to segment a set of discrete 3D range measurements is presented.

3) Finally, the research develops a methodology to support navigation in environments previously infeasible for autonomous vehicles due to lack of position updates. This problem is well known in the navigation field; while Global Positioning Systems (GPS) provide excellent positional information, their signals can become unavailable in a wide variety of conditions, such as indoor or underground localities, dense urban settings, or jammed signal environments.
Current research in robotic manipulation rarely addresses the concept of operations within an unknown environment, and virtually never attempts navigation in the presence of non-static objects. The ability to extend the navigation solution beyond these limitations extends the possibilities for autonomous navigation and advances the field of navigation. The current algorithm cannot provide a navigation solution for an indefinite time period; it can extend the feasible extent of navigation without benefit of GPS positioning.

While this research could not possibly claim to solve the problem of autonomous navigation, it represents an important step towards the vision of developing a machine to emulate cognitive navigation. As we see farther only by standing on the shoulders of giants, it is hoped that this research will someday enable another researcher to see the achievement of true autonomous navigation.
Dedication

This is my fifth academic degree.

In the past I have dedicated research papers to my partner, my children and my parents.

I acknowledge them in the following section of this dissertation.

But this one I did for me...
ACKNOWLEDGMENTS

Foremost, I must acknowledge Dr. Dorota Grejner-Brzezinska, who was willing to take a chance when few others dared, who gave independence and expressed confidence when others wanted micromanagement, and served as an inspirational leader by example.

I would like to acknowledge Dr. Charles Toth of The Ohio State University, whose support in the collection and processing of data was invaluable, and Dr. Ralph von-Frese, whose lectures in gravity and magnetism sparked a career.

I would like to acknowledge Dr. Jacob Campbell and Dr. Michael Miller of the Air Force Research Laboratories, Wright Patterson Air Force Base, whose vision provided the genesis for the research.

I would like to acknowledge Dr. Rolf Hansen of Omnitech AS, Norway, who kindly scanned his landmark dissertation on acoustic imagery that I could ferret out the underlying principles of the technology, and William Woodward of UrsaNav, Inc. for data, contacts, and support relative to 3D sonar imaging.
I would like to acknowledge my parents, whose support and love was always unconditional, and my children, whose accomplishments as adults give me pride.

Finally, I would like to acknowledge my partner and dearest friend, Victoria Pennington, without whom this effort would not have been achieved.
VITA

December 30, 1964......................Born – Moscow, Idaho

1984.....................................A.S. Industrial Engineering,
                          Columbus State Community College

                          Superior Die, Tool & Machine Company

1989 – 1991.........................Supplier Quality Engineer
                          BMY/WVD

1991 – 1994.........................Quality Assurance Manager
                          GPAX International

1994 – 1995.........................Quality Assurance Manager
                          Vanner-Weldon Incorporated

1995 – 2003.........................Division Manager, Engineering & QA/ISO
                          YSK Corporation

1996.....................................B.S. Mechanical Engineering
                          Franklin University, Summa cum Laude

2000.....................................Master’s in Business Administration
Franklin University, Summa cum Laude

2003 – 2005 ……………………Senior Manager of Engineering
American Showa Corporation

2005 – 2009 ……………………Graduate Research Assistant
The Ohio State University

2007……………………………………Masters of Science, Geodetic Science
The Ohio State University, Magna cum Laude

2009 – Present …………………… Lead Geophysical Scientist
U.S. Department of Defense

PUBLICATIONS & PRESENTATIONS


Markiel, J.N. and M. Earwood. “Platform Trajectory Solutions in DGPS Challenged Environments”, ASPRS, PECORA 18, November 14~17, 2011, Herndon, VA


Woodward, W. “Going Deep: Blending GNSS with 3D Sonar Imaging for Underwater Applications”, Cover Article, Inside GNSS, June 2010 (Contributing author)


FIELDS OF STUDY

Major Field: Geodetic Science
# TABLE OF CONTENTS

Abstract.......................................................................................................................... iii

Dedication......................................................................................................................... vi

Acknowledgments............................................................................................................ vii

Vita.................................................................................................................................... ix

List of Tables.................................................................................................................... xvii

List of Figures.................................................................................................................. xix

List of Abbreviations..................................................................................................... xxii

Chapters:

1  Introduction.................................................................................................................. 1

2  3D Data and Algorithm Conception........................................................................... 7

   2.1  3D Ranging Data.................................................................................................... 7

   2.2  Relation to Image Processing................................................................................. 9

   2.3  Previous Work....................................................................................................... 13

      2.3.1.  General Algorithms for Laser Ranging (1D/2D)................................. 14

      2.3.2.  Three Dimensional Ranging Algorithms............................................. 21
2.4 Nature of the Problem........................................................................................................25
2.5 Conceptualizing a Solution................................................................................................28
2.6 Conceptualizing Feature Matching.....................................................................................32
  2.6.1. Conceptual 3D Feature Extraction........................................................................37
  2.6.2. Eigenvectors........................................................................................................38

3 Algorithm..........................................................................................................................41
  3.1 Overview of the Algorithm for Navigation from 3D Ranging Imagery
    (NAVRI).......................................................................................................................39
  3.2 Hypothesis Testing of Two Distributions.......................................................................45
  3.3 Image Segmentation......................................................................................................52
    3.3.1. Statistical Theory Supporting Eigenvector Segmentation...............................56
  3.4 Feature Extraction.........................................................................................................68
  3.5 Coarse Matching of Two Coordinate Frames...............................................................68
  3.6 Feature Matching..........................................................................................................71
  3.7 Determination of Position............................................................................................73

4 Overview of Supporting Technologies..............................................................................75
  4.1 Flash Laser Ranging.......................................................................................................75
  4.2 3D Kinect....................................................................................................................78
  4.3 3D Sound Raising........................................................................................................79
  4.4 Inertial Navigation System............................................................................................88

5 Experimental Data Acquisition.........................................................................................91
  5.1 3D Flash LADAR........................................................................................................91
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5.2</td>
<td>3D Sonar</td>
<td>99</td>
</tr>
<tr>
<td>5</td>
<td>5.3</td>
<td>3D Kinetic</td>
<td>110</td>
</tr>
<tr>
<td>6</td>
<td>6.1</td>
<td>Application of the Algorithm to the Experimental Data and Results</td>
<td>117</td>
</tr>
<tr>
<td>6</td>
<td>6.1</td>
<td>Analysis of the NAVRI with 3D Flash LADAR Data</td>
<td>125</td>
</tr>
<tr>
<td>6</td>
<td>6.2</td>
<td>Demonstration of Trajectory with 3D Sonar Data</td>
<td>137</td>
</tr>
<tr>
<td>6</td>
<td>6.3</td>
<td>Demonstration of Trajectory with 3D Kinetic Data</td>
<td>150</td>
</tr>
<tr>
<td>7</td>
<td>7.1</td>
<td>Conclusions and Observations</td>
<td>156</td>
</tr>
<tr>
<td>7</td>
<td>7.1</td>
<td>The NAVRI Algorithm</td>
<td>156</td>
</tr>
<tr>
<td>7</td>
<td>7.2</td>
<td>The Technologies</td>
<td>159</td>
</tr>
<tr>
<td>7</td>
<td>7.3</td>
<td>Future Research</td>
<td>161</td>
</tr>
<tr>
<td>7</td>
<td>7.4</td>
<td>Finis est Principium</td>
<td>165</td>
</tr>
<tr>
<td></td>
<td></td>
<td>List of References</td>
<td>169</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Appendices</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Absolute Orientation – Horn’s Closed Form Solution</td>
<td>184</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Seven Primary Types of Data Characterizations for Image Data</td>
<td>12</td>
</tr>
<tr>
<td>2.2 Matching Results of Common Algorithms</td>
<td>19</td>
</tr>
<tr>
<td>2.3 Matching Results of Line Extraction Algorithms</td>
<td>20</td>
</tr>
<tr>
<td>2.4 Primitive Entities by Dimension</td>
<td>36</td>
</tr>
<tr>
<td>4.1 Echoscope Performance Specifications</td>
<td>87</td>
</tr>
<tr>
<td>4.2 Echoscope Physical Specifications</td>
<td>87</td>
</tr>
<tr>
<td>4.3 Echoscope Interface Specifications</td>
<td>87</td>
</tr>
<tr>
<td>5.1 3D Flash LADAR Test - Equipment Specifications</td>
<td>92</td>
</tr>
<tr>
<td>5.2 Honeywell HG1700 Specifications</td>
<td>94</td>
</tr>
<tr>
<td>5.3 Summary of Experimental Runs</td>
<td>98</td>
</tr>
<tr>
<td>5.4 Octopus F180R Specifications</td>
<td>101</td>
</tr>
<tr>
<td>5.5 3D Kinect Test - Equipment Specifications</td>
<td>111</td>
</tr>
<tr>
<td>6.1 Magnitude of Error Ellipses Provided by INS, LIDAR Run #10</td>
<td>122</td>
</tr>
</tbody>
</table>
6.2 Positional results for INS during 3D Flash LADAR Run #10 (reference Table 5.3), positions derived from HG1700 IMU data only………………………… 132
6.3 Positional results for INS during 3D Flash LADAR Run #10 (reference Table 5.3), positions derived from NAVRI updates………………………… 132
6.4 Approximate Positional Results for OSU Supercomputing Trajectory…… 151
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>10</td>
</tr>
<tr>
<td>3.1</td>
<td>45</td>
</tr>
<tr>
<td>3.2</td>
<td>47</td>
</tr>
<tr>
<td>3.3</td>
<td>58</td>
</tr>
<tr>
<td>4.1</td>
<td>82</td>
</tr>
<tr>
<td>4.2</td>
<td>83</td>
</tr>
<tr>
<td>4.3</td>
<td>84</td>
</tr>
<tr>
<td>4.4</td>
<td>86</td>
</tr>
<tr>
<td>5.1</td>
<td>93</td>
</tr>
<tr>
<td>5.2</td>
<td>93</td>
</tr>
<tr>
<td>5.3</td>
<td>95</td>
</tr>
<tr>
<td>5.4</td>
<td>96</td>
</tr>
<tr>
<td>5.5</td>
<td>99</td>
</tr>
<tr>
<td>5.6</td>
<td>101</td>
</tr>
</tbody>
</table>
6.10 XY Plot of Position - Reference and NAVRI Based Solutions, Panama Canal Traverse..........................140
6.12 XZ Plot of Position - Reference and NAVRI Based Solutions, Panama Canal Traverse..........................141
6.13 Panama Canal Survey Traverse and Results, 3D..........................142
6.14 XY Plot of Position - Reference and NAVRI Based Solutions, Canadian Traverse..........................144
6.15 XZ Plot of Position - Reference and NAVRI Based Solutions, Canadian Traverse..........................145
6.16 Canadian Canal Survey Traverse and Results, 3D..........................146
6.17 3D Sonar, ΔX (Actual Trajectory minus NAVRI Solution)....................147
6.18 3D Sonar, ΔY (Actual Trajectory minus NAVRI Solution)....................148
6.19 3D Sonar, ΔZ (Actual Trajectory minus NAVRI Solution)....................149
6.20 “Box” Trajectory, Indoor Facility..................................................153
6.21 3D Kinect, NAVRI positional uncertainty by frame, Alternate indoor facility data acquisition..................................................153
**LIST OF ABBREVIATIONS**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1D</td>
<td>One Dimensional</td>
</tr>
<tr>
<td>2D</td>
<td>Two Dimensional</td>
</tr>
<tr>
<td>3D</td>
<td>Three Dimensional</td>
</tr>
<tr>
<td>ALS</td>
<td>Airborne Laser Scanner</td>
</tr>
<tr>
<td>CA</td>
<td>Coarse Acquisition</td>
</tr>
<tr>
<td>CAT</td>
<td>Computerized Axial Tomography</td>
</tr>
<tr>
<td>CCD</td>
<td>Charge Coupled Device</td>
</tr>
<tr>
<td>CFM</td>
<td>Center for Mapping</td>
</tr>
<tr>
<td>CPD</td>
<td>Cumulative Probability Distribution</td>
</tr>
<tr>
<td>DATMO</td>
<td>Detection and Tracking of Moving Objects</td>
</tr>
<tr>
<td>DEM</td>
<td>Digital Elevation Model</td>
</tr>
<tr>
<td>ECEF</td>
<td>Earth Centered, Earth Fixed</td>
</tr>
<tr>
<td>EKF</td>
<td>Extended Kalman Filter</td>
</tr>
<tr>
<td>FOG</td>
<td>Fiber Optic Gyroscope</td>
</tr>
<tr>
<td>GB</td>
<td>Giga-Byte</td>
</tr>
<tr>
<td>GLONASS</td>
<td>GLObal'naya NAvigatsionnaya Sputnikovaya Sistema</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
</tbody>
</table>
IMU  Inertial Measurement Unit
INS  Inertial Navigation System
kB   Kilo-Byte
KF   Kalman Filter
LADAR Laser Distance and Ranging
LIDAR Light Detection and Ranging
LLE  Local Linear Embedding
MEMS Micro-Electro-Mechanical Systems
MRI  Magnetic Resonance Imagery
NAVRI Navigation from Ranging Imagery
NED  North-East-Down
NND  Nearest Neighbor Distance
OLS  Ordinary Least Squares
OR   Orthogonal Regression
OSU  The Ohio State University
PCA  Principal Component Analysis
PDF  Probability Density Function
PL   Pseudolite
ppm  Parts per Million
RGB  Red-Green-Blue
RT   Real Time
SFEM Segmentation, Feature Extraction, and Matching
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT</td>
<td>Scale Invariant Feature Transform</td>
</tr>
<tr>
<td>SLAM</td>
<td>Simultaneous Location and Mapping</td>
</tr>
<tr>
<td>SONAR</td>
<td>Sound Navigation and Ranging</td>
</tr>
<tr>
<td>SPIN</td>
<td>Satellite Positioning and Inertial Navigation</td>
</tr>
<tr>
<td>TLS</td>
<td>Terrestrial Laser Scanner</td>
</tr>
<tr>
<td>TOF</td>
<td>Time of Flight</td>
</tr>
<tr>
<td>UIS™</td>
<td>Underwater Inspection System (patented trademark of Coda Octopus)</td>
</tr>
<tr>
<td>XTF</td>
<td>eXtended Triton Format</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

Indoor, underground, urban, and underwater environments present unique challenges for navigation. The integration of Global Positioning Systems (GPS) with inertial measurement units (IMUs) becomes inefficacious once a person or autonomous mobile unit moves into an environment which restricts or blocks access to the GPS signal [Lachapelle, 2004], [Dedes and Dempster, 2005], [Miller et al., 2008]. Humans possess the ability to utilize landmarks and other visual references to enable coarse spatial referencing which permits navigation in such conditions, provided that sufficient distinguishable features are available along the traversed route [Werner et al., 1997]. In contrast, mobile autonomous units are limited in their ability to generate cognitive maps and require different methods for construction of spatial referencing [Bergholm, et al., 1997], [Dixon and Henlich, 1997], [Madhavan et al., 1999], [Jacobsen, 2000], [Dupuis et al., 2005].

A primary concern is the ability to maintain and track spatial position while the navigation or mapping process occurs [Lumelsky, and Skewis, 1990], [Jansfelt, 2001], [DeSouza and Kak, 2002]. While inertial navigation systems (INS) provide excellent
short term solutions for determining attitude and position, the limitations of these units in terms of drift and uncertainty error are well understood [Weston & Titterton, 2000]. The traditional GPS/INS tightly coupled system cannot be relied upon to provide an accurate position over extended temporal intervals in conditions where the GPS signal is absent and the inertial system is free to accumulate error without correction [Jekeli, 2000].

Laser ranging techniques have traditionally been utilized from airborne or satellite platforms for the purpose of mapping. Recent technical developments in Flash LADAR and Terrestrial Laser Scanning (TLS) devices are extending the reach of remote sensing to ground based applications [Surmann et al., 2001], [Brenneke et al., 2003], [Campbell et al., 2003], [Stettner et al., 2004], [Campbell et al., 2006], [Gajdamowicz et al., 2007]. These newer laser technologies are capable of acquiring several thousand ranging points per second with millimeter accuracy [Weingarten et al., 2004], [Anderson and Kelly, 2005], [Glennie, et al., 2006], [Halterman and Bruch, 2010]. Inclusion of these land based laser devices on a mobile unit enables the acquisition and reconstruction of the sensor environment in 3D with high accuracy [Guivant et al., 2000], [Surman et al., 2003], [Cole and Newman, 2006], [Newman et al., 2006]. Similar techniques in 3D sonar ranging are permitting similar capability for sensory acquisition in underwater environments [Kao and Probert, 2000], [Hansen et al., 2005], [Costa and Battista, 2008].

Flash LADAR cameras can be utilized to acquire 3D data based upon the return reflection of an emitted light pulse. Details related to the Flash LADAR technology can
be found in [Lange, 2000], [Oggier et al., 2003]; in simplest terms the device measures the time of flight for a modulated laser emission and recovers a distance measurement with respect to the speed of light. Kinect operates along similar principles, details can be found in [Newcombe, et al., 2011] and [Shotton, et al., 2011]. To ensure clarity, the term “Flash LADAR” will henceforth refer to the technology produced by SwissRanger or CSEM, while the term “Kinect” will refer the system produced by Microsoft. 3D Sonar operates in a similar manner, but utilizes the emission of sound rather than light [Kao and Probert, 2000]. Details of these technologies are provided in Chapter Four. The resulting output is a triplet of Cartesian coordinates for each pixel in the image, enabling the reconstruction of the object space in 3D. If the initial starting point of the mobile platform is known, the position of features extracted from the 3D image can be established since the distance values are known relative to the coordinate origin of the acquisition sensor. Additionally, it is possible to recover not only the position in terms of coordinate translations, but the relevant 3D orientation since the rotational changes in each axis can be determined as well. When the platform moves to a new location, the same features can be utilized to triangulate (or trilaterate) the new position, provided that the same features can be identified in the subsequent image [Fletcher et al., 2007], [Veth et al., 2008], [Gray and Veth, 2009]. The objective is to utilize static features for the purposes of location as moving features would introduce errors to the navigation solution.

The challenge is three-fold; first, the recovery of features from a time series of 3D images given no a-priori information about the environment, secondly to determine which of the
extracted features are static, and third, to locate the same static features from temporally separated images in the presence of motion.

While extensive literature exists in the extraction of features from 2D images [Canny, 1986], [Stein and Medioni, 2002], [Karaman et al., 2005] direct extension of these techniques is complicated by the presence of motion in the third axis and necessitates the development of new algorithmic approaches for feature extraction [Hoover et al., 1996], [Mikolajczyk and Schmid, 2005]. The algorithm becomes more complex if information related to the scene is unknown; lacking knowledge of the object space reduces the number of constraints which may be applied to reduce the domain of the image space. In our instance, underlying assumptions are:

- Features exist in the acquired imagery
- Sufficient correspondence exists for static features in two temporally spaced images
- Sufficient static features exist to enable determination of position and orientation

It is important to note that our objective is not to extract entities from the imagery as distinct physical objects; as an example, it is not necessary to distinguish between a telephone and the desk. The fact that both “features” exist and are static is sufficient. Secondly, our objective does not require object recognition; whether the surface is a wall
or a picture is immaterial so long as the feature is static and can be matched between the images.

An important issue is the need for reduction of the problem dimensionality, primarily due to the significant amount of data acquired by laser ranging systems. As an example, the Flash LADAR camera utilized in this study operates at 5 Hz frame rate; each frame is a 700 kilobyte (kB) image of some 25,000 laser points. The camera is capable of frame rates up to 30 Hz; thus acquiring data at nearly 1 gigabyte (GB) per minute. To enable the efficient, (near)-real time processing of the imagery to establish position/orientation necessitates that the image space be quickly reduced to a computationally tractable set of features.

In this dissertation the development of an alternative method for determining position is introduced which is based on using 3D features extracted from 3D ranging imagery. The algorithm presented enables this position to be determined in unknown environments and in the presence of dynamic (moving) objects.
This dissertation is organized as follows.

In Chapter Two, the problem of navigation by 3D ranging data is defined. A brief overview of related research efforts is then provided. Next, a conceptualization of the algorithm is established to provide an approach to achieve the objective of navigation. Finally, an innovative methodology for feature segmentation is outlined. Chapter Three develops the navigation algorithm in detail. In Chapter Four, the technology utilized in the process is discussed to provide an overview of system elements. Chapter Five explains the experimental setup and data acquisition process which will be utilized to test the algorithm performance. Chapter Six discusses the application of the navigation algorithm to the 3D datasets, along with the related results. Finally, Chapter Seven provides some conclusions and observations related to this line of investigation; and indicates as well opportunities for future research related to 3D ranging imagery.
CHAPTER 2
3D DATA AND ALGORITHM CONCEPTION

Our objective is to navigate in a GPS challenged environment based upon information derived from 3D ranging data. In considering this objective, it is first essential to study the nature of 3D ranging data. Existing literature related to the topic of navigation via 3D ranging data is reviewed. The specific problems related to the objective are outlined, followed by a general conception of exploitation. The chapter finishes with a discussion of feature extraction methodology to support the conceptual algorithm.

2.1 3D Ranging Data

3D ranging data are the product of an active sensor utilized to acquire ranging measurements from the sensor to objects in the sensor environment. The 3D sensor emits energy in the form of light or sound; the resulting wave is absorbed, reflected, or refracted by the surfaces of objects in the environment. Waves that are reflected are sensed by the 3D sensor on their return; utilizing basic principles of light (or sound), the time of flight can be determined with substantial accuracy. This measurement provides the distance (range) from the surface of reflection to the sensor; implicit in this
measurement is the assumption that the range is not the product of multiple reflections. Removing such spurious measurements from the acquisition process is accomplished by sophisticated engineering techniques [Lange, 2000], [Murino and Trucco, 2000]. If the metrics of the sensor are known, the angles of return can be identified relative to the sensor plane, in similar manner to the operation of the photogrammetric camera, and with commensurate accuracy relative to the precision of the camera metrics. Evaluating the performance of a 3D sensor is beyond the scope of this dissertation; with particular respect to cameras, the reader is referred to [Light, 1992], [Clarke and Fryer, 1998], [Karras and Mavrommati, 2001] for calibration of metric and non-metric cameras and [Foote et al., 2005] for sonar arrays.

The output of the acquisition process is therefore a range and two defined angles relative to a known direction, enabling the position of the reflected surface relative to the sensor to be established within a defined error. The coordinate frame has the $xy$ plane coincident with the image plane and the $z$ axis orthogonal to that plane and extending through the focal point. The origin of the sensor is that point where the $z$ axis intersects the $xy$ plane. The $x$ axis defines the horizontal horizon, the $y$ axis defines “up” in a local sense, and $z$ axis is forward to the image space. Rotation about the $y$ axis (direction of $z$ axis) defines the azimuth of the camera coordinate frame. Transformation of the spherical form to a Cartesian schema based upon the sensor focal point can be accomplished by equation 2.1.
\[
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix} = r 
\begin{bmatrix}
sin(\alpha)cos(\gamma) \\
sin(\alpha)sin(\gamma) \\
cos(\alpha)
\end{bmatrix}
\] (2.1)

In this equation, \( \alpha \) is the angle from the vertical axis, \( \gamma \) is the azimuth angle, \( r \) is the range measurement, and \( x, y, z \) reflect the local (sensor) Cartesian coordinate frame of reference.

No sensor can capture the environment as a continuous representation; instead, the spacing of range measurements defines a set of discrete measurements. The resolution of the sensor is therefore an important consideration since it defines the level of discretization which will exist in the resulting representation of the object space [Schenk, 1999], [Gao, 2009].

2.2 Relation to Image Processing

The discussion in paragraph 2.1 indicates that the output of a 3D sensor is a point cloud representing a set of discrete points, measured with respect to a local coordinate frame originating from the camera. These measurements reflect the discrete representation of a continuous object space filled with features. Since the measurements are acquired on a near simultaneous basis, each epoch of acquisition generates a set of range measurements which shall be referred to as an image, since the set of range measurements can be easily
displayed visually to emulate a traditional intensity based photograph, as shown in Figure 2.1.

Figure 2.1. Example of a 3D Ranging “Image”

In our case, there are three general steps in image processing: segmentation, feature extraction, and image matching. It should be noted that different sources of literature are not consistent in the ordering of these steps; pattern classification algorithms tend to focus first on the extraction of features, followed by a segmentation step once the features are available [Duda and Hart 2001], while the majority of literature related to navigation,
computer science, and other related fields related to extracting information from sensor acquired data tends to view segmentation as the first phase, followed by feature extraction, and then linking two (or more) images based upon matching features. In some instances, the authors either fail to (or neglect to) distinguish between segmentation and feature extraction; others treat these steps as a “resolved problem” and focus on higher order activities such as object recognition, object tracking, and obstacle avoidance that stem from the basic steps. The lack of consistent terminology and meaning across the literature infers the strong necessity to define these processes in the context of the research at hand.

In the segmentation step, a variety of methods are utilized to partition the image into sets of spatial similarity. This can be accomplished by a wide variety of methods; Gao [2009] outlines seven primary types of data characterizations, as shown in Table 2.1. The outputs of the segmentation step are points, or clusters of points.

Feature extraction is a process of dimensionality reduction, implemented to improve the computational tractability of higher order operations. The feature extraction process attempts to classify segmented data into common structures, such as lines, edges, corners, planes, or surfaces. A key element in distinguishing segmentation from feature extraction is the ability to derive a mathematical expression which defines the segmented data to a specific classification; a linear feature, a spherical blob, etc.
In the third step, extracted features from two (or more) images are compared to enable commonality (matching) between the two data sets. This may be accomplished for a wide variety of objectives, including change detection, co-registration, or feature matching itself. The level of complexity arising from the matching process is therefore highly correlated with the objective; the process of co-registration being perhaps the most computationally difficult owing to the need of external information. This will be explored further in Section 2.4.

<table>
<thead>
<tr>
<th>Photo Element</th>
<th>Description / Math Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tone / Color</td>
<td>Digital Number</td>
</tr>
<tr>
<td>Size</td>
<td>Number of spatially continuous pixels</td>
</tr>
<tr>
<td>Shape</td>
<td>Hard to area / perimeter ratio size, perimeter</td>
</tr>
<tr>
<td>Shadow</td>
<td>Digital Number</td>
</tr>
<tr>
<td>Texture</td>
<td>Standard deviation of digital number, or spatial auto-correlation</td>
</tr>
<tr>
<td>Pattern</td>
<td>Spatial adjacency</td>
</tr>
<tr>
<td>Location / Association</td>
<td>Logical (Boolean) expression</td>
</tr>
</tbody>
</table>

Table 2.1. Seven Primary Types of Data Characterizations for Image Data
(from Gao, 2009)
2.3 Previous Work

The extent of papers related to the processes of segmentation, feature extraction, and image matching (SFEM) is daunting. Papers related to image processing can be found across a wide range of fields, including navigation, robotics, computer science, computer vision, geographic science, various earth sciences (such as geology, biology, hydrology), mining, marine science, avionics, space technologies, civil engineering, mechanical engineering, medical science, medical technologies, neural networks, artificial intelligence, virtual reality, mathematics, statistics, economics, and cinematography - without even providing an exhaustive list. Terminology and methodology vary widely across the various disciplines; environments, objectives, and outcomes are similarly extensive. Parallel efforts in Simultaneous Location and Mapping (SLAM) and Detection and Tracking of Moving Objects (DATMO) cross several fields of research, most notably in the robotics and autonomous vehicle communities [Thrun, et al., 2000], [Montemerlo, et al., 2002], [Brenneke, et al., 2003], [Cole and Newman, 2006], [Wang and Thrope, 2002], [Wang, 2004] and [Wang, et al., 2009]. What can be said is that there is a common need to transform acquired data from some sensory device and convert the data to some level of information suitable for a particular purpose. Dozens of algorithms (if not more) exist to accomplish this basic goal; the applicability of the algorithm can range from having narrow application to wide flexibility depending on the intent, nature, and purpose of a particular approach. Algorithms developed in one field are modified in turn to meet particular challenges in another field of study. What are widely accepted are two basic
conditions; first, no single algorithm provides a “universal” solution to the problem(s) of segmentation, feature extraction, and image matching, and secondly, that implementation without human intervention has not yet been achieved [Costa and Cesar, 2001], [Russ, 2002], [Gonzalez and Woods, 2008].

2.3.1 General Algorithms for Laser / Sonar Ranging (1D/2D)

A surprising number of studies have simply utilized existing feature extraction algorithms developed for 1D and/or 2D applications on the 3D dataset. This may be in part due to the focus on robot navigation or automated mapping rather than feature extraction. Examples of algorithmic approaches for range based navigation include:


- Line/Edge Detection – [Lee and Labdgrebe, 1993a], [Lee and Labdgrebe, 1993b], [Arras and Siegwart, 1997], [Adams and Kerstens, 1998], [Koksal et al., 1998], [Arras and Tomatis, 1999], [Thrun, 2001], [Zhao and Shibasaki, 2001], [Miura et al., 2002], [Kim et al., 2003], [Castro et al., 2004], [Wang, 2004], [Wulf et al., 2004], [Tapus and Siegwart, 2005], [Nguyen et al., 2007], [Gajdamowicz et al., 2007], [Dolgov and Thrun, 2009], [Lee et al., 2010]
Point descriptors represent the characterization of a particular position triplet \((x, y, z)\) by means of additional data. The availability of such data provides content associated with a particular spatial location on an image. The presence of content associated with a particular position enables the opportunity to match points in two images based upon that characteristic [Schenk, 2000].

Edge pixels are characterized by abrupt changes in the intensity of the data, and edges are represented by connected sets of edge pixels [Gonzalez and Woods, 2008]. An extensive variety of algorithms exist to exploit changes in data characterizations (reference Table 1). The segmented edge segments are often created with a Canny edge detector [Canny, 1986] and/or a Hough Transform [Duda and Hart, 1972], [Ballard, 1981] to enable the extraction of lines or boundaries of elements. As will be discussed momentarily, these methods, while very useful and widely implemented, are not foolproof, and require extensive interaction by the operator to develop a meaningful result.

Extensive literature exists related to the identification, extraction and matching of both point and edge descriptors [Jähne, 1997], [Costa and Cesar, 2001], [Russ, 2002], [Biber, 2003], [Nixon and Aguado, 2004], [Gao, 2009] and the methodologies have been studied extensively. An exhaustive description of the many algorithms available would fill several large books; analysis of their strengths and weaknesses several more yet. It is important to note that no single method has been identified as the “best” feature
extraction algorithm for all applications [Hoover et al., 1996], [Gonzalez and Woods, 2008].

One concern is the lack of positioning information resulting from these feature extraction activities; while 3D scene reconstructions are utilized for navigation, the objective has been to move through an environment rather than determine precise positional information within that environment. Based upon a comprehensive search by the author, very few research efforts have been conducted on the segmentation of true 3D range imagery (as opposed to 1D or 2D mosaic scans), or the related subject of positioning the sensor based upon the same. Feature extraction algorithms designed to exploit the 3D data set are similarly lacking; the technology simply has not yet received extensive focus from the academic community. The few available research activities will be discussed in Section 2.3.2.

The majority of 3D data algorithms have been associated with the medical community for the segmentation of imagery from advanced scanners, such as Computerized Axial Tomography (CAT) or Magnetic Resonance Imaging (MRI). In these instances, the imagery has been treated as a 2D image resulting from 3D “slices” and processed for feature extraction without regard for matching or positioning.

Examples of 3D algorithmic approaches include:
• Planes - [Forsberg et al., 1995], [Lang and Pai, 1999], [Liu et al., 2001], [Dorninger and Nothegger, 2007], [Campbell et al., 2003], [Schindler and Biscof, 2003], [Uijt de Haag et al., 2006], [Uijt de Haag et al., 2008]
• Shapes - [Terzopoulos, 1988], [McInerney and Terzopoulos, 1993], [El-Hakim et al., 1997], [Thrun et al., 2000], [Forsman, 2001]
• Surfaces - [Huber, 2001], [Karbacher et al., 2001], [Stulp et al., 2001], [Montemerlo et al., 2002], [Hähnel et al., 2003]

All of these algorithms demonstrate varying levels of success within highly constrained environments, and typically require extensive human intervention to generate acceptable results. Other issues include the use of supervised environments, a-priori maps, or predetermined geometric landmarks (such as trees modeled as cylinders). While each enables the algorithm to successfully execute within certain constraints, none of these can be considered as a “universal” algorithm to support autonomous navigation.

Another concern is the error associated with the utilization of such methods. Hoover et al., [1996] investigated a dozen algorithms published in literature (many of which are still utilized in current applications) and identified that each algorithmic study involved at most eleven images for testing purposes. Extensive testing by Hoover et al. [1996] found
that in comparison to visual verification of the images being processed, all segmentation algorithms tested failed to detect more than 80% of available features. More recently, Mikolajczyk and Schmid [2005] tested ten algorithms as shown in Table 2.2. The “recall” column reflects the ratio of correct matches to the number of available correspondences, while the 1-precision column defines the number of false matches with respect to the total number of matches. The final column reflects the number of actual correct matches. The formulas for these values are reflected in equations 2.2 and 2.3.

$$\text{Recall} = \frac{\text{correct matches}}{\text{available correspondences}}$$  \hspace{1cm} (2.2)

$$1 - \text{precision} = \frac{\text{false matches}}{\text{#correct matches} - \#false matches}$$  \hspace{1cm} (2.3)

Nguyen et al. [2007] tested a number of line extraction algorithms, as shown in Table 2.3. The results clearly indicate significant issues, with no single algorithmic method exceeding 85% in terms of accuracy.
<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Recall</th>
<th>1-Precision</th>
<th>Number of nearest neighbor correct matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOH</td>
<td>0.25</td>
<td>0.52</td>
<td>192</td>
</tr>
<tr>
<td>SIFT</td>
<td>0.24</td>
<td>0.56</td>
<td>177</td>
</tr>
<tr>
<td>Shape Context</td>
<td>0.22</td>
<td>0.59</td>
<td>166</td>
</tr>
<tr>
<td>PCA-SIFT</td>
<td>0.19</td>
<td>0.65</td>
<td>139</td>
</tr>
<tr>
<td>Moments</td>
<td>0.18</td>
<td>0.67</td>
<td>133</td>
</tr>
<tr>
<td>Cross Correlation</td>
<td>0.15</td>
<td>0.72</td>
<td>113</td>
</tr>
<tr>
<td>Steerable Filters</td>
<td>0.12</td>
<td>0.78</td>
<td>90</td>
</tr>
<tr>
<td>Spin Images</td>
<td>0.09</td>
<td>0.84</td>
<td>64</td>
</tr>
<tr>
<td>Differential Invariants</td>
<td>0.07</td>
<td>0.87</td>
<td>54</td>
</tr>
<tr>
<td>Complex Filters</td>
<td>0.06</td>
<td>0.89</td>
<td>44</td>
</tr>
</tbody>
</table>

Table 2.2. Matching Results of Common Algorithms

(from Mikolajczyk and Schmid, 2005)
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>True position (%)</th>
<th>False position (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Split - Merge + Cluster</td>
<td>83.9</td>
<td>7.2</td>
</tr>
<tr>
<td>Incremental</td>
<td>76.0</td>
<td>3.7</td>
</tr>
<tr>
<td>Incremental + Clustering</td>
<td>77.6</td>
<td>4.0</td>
</tr>
<tr>
<td>Line Regression</td>
<td>75.3</td>
<td>9.6</td>
</tr>
<tr>
<td>LR + Clustering</td>
<td>75.6</td>
<td>7.7</td>
</tr>
<tr>
<td>RANSAC</td>
<td>76.0</td>
<td>28.8</td>
</tr>
<tr>
<td>RANSAC + Clustering</td>
<td>70.0</td>
<td>9.2</td>
</tr>
<tr>
<td>Hough Transform</td>
<td>84.1</td>
<td>36.0</td>
</tr>
<tr>
<td>Hough Transform + Clustering</td>
<td>80.6</td>
<td>12.5</td>
</tr>
<tr>
<td>Expectation Maximization</td>
<td>74.4</td>
<td>43.4</td>
</tr>
<tr>
<td>Expectation Maximization + Clustering</td>
<td>77.5</td>
<td>18.6</td>
</tr>
</tbody>
</table>

Table 2.3. Matching Results of Line Extraction Algorithms (from Nguyen et al., 2007)
2.3.2 Three Dimensional Ranging Algorithms

The introduction of 3D Ranging (laser or sonar) is a relatively new technological advance, appearing within the past decade. Extensive historical literature is not apparent even within the scope of a detailed search; past efforts related to laser ranging typically related to a single beam and/or strip of data which required multiple strips of temporally spaced data to achieve object space reconstruction. The bulk of research related to 3D laser ranging has occurred in tandem with the research for this dissertation over the period of the past five years. Within this time frame, several researchers have investigated the utilization of 3D laser ranging sensors where reference maps exists a-priori, typically for mapping purposes, or navigation in a known environment. Since the objective of this study is to navigate in environments where no a-priori knowledge of the environment exists, these research activities are not germane.

Current parallel attempts to navigate based upon 3D laser ranging data are the topic of research by several authors. In brief summary, while a number of research directions have been subject to analysis, no single published algorithmic approach has appeared to resolve, in a universal sense, the problem of autonomous navigation based upon 3D ranging imagery.

Campbell et al., [2003], [Uijt de Haag et al., 2006], [Uijt de Haag et al., 2008], utilized extracted lines or planes to support positioning. Investigations by the author in the early
development of this dissertation showed that small perturbations (on the order of a few millimeters) in the point cloud would rapidly induce errors to the navigation solution and rapidly degrade the solution; Uijt de Haag et al. [2008] reported similar concerns with errors up to 0.4 meters after 9 seconds. As such, navigation from line and/or planar features is demonstrably limited in practical application for extended navigation in GPS challenged environments.

Crosilla and Beinat [2007] proposed a form of Generalized Procrustes Analysis [Gower, 1975] for matching 3D LiDAR point clouds. Their algorithm utilizes a similar principal of utilizing least squares adjustment between two sets of 3D points to establish the transformation between point sets drawn from different coordinate frames. The algorithm first assumes that the number of corresponding points is identical in both reference frames, which is clearly not the case if dynamic features are present. Secondly, the algorithm starts with a set of known static points and then iteratively adds points to the solution space and “tests” the least squares solution to evaluate if the point should be kept as part of the solution set. This approach is effective in the identification of deformable or non-static features, but computationally represents issues if scaled to large point sets.

Hess [2008] utilized point clouds and Gabor Filters [Nixon and Aguado, 2004] to cluster pixels and check for motion based upon Gaussian processes; however, his process was intended to identify the heading and velocity of moving objects in the field of view, not to enable navigation of the sensor platform.
Wang et al. [2009] utilized the intensity images from the laser ranging camera, but not the ranging data itself. Their algorithm utilized speeded up robust features (SURF) to segment and extract features. The results were then fed into a newer SLAM algorithm by Hu et al. [2009] to locate landmarks in an unsupervised scene and enable ongoing position of the sensor. Once intensity based landmarks are established, the corresponding range measurements are utilized by the SLAM algorithm to enable the determination of position and attitude. The results of this methodology are highly encouraging and suggest that a fusion between the 3D range data and the corresponding intensity data could be highly beneficial. This idea was suggested by the author of this thesis independently during conference proceedings; but was not incorporated in the present algorithm. As noted by Wang et al. [2009], the nature of the 3D Flash LADAR sensor (pre-2009) generates exceptionally noisy imagery, both from the intensity and ranging outputs. The experience of this dissertation suggests that intensity images are exceptionally susceptible to noise owing to background electro-magnetic radiation (lighting, sunlight, body heat, etc.) and the induced noise precluded incorporation until such time that the sensor technology improves.

The approach outlined by Guomundsson [2006] is conceptually similar to that proposed by the research presented in this dissertation, but not identical in methodology or implementation. Guomundsson utilized surface normals to segment the image, followed by construction of planar features via Principal Component Analysis (PCA) [Shlens,
2005] or Local Linear Embedding (LLE) [Chang and Yeung, 2005]. The LLE methodology was identified to be superior in performance (to PCA) based upon testing; however, the experiments were conducted in a highly controlled environment. The segmentation step is similar to the one proposed by this thesis, but the methods for feature extraction and matching are not identical and require human intervention of several parameters to enable the ongoing positional estimation of the sensor platform.

Another very recent paper [Lee et al., 2010] involves the use of a Velodyne 3D laser scanning device. Details related to the segmentation process are not provided, but stated to be “using any standard range image segmentation method”; interestingly, the aforementioned paper by Hoover et al. [1996] is referenced to define these “standard methods”, despite the poor results reflected therein! The primary focus by Lee et al., [2010] is the implementation of a Probability Hypothesis Density [Vo and Ma 2006] to facilitate tracking of motion objects. The experiments were completed without an available ground truth to reference, thus leaving assessment of the algorithm somewhat difficult. The authors did note that under conditions of highly cluttered environments and low signal-to noise ratios, the algorithm did not perform well. Mitigation strategies were theorized but no further testing or data were available in the noted paper.

Finally, Unel et al., [2010], has proposed the utilization of 2D “slices” of 3D data sets, followed by the derivation of algebraic curves to characterize the resulting shapes. The authors only utilized the algorithm for the purpose of object recognition and did not
expand the paper to include localization. The paper closely resembles work proposed in 2D by Taubin and Cooper [1992] and continued for the 3D instance [Markiel, 2007], utilizing implicit polynomials for feature matching. While highly effective in terms of accurate modeling, the methodology was identified to suffer from computational complexity and susceptibility to noise in terms of a navigation solution.

2.4 Nature of the Problem

The method for establishing the nature or existence of a feature has yet to be defined, nor the means to segment the range images. For the moment, let it be assumed that a methodology exists to extract features from a given 3D ranging image; the details will be covered in Chapter 3; lacking such detail does not hinder the present discussion of algorithmic conception.

At an acquisition epoch, $t_u$, the sensor acquires an array of $n \times m$ range values which contain an unknown number of features $p$. Let the coordinate frame of the sensor at $t_u$ consist of three orthogonal axes, with the $(i,j)$ axes parallel to the sensing plane, and the $k$ axis in the direction of the focal point. Call this coordinate reference frame $C_{t(u)}$.

At time $t_{u+1}$, the sensor will acquire an additional set of $n \times m$ range values containing $q$ unknown features. During the time interval $\Delta t = (t_{u+1} - t_u)$, the sensor will move to a new
(orthogonal) coordinate reference frame $C_{(u+1)}$, where the relationships between the two coordinate frames is defined by:

$$C_{(u+1)} = sRc_{(u)} + T$$  \hspace{1cm} (2.4)\

where $s$ is a scale factor between the acquired images, $R$ is a 3x3 directional cosine matrix of rotation, and $T$ is a 3x1 matrix of translation.

If $s$, $R$ and $T$ were known, the $p$ features from the image acquired at $t_u$ would be transformed to the same coordinate framework as image $t_{u+1}$. Let these features be $\tilde{p}$ where the tilde indicates the transformed condition. It would then be possible to compare the $\tilde{p}$ features from the first image to the $q$ features from the second image, and identify which features were common (matching) between the two images. The first problem arises from the condition that $s$, $R$ and $T$ are unknown; we lack sufficient information to place both sets of features into the same coordinate frame of reference.

Alternatively, if the matching features between the images were known, and the matching features were known to be exhibiting static behavior, the $xyz$ locations of the matching features could be utilized to determine $s$, $R$ and $T$. The second problem arises from the condition that we do not know which features within sets $p$ and $q$ are static in nature. The third problem is that we do not know which static features from $p$ are present in set $q$. The final problem is that we need information from images $t_{u+1}$ and $t_u$ in order to establish
which features are static, but static features are required to place both images in the same coordinate frame and enable evaluation of static behavior.

The dilemma is somewhat “chicken and egg” in nature; we require knowledge of static and matching features to determine the rotational and translating matrices, but determination of the static and matching features requires knowledge of the rotational and translational matrices initially. A similar problem occurs frequently in the mathematical theory of partial derivatives; the underlying (unknown) function of two or more independent variables is sought. Resolving these types of problems requires, in general, three pieces of information: a set of initial assumptions about the equation to be modeled, a set of initial conditions (constraints), and an iterative process of analysis involving circular reference until the residual error between the modeled equation and the derived result meets some acceptable level of error [Boyce and DiPrima, 1992]. To resolve the problem at hand, it becomes reasonable to pursue a comparable process to identify an iterative solution.
2.5 Conceptualizing a General Solution

For sake of clarity, the four problems identified in the previous paragraph are stated below.

1. The scaling factor $s$, rotational matrix $R$ and translational matrix $T$ are unknown between two images acquired at two epochs separated by time $\Delta t$.

2. The presence of static features within feature sets $p$ and $q$ is unknown, with $p$ and $q$ representing the features (static and non-static) present in each respective image.

3. The number of features from set $p$ which match features in set $q$ is unknown.

4. Determination that a feature is static requires information from both images, but only if both images can be placed in the same coordinate frame.

Let us first consider the assumption that an initial estimate is available to establish the $s$, $R$ and $T$ matrices; let these estimates be $\tilde{s}$, $\tilde{R}$ and $\tilde{T}$, respectively, with the tilde again indicating the estimated nature of scalar variable $s$ and the matrices $R$ and $T$. These initial estimates may be utilized to transform the set of static features from set $p$ into the coordinate frame containing feature set $q$ by means of equation 2.4.

Next, let there be some number of static features $r$, which match between sets $p$ and $q$; for the moment setting aside the issue of how the matching will occur. Since the initial estimates for $\tilde{s}$, $\tilde{R}$ and $\tilde{T}$ will contain some error, the match between these static features will exhibit a residual error in position. If the $\tilde{s}$, $\tilde{R}$ and $\tilde{T}$ matrices are iterated by
introducing small changes to the underlying base matrices, each of the iterations will induce a change to the residual error, either increasing or decreasing the associated positional displacement. As the iterated $\tilde{s}$, $\tilde{R}$ and $\tilde{T}$ matrices reduce the residual error, the global mismatch between static features will be minimized and the two images will begin to “nest”. The scaling factor $\tilde{s}$, rotational matrix $\tilde{R}$, and the translational matrix $\tilde{T}$ can now be determined by minimizing the residual differences in location between static features. The number of static features is not required to be known a-priori; if they exist, the minimization process will converge to a global minimum.

Resolving the $s$, $R$ and $T$ matrices by means of this functional minimization incorporates an implicit assumption that the number of static features is greater than the corresponding number of dynamic features. It is critical to consider the impact if this is not the case.

Consider a pair of images; containing feature sets $p$ and $q$, respectively, as before. Let the number of static features matching between the images be set $r_s$, and the corresponding number of dynamic features be contained in set $r_d$, with $r_d > r_s$. Now two instances can occur; in the first, at least half the dynamic features do not possess a common vector direction and magnitude, and the second, where at least half of the dynamic features are moving along a vector of common direction and magnitude.

In the first instance, any attempt to minimize the positional location of one dynamic feature will increase the positional error of features moving in a differing vector. This will
increase the overall global error. The global solution will occur when the average of the residual errors becomes a minimum. In this instance, the majority of the features will exhibit a residual error and the static features will now appear to have shifted position, introducing a bias. In the worst case, the static features could become interpreted as being dynamic, causing the loss of such features between images and derived sensor position to diverge from the actual location. To avoid such a decision, it is necessary to evaluate the iterated output from the feature matching process. One possible method is to simply count the number of “matching” features with positional error less than some small error value \( \epsilon \), and ensure that this is both larger than the “matching” features with larger residuals, and similar to the previous set of matched “static” features. A second method is to require a minimum number of “matching” features to exhibit a minimum error in position. A third possibility is to compare the residual error from the previous image set comparison to the current global residual error; a large increase in residual error would serve as a flag to either re-evaluate the matching process, or discard the current image and acquire new information.

The second possible instance can occur if the majority of dynamic points possess the same direction with common magnitude. The global minimum will occur when all of the dynamic features become matching, causing the static features to become classified as moving. If the majority of the “dynamic” points are in fact static features, the residual error of all “dynamic” (but actually static) points will be nearly identical, thus enabling the condition to be identified by simple analysis of the residuals for “dynamic” points. If
the alternative condition exists (the number of static points is less than the number of true dynamic points exhibiting error), the solution will converge to the dynamic set of features which appear to be static. Fortunately, the existence of this instance is unlikely, as a simple solution to the condition is difficult to envision with respect to two sequential images. The easiest solution is to enable a longer “memory” of static positions between multiple frames and introduce higher order heuristics to overcome this potential failure mode. Given expected likelihood of two mutually independent events (dynamic features outnumber static features, and the dynamic features are co-incident in a vector sense) to be small, the algorithm developed for this dissertation does not incorporate such advanced fail safes.

This analysis indicates that the four problems outlined at the initial start can be overcome by introducing an initial guess for the rotational matrix $R$ and translational matrix $T$, and then iterating the solution until the global minimization converges by matching static features. The features exhibiting minimal residual errors can then be utilized to update the sensor position. With one exception, simple checks can confirm the process to reduce spurious results. The remaining issue is the extraction of features to facilitate the matching process.
2.6 Conceptualizing Feature Matching

We seek to identify static features from, at a minimum, a pair of 3D range images. To accomplish the extraction and matching of features from the images requires the development of a mathematical representation of the features based upon the discrete points which represent the continuous reality. Direct comparison of range measurements is not sufficient to achieve this objective. A single range measurement can be easily obscured, either by intervening objects or noise in the acquisition process. A solitary range measurement on a flat surface, or along a rotating edge, may not indicate the motion of the surface between frames since an identical range measurement can occur within a small spatial offset. Instead, the range measurements must be characterized by sufficient descriptors such that, when compared, it becomes possible to achieve the objectives of extraction and matching.

The real world can be modeled as a four dimensional space with time as the fourth aspect, however, the dimensional space in which the objects exist can vary mathematically based upon the perspective of coordinate framework. Inclusion of the time element permits the extraction methodologies to include motion. Objects can be mathematically considered as a collection of primitives; that is, they can be considered as a set of points, lines, surfaces and volumes. Besl [1988a] provides a concise summary of the possible selections as shown in Table 2.4. In this table:
- Variables are denoted \((x, y, z, t)\)
- \(\mathbb{R}^1\) indicates real numbers of coordinate frame of dimension size 1
- \(\mathbb{R}^2\) indicates real numbers of coordinate frame of dimension size 2
- \(\mathbb{R}^3\) indicates real numbers of coordinate frame of dimension size 3
- \(\mathbb{R}^4\) indicates real numbers of coordinate frame of dimension size 4
- \([a, b]\) indicates the set of real numbers bounded by \(a\) and \(b\), with \(a \leq b\)
- Symbols \(u, v\) represent the mapping of variables to the parameter domain, such as a unit interval, unit square, or unit cube.
- \(\vec{f}, \vec{g}\) indicate vector functions of one or more variables (per above), while \(f\) and \(g\) without the overbar arrows indicate functions other than strictly vectorial
- The symbol \(\Omega\) denotes a region in the set space, and \(\partial \Omega\) reflects the gradient
- \(V\) indicates the volume of a three dimensional region, and \(\partial V\) reflects the gradient
- \(H\) represents the implicit definition of the space-time volume trajectory.

The selection of representation for range measurements carries with it an associated increase in complexity. Point representations can be expressed as a single value (range) at a grid position. Lines representations require multiple values; two single value endpoints, or a single point, length, and direction. Planar surfaces require four parameters, while non-planar surfaces require increasing levels of non-linear description. Increased complexity is not only computationally challenging, but can also lead to defining objects such that the matching process becomes unstable in the presence of perturbations at the
level of measurement resolution for the acquisition platform. [Besl, 1988b], [Costa and Cesar, 2001], [Markiel, 2007].

Each of the possible representations therefore contains a tradeoff between increasingly accurate representations of the continuous object space at the cost of intensifying computational loads and greater instability in the presence of small perturbations.

The matching process can be evaluated in terms of three measurable statistics; repeatability, localization, and distinctiveness [Canny, 1986], [Schmidt, et al., 2000]. Repeatability refers to the geometric stability of a point, or the ability to locate the same point in two images. Localization refers to the spatial positioning of the repeatable point between images. Finally, distinctiveness determines the statistical likelihood of identifying the point content from a set of descriptor points. Canny [1986] showed that improvements to repeatability could be accomplished by smoothing the image by implementation of appropriate filtering, but the localization of the resulting points would degrade. Schmidt et al., [2000] found that repeatability suffered in the presence of scale changes, and that the distinctiveness of information varied across a variety of tested algorithms.

The inability to distinguish point features under the influence of scale variations inspired (in part) Lowe [2004] to develop his Scale Invariant Feature Transform (SIFT) algorithm. The SIFT algorithm has been primarily applied to 1D and 2D imagery to date; the author
of this dissertation is not aware of any research efforts to apply SIFT to 3D datasets for the expressed purpose of positioning. Analysis at the Satellite, Positioning, and Inertial Navigation (SPIN) laboratory [Markiel, 2009] supported well published results by Lowe and others [Delponte et al., 2006], [Bauer et al., 2007], [Bakken, 2007], [Campbell, 2008], [Yang et al., 2009] in regards to the exceptional performance of SIFT with respect to both repeatability and extraction of feature content. The algorithm is remarkably robust to most image corruption schema, although white noise above 5% does appear to be the primary weakness of the algorithm. The algorithm suffers in three critical areas with respect to providing a 3D positioning solution. First, the algorithm is difficult to scale in terms of the number of descriptive points; that is, the algorithm quickly become computationally intractable for large numbers (>5,000) of image pixels.

Secondly, the matching process is not unique; it is exceptionally feasible for the algorithm to match a single point in one image to multiple points in another image. Finally, since the algorithm loses spatial positioning capabilities to achieve the repeatability, the ability to utilize matching features for triangulation or trilateration becomes impaired. Owing to the noted issues [Markiel, 2009], SIFT was not found to be a suitable methodology for real time positioning based upon Flash LADAR datasets.
<table>
<thead>
<tr>
<th>Dimension</th>
<th>1-D Space</th>
<th>2-D Space</th>
<th>3-D Space</th>
<th>4-D Space-Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-D Entity</td>
<td>Point</td>
<td>Point</td>
<td>Point</td>
<td>Point</td>
</tr>
<tr>
<td>x ∈ ℝ</td>
<td>(x, y) ∈ ℝ²</td>
<td>(x, y, z) ∈ ℝ³</td>
<td>(x, y, z, t) ∈ ℝ⁴</td>
<td></td>
</tr>
<tr>
<td>1-D Entity</td>
<td>Internal</td>
<td>Planar Curve</td>
<td>Space Curve</td>
<td>Point Trajectory</td>
</tr>
<tr>
<td>Parametric</td>
<td>[a, b] ⊂ ℝ</td>
<td>( \vec{f}(u) \in \mathbb{R}^2 )</td>
<td>( \vec{f}(u) \in \mathbb{R}^3 )</td>
<td>( \dot{\vec{f}}(u) \in \mathbb{R}^3 )</td>
</tr>
<tr>
<td>Implicit</td>
<td>( g(z) \leq 0 )</td>
<td>( g(y, z) = 0 )</td>
<td>( g(x, y, z) = 0 \in \mathbb{R}^3 )</td>
<td>( \ddot{g}(x, y, z, t) = 0 \in \mathbb{R}^3 )</td>
</tr>
<tr>
<td>2-D Entity</td>
<td>_</td>
<td>Region</td>
<td>Surface</td>
<td>Curve Trajectory</td>
</tr>
<tr>
<td>Parametric</td>
<td>( \Omega = \mathbb{R}^2 )</td>
<td>( \vec{f}(u, v) \in \mathbb{R}^3 )</td>
<td>( \vec{f}(u, t) \in \mathbb{R}^3 )</td>
<td>( \dot{\vec{f}}(u, t) \in \mathbb{R}^3 )</td>
</tr>
<tr>
<td>Implicit</td>
<td>( g(x, y) \leq 0 )</td>
<td>( g(x, y, z) = 0 )</td>
<td>( \ddot{g}(x, y, z, t) = 0 \in \mathbb{R}^2 )</td>
<td></td>
</tr>
<tr>
<td>Boundary</td>
<td>_</td>
<td>( \partial \Omega = \text{Planar Curve} )</td>
<td>( \vec{f}(u, v) \in \mathbb{R}^3 )</td>
<td>( \dot{\vec{f}}(a, t), \dot{\vec{f}}(b, t) )</td>
</tr>
<tr>
<td>3-D Entity</td>
<td>_</td>
<td>_</td>
<td>Volume</td>
<td>Surface Trajectory</td>
</tr>
<tr>
<td>Parametric</td>
<td>_</td>
<td>( V \subset \mathbb{R}^3 )</td>
<td>( \vec{f}(u, v, t) \in \mathbb{R}^3 )</td>
<td>( \dot{\vec{f}}(u, v, t) \in \mathbb{R}^3 )</td>
</tr>
<tr>
<td>Implicit</td>
<td>_</td>
<td>( g(x, y, z) \leq 0 )</td>
<td>( g(x, y, z, t) = 0 )</td>
<td>_</td>
</tr>
<tr>
<td>Boundary</td>
<td>_</td>
<td>( \partial V = \text{Surface} )</td>
<td>( \vec{f}(u, v, t); (u, v) \in \partial \Omega )</td>
<td>_</td>
</tr>
<tr>
<td>4-D Entity</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>Volume Trajectory</td>
</tr>
<tr>
<td>Parametric</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>( H \subset \mathbb{R}^4 )</td>
</tr>
<tr>
<td>Implicit</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>( g(x, y, z, t) \leq 0 )</td>
</tr>
</tbody>
</table>

Table 2.4. Primitive Entities by Dimension (from Besl; 1988a)
2.6.1 Conceptual 3D Feature Extraction

The previous discussions lead to the inescapable conclusion that a new type of feature extraction algorithm is needed which can leverage the addition of information provided by 3D sensors to achieve segmentation. The goal is to segment the image to obtain features, yet remain computationally tractable when presented with large datasets and limited windows for processing in real (or near real) time navigation. Ideally, the segmented features will possess a descriptor, or set of descriptors, which will enable a matching process to a future set of acquired features, while providing robust results in terms of repeatability, localization, and distinctiveness metrics.

All of the points in a single image may be considered part of a single large collection of points which are not immediately separable with application of an algorithmic approach to approximate the cognitive capabilities of a human being. When 3D images are spaced very close temporally, the two sets of points may be, in general, considered as two samplings of a larger, unknown distribution of discrete measurements which remains incomplete, since even the sum of the two discrete samplings cannot fully represent the continuous reality of the object space.

Without loss of generality, it is possible to separate the discrete points into three categories; surface points, transition points, or noise points. Surface points are defined as being part of a smoothly continuous function, such as along a wall, a telephone, or a
person. In contrast, transitional points are defined as those points where a particular surface evidences an abrupt change in inflection to another surface; for example, the transition from the wall to the floor. It is important to note that transitional points may be located on the surface of an object, but they will contribute to the segmentation of transitional points during the classification portion of the algorithm. Finally, noise points are erroneous measurements due to a wide variety of environmental and processing factors.

These categories serve to provide both repeatable and localized features. The final requirement is the ability to generate a distinct descriptor for each feature. We seek a descriptor which is repeatable, localized, and distinct in the presence of 3D motion. Instead of a single descriptor, such as intensity or range, what is envisioned is a triad of descriptors which uniquely define a point in 3D space, can be mathematically transformed to different coordinate frames of reference, and enable a matching process between points of interest. A feasible candidate for this purpose is an eigenvector triad.

2.6.2 Eigenvectors

The utilization of eigenvectors for segmentation and matching is well established in the field of computer vision; reference [Scott and Longuet-Higgins, 1991], [Sclaroff and Pentland, 1995], [Ghita and Whelan, 1997], [Weiss, 1999], [Campbell and Flynn, 1999]. More recently, Park et al., [2000] found matching of segmented features with eigenvector
descriptors to be fast, robust, and efficient while demonstrating invariance to translation, rotation, and scaling.

Consider the \(xyz\) triad that is associated with a particular point \(p_{i,j}\), where \((i, j)\) refer to the row and column of the 3D image, and the related \(xyz\) for points \(p_{i-1,j}\) and \(p_{i+1,j}\). Linearly fitting a line to these three points in an orthogonal least squares sense can be accomplished as per the following steps. Define \(X_i\) to be the three sample points and let the fitted line be:

\[
L = sD + A
\]  
(2.5)

In this equation, \(A = \frac{1}{m} \sum_{i=1}^{m} X_i\) is the average for the three sample points and \(D\) is a 3x1 vector of unit length along the diagonal with scaling factor \(s\). Next, let:

\[
X_i = A + d_iD + s_iD_i^\perp
\]  
(2.6)

Where \(d_i = D \cdot (X_i - A)\), and \(D_i^\perp\) is a unit length vector perpendicular to \(D\), with scaling coefficient \(s_i\). Defining \(Y_i = (X_i - A)\), the vector from \(X_i\) to its projection onto line \(Y_i\) is defined by the equation:

\[
Y_i - d_iD = s_iD_i^\perp
\]  
(2.7)
Since \( s_i^2 = (Y_i - d_iD)^2 \), the minimization of the summation of the scaling coefficients represents the least squares fit of the line to the data points \( X_i \). Therefore, the function \( F(A, D) \) can be solved by:

\[
F(A, D) = D^T (\sum_{i=1}^{n} (Y_i \cdot Y_i^T) - Y_iY_i^T)D = D^T M(A)D
\] (2.8)

Here \( D \) and \( Y \) are vectors of size 3x1 and \( M(A) \) is a matrix of size 3x3. The solution to \( D^T M(A)D \) is the quadratic form of the matrix [Wilkinson, 1965] where the minimum is the smallest eigenvalue of \( M(A) \) and the corresponding unit length eigenvector \( D \) enables the construction of the least squares line.

\[
M(A) = \delta \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} \sum_{i=1}^{n} (x_i - a)^2 & \sum_{i=1}^{n} (x_i - a)(y_i - b) & \sum_{i=1}^{n} (x_i - a)(z_i - c) \\ \sum_{i=1}^{n} (x_i - a)(y_i - b) & \sum_{i=1}^{n} (y_i - b)^2 & \sum_{i=1}^{n} (y_i - a)(z_i - b) \\ \sum_{i=1}^{n} (x_i - a)(z_i - c) & \sum_{i=1}^{n} (y_i - a)(z_i - b) & \sum_{i=1}^{n} (z_i - c)^2 \end{bmatrix}
\] (2.9)

Where

\[
\delta = \sum_{i=1}^{n} (x_i - a)^2 + \sum_{i=1}^{n} (y_i - b)^2 + \sum_{i=1}^{n} (z_i - c)^2
\] (2.10)

Since we are fitting three points, \( n = 1, 2, 3 \). The resulting eigenvector, therefore, enables the least squares fit of a line to the triplet of points which acts as a descriptor for the target point.
CHAPTER 3

ALGORITHM FOR NAVIGATION FROM 3D RANGE IMAGERY

3.1 Overview of the Algorithm for Navigation from 3D Ranging Imagery (NAVRI)

In this chapter, the conceptual algorithm discussed in the previous chapter is provided in a more complete mathematical foundation, and the sequential nature of its implementation is provided. The results from the NAVRI algorithm enables 3D ranging imagery to be leveraged to provide relative position and orientation updates, thus achieving navigation in a GPS challenged environment.

As a general overview, the NAVRI algorithm first utilizes the distribution of nearest neighbor distances (NND) to segment surface edges from two temporally spaced images. After the segmentation phase, a minimum edge consists of three adjacent pixels from the jth column. The center pixel is the edge; the above and below pixels are the adjacent neighbors, and each of the three pixels has an associated \( xyz \) triplet. The objective is to locate a pixel in the second image which matches the first edge pixel. What we seek is the function involving scale, rotation, and translation \( f(s, R, T) \) which, if the edge pixels are matching, would place the pixels from both images into the same coordinate frame as the
second. When the function \((s,R,T)\) has been refined, the position and orientation results, along with associated error estimates, are returned to the Extended Kalman Filter as a substitute for the GPS positional information.

Resolving the function \((s,R,T)\) is accomplished by first acquiring additional information from an external sensor, such as an IMU. This information (processed by means of the INS) provides an initial estimate of the desired function \(f(s,R,T)\) but also reflects uncertainty, reflected by estimates of positional and rotational error provided along with the mean values for translation and rotation. It is therefore possible to constrain the solution space; if the edge pixel from the first image is transformed to the coordinate system of image two, and if the edge pixel in image two matches this pixel, the corrected position for each of the three pixels from image one must lie within these error ellipses, and the eigenvectors of the edge pixel (as calculated from the three pixel set) should be identical to the eigenvector signature of the edge pixel in image two.

By randomly selecting \(xyz\) coordinates within the boundary of the error ellipses, it is possible to derive multiple possible eigenvector signatures, each defining the location of the edge pixel from image one in a coordinate frame approximating the second image coordinate frame. By choosing the eigenvector of the image one edge pixel which best matches the eigensignature of edge pixel of the second image, the functional \(f(s,R,T)\) can be refined to provide the best possible match. If an eigenvector match cannot be established, the hypothesis is that the edge pixels do not match.
By applying the $xyz$ coordinate shifts to every edge pixel being considered for matching, it is possible to find the coordinate shift which creates the largest number of matching edge pixels in the global sense. As will be discussed in Section 2.5, this implicitly assumes that the number of static edge pixels outnumber the dynamic edge pixels. The consequences if this is not the case, along with mitigation strategies, will be covered in Section 2.5 of this chapter.

After the maximum number of matching edge pixels has been identified (which globally minimizes the difference between eigenvector signatures), the $xyz$ coordinates of all “matching” edge pixels are utilized in Horn’s Method [Horn, 1987] to establish the refined functional $f(s,R,T)$ which best transforms image one pixels into the coordinate frame of image two. The translation values are utilized to provide positional updates, while the rotation matrix is utilized to provide orientation updates.

The NAVRI algorithm consists of several steps, or modules, as outlined in the following bullets.

- Hypothesis Testing of Two Images
- Segmentation
- Feature Extraction
- Coarse Matching of Two Coordinate Frames
- Feature Matching
• Determination of Position

A pictorial overview of the NAVRI algorithm is shown in Figure 3.1 below.

In each instance, two 3D “images” have been acquired, each of size \(n \times m\); the first image consisting of \(p\) features, the second image consisting of \(q\) features. Note that \(p\) is not necessarily equal to \(q\); in reality, it is fully anticipated that this will not be the case. Furthermore, it is given that information from an INS is available to provide a coarse estimate of the orientation and position changes between epochs of interest (those corresponding to the epochs of each image). While the algorithm would remain technically feasible without the INS estimate, the inability to constrain the search space would likely become rapidly intractable in a computational sense for even moderate sets of pixels, say <5,000 in set size.

The \(p\) features in the first image consist of static features \(p_s\) and dynamic features \(p_d\), with

\[
p = p_s \cup p_d
\]

(3.1)

And the similar condition for the \(q\) features in the second image.

It is important to note that it is assumed that features do exist, but the specific quantity of features in \(p_s, p_d, q_s, q_d\) is unknown at time of acquisition.
Figure 3.1 General Overview of the NAVRI Algorithm

3.2 Hypothesis Testing of Two Images

As previously noted in Chapter Two, each image represents a discrete set of measurements which approximate the continuous reality. The hypothesis is that the two images, separated by time interval $\Delta t$, represent samples of a larger, unknown distribution. In the strictest sense, if any dynamic features are present, they will obscure portions of the environment in one image but not the same portions in a second image, thus violating the hypothesis that the two images are samples of a single large
distribution. In a practical sense, while some changes (dynamic features) may occur between the two images, if the two samples are statistically similar based upon comparison of their variances, then the hypothesis is true (the two images are similar), the content (features) of the two images can be utilized for SFEM activities. To conduct such a test, it is first necessary to establish a statistical distribution for each image from which a suitable statistical vector (mean and variance) can be derived; the statistical vector $\theta$ can then be subjected to standard analysis of variance techniques.

The distribution which will be utilized is based upon nearest neighbor distances (NND). In addition to the current hypothesis testing, this distribution will be utilized in future modules to derive threshold limits for various operations.

Consider a target pixel at position $(i, j)$ in the image. Surrounding this target pixel are eight neighboring pixels; each pixel (including the target) contain a $xyz$ triplet of positional information. However, as will be discussed in Chapter 4, the “pixels” are located onto an array at fixed distances from the principal point of the image sensor; thus the $xy$ distances are a function of the array spacing and the measured range distance, and do not represent an independent variable. This condition is illustrated in Figure 3.2; the distance from the sensor origin to the $(i, j)$ pixel on the sensor array is fixed in terms of the $x$ and $y$ axis, while the range measurement ($z$) is unique.
The nearest neighbor distance can therefore rely exclusively on the range distance, rather than the full, non-linear, Euclidean distance measure. The average distance from the target pixel to its eight nearest neighbors is therefore:

\[
\bar{d}_{ij} = \frac{1}{8} \sum_{k=1}^{8} (z_{tp} - z_k)
\]  

(3.2)

Where \( \bar{d}_{ij} \) is the average distance for the \((ij)^{th}\) pixel, \(z_{tp}\) is the range distance of the target pixel, and \(z_k\) refers to the range distance for each of the eight surrounding pixels.
Repeating this exercise from the \((2, 2)\) position to the \((n-1, m-1)\) position, a distribution of average NND measurements in generated. It must be noted that this excludes the outermost row and column of range data; this presents no limitation on the algorithm so long as these range measurements are excluded from the matching process. The resulting Cumulative Probability Distribution (CPD) is expressed as:

\[
P\{d_n\} = Pr\{D \leq d_n\} = \sum_{i=0}^{n} p\{d_i\}
\]  

(3.3)

Where \(P\{d_n\}\) is the probability that \(D\) is less than or equal to \(d_n\) and \(p\{d_i\}\) is the probability of the \(i^{th}\) distance [Brownlee, 1965]. Since the hypothesis is that this sample distribution is representative of a continuous distribution of range values, equation 3.3 can be extended to a Probability Density Function (PDF) [Brownlee, 1965]:

\[
P\{d_n\} = \int_{-\infty}^{+\infty} p\{x\} \; dx = 1
\]  

(3.4)

Now that the distribution has been derived, it is possible to generate the expectation and variance of the distribution. The NND is a set of sample averages; by the Central Limit Theorem, for sufficiently large numbers of samples, the NND will represent a normal distribution, regardless of the underlying data structure [Hamburg, 1974]. The first and second moments are reflected in equations 3.5 and 3.6 respectively [Brownlee, 1965] where \(\xi\) represents the mean of a theoretical normal distribution with variance \(\sigma^2\) and
standard deviation $\sigma$. Note that we have used $\xi$ in order to reserve $\mu$ for future use, as will be shown momentarily.

$$E[D] = \int_{-\infty}^{+\infty} xp(x)dx = \xi \int_{-\infty}^{+\infty} p(x)dx + \int_{-\infty}^{+\infty} (x - \xi) \frac{1}{\sigma \sqrt{2\pi}} e^{-(x-\xi)^2/2\sigma^2} dx$$  \hspace{1cm} (3.5)

$$E[D^2] = \int_{-\infty}^{+\infty} x^2 \cdot \frac{1}{\sigma \sqrt{2\pi}} e^{-(x-\xi)^2/2\sigma^2} dx$$  \hspace{1cm} (3.6)

Special note should be made with respect to the standard deviation; since the nearest neighbor distribution is formed from sample means rather than individual samples, the actual deviation is [Hamburg, 1974]:

$$\sigma_\bar{x} = \sqrt{1 - \frac{n}{N} \frac{\sigma_x}{\sqrt{n}}}$$  \hspace{1cm} (3.7)

Where $n$ is the sample size, $N$ is the total number of population elements, $\sigma_x$ is the standard deviation of the population, and $\sigma_\bar{x}$ is the standard deviation of the sample average NND.

The NND is, therefore, determined for each image, followed by the derivation of the first and second moment for each PDF, yielding $\theta_p = \{\mu_p, \sigma_p\}$ and $\theta_q = \{\mu_q, \sigma_q\}$, where $\mu$ has been substituted for $x$ to reflect the mean of the NND, and the subscripts $p$, $q$ refer to the two images.
The next step is to test the hypothesis that $\sigma_p$ and $\sigma_q$ sample standard deviations are equivalent representations of the same larger and unknown distribution. This can be accomplished by the well-known Fisher’s Test, or simply an F-test. Invoking a two-sided test [Brownlee, 1965], the hypothesis $H_0: \sigma_p = \sigma_q$ is accepted if:

$$\frac{\sigma_p}{\sigma_q} < F_{\alpha/2}$$  \hspace{1cm} (3.8)

And

$$\frac{\sigma_p}{\sigma_q} > F_{1-\alpha/2}$$  \hspace{1cm} (3.9)

$F_{\alpha/2}$ in the equation 3.8 represents the level of statistical significance to test the null hypothesis against. The final question arises as to the selection of statistical significance $\alpha$. The first image has degrees of freedom $p$, the second image $q$. Since the 3D ranging image contains a total of $n \times m$ pixels with count greater than 120, a quick look at a standard $F$-test lookup table quickly shows that for sample sizes greater than 120 all $F$-tests converge to one (1) [Brownlee, 1965]. This therefore renders the selection moot owing to the large sample size available in 3D range images, which have 15,000 pixels or more. However, implementation of such a tight measure is not feasible since dynamic features may be present in (or between) the two images. Accepting the realistic
probability that dynamic features may be present requires the acceptance that $\frac{\sigma_p}{\sigma_q} \neq 1$. The objective of the module is to verify that the distributions are not radically different; that is, that the probability is high that matching features will be present between the images, but not all features will match. Based upon this modified adaptation to accommodate the realistic nature of the acquired environment, at the level of significance of $\alpha = 99.9\%$, a reduced $F$-statistic of 1.76 ($p = q = 120$) [Brownlee, 1965] provides a reasonable balance.

If the hypothesis is accepted, then the two images can be reasonably considered as samples from a continuous distribution based upon the $F$-test, else, the second image has radically changed. The situation now returns outside of the navigation algorithm and enters the realm of platform heuristics; the platform might be ordered to return the previous position (to confirm location), wait for an additional image to be acquired at $\Delta t_{u+2}$, or some other set of instructions. In any event the algorithm indicates the insufficiency between the acquired images and stops until additional information can be obtained.

With respect to the current implementation, the algorithm attempts to compare the first image to two additional images (total of three). If two images cannot be found to be matching, the algorithm has no choice but to “reset” to the last information provided by the INS and proceed again based upon the most recent image frame. We again note the
necessity of having an INS present; in this instance, the INS provides a navigation solution when the camera information is insufficient.

3.3 Image Segmentation

If the images are found to be statistically similar, the algorithm proceeds to segment the images. The segmentation module involves two pieces of information; the NND from the previous module, and the eigenvector/eigenvalue descriptor derived for each pixel.

Again consider the $i, j$ pixel drawn from the first image, along with the row immediately above and below the target ($i, j$) pixel; that is, pixels $(i-1, j), (i, j), (i+1, j)$. As per Section 2.6.2 in the previous chapter, the eigenvalue ($\alpha_{i,j}$) and eigenvector ($\lambda_{i,j}$) for the target pixel are easily derived from minimization of orthogonal regression. Since the eigenvalue “stretches” the eigenvector in a particular direction, the “endpoints” of the eigenline are therefore obtained by multiplying the scaling factor times the eigenvector; i.e., $(\alpha\lambda)_{i,j}$ and $-(\alpha\lambda)_{i,j}$. The “eigen-distance” between the endpoints is therefore $\delta_{ij} = 2(\alpha\lambda)_{i,j}$, since the endpoints are equally displaced from the centroid of the three points $A = \frac{1}{m} \sum_{i=1}^{m} X_i$, reference Chapter 2, Section 2.6.2). Proceeding as before, the eigen-distance is determined for each range pixel from the $(2, 2)$ location to the $(n-1, m-1)$ pixel, excluding the outer rim, as before.
Now note that at any selected sigma from the NND distribution, a corresponding distance can be selected. Call this distance $\beta$, representing the threshold below which two distances will be considered “identical”; above this value, the two distances will be considered “different”. The algorithm now proceeds sequentially, first row by row, then column by column, comparing the eigen-distances $|\delta_{i,j} - \delta_{i,j-1}|$ (for adjacent pixels within a row), or $|\delta_{i,j} - \delta_{i-1,j}|$ (for pixels within columns) to the threshold parameter $\beta$. If $\delta_{i,j} \leq \beta$, the pixels are merged together, else, the pixels remain independent. Merged pixels are considered surface points, while independent pixels are considered to be either transition or noise pixels.

Logically, if $\beta$ is small, few pixels will merge, and most pixels will be classified as transition/noise pixels. If $\beta$ is large, a majority of the pixels will merge and very few pixels will be classified as transition/noise pixels. The question is how to determine the appropriate threshold, such that pixels are appropriately classified by the algorithm.

With respect to the NND, it is possible to count; based upon a given sigma (distance), the number of pixels greater than and less than the threshold. Let the number pixels greater or equal to the threshold be $\gamma^+$, and the number less than the threshold be $\gamma^-$. Let the count of merged pixels from the image be $\delta^+$ and $\delta^-$ respectively. If the algorithm starts at one extreme (say $\beta = 0.5\sigma$) and iterates to progressively larger distances, the pixel counts for $\gamma^+, \gamma^-, \delta^+$ and $\delta^-$ will change based upon each possible value of $\beta$. What the
algorithm seeks is the minimization of the difference between the counts of un-merged pixels and expected count of NND pixels with distances larger than $\beta$, that is:

$$\varepsilon_{min} = |\gamma^- - \delta^-|$$  \hspace{1cm} (3.10)

When this condition occurs, the selected $\beta$ is considered to be the appropriate threshold parameter for the image, and the resulting segmentation of pixels occurs to separate surface pixels from transition/noise pixels. Note that $\beta$ will be selected independently for each image, because the threshold is determined uniquely from the image range data, rather than a-priori value(s). After this step, a median filter is utilized to remove noise pixels from the image. The resulting product is a binary image with all transitional pixels flagged with a one (1) and all merged/noise pixels flagged as zero (0). Each flagged transitional pixel carries meta-data in terms of the $xyz$ location triplet and the eigenvector/eigenvalue signature of the pixel.

In general, the number of transitional pixels is significantly less than the available $(n, m)$ pixels in the image, enabling a significant reduction in the dimensionality of the matching process, each possessing sufficient quantity to enable a highly redundant solution based upon matched points. If necessary, the algorithm could be modified to require a certain number of transitional pixels to be present in order to proceed to the matching process. Based upon testing to date, this has not been a concern. Another option could be to focus on the surface pixels rather than the transitional pixels. This option has some merit as
transitional pixels are typically the edges of surfaces and exhibit less than robust behavior. The tradeoff is a higher number of pixels to match in the subsequent steps, presuming that the merged pixels outnumber the transitional pixels (again the typical condition). In the current algorithm, the focus has been to utilize the transition pixels to achieve greater computational efficiency, particularly due to the aforementioned size of the data stream.

The utilization of the NND in conjunction with the eigen-distance to enable image segmentation is one of the important contributions of this paper. It is worth considering why the threshold schema involving the NND and the eigen-distance works.

Geometrically, if the distance from the target pixel to its adjacent pixels is (nearly) identical, the likelihood is that the three pixels represent the same surface in a continuous reality. Inversely, if the distances are not identical, the likelihood is that the pixels represent a point of inflection, representing a transition from one surface to another.

Mathematically, the eigenvector calculated across the three points represents one of the three axes in a Principal Component Analysis (PCA) [Duda et al, 2001] of the 3D dataset; that is, the eigenvector represents the dataset comprised of three points. The comparison of the PCA to the NND is closely related to the application of Orthogonal Regression (OR) to support dataset analysis for both outlier rejection and error estimation [Jackson and Dunlevy, 1988], [Leng, et. al, 2007]. The classic dilemma is that PCA can represent
the data point, but lacks the ability to discriminate the data into different classes [McCuen, 1985]. The statistical theory relating PCA and OR, including the relevant error analysis, will be developed momentarily in sub-section 3.3.1.

By merging eigenvector with similar representations, the algorithm attempts to separate the pixels (ranges) into the three classes of surface, transition, or noise. The NND distribution provides the capability to discriminate the data into different classes by controlling the extent of segmentation. Thus, the combination of the two methodologies enables the image to be segmented, using heuristics derived statistically from the dataset, in a robust process.

3.3.1 Statistical Theory Supporting Eigenvector Segmentation

Material supporting this section is drawn from [Brook and Arnold, 1985], [Draper and Smith, 1981], and [Montgomery and Peck, 1982] (pages as per bibliography), as well as papers by [Jackson and Dunlevy, 1988], and [Leng, et. al, 2007].

Consider the linear regression of the 3D system fashioned by three pixels, each possessing a triad of both positions and associated standard deviations \((x, y, z, \sigma_x, \sigma_y, \sigma_z)\). The objective is to create a hyperplane which fits the set of points with minimal error with respect to a particular direction of orientation for the system. Generalizing, there is a linear hyperplane (or set of hyperplanes):
\[ n^T \tilde{x}_i + d = n_1 \tilde{x}_{i1} + n_2 \tilde{x}_{i2} + \cdots + n_m \tilde{x}_{im} + d = 0, \quad i = 1, 2, \ldots n \quad (3.11) \]

Where \( n = (n_1, n_2, \ldots, n_m) \in \mathbb{R}^m \) contains the regression coefficients of \( n \) data vectors \( \tilde{x} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_m)^T \), and \( d \) is offset of the hyperplane with respect to the system origin. The vector \( n \) is in the direction of the normal to the associated hyperplane, and \( |d|/|n| \) represents the perpendicular distance from the hyperplane to the origin.

It is further assumed that each of the data points is perturbed by a random noise vector \( \delta \tilde{x}_{i=1,2,\ldots,n} \in \mathbb{R}^m \), thus each range measurement is expressed by:

\[ x_i = \hat{x}_i + \delta \hat{x}_i. \quad (3.12) \]

In the method of Ordinary Least Squares (OLS) it is necessary to impose an external coordinate system on the system, which then defines a-priori both axis definition and direction of minimization. Orthogonal Regression (OR) computes the coordinate system on the basis of least squares fit. Figure 3.3 shows the difference between the methods; OLS minimizes the vertical/horizontal distance from the data point to the regression line (3.2a, 3.2b), while OR minimizes the orthogonal distance (3.2c). OLS presumes that one variable (the dependent variable along the y axis) possesses error, while all remaining variables are independent and possess known accuracies. By contrast, OR permits the inclusion of error of all variables in all directions and yields the “best” relationship between the variables of interest. It is therefore evident that OR is better suited to the
problem of fitting the hyperplane since it accommodates error in all dimensions simultaneously.

Figure 3.3. Comparison of Ordinary Least Squares (a,b) to Orthogonal Regression (c)

The OR can be achieved by minimizing the sum of squared perpendicular distances, as shown by equation 3.13:

$$\varepsilon(\hat{x}, d) = \sum_{i=1}^{n} \frac{(n^T x_i + d)^2}{|x_i|^2}$$  \hspace{1cm} (3.13)

In this equation, the vector $x_i$ reflects the data points $x_i = (x_{i1}, x_{i2}, \ldots x_{im})^T$, $\hat{x}$ is a vector of regression coefficients to be estimated, $d/|x|$ is the perpendicular distance from the hyperplane to the origin, and the function $\varepsilon(x, d)$ reflects the minimization function.
To resolve equation 3.13, let \( n \) be unit vector, thus eliminating the denominator term in (3.13), generating:

\[
\varepsilon(x, d) = \sum_{i=1}^{n} (n^T \vec{x}_i + d)^2, \quad |n|^2 = 1
\]  

(3.14)

The hyperplane must pass through the centroid [Wilkinson, 1965], and so \( d = -n^T \bar{x} \). Thus we can eliminate \( d \) by centering the data points, as will be shown momentarily.

Expressing the constraint as a Lagrange multiplier (\( \gamma \)), we obtain:

\[
\varepsilon^*(x, d) = \varepsilon - \gamma (n^T n - 1) = 0
\]  

(3.15)

In equation 3.15, the asterisk (*) is used to distinguish the functional \( \varepsilon^* \) from the variable \( \varepsilon \). Setting the partial derivatives of the Lagrangian to zero indicates that the hyperplane must pass through the centroid of the data, as shown in equation 3.16, with \( \bar{x} \) indicating the mean value of the data points.

\[
\frac{\partial \varepsilon^*}{\partial d} = 0 = \sum_{i=1}^{n} (2n^T x_i + 2d) \rightarrow d = -\frac{1}{n} \sum_{i=1}^{n} (n^T x_i) = -n^T \bar{x}
\]  

(3.16)

Centering the data points with \( v_i = x_i - \bar{x} \), equations 3.17a and 3.17b are obtained, which indicates that the solution for \( x \) is the eigenvector of \( A = \sum_i v_i v_i^T \) corresponding to the smallest eigenvalue [Wilkinson, 1965].
\[
\varepsilon^*(x) = \sum_{i=1}^{n}(n^T \nu_i)^2 - \gamma(n^T n - 1) = n^T (\sum_{i=1}^{n} \nu_i \nu_i^T)n - \gamma(n^T n - 1)
\]

(3.17a)

\[
\frac{\partial \varepsilon^*}{\partial n} = 0 = 2A n - 2\gamma n \rightarrow A n = \gamma n
\]

(3.17b)

Thus \( n \) is a unit eigenvector of \( A \) corresponding to the eigenvalue \( \gamma \); to determine which eigenvalue, we substitute back into \( \varepsilon \).

\[
\varepsilon = n^T A n = n^T \gamma n = \gamma
\]

(3.17c)

Classically, the minimum eigenvalue method is utilized to remove outliers from the dataset under study. The usual approach is to iteratively remove data points individually, recalculate the \( A \) matrix, and evaluate for improvement to the magnitude of the lowest eigenvalue. An alternative approach is to evaluate the residual error associated with each data element after the iteration is complete and continue until some threshold minimum error value is achieved.

In our present analysis, the problem is recast; given the calculated eigen-distance and the NND distribution, what does the eigen-solution suggest for pixel classification? The hypothesis of the algorithm is that eigen-distances greater than the threshold \( \beta \) reflect transition pixels (edges). The justification for this hypothesis can be illustrated by
completing an error analysis of the noise perturbation defined in equation 3.12, and develop a confidence estimate in the computation of $x$. Our error analysis is tailored to the 3D nature of the dataset, but follows in spirit the 2D motion analysis outlined by Weng, et. al, [1989].

Let the perturbation $\delta x_i$ have zero mean and variance $\sigma^2$, and be modeled as independent Gaussian noise. Then, with each pixel possessing independent errors:

$$E\{\delta x_i\} = 0, \ E\{x_i\} = E\{x_i + \delta x_i\} = \bar{x}_i, \ Var\{x_i\} = Var\{\delta x_i\}$$ (3.18a)

$$\Gamma_x = E\{\delta x_i \delta x_i^T\} = \sigma^2 I$$ (3.18b)

In equation 3.18 I use $\Gamma_x$ to indicate the covariance function of the data points.

The perturbation induces a shift to the centered data points $v_i$.

$$v_i = x_i - \bar{x} = (\bar{x}_i - \bar{x}) + (\delta x_i - \delta \bar{x}) = \bar{v}_i + \delta v_i$$ (3.19)

By centering the data the noise of each individual point influences the noise of the centroid, and the covariance matrix for $\delta v$ must differ from the covariance matrix of $\delta x$.

$$E\{\delta v_i \delta v_i^T\} = E\{(\delta x_i - \delta \bar{x})(\delta x_i - \delta \bar{x})^T\} = E\{\delta x_i \delta x_i^T\} - E\{\delta \bar{x} \delta x_i^T\} - E\{\delta x_i \delta \bar{x}^T\} +$$

$$E\{\delta \bar{x} \delta \bar{x}^T\} = \sigma^2 I - \frac{1}{n} \sigma^2 I - \frac{1}{n} \sigma^2 I + \frac{1}{n^2} (n\sigma^2 I) = \frac{n-1}{n} \sigma^2 I, \quad i = j$$ (3.20a)
\[ E\{\delta v_i \delta v_j^T\} = E\{(\delta x_i - \delta \bar{x})(\delta x_j - \delta \bar{x})^T\} = E\{\delta x_i \delta x_j^T\} - E\{\delta \bar{x} \delta x_j^T\} + E\{\delta \bar{x} \delta \bar{x}^T\} = 0 - \frac{1}{n} \sigma^2 I - \frac{1}{n} \sigma^2 I_m + \frac{1}{n^2} (n \sigma^2 I) = -\frac{1}{n} \sigma^2 I, \quad i \neq j \] (3.20b)

The covariance matrix for \( v_i \) is therefore expressed as per equation 3.21a, and for \( v_i \) and \( v_j, \ i \neq j \), as per equation 3.21b [Wilkinson, 1965]. Note that we have used \( \Upsilon \) to distinguish the covariance matrices for the two possible results.

\[ \Gamma_v = E\{\delta v_i \delta v_j^T\} = \frac{\sigma^2}{n}(n - 1)I = \frac{n-1}{n} \Gamma_r, \quad i = j \] (3.21a)

\[ \Upsilon_v = E\{\delta v_i \delta v_j^T\} = -\frac{\sigma^2}{n}I = -\frac{1}{n} \Gamma_r, \quad i \neq j \] (3.21b)

This indicates that when the number of points “n” is large, the covariance matrix for centered data tends towards the covariance matrix for data which is not centered, since \((n - 1)/n \to 1\) and \(-1/n \to 0\). In matrix form, \( V = [v_1|v_2|...|v_n] \) and \( \delta V = [\delta v_1|\delta v_2|...|\delta v_n] \).

The impact of the perturbation to the \( A = VV^T \) matrix due to the noise in \( V \) is therefore given by:

\[ A = (\tilde{V} + \delta V)(\tilde{V} + \delta V)^T = \tilde{V} \tilde{V}^T + \tilde{V} \delta V^T + \delta V \tilde{V}^T + \delta V \delta V^T = \hat{A} + \delta \hat{A} \] (3.22a)
Where the “hat” indicates the unperturbed variable. Writing \( A = \hat{A} + \delta A \) and \( \hat{A} = \tilde{V} \tilde{V}^T \) and using a first order approximation [Weng, et. al, 1989]:

\[
\delta A \approx \tilde{V} \delta V^T + \delta V \delta V^T
\]  

(3.23)

Let the vector \( \hat{u}_1 \) contain the \( m \) eigenvectors which correspond to equation 3.17. Let the \( j^{th} \) eigenvector be associated with the solution after perturbing outliers have been removed. Since \( \hat{A} \hat{u}_j = \hat{\lambda}_j \hat{u}_j \), in the noise free solution, \( \hat{\lambda}_j \) must be equal to zero. Since equation 3.23 reflects a symmetric matrix, the first order change in the unit axis is given by [Wilkinson, 1965] as:

\[
\delta u_1 = - \sum_{k=2}^{m} \left( \frac{\hat{u}_1^T \delta \hat{A} \hat{u}_1}{\hat{\lambda}_k} \right) \hat{u}_k = - \left( \sum_{k=2}^{m} \frac{\hat{u}_k \hat{u}_k^T}{\hat{\lambda}_k} \right) \delta \hat{A} \hat{u}_1
\]  

(3.24)

Since the noise free residuals require that \( \hat{u}_1^T \hat{v}_i = 0 \) (all the points lie on the hyperplane), \( \tilde{V}^T \hat{u}_1 = 0 \), and therefore:

\[
\delta \hat{A} \hat{u}_1 = (\tilde{V} \delta V^T + \delta V \delta V^T) \hat{u}_1 = \tilde{V} \delta V^T \hat{u}_1
\]  

(3.25)

Thus, the induced perturbation to the unit axis is:
\[
\delta u_1 = \hat{B} \widehat{\nu V^T} \hat{u}_1 = \hat{B} \sum_{i=1}^{n} \hat{v}_i (\delta v_i^T \hat{u}_1), \quad \hat{B} = -\left(\sum_{k=2}^{m} \frac{\hat{u}_k \hat{u}_k^T}{\lambda_k}\right)
\] (3.26)

The covariance of the eigenvector matrix can be determined as follows:

\[
\Gamma_{u_1} = E\{\delta u_1 \delta u_1^T\} = E\{\hat{B} \widehat{\nu V^T} \hat{u}_1 \hat{u}_1^T \delta V \widehat{\nu}^T \hat{B}^T\}
\] (3.27a)

\[
= \hat{B} E\{(\sum_{i=1}^{n} \hat{v}_i (\delta v_i^T \hat{u}_1))(\sum_{j=1}^{n} \hat{v}_j^T (\delta v_j^T \hat{u}_1))\}\hat{B}^T
\] (3.27b)

\[
= \hat{B} \left[\sum_{i=1}^{n} \hat{v}_i \left(\sum_{j=1}^{n} \hat{v}_j^T E\{\delta v_i \delta v_j^T\} \hat{u}_1\right)\right] \hat{B}^T
\] (3.27c)

From the previous analysis:

\[
E\{\delta v_i \delta v_j^T\} = \begin{cases} 
\sigma^2 \left(1 - \frac{1}{n}\right) I, & i = j \\
-\sigma^2 \frac{1}{n} I, & i \neq j
\end{cases}
\] (3.28a)

Since \(\hat{u}_1^T \hat{u}_1 = 1\),

\[
\hat{u}_1^T E\{\delta v_i \delta v_j^T\} \hat{u}_1 = \begin{cases} 
\sigma^2 \left(1 - \frac{1}{n}\right), & i = j \\
-\sigma^2 \frac{1}{n}, & i \neq j
\end{cases}
\] (3.28b)

Since \(\sum_{j=1}^{n} \hat{v}_j = 0\),

64
\[
\sum_{j=1}^{n} \hat{v}_j^T \hat{u}_1^T E\{\delta v_i \delta v_j^T\} \hat{u}_1 = \sigma^2 \hat{v}_j^T - \frac{\sigma^2}{n} \sum_{j=1}^{n} \hat{v}_j^T = \sigma^2 \hat{v}_j^T \quad (3.29a)
\]

\[
\sum_{i=1}^{n} \hat{v}_i (\sum_{j=1}^{n} \hat{v}_j^T \hat{u}_1^T E\{\delta v_i \delta v_j^T\}) \hat{u}_1 = \sigma^2 \sum_{i=1}^{n} v_i v_i^T = \sigma^2 A \quad (3.29b)
\]

From the previous, \( \hat{A} \hat{u}_j = \hat{\lambda}_j \hat{u}_j \) and \( \hat{u}_i^T \hat{u}_j = \delta_{ij} \),

\[
\Gamma_{u_1} = \sigma^2 \hat{B} \hat{A} \hat{B}^T = \sigma^2 \left( \sum_{k=2}^{m} \frac{\hat{u}_k \hat{u}_k^T}{\hat{\lambda}_k} \right) \left( \sum_{l=2}^{m} \frac{\hat{\lambda}_l \hat{u}_l \hat{u}_l^T}{\hat{\lambda}_l} \right) \quad (3.30a)
\]

\[
= \sigma^2 \left( \sum_{k=2}^{m} \frac{\hat{u}_k \hat{u}_k^T}{\hat{\lambda}_k} \right) \left( \sum_{l=2}^{m} \frac{\hat{\lambda}_l \hat{u}_l \hat{u}_l^T}{\hat{\lambda}_l} \right) = \sigma^2 \sum_{k=2}^{m} \frac{\hat{u}_k \hat{u}_k^T}{\hat{\lambda}_k} = -\sigma^2 \hat{B} \quad (3.30b)
\]

With the covariances of \( \Gamma_v \) and \( \Gamma_{u_1} \) in hand, it is now possible to analyze the residual error owing to the noisy solution. Let the vector \( \ell^T = (\ell_1, \ell_2, ..., \ell_n)^T = u_1^T V \) which reflects the residual of fit between the \( i^{th} \) data point and the eigenvectors \( V = (|v_1|, |v_2|, ..., |v_n|) \). We begin with this residual variance:

\[
l_i = \hat{l}_i + \delta l_i = (\hat{u}_i + \delta u_i)^T (\hat{\nu}_i + \delta \nu_i) = \hat{u}_i^T \hat{\nu}_i + \hat{u}_i^T \delta \nu_i + \hat{\nu}_i^T \delta u_i + \delta u_i^T \delta \nu_i \quad (3.31)
\]
where \( \hat{\ell}_i = 0 \) represents the unperturbed residual fit between the data and the hyperplane set. To obtain an expression for the perturbation, we ignore the second order terms and observe that:

\[
\delta l_i = \hat{u}_i^T \delta v_i + \hat{v}_i^T \delta u_1
\]  

(3.32)

The expectation is zero \((E[\delta l_i] = 0)\), and the associated variance is easily determined to be:

\[
Var[\delta l_i] = E\{(\hat{u}_i^T \delta v_i + \hat{v}_i^T \delta u_1)^2\} = E\{(\hat{u}_i^T \delta v_i)^2 + (\hat{v}_i^T \delta u_1)^2 + 2(\hat{v}_i^T \delta u_1)(\hat{u}_i^T \delta v_i)\}
\]  

(3.33a)

\[
= \hat{u}_i^T \Gamma_v \hat{u}_1 + \hat{v}_i^T \Gamma_u \hat{v}_1 + 2\hat{v}_i^T E\{\delta u_1 \delta v_i^T\} \hat{u}_1
\]

(3.33b)

\[
= \frac{n-1}{n} \sigma^2 \hat{u}_i^T I \hat{u}_1 - \sigma^2 \hat{v}_i^T \hat{B} \hat{v}_1 + 2\hat{v}_i^T E\{\hat{B} (\Sigma_{j=1}^n \hat{v}_j (\delta v_j^T \hat{u}_1) \delta v_i^T)\} \hat{u}_1
\]

(3.33c)

\[
= \frac{n-1}{n} \sigma^2 \hat{u}_i^T I \hat{u}_1 + \sigma^2 \hat{v}_i^T \left( \sum_{k=2}^m \frac{\hat{u}_k \hat{u}_k^T}{\lambda_k} \right) \hat{v}_1 + 2\hat{v}_i^T \hat{B} \left( \sum_{j=1}^n \hat{v}_j \hat{u}_1^T E\{\delta v_j \delta v_i^T\} \hat{u}_1 \right)
\]

(3.33d)

From the previous analysis, \( \hat{u}_1^T \hat{u}_1 = 1 \) and with equation 3.29a, we derive:

\[
Var[\delta l_i] = \frac{n-1}{n} \sigma^2 + \sigma^2 \sum_{k=2}^m \frac{(\hat{v}_i^T \hat{u}_k)^2}{\lambda_k} + 2\sigma^2 \hat{v}_i^T \hat{B} \hat{v}_i
\]

(3.34a)
$$\sigma^2 \left( \frac{n-1}{n} + \sum_{k=2}^{m} \frac{(v_i^T u_k)^2}{\lambda_k} - 2 \sum_{k=2}^{m} \frac{(v_i^T u_k)^2}{\lambda_k} \right) = \sigma^2 \left( \frac{n-1}{n} - \sum_{k=2}^{m} \frac{(v_i^T u_k)^2}{\lambda_k - \lambda_1} \right)$$ (3.34b)

Equation 3.34b is an extremely interesting result – the variance for the residual is the combination of the constant term $\sigma_i^2 = (n - 1) \sigma^2 / n$ which is dependent on the variance of the original data, and a variable term which is dependent on the location of the data with respect to centroid. This a reasonable result; points further away from the centroid have a greater influence to “tilt” the hyperplane. This also indicates that the residual error distribution is different for each and every point being fitted to the hyperplane!

If the pixels exhibit a low residual, they will all be “close” to the hyperplane and can be considered part of the same surface. As the “endpoints’ move further away from the fitted hyperplane, the increased residual error will increase, and the centroid can be considered as a transition point between two planes. Very large residual errors can be quickly classified as noise. Since each pixel possesses a unique residual distribution error arising from the hyperplane fitting (eigen-distance), the comparison to the overall distribution of nearest neighbors represents a comparison of residual distribution errors against the expected error for a particular threshold value.
3.4 Feature Extraction

The output from the previous module is a binary image with supporting metadata (eigenvector signatures) for each transitional pixel. In the feature extraction module, the well published Canny Edge Detector [Canny, 1986] is utilized to extract the edges (features) from the binary image, followed by application of the equally well established Hough Transform [Duda and Hart, 1972], [Ballard, 1981]. These steps are not strictly essential to the algorithm, but could assist future applications which seek to track specific features as objects. Extraneous pixels are discarded by simple filtering, leaving distinct edges for matching; each pixel of the edge possessing a distinct eigen-signature and \(xyz\) positional triplet.

Two images have now been segmented, and possess extracted features (pixels) with distinct signatures for matching. The challenge is two-fold; to place both images in the same coordinate frame of reference, and secondly, to accomplish this by utilizing only static features. It is worth noting that while the features have been extracted, at this point it remains unknown which pixels are static or dynamic.

3.5 Coarse Matching of Two Coordinate Frames

Matching the extracted features between the two images would still be a considerable computational exercise if approached in brute force. Both the rotational matrix \(R\) and the
translational vector $T$ are unknown; an infinitum of possible combinations exist that could define the transformation defined by equation 2.4. The number of static and dynamic features is unknown in both images; thus it is unknown which features can be utilized to facilitate the transformation. To provide a computationally tractable solution, additional information is required to enable constrained regions for the solution space.

Two critical inputs are available to leverage an improved computational reality. First, the onboard INS can provide a reasonable estimate of attitude and azimuth from the time of first image acquisition to the time of the second image acquisition. The input consists of the raw gyroscope and accelerometer signal, the INS algorithm utilizes these inputs to establish orientation and position. The available interim (coarse) solution from the INS is provided as [Sun, 2009]:

$$[t, d_x, d_y, d_z, \sigma_x, \sigma_y, \sigma_z, \psi, \theta, \phi, \sigma_\psi, \sigma_\theta, \sigma_\phi]$$  \hspace{1cm} (3.35)

Where:

- $t$ is the time tag linking the INS solution to a particular image frame
- $d_x, d_y, d_z$ are the positional offsets from $t_i$ to $t_{i+1}$
- $\sigma_x, \sigma_y, \sigma_z$ are the error estimates associated with the positional offsets
- $\psi, \theta, \phi$ are the orientation offsets from $t_i$ to $t_{i+1}$
- $\sigma_\psi, \sigma_\theta, \sigma_\phi$ are the error estimates associated with the orientation offsets
Provided that the errors are small, equation 2.4 can be re-written as:

$$\tilde{C}_{i+1} = \begin{bmatrix} 1 & -\varphi & -\theta \\ \varphi & 1 & \psi \\ \theta & -\psi & 1 \end{bmatrix} C_i + \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix}$$

(3.36)

Where the ~ indicates that the solution is based upon the estimated matrices (as indicated by the null subscript) from the first epoch $i$ to the next epoch $i+1$.

Recall that the segmented pixels from the first image consist of a set $p = p_s + p_d$, with $p_s$ the (unknown) number of static pixels and $p_d$ the (unknown) number of non-static, or dynamic, pixels. Applying equation 3.36 to each transitional pixel in the set $p$, along with the pixels surrounding them in a 5x5 window, the $p$ features from the initial image are transformed into $\tilde{p}$.

The motion from the coordinate frame $C_i$ to coordinate frame $C_{i+1}$ can result in an offset of pixels between the two images along the edges of the image. This example is simplistic owing to the planar offset; in practice, both rotation and translation must be considered when evaluating “lost” pixels.

The offset regions reflect areas no longer visible or not visible in the image one but visible in image two. These regions are set to zero in the binary image to preclude the matching of features where no matches can exist.
The second constraint, which can be invoked, involves the potential motion of the sensor platform during the time period $\Delta t = t_{i+1} - t_i$. Owing to the high rate of sensor acquisition, the distance and orientation change in the sensor platform is limited for velocities on the order of a few kilometers per hour. Given a conservative estimate for rate of change, the offsets can be reasonably estimated with error limited to a few pixels in the worst case.

3.6 Feature Matching

The available data has been reduced to a subset of features drawn from each image. The features $p$ from the first image have been transformed into the coordinate frame of the next image containing features $q$ with some error owing to the uncertainties associated with the coarse INS solution. Features $p$ are now referenced as $\tilde{p}$; the tilde again indicating the estimated (coarse) transformation of features $p$ into the coordinate frame of the second image. The goal is to refine the transformation such that the error between matching static features $\tilde{p}_s$ and $q_s$ is minimized. The feature matching module accomplishes two simultaneous activities; first, the separation of static features, and secondly the estimation of final perturbations to the estimated rotation matrix $\tilde{R}$ and translation vector $\tilde{T}$ to place the static features $p_s$ in the coordinate frame of $q$ containing static features $q_s$, such that the difference in position between the matching features $\tilde{p}_s$ and $q_s$ is minimized.
Consider the set of static points \( \tilde{p}_s \) as a structure in space. Owing to the initial estimate of rotation and translation from the INS, if a similar set of static point’s \( q_s \) lie within “close” vicinity, then small perturbations to the estimated rotation matrix \( \tilde{R} \) and translation vector \( \tilde{T} \) will minimize the differences between matching features.

Consider the arbitrary transition feature \( p \) located at the \( ij \) pixel; call this \( \tilde{p}_{ij} \), with positional location \( (\tilde{x}\tilde{y}\tilde{z})_{ij} \), and eigenvector signature \( \tilde{\lambda}_{ij} \). The hypothesis is that if feature is static, an equivalent feature \( q_{ij} \) must lie within the small 5x5 window surrounding the target pixel and must possess the same eigenvector signature within some small amount of error \( \varepsilon \).

A critical question relates to the magnitude of the error \( \varepsilon \) between two eigenvector signatures; i.e., at what value should two slightly different eigenvector signatures be considered the same (or different)? Recall that from the segmentation step (reference paragraph 3.3) the threshold for merging pixels was based upon distances derived from the endpoints of the eigenvector. If the eigen-distances are larger than this threshold, the associated pixel was considered a transition pixel (the pixels currently being matched). This threshold serves as the upper limit for matching two transition pixels. The algorithm iterates down from \( \sigma_{\varepsilon} \), (the sigma associated with \( \varepsilon_{\text{min}} \) from equation 3.20) towards zero. In practice, at some point a significant number of transition pixels cease to match. The value \( \beta \) where the matching of transition pixels rapidly changes is called the \textit{threshold} value. The sigma level before this value (where many transition pixels match) is called
the *pre-drop* value, and the sigmas greater than the threshold value (where fewer new transition pixels match) is called the *drop* value. These values are not known in advance, but determined on an iterative basis the count of matching transition pixels at a given sigma distance. The threshold value to control the matching process can be assumed to be:

\[
\sigma_{pd} < \sigma_t < \sigma_d
\]  

(47)

Where the subscripts are defined as:

- \(pd\) – pre-drop value
- \(t\) – threshold value
- \(d\) – drop value

Ideally, the algorithm would prefer to utilize only the best matching static pixels (features); realistically, the algorithm requires a number of well distributed points to obtain a robust solution for position. A simple solution is to evaluate the number of available points and their associated dilution of precision (DOP) at both \(\sigma_{pd}\) and \(\sigma_d\) and select the best condition based upon both criteria.
3.7 Determination of Position

The output of the previous step is a set of matching static pixels (features) with known $xyz$ coordinates in two different coordinate frames. Recovering the transformation between these coordinate frames may be accomplished by absolute orientation estimation. A closed form solution to this problem has previously been derived by Horn [1987]; details of the method are provided in Appendix A. Applying the method suggested by Horn permits the calculation of the least squares solution of the scale, rotation, and translation parameters $\tilde{s}, \tilde{R},$ and $\tilde{T}$ required to transform features from the first image into the coordinate frame of the second image, such that the residual error in eigenvector signature between matching static features is minimized. These scale, rotation, and translation parameters, along with associated error estimates, which can then be passed to the INS as a rectified position and attitude update to the original INS estimated solution.
CHAPTER 4
OVERVIEW OF SUPPORTING TECHNOLOGIES

This chapter outlines the technologies which are utilized to acquire the 3D data during a variety of experimental data acquisitions which will be discussed in Chapter 5. These include 3D Flash LADAR, 3D Kinect, 3D sonar, the supporting Inertial Navigation System (INS) and associated Extended Kalman Filter (EKF). Note again that I distinguish between the term 3D Flash LADAR (referring to the CSEM device) and the 3D Kinect (Microsoft device).

4.1 3D Flash LADAR

Theory

Flash Laser Distance Array (LADAR) is a relatively new technology arising from the doctoral dissertations of Spirig [1997] and Lange [2000]. The methodology is based upon the time of flight for a light beam. The Flash LADAR unit emits energy in the eye safe laser portion of the electromagnetic spectrum; the resulting waveform is absorbed, refracted, or reflected by 3D objects in the environment. Waveforms which are reflected
pass through the focal lens of the LADAR camera and the energy is then captured on a 
CCD array. Internally, a waveform identical to the emitted energy is produced on a 
simultaneous basis. The resulting offset between the reflected energy waveform and the 
base waveform represents a phase offset in time. Since the speed of light is well known, 
the distance traveled by the emitted wave is twice the distance between the camera and 
the reflecting object.

The initial pulsed waveform emitted is sinusoidal in theory, although some departures 
from this basic assumption will occur via natural variances inherent to the propagation of 
the electro-optical device. The emitted signal is modulated in amplitude; the 3D 
environment modulates the signal in phase, and a portion of the energy is reflected back 
to the sensor and retrieved. The amplitude and phase of the captured signal can be 
retrieved by the synchronous demodulation of the captured signal within the detector. The 
demodulation is accomplished by cross-correlation of the received (captured) signal with 
the original modulated signal. The cross-correlated function is measured at discrete 
temporal instances (phases) to enable the determination of the phase offset.

While this explains the 3D Flash LADAR conceptually, the actual implementation of the 
technology is complex enough to warrant additional exposition. The interested reader is 
referred to Lange [2000] for details related to the technology.
Technology

In the 3D Flash LADAR camera, each sensor (pixel) on the array must have the capability to generate the noted amplitude and phase values based upon a received signal. The 3D Flash LADAR sensors are photovoltaic, consisting of a region where conductivity is due to electrons and a separate region where conductivity is due to holes. When optical energy is received by the sensor, holes are generated in the latter material, creating a voltage that can be measured (sampled) as a waveform [Ready, 2008].

The physical sensors must accomplish four primary tasks [Lange, 2000].

1. Energy conversion
   - Conversion of photons into electron hole pairs
2. Fast separation
   - Avoidance of temporal blurring of the received time critical information
3. Repeated, noise free addition
   - Improvement of the signal to noise ratio (SNR) and insensitivity to spurious frequencies
4. In-pixel storage
   - Storage of acquired sampling points
A common sensor utilized to accomplish these demands is a charge-coupled device (CCD). In simplest essence, the CCD consists of a sensor and a photogate; the photogate exists to time the passing of material (in these instance photons) past a given location. By establishing an array of such sensors, the 3D Flash LADAR camera can recover range information from the environment subject to limitations in the power of the emitted signal, the sensitivity of the CCD element, and the field of view permitted by the optical (lens) elements acquiring the reflected signal from the environment.

The camera is therefore a complex system to accomplish a simple function; emit a light source, capture the reflected energy, and utilize the phase delay to determine distance based upon TOF.

4.2 3D Kinect

Theory & Technology

Near the end of this dissertation, the SPIN laboratory at The Ohio State University acquired a Kinect sensor for experimental purposes. Kinect is a proprietary technology owned by Microsoft Corporation and relatively few details are available regarding the hardware and software underlying this device. The Kinect unit consists primarily of a 640x480 pixel array of infrared sensors [Newcombe, et al., 2011], [Shotton, et al., 2011], which appear to acquire 3D ranging data in a manner virtually identical to the 3D Laser
Ranging Camera discussed in Section 4.1 above [Primesense, 2010]. In addition to the laser ranging, the unit also acquires an intensity image, akin to a standard digital camera image, on a simultaneous basis with the range data [Primesense, 2010]. The operational range of the sensor is 0.8–3.5 meters, with a range resolution of 1 cm at 2m distance [Primesense, 2010], [Toth, 2012].

4.3 3D Sound Ranging

Theory

3D sonar is similar to 3D Flash LADAR in the emission of a wave form (acoustic sound) followed by the recovery of wave forms reflecting from features in the environment. The processing methods and techniques are different and require further detail and discussion to ensure an awareness of acoustic imagery.

The emission of a sound wave in a fluid is to esonify the environment. The resulting reflections are received from multiple directions by a 2D array of receiving sensors. The sensors recover the phase, time, and amplitude of the received signal. A variety of algorithms exist to transform the acquired signal characteristics into an acoustical image. One advanced methodology is transformation by means of spectral decomposition [Hansen, 2002], [Hansen, 2008], [Cunningham, et al., 2008].
Beamforming is the combining of acoustic signals from multiple transducers into beams, followed by the detection of the energy in each beam [Sutton, 1979]. The combined signals can be generated and scanned over a volume; the received energy is measured and displayed in an appropriate geometry. In holographic acoustic imagery, the beams are processed simultaneously. The sound (pressure) spatially sampled by a 2D array of sensors is immediately converted to a stable set of direct current values called a hologram [Sutton, 1979].

A more recent holographic method has been developed by Hansen [1990] and expanded by Hansen [1992, 2002, 2008], and [Hansen, et al., 2005] and [Murino and Trucco. 2000]. Many thanks are extended to Dr. Hansen (see acknowledgements) for his support in providing information related to this methodology. This methodology utilizes a specialized matrix approach to the inversion process which significantly simplifies the recovery of the acoustic image. For details related to the implementation of this method, the reader is referred to Hansen [1992].

**Technology**

The Coda Echoscope is a proprietary product of CodaOctopus, which owns Omnitech AS of Bergen, Denmark. Details related to the equipment are not readily available owing to the protected and sensitive nature of the product. The following information was kindly
provided by Mr. William Woodward of UrsaNav, Inc., the sole US supplier of Coda Echoscope technology [Woodward, 2010], [Coda Octopus, 2009].

The Echoscope 3-D sonar is unlike conventional multi-beam or scanning sonar that typically use post-processing to combine scanned images and creates a 3-D image. Rather than using a narrow fan of acoustic beams, the Echoscope simultaneously forms over 16,000 acoustic beams to fill a 3-D volume. This process is accomplished in real-time, within the Echoscope, at a rate of up to 12 frames per second. The images provided by the Echoscope, like those shown in Figure 4.1, appear to be taken by an underwater camera, not sonar. With a frame rate of up to 12 frames per second, the Echoscope is capable of capturing moving objects, such as dolphins and divers, and displaying them in a format that compares to a video camera.
The Echoscope accepts real-time navigational data, including heading, pitch, roll, acceleration, position, and speed from the INS. The geo-referenced ping data allows the creation of overlaid mosaic datasets adding each ping to the 3-D scene in real time to build an ultra-high-definition map on the fly, as shown in Figure 4.2. The colored scale bar on the left hand side of both figures is in meters. The scale on the right hand edge of 4.2 represents the orientation of the sensor head in space and is not relevant to the ranging measurement. This image is for illustrative purposes only; details related to the exact setup during acquisition are not available.
Each output ping from the Echoscope (shown in Figure 4.3) ensonifies the entire $50^\circ \times 50^\circ$ volume of water to a maximum range of approximately 150 meters. The sonar transducer emits the sonar waveform at 375 Hz; reflected sound waveforms are recovered by the sensor array. The tilt sensor is utilized to supply orientation information to support image mosaicking of successive sonar imagers. Phased-array technology is used in the planar receive array to form a large number of beams in both the horizontal and vertical directions; allowing range and target strength data to be acquired for each point on a
128x128 grid that represents the volume being ensonified. This is illustrated in Figure 4.4.

The Echoscope transmit array forms 128 beams vertically and 128 horizontally; the receive array is composed of 48 x 48 channels. This configuration produces a 50° x 50° field of view, which is decomposed into 16,384 beams, each with a horizontal and vertical width of around 0.4 degrees. Echoscope performance specifications are outlined in Table 4.1.
The Echoscope sonar head houses the sonar projectors and the receive array panel. The beam-forming, focusing, and first-pass seabed detection is carried out in the sonar head. Data is output from the sonar head over a copper connection cable to the Data Interface Unit (DIU). Power (24 VDC) for the sonar head is supplied via the same cable, as well as from the DIU. The Echoscope physical specifications are outlined in Table 4.2 and interface specifications are outlined in Table 4.3.
Figure 4.4. An Example of One ‘White’ Ensonfication Grid point.
<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>375 kHz</td>
</tr>
<tr>
<td>Number of beams</td>
<td>128 x 128 (16,384 total)</td>
</tr>
<tr>
<td>Maximum range*</td>
<td>150 m (500 ft)</td>
</tr>
<tr>
<td>Minimum range</td>
<td>1 m (3 ft)</td>
</tr>
<tr>
<td>Range resolution</td>
<td>3 cm (1.2”)</td>
</tr>
<tr>
<td>Ping rate</td>
<td>Up to 12 Hz</td>
</tr>
<tr>
<td>Angular coverage</td>
<td>50° x 50°</td>
</tr>
<tr>
<td>Beam spacing</td>
<td>0.39°</td>
</tr>
</tbody>
</table>

Table 4.3. Echoscope Performance Specifications, courtesy of UrsaNav, Inc. and Coda Octopus

* Actual range is dependent on pulse length, target size, and target strength.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions (h x w x d)*</td>
<td>380 mm x 300 mm x 160 mm (15” x 11.8” x 6.3”)</td>
</tr>
<tr>
<td>Weight in air</td>
<td>22 kg (48 lb)</td>
</tr>
<tr>
<td>Weight in water</td>
<td>12 kg (26 lb)</td>
</tr>
<tr>
<td>Power consumption</td>
<td>3 to 6 A at 24 VDC</td>
</tr>
<tr>
<td>Depth rating</td>
<td>600 m</td>
</tr>
</tbody>
</table>

Table 4.2. Echoscope Physical Specifications, courtesy of UrsaNav, Inc. and Coda Octopus

* Excluding connectors

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sonar head control unit</td>
<td>Serial RS232 and Ethernet</td>
</tr>
<tr>
<td>Control unit to top end PC</td>
<td>Ethernet</td>
</tr>
</tbody>
</table>

Table 4.3. Echoscope Interface Specifications, courtesy of UrsaNav, Inc. and Coda Octopus
4.4 Inertial Navigation System

Theory

The Inertial Navigation System (INS) is a system comprised of two primary elements. The first is the inertial measurement units (IMU’s) which utilizes gyroscopes to measure changes in angular rate, and accelerometers to measure changes in linear acceleration. The second is the algorithm utilized to derive position and orientation from the IMU data. By integrating the changes (in acceleration and angular rate) twice, it is possible to recover the change in position and attitude. By sampling these changes (and correspondingly integrating the associated deltas) it is possible to generate high accuracy position and attitude information [Jekeli, 2000]. However, the INS is subject to a number of errors which steadily degrade the accuracy of the solution for location and orientation. GPS positional updates are utilized to correct these systemic errors; when GPS is not available, the goal is to provide a substitute set of information to enable ongoing corrections to the navigation solution.

In the context of this dissertation, the INS system is treated as a “black box” technology which accepts updates in position and orientation derived from 3D ranging imagery. As such, full details related to INS systems will not be covered here. The interested reader is referred to Jekeli [2000].
Implementation

Implementation of the EKF in this research is accomplished by leveraging the AIMS-Pro™ software system. The earlier version of AIMS-Pro™ system was developed internally at The Center for Mapping at The Ohio State University, Columbus Ohio. It was redeveloped and re-implemented by the Satellite Positioning and Inertial Navigation (SPIN) Laboratory at The Ohio State University. AIMS-Pro™ (hereafter abbreviated to APro) is an advanced EKF capable of integrating multiple sensors within the base GPS/INS integration schema [Grejner-Brzezinska et al., 1998], [Grejner-Brzezinska, 1999], [Grejner-Brzezinska et al., 2011].

The AIMS Pro is capable of GNSS stand-alone post-processing (real time (RT) applications potential), precise positioning, and velocity determination, as well as the integration of Pseudo Lites (PL), Terrestrial Laser Scanners, and other sensors. In GNSS precise positioning and velocity determination, GPS system (of data in two frequencies) is mainly supported; GLObal'naya NAvigatsionnaya Sputnikovaya Sistema (GLONASS) and the European GNSS system (GALILEO) support is planned in future releases. For INS, the navigation grade, tactic grade and consumer grade IMU technologies are supported via different dynamic models. TLS sensors include terrestrial line laser scanner and Flash LADAR.
AIMS Pro can provide 7 navigational parameters in ECEF coordinate system including time, 3D position, velocity and 3D attitude with accuracies dependent on the quality of input data. The relative positioning accuracy in local coordinate system can reach a few centimeters for continuous GP and a tactical (or navigation) grade IMU [Sun, 2007].
CHAPTER 5
EXPERIMENTAL DATA ACQUISITION

5.1 3D Flash LADAR

Equipment and Setup

In conjunction with the Air Force Research Laboratories, the Center for Mapping at The Ohio State University developed a data collection process involving multiple sensors, including a CSEM/Mesa Flash LADAR camera. The purpose of this effort was the acquisition of 3D LADAR data, along with GPS/INS data, to support testing of algorithms developed to exploit 3D ranging imagery for the purpose of navigation in GPS challenged environments. A summary of the sensors appropriate to this dissertation is shown in Table 5.1, while the experimental setup is indicated in Figure 5.1. The equipment is mounted to a backpack frame which enables the system to be carried by a single individual. A follow cart contains the personal computer, power supply, and data recorders (Figure 5.2). All system data inputs were tagged with reference to the GPS time to permit fusion of the information during analysis and testing.
For the purposes of this dissertation, only the 3D Flash LADAR and INS (HG1700) system elements are relevant to the data collection. The remaining sensors were utilized for other experiments conducted in tandem with the 3D Flash LADAR experiment, but unrelated to this research.

<table>
<thead>
<tr>
<th>No.</th>
<th>Device</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MESA/CSEM 3000</td>
<td>Flash LADAR Camera</td>
</tr>
<tr>
<td>2</td>
<td>Honeywell HG1700</td>
<td>Inertial Navigation Unit</td>
</tr>
</tbody>
</table>

Table 5.1. 3D Flash LADAR Test - Equipment Specifications

Details related to IMU are documented at Honeywell [Honeywell, 2010]. The Honeywell HG1700 is classified as a tactical grade IMU which utilizes a Fiber Optic Gyroscope (FOG) system in conjunction with a triad of orthogonal accelerometers. Details related to the performance characteristics of the unit are provided in Table 5.2 as reported by Brown [2004].

92
Figure 5.1. 3D Flash LADAR - Equipment Setup

Reference Table 5.1 for item descriptions

Figure 5.2. 3D Flash LADAR - Equipment & Cart Collecting Data
To permit the validation of the experiment, it was necessary to establish a network of ground truth points. First, a known geodetic reference point external to the Center for Mapping (CFM) at The Ohio State University was utilized to establish a long term GPS positional fix. Second, a traverse was then established inside the building based upon the original base point, using a total station. Finally, internal points along the traverse were measured from the surveyed points.

A visual representation of the traverse, survey points, and location of the wall targets along the path is provided in Figure 5.3 and 5.4, respectively. Survey points A, B, C, D, F,

<table>
<thead>
<tr>
<th>HG1700 Units Specification</th>
<th>Gyroscopes</th>
<th>Accelerometers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Range</td>
<td>± °/s</td>
<td>± g</td>
</tr>
<tr>
<td>Scale Factor Accuracy (1 σ)</td>
<td>ppm</td>
<td>ppm</td>
</tr>
<tr>
<td>Scale Factor Linearity (1 σ ±800°/s)</td>
<td>ppm</td>
<td>ppm</td>
</tr>
<tr>
<td>Bias (1 σ)</td>
<td>°/hour</td>
<td>mg</td>
</tr>
<tr>
<td>Angular Random Walk (max)</td>
<td>°/√hour</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.2. Honeywell HG1700 Specifications
H, and I were established as the primary traverse, while the remaining points were measured with reference to these controls. The precision in the east/north directions was 0.01 meters and 0.005 meters in the vertical.

Figure 5.3 CFM Experimental Area with Target Locations
Data Acquisition

A series of data collection activities were conducted from August 21st to August 26th, 2007. The data collection processes consisted of the following:

- The mobile pack was located at the GPS “base point” located in front of the Center for Mapping. This position was maintained for a minimum of 5 minutes to establish a solid GPS positional fix.
• Several exterior loops (as per Table 5.3) were completed in the outside parking lot of the Center, followed by a return to the “base point”.

• Several interior loops were completed inside the building (as per Table 5.3), either in a dynamic state (continuous walking), or a semi-static state (stopping at each surveyed point).

• In different runs, non-static features were introduced during the loop, typically consisting of a person (or persons) entering the field of view. In other cases furniture was shifted during the acquisition process. Instances where non-static features were present are noted in the last column of Table 5.3.

A summary of the experimental runs is shown in Table 5.3. In experimental run #11, the camera was kept at a fixed location and rotated 180° counter-clockwise, 360° clockwise, and finally 180° counter-clockwise. An example trajectory is shown in Figure 5.5
<table>
<thead>
<tr>
<th>Run #</th>
<th>Exterior Loops</th>
<th>Interior Loops</th>
<th>Static - S</th>
<th>Dynamic - D</th>
<th>Non-Static Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>D</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>D</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>S</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>D</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td>D</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>D</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>2</td>
<td>D</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>2</td>
<td>D</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>1</td>
<td>S</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1</td>
<td>S</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>-</td>
<td>-</td>
<td>S &amp; D</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.3. Summary of Experimental Runs
5.2 3D Sonar

Equipment and Setup

The experimental setup for 3D Sonar data collection is shown in Figure 5.6, which shows the Coda Octopus Underwater Inspection System (UIS™). The system consists of the 3D sonar system (Echoscope) as shown in the upper right hand image, two GPS antennae and a inertial measurement unit (IMU) as shown in the top left image, and the data integration system (not shown), which incorporates both the inertial navigation system
(GPS/INS) and geo-referencing of all 3D Sonar data points. The GPS/INS system enables the geo-referencing of 3D sonar pings collected from surface operations (the sonar hanging from the side of a surface vessel). In underwater operations the GPS ceases to provide information owing to the lack of signal, while the INS provides coarse “frame to frame” updates, in similar fashion to the INS described in the previous section on 3D LADAR.

The UIS™ system utilizes Novatel GPS-701-GGL and GPS-702-GGL antenna to acquire positional information from the GNSS and GLONASS satellites. The 701 unit acquires information on the L1 frequency, while the 702 unit acquires both L1 and L2 frequencies. Detailed information related to the antenna is available from Novatel [Novatel, 2010]. The system is capable of positional accuracy on the order of <10 mm, with drift error of <1.0 ppm. The inertial navigation system contains an inertial measurement unit (IMU) which is the Octopus F180R remote inertial measurement unit; specifications and details are reflected in Table 5.4 [Coda Octopus, 2010]. Details related to the GPS/INS integration in the F180R are identical to the system reported by Ford, et al [2003].
Figure 5.6. Coda Octopus Underwater Inspection System (UIS™)

Picture courtesy of UrsaNav, Inc, and Coda Octopus

[Coda Octopus Products, 2009]

<table>
<thead>
<tr>
<th>F180R Units Specification</th>
<th>Gyroscopes</th>
<th>Accelerometers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Units</td>
<td>± °/s</td>
<td>± g</td>
</tr>
<tr>
<td>Specification</td>
<td>30</td>
<td>110</td>
</tr>
</tbody>
</table>

| Gyroscopes | Operating Range | ± °/s | 30 |
| Scale Factor Accuracy (1 σ) | ppm | <2500 |
| Scale Factor Linearity (1 σ ±800°/s) | ppm | N/A |
| Bias (1 σ) | °/hour | <15 |
| Angular Random Walk (max) | °/√hour | N/A |

| Accelerometers | Operating Range | ± g | 110 |
| Scale Factor Accuracy (1 σ) | mV/°/sec | 18.2 |
| Scale Factor Linearity | % of Scale | 1 |
| Bias (1 σ) | °/sec | 0.1 |
| Vertical Random Walk (max) | °/Hour | 0.2 |

Table 5.4 Octopus F180R Specifications [Coda Octopus, 2010]
Experiment Area

Experimental data to test the applicability of the algorithm developed in Chapter Three to 3D sonar was collected by UrsaNav, Incorporated. The UIS™ system outlined previously was deployed in the Panama Canal in the near vicinity of 8°59’39.06”N, 79°35’22.52”W. A satellite view of the region is illustrated in Figure 5.7; Figure 5.8 shows the area from the water level, and Figure 5.9 reflects the mosaicked result of multiple acoustic images acquired by the UIS™. In figure 5.9, the colored scale bar on the left hand side of both figures is in meters. The scale on the right hand edge of figure 5.9 represents the orientation of the sensor head in space and is not relevant to the ranging measurement. This image is for illustrative purposes only; details related to the exact setup during acquisition of this mosaic series of images are not available.

The data were collected during a traverse down the noted region, with all data time tagged to assure interoperability. All data acquired by the UIS™ are collected in a proprietary .xtf format. Details related to the .xtf format can be located in the eXtended Triton Format (XTF) document [Triton, 2006]; the current revision at time of writing is Revision 26.
Figure 5.7. 3D Sonar Survey Region, Panama Canal

Picture courtesy of UrsaNav, Inc., 2010
Figure 5.8. 3D Sonar Survey Area, Water Level, Panama Canal

Picture courtesy of UrsaNav, Inc., and Coda Octopus Products, 2010
Details related to the Coda Octopus implementation of the XTF standard are proprietary and cannot be released in forum of this document; germane to this study is the presence of GPS/INS data, geo-referenced 3D sonar data, and corrections for dynamic motion of the vessel owing to motion in the water.

Near the end of this dissertation, UrsaNav kindly donated a second (smaller) 3D sonar survey completed in Canada. The sonar survey area is indicated in Figure 5.10 below. The data acquisition process was identical to the Panama Canal study previously discussed and provides a second demonstration of the 3D navigation algorithm in tandem with 3D sonar data.
For each study region, the reference position of the 3D Sonar unit was determined by means of the Coda Echoscope system, which leverages a GPS/INS integrated solution to establish positioning for the unit. The reference solutions for each experiment are shown by the red line in Figures 5.11 and 5.12 respectively. Afterwards, a simulation was run
with the GPS data removed from the GPS/INS solution whenever a position/orientation solution was available from the 3D ranging imagery algorithm outlined in Chapter 3. The 3D ranging imagery position/orientation information was utilized in lieu of the GPS; these positions are marked in Figures 5.11 and 5.12 with blue crosses. In both figures, the xy-axes represent the trajectory position in meters, while the z-axis shows the depth from the sensor head to the bottom of the canal.

The survey traverse for the Panama Canal 3D sonar survey is plotted in Figure 5.11 which reflects both the actual reference location based upon the GPS/INS solution (dark red line) and the positions established via 3D sonar imagery (blue crosses). Details related to the established positions (blue crosses) will be discussed in Chapter 6; the essential point for explanation of the data acquisition environment is to note the extent and motion involved. In totality, the total traverse measured approximately 200 meters with a platform velocity of ~3 meters/second.

The Canadian survey is plotted in Figure 5.12. Total traverse is approximately 80 meters over a temporal period of three and a half minutes. The platform was “drifting” at an approximate rate of 0.4 meters/second. The details related to the established positions will also be discussed in Chapter 6.
Figure 5.11. Panama Canal Survey Traverse and Results
5.3 3D Kinect

**Equipment and Setup**

The 3D Kinect equipment setup was developed and implemented by the SPIN Laboratory at The Ohio State University in a fashion reminiscent of the 3D LIDAR equipment setup, as can be seen by comparing Figure 5.13 (3D Kinect Setup) with early Figures 5.1 and 5.2 (3D LADAR). From the viewpoint of this dissertation, the key elements of the setup are the Kinect Sensor and the HG1700 IMU (reference Table 5.5).

Unfortunately, the HG1700 IMU data were not acquired during the data acquisition process and the substitute micro-electronic measurement sensor (MEMS) IMU was of such poor quality that the drift within a few seconds of GPS signal loss was so great as to render the information useless for the purpose of providing coarse adjustment between range images.

To overcome this limitation, the intensity images acquired simultaneously with the range data by the Kinect unit [Primesense, 2010] were utilized to provide this crucial information. The two intensity images were processed utilizing the Scale Invariant Feature Transform (SIFT) algorithm [Lowe, 2004] to identify matching features between the pair of 2D intensity images.
This method has several drawbacks as previously discussed in Chapter Two, Section 2.6; namely, the matching process is not unique, and the SIFT algorithm loses spatial positioning capabilities to achieve the repeatability required for establishing a navigation solution. As the intensity images were the only possible sources of coarse adjustment
data, the algorithm was leveraged to facilitate processing. The matching SIFT features were correlated with the associated range pixel measurements, and these range measurements were utilized in Horn’s Method [Horn, 1987] to provide the coarse position and orientation between consecutive range image frames. The 3D range matching algorithm described in Chapter 3 then proceeds normally as described. The schematic overview of the method is illustrated in Figure 5.14.

Figure 5.14 Schematic of Adapted 3D Range Matching Algorithm Used for Kinect Dataset
The use of SIFT to provide the initial matching between the images entails the acceptance of several critical issues, beyond the limitations discussed in Chapter 2. First, since the SIFT algorithm is matching 2D features on the intensity image; there is no guarantee that the matched features represent static elements in the field of view. As an example, SIFT can easily “match” the logo on a shirt worn by a moving person; since the input data will include the position of non-static elements, the resulting coarse adjustment may possess very large biases (in position). As the magnitude of these biases increases, the ability to constrain the search space may be infeasible, resulting in either the inability to generate eigenvector matches (worst case) or a longer search time (best case). Since the 3D Range Matching Algorithm checks the two range images for consistency before the matching process begins, this can be largely mitigated in implementation. Secondly, the returned locations for SIFT features are non-integer, thus the correlation to the range pixel image will inherently possess an error of ±1 pixel (row and column). The impact of this error is that range pixels utilized to facilitate the coarse position/orientation may in fact not be correct, the “correct” range pixel to be matched may not be the one selected. This will result in larger errors during the initial (coarse) adjustment process. Third, the uncertainty of the coarse adjustment is not known, so a-priori estimates of the error ellipse must be made to establish the eigenvector search space. The size and extent of these error ellipses is not defined on-the-fly by the data, which reduces one of the key elements of the algorithm outlined in Section Three. Fourth, conditions occur where the intensity features have no associated range measurement. This can result from the feature being out of range of the ranging sensor; other environmental issues (such as bright lighting) may also
cause the loss of range data for a given intensity feature. This reduces the effective use of SIFT features for coarse alignment. However, using the intensity images does demonstrate the ability of the 3D Range Matching algorithm to generically utilize coarse adjustment information and refine the result to provide a navigation solution.

Experiment Area

The data were acquired at The Ohio State University Supercomputing Center and entailed trajectories inside the corridors of the facility. The two trajectories are illustrated in Figure 5.15 below; the trajectory processed for this dissertation is the North-South (vertical dashed line in the figure).
A follow up study conducted at a different facility was completed using the same platform and methodology. In this study, a complete traverse was completed indoors forming a “box” or square trajectory, which returned to the original entrance point. A plot of the trajectory results is provided in Figure 5.16 below.
It is critical to note in both instances that survey data (reference positions) were not provided as part of these data sets. While the ability of the algorithm to achieve frame to frame positioning is accomplished, the accuracy of the resulting positions cannot be definitively evaluated, although some crude estimates are possible, as will be discussed in Chapter 6.
CHAPTER 6

Application of the NAVRI Algorithm to the Experimental Data and Results

In this chapter the NAVRI algorithm discussed in Chapter 3 is demonstrated on several types of 3D ranging datasets, including 3D Flash LADAR, 3D Sonar, and 3D Kinect based acquisitions.

To begin, I restate the highlights of the algorithm discussed in Chapter 3. In brief summary, the algorithm first utilizes the distribution of nearest neighbor distances (NND) to segment surface edges from two temporally spaced images. After the segmentation phase, a minimum edge consists of three adjacent pixels from the $jth$ column. The center pixel is the edge; the above and below pixels are the adjacent neighbors, and each of the three pixels has an associated $xyz$ triplet. The objective is to locate a pixel in the second image which matches the first edge pixel, as shown in Figure 6.1. What we seek is the function involving scale, rotation, and translation $f(s, R, T)$ which, if the edge pixels are matching, would place the pixels from both images into the same coordinate frame as the second. When the function $(s, R, T)$ has been refined, the position and orientation results, along with associated error estimates, are returned to the Extended Kalman Filter as a substitute for the GPS positional information.
Resolving the function \((s, R, T)\) is accomplished by first acquiring additional information from an external sensor, such as an IMU. This information (processed by means of the INS) provides an initial estimate of the desired function \(f(s, R, T)\) but also reflects uncertainty, reflected by estimates of positional and rotational error provided along with the mean values for translation and rotation. It is therefore possible to constrain the solution space; if the edge pixel from the first image is transformed to the coordinate system of image two, and if the edge pixel in image two matches this pixel, the corrected position for each of the three pixels from image one must lie within these error ellipses, as shown in Figure 6.2, and the eigenvectors of the edge pixel (as calculated from the three pixel set) should be identical to the eigenvector signature of the edge pixel in image two.

Figure 6.1 Edge Matching Between Images
By randomly selecting $xyz$ coordinates within the boundary of the error ellipses, it is possible to derive multiple possible eigenvector signatures (the red arrows in Figure 6.3), each defining the location of the edge pixel from image one in a coordinate frame approximating the second image coordinate frame. By choosing the eigenvector of the image one edge pixel which best matches the eigensignature of edge pixel of the second image (the purple arrow in Figure 6.3); the functional $f(s, R, T)$ can be refined to provide the best possible match. If an eigenvector match cannot be established, the hypothesis is that the edge pixels do not match.

Figure 6.2 3D Error Ellipses and Eigenvector in the Image Two Coordinate Frame
By applying the $xyz$ coordinate shifts to every edge pixel being considered for matching, it is possible to find the coordinate shift which creates the largest number of matching edge pixels in the global sense. As noted in Chapter 2, Section 2.5, this implicitly assumes that the number of static edge pixels outnumber the dynamic edge pixels; various methodologies (as discussed in 2.5) can be utilized to mitigate the instance where dynamic pixels outnumber static pixels.

After the maximum number of matching edge pixels has been identified (which globally minimizes the difference between eigenvector signatures), the $xyz$ coordinates of all “matching” edge pixels are utilized in Horn’s Method [Horn, 1987] to establish the refined functional $f(s,R,T)$ which best transforms image one pixels into the coordinate frame of image two.
This evaluation shows that the only variable defined \textit{a-priori} is the number of possible eigenvector solutions to be generated for matching. All other thresholds and variables are defined dynamically on-the-fly based upon statistics derived from the 3D ranging data.

What then is the impact of changing the number of possible eigenvector solutions? To investigate, the data from the 3D Flash LADAR experiment was analyzed. In specificity, trajectory #10 (Reference Table 5.3) was executed with the algorithm multiple times, the sole change being revising the program for the number of randomly generated eigenvector solutions for values \( n = \{5, 10, 20, 30\} \) and reviewing the changes to the program outputs. With some initial surprise, it was found that \textit{no} meaningful (to three decimal places) changes occurred between these trials.

The reason for this lack of sensitivity to the parameter is three-fold: first, the error ellipses provided by the INS during the 3D Flash LADAR (MESA/CSEM) are small (on the order of the millimeter level), secondly, the \( xyz \) coordinates of “matching” edge pixels are used for determining the functional, not the eigenvectors, and third, the volume of matching edge pixels is quite large (several thousand) providing a highly redundant solution set to the Horn method.

Table 6.1 reflects the small magnitude of error ellipses provided from the INS during the 3D Flash LADAR data acquisition; the translation errors are on the order of 1 centimeter or less, while the rotation errors are less than 0.1° in all axis. The generation of multiple
eigenvector solutions within the constraint for small error ellipses creates nearly exact duplicates, and as such, the difference between the potential solutions is slight.

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation Mean Error (m)</td>
<td>0.013</td>
<td>0.013</td>
<td>0.012</td>
</tr>
<tr>
<td>Translation Std Dev (m)</td>
<td>0.005</td>
<td>0.005</td>
<td>0.005</td>
</tr>
<tr>
<td>Rotation Mean Error (°)</td>
<td>0.017</td>
<td>0.019</td>
<td>0.075</td>
</tr>
<tr>
<td>Rotation Std Dev (°)</td>
<td>0.069</td>
<td>0.080</td>
<td>0.126</td>
</tr>
</tbody>
</table>

Table 6.1 Magnitude of Error Ellipses Provided by INS during 3D Flash LADAR Run #10 (reference Table 5.3)

Figure 6.4 shows the number of available pixels for matching for each frame. The total number of pixels available with the CSEM Flash LIDAR is 144x176 = 25344 pixels, thus the edge pixels represent a smaller subset of ~10,348 pixels on average, yet still provide considerable redundancy.

Since relatively few edge pixels are dropped as a result of the eigenvector matching process, the sheer number of matching edge pixels provides great redundancy for the final derivation of the $sRT$ function using Horn’s method. Since Horn’s method is a least squares minimization process, the final positions (frame to frame) are unchanging regardless of the eigenvector iterations utilized.
If a less accurate IMU were utilized to support navigation efforts, this could potentially be source of concern, and a study would be required to identify the optimal number of iterations (of eigenvector solutions) for a given size of error ellipse. Since the IMU’s utilized for both the LADAR and SONAR experiments were of high quality, further investigation is unnecessary with respect to demonstrating the algorithm.

We tested the algorithm on several 3D ranging datasets, one acquired from Flash LADAR, one acquired by Kinect, and two acquired from Sonar. The remainder of this chapter is organized as follows:
• The first section illustrates the frame by frame separation of static and non-static features. This section also demonstrates the determination of motion “frame to frame” as compared to the reference INS solution.

• The second section demonstrates the trajectory solution using 3D SONAR dataset and compares the result to a reference GPS/INS trajectory solution.

• The third section demonstrates the trajectory solution using the 3D KINECT dataset.

One original objective of this dissertation research was to fully integrate the NAVRI solutions within the schema of a tightly coupled EKF, blending GPS/INS and NAVRI solutions (when GPS is available) and INS/NAVRI solutions (when GPS is challenged). The development of the tightly coupled EKF was outside the scope of research to develop the NAVRI algorithm and was not part of this dissertation effort. Since the desired EKF is not currently available, this objective was not realized in the course of this research.
6.1 Analysis of the NAVRI with 3D Flash LADAR Data

In this chapter we first analyze the ability of the NAVRI algorithm to separate static and non-static features from a time series of 3D Flash LADAR images. Next, the impact of position and orientation updates from the NAVRI algorithm is analyzed by comparing the results with INS data only and the results of INS+NAVRI.

Test for Separation of Static and Non-Static Features

Figure 6.5 illustrates a sequential series of 3D Flash LADAR frames acquired in run #10 (reference Table 5.3). In frame 6.5a, the hallway is empty, in frames 6.5b to 6.5d, a person moves into the field of view and proceeds down the hallway. Our goal is to examine (visually) the extracted non-static features from this image series and determine if the algorithm can successfully separate the static and non-static range points (pixels).

The algorithm acquires two successive images and determines the nearest neighbor distribution (NND) for both images. A simple $F$-test is utilized to determine whether the two images can be considered as samples from the same continuous distribution. In the instance of frame 6.5a to 6.5b, this is not the case, and the algorithm would normally return an inconclusive result to the EKF. In this instance, the $F$-test result is bypassed to enable the verification of static/non-static pixel separation. The remaining frames 6.5b to 6.5c, and then 6.5c to 6.5d, the $F$-test holds; the hypothesis that the two images are
similar enough to be considered samples of the same distribution is accepted and the algorithm proceeds to execute the SFEM process as described in Chapter 3.

The eigenvector signatures are derived for all pixels from the \((2, 2)\) to the \((n-1, m-1)\) pixel, i.e., all pixels except the outermost edges. This is completed for both images. Based upon the eigenvector signatures, the pixels in each image are separated into surface, edge, and noise pixels as explained in Chapter 3.

Next, the NAVRI algorithm requests the EKF for a coarse estimate of motion from the previous frame to the current frame based upon the HG1700 IMU data. The NAVRI algorithm then generates thirty random eigenvector solutions within the 3D error ellipse defined by the EKF uncertainty estimates. Applying each of the thirty possible solutions to all edge pixels, the algorithm attempts to minimize the global difference between edge pixels in the image pair. Edge pixels which do not match are considered “non-static” and are stored (for the purpose of this analysis) in a separate matrix. This matrix therefore represents pixels which are classified as “edge” pixels but did not match between image pairs and are therefore classified as non-static.
6.5a. 3D Flash LADAR image, empty hallway
6.5b. 3D Flash LADAR image, with non-static feature (moving person) next frame from Figure 6.5a
6.5c. 3D Flash LADAR image, with non-static feature (moving person) next frame from Figure 6.5b
6.5d. 3D Flash LADAR image, with non-static feature (moving person) next frame from Figure 6.5c

Figure 6.5. 3D LADAR Acquisition of Non-Static Features
The results of this test are reflected in Figure 6.6. In each frame comparison, the edges associated with the moving person (non-static edges) are clearly identifiable, along with some “extra” features. The reason for the presence of these “false” non-static features required some investigation. After much study, it became apparent that the lines along the right hand edge of each image were the reflection of the laser beams off the small metal strips between successive wall panels. Owing to the high reflectivity of these features they induce a highly variable range distance and thus appear to be “moving”. At the same time, they possess sufficient spatial extent (width) to appear as a surface rather than simple isolated noise pixels. The smaller “circles” are artifacts of reflection from various surfaces, including the “duct tape” utilized to mark survey points on the floor.

The analysis to determine the ability of the NAVRI algorithm to separate static and non-static features was completed over ten image frames drawn from 3D Flash LADAR experiment runs #9 and #10 (ten frames from each experiment, reference Table 5.3 for list of runs). The results for these investigations are consistent with those shown above.

The erroneous classification of grouped noise pixels as non-static edges is not critical since the objective is to identify static features for the purpose of location. This issue of classification could become an issue only in the instance where the number of non-static
features outnumbers the static features, and the non-static features are all coincident, as previously discussed in Chapter 2, Section 2.5. Given the unlikely nature of both events (non-static greater than static and coincident vector motion); we conclude that the algorithm successfully eliminates non-static features, whether they are actually moving (person) or only appear to be moving (scintillation of the laser beam).

Impact of NAVRI Updates to the INS position/orientation solution

The next question is if the positional updates derived by matching static features are meaningful to the determination of trajectory. For this purpose, the offset induced by each set of images was determined over a span of a) images 6.5a~6.5d (which correspond to images 2~5 of the acquisition set), b) the 31\textsuperscript{st} and 32\textsuperscript{nd} image set of the same trajectory collect, c) the 539\textsuperscript{th} and 540\textsuperscript{th} image set, and finally, d) the 896\textsuperscript{nd} and 897\textsuperscript{rd} image set.

The choice of sets was selected at random from 900 available image frames and provides some indication of the “corrections” provided by the 3D ranging dataset over an extended temporal period of ~2 minutes. The results are reflected in Tables 6.2 and 6.3 respectively.
6.6a. Extracted non-static edges, image 6.5a to 6.5b

6.6b. Extracted non-static edges, image 6.5b to 6.5c

6.6c. Extracted non-static edges, image 6.5c to 6.5d

Figure 6.6 Extracted Non-Static Edges
In Table 6.2, the results show the position reported by the EKF, relying solely on the IMU data for the duration of the trajectory. The “position $\sigma$” indicates the EKF uncertainty in the position for each direction. Initially, the offsets induced by the IMU only solution are small and generally insignificant. The IMU errors grow quickly however, after 1.8 minutes (540 frames collected at 5 Hz frame rate) the difference between the solutions is 0.376 meters in the X, 2.64 meters in the Y axis (corresponding to forward/back and right/left) and 2.14 meters in the Z (up/down) axis. The results for the frames 896~897 are clearly poorer.

In Table 6.3, the position in each coordinate axis as returned by the EKF is reported. This is based upon the 3D Ranging LADAR solution. The “position $\sigma$” indicates the EKF uncertainty in the position for each direction. When the 3D Ranging solution is applied to update the position, the resulting drifts remain on the order of $<30$ millimeters in the X and Y axis and under 60 millimeters in the Z (vertical).
Table 6.2 Positional results for INS during 3D Flash LADAR Run #10 (reference Table 5.3), positions derived from HG1700 IMU data only

<table>
<thead>
<tr>
<th>Image</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without Position Updates (INS Only)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image 31~32</td>
<td>-0.0220</td>
<td>0.1680</td>
<td>-0.1400</td>
</tr>
<tr>
<td></td>
<td>0.0150</td>
<td>0.0150</td>
<td>0.0040</td>
</tr>
<tr>
<td>Image 539~540</td>
<td>-0.3760</td>
<td>2.6400</td>
<td>-2.1400</td>
</tr>
<tr>
<td></td>
<td>0.1820</td>
<td>0.1820</td>
<td>0.0170</td>
</tr>
<tr>
<td>Image 896~897</td>
<td>-1.2390</td>
<td>6.7780</td>
<td>-4.4250</td>
</tr>
<tr>
<td></td>
<td>0.4100</td>
<td>0.4100</td>
<td>0.0370</td>
</tr>
</tbody>
</table>

Table 6.3 Positional results for INS during 3D Flash LADAR Run #10 (reference Table 5.3), positions derived from NAVRI updates

<table>
<thead>
<tr>
<th>Image</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>With 3D LADAR Position Updates</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image 31~32</td>
<td>0.0090</td>
<td>0.0300</td>
<td>-0.0520</td>
</tr>
<tr>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Image 539~540</td>
<td>0.0150</td>
<td>-0.0090</td>
<td>-0.0580</td>
</tr>
<tr>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Image 896~897</td>
<td>0.0340</td>
<td>0.0050</td>
<td>-0.0320</td>
</tr>
<tr>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
These results indicate that the 3D eigenvector matching algorithm is enabling the overall uncertainty of position to be reduced over the time period. The next question is if the offsets provided by the algorithm are correct, that is, do they permit the system to maintain the proper position with respect to known positions along the trajectory? To answer this question, we compare the generated trajectory (from INS and the 3D Ranging Data) to the expected trajectory based upon traditional survey data.

Conclusions for application of the NAVRI algorithms to 3D Flash LADAR data

The analysis indicates that the NAVRI algorithm, as applied to 3D Flash LADAR data, is capable of accomplishing two key goals. First, the separation of static and non-static features, and second, the derivation of position and orientation based upon the static features.

Unfortunately, the LADAR data becomes saturated when the unit is placed too close to a reflecting surface (as indicated by Figure 6.8) which effectively “blinds” the sensor. No features are available to facilitate matching edges and the navigation algorithm is paralyzed. In the instance of the LADAR acquisition, the unit experienced outages in the corners of the trajectory for upwards of 90 seconds. Because of this condition, the IMU was free to drift without compensation during the LADAR trajectory. While the navigation algorithm can correct the relative motion from one frame to another, the
algorithm is dependent on the INS to provide reasonable values for positional offset between epochs of extended “outage”.

As can be seen from Figure 6.7, the uncompensated IMU drift is so extensive as to render the data nearly useless; in a period of 2.3 minutes the IMU drifts ~2500 meters in Easting, ~400 meters in Northing, and ~1700 meters in the vertical. Thus, the ability to create a trajectory based upon post-analysis data is infeasible with the 3D Flash LADAR datasets which were acquired for this experiment.

Figure 6.7 Uncompensated IMU Drift, LADAR Acquisition #10
It is interesting to review the attempt to correct the extensive IMU drift on a post-processing basis. The results of this effort are reflected in Figure 6.9. The red dots indicate the information coming from the EKF (filtered IMU) while the blue dots show the solution derived by the NAVRI algorithm and provided to the INS as a substitute GPS positional update. While the trajectory does not match the surveyed points along the trajectory (shown in green with diamonds for the survey points), the results do indicate that the combined solution is capable of preventing the rapid IMU drift shown in Figure 6.7. This suggests that if the two system elements (IMU and NAVRI) could be tightly integrated, the results are likely to minimize trajectory errors compared to a GPS/INS reference solution. In the following section, this will be accomplished with the 3D Sonar data.

![Figure 6.8 Example of LADAR Image “Saturation”](image-url)
Figure 6.9 Post-Processing LADAR Trajectory
6.2 Demonstration of NAVRI algorithm with 3D Sonar Data

The 3D sonar data sets were acquired in an environment of full GNSS availability; thus, the IMU was compensated throughout the survey, either from the 3D navigation algorithm, or by GNSS position when the NAVRI algorithm was unable to provide a positional fix.

The significant drawback to 3D sonar data is the frequent lack of features to facilitate image matching and thus positioning. A typical 3D sonar “image” reflects the all too common scarcity of features along the muddy bottom of canals. The unfortunate reality is that features are not available often enough to facilitate an entire traverse based solely on the IMU and the 3D navigation algorithm. When features are present, the 3D navigation algorithm was found to be capable of maintaining position within a design specification of 10 meters CEP.

Figure 6.10 (Panama Canal Survey) indicates the position of the sensor during movement of the ship past of the lock, shown in solid red on the diagram. The position of the sensor as determined from 3D sonar imagery is shown by the blue cross marks. Movement of the ship is from left to right, and then returning along the parallel track. In this experiment, the NAVRI algorithm was utilized to determine position from a known GPS
location until the drift in position exceeded 1 meter during a period where NAVRI solutions were not available due to data scarcity; the true GPS position was then utilized to restart the algorithm from a new “known” position. The initial tracking was quite good until the center portion of the travel, where lack of features prohibited the algorithm from determining position. This condition is quite problematic since it reflects a rather extended period where the navigation system must rely upon the inertial system for positioning. The error rate in the inertial system could easily grow to unacceptable levels within this time period.

Figure 6.11 (Panama Canal Survey) indicates the position in the xz plane, in order to show the error in the range axis. The positioning error is larger in the range direction, approximately +/- 4 meters from the reference solution. The points with large differences from the reference solution are largely a function of available points; a few points with large range differences are problematic for estimation.

Figure 6.12 (Panama Canal Survey) shows the total solution in 3D sense and provides a complete overview of the positional capabilities of the integrated INS/Echoscope/NA VRI. Further research is needed to improve the capability to identify features in the center areas of the traverse. A first possible solution is the utilization of mosaicked sonar images in tandem with the range measurement data; the combination of intensity based imagery with range based feature extraction may provide greater capabilities to support pixel separation. A second option is the use of finer iterations
during the segregation processing to avoid overshooting the critical thresholding point for merging of surface pixels while maintaining sufficient transitional pixels. The figures do indicate that positioning by 3D sonar features is quite feasible, provided that the features exist for extraction. Overall 3D average position error is \(~0.2\) meters with an associated RMS of \(~0.8\) meters. While careful filtering would be necessary to avoid spurious solutions, the algorithm can provide an accurate position by matching static features.
Figure 6.10 XY Plot of Position - Reference and NAVRI Based Solutions, Panama Canal Traverse
Figure 6.11 XZ Plot of Position - Reference and NAVI Based Solutions, Panama Canal Traverse
Figure 6.12 Panama Canal Survey Traverse and Results, 3D
The Canadian survey results were processed identically to the Panama Canal survey and the results are reflected in Figures 6.13 to 6.15 respectively. The ability to recover matching images was clearly reduced as evidenced by the sparsely spaced blue crosses. However, where features were present along the traverse to permit matching, the resulting position is less than 1 meter in the X and Y directions (Figure 6.16 and 6.17 respectively), and less than 1.5 meters in the Z (vertical) (Figure 6.18) with the exception of frame 37, which reaches 4.1 meters error in the vertical.
Figure 6.13 XY Plot of Position - Reference and NAVRI Based Solutions, Canadian Traverse
Figure 6.1: XZ Plot of Position - Reference and NAVRI Based Solutions, Canadian Traverse
Figure 6.15 Canadian Survey Traverse and Results, 3D
Figure 6.16 3D Sonar, ΔX (Reference Trajectory minus NAVRI Solution)
Figure 6.17. 3D Sonar, ΔY (Actual Trajectory minus NAVR1 Solution)
Figure 6.18 3D Sonar, ΔZ (Actual Trajectory minus NAVI Solution)
6.3 Demonstration of Trajectory with 3D Kinect Data

We first consider the trajectory inside the Supercomputing Center at The Ohio State University. The general path of the platform is indicated in Figure 5.15 in Chapter 5.

The key point in the overview is the need to provide coarse positioning information to the 3D matching algorithm to constrain the search space for matching eigenvector signatures. Since quality IMU data are not available, the matching SIFT features from the intensity images were correlated with the associated range pixel measurements, and these range measurements were utilized in Horn’s Method [Horn, 1987] to provide the coarse adjustment between consecutive range image frames. The 3D range matching algorithm described above then proceeds normally.

Utilizing the combined intensity images (for coarse adjustment via SIFT) and the 3D ranging data, a trajectory was derived for travel inside the building at the OSU Supercomputing Facility. There is a finite interval between exiting the building and recovery of GPS signal lock during which the range acquisition was not available; thus the total extent of travel distance during GPS signal outage is not precisely identical to the travel distance where 3D range solutions were utilized for positioning. We estimate the distance from recovery of GPS signal to last known 3D ranging-derived position to be approximately 3 meters. Based upon this estimate, the travel distance inside the building should be approximately -53.5 meters (forward), 9.5 meters (right), and 0.75 meters
Based upon these estimates, the total misclosure based upon 3D range derived positions is provided in Table 6.4. The asterisk in the third row indicates the estimated nature of these values.

<table>
<thead>
<tr>
<th></th>
<th>Right/Left</th>
<th>Forward</th>
<th>Vertical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average positional uncertainty (m)</td>
<td>0.2</td>
<td>0.4</td>
<td>0.1</td>
</tr>
<tr>
<td>Standard deviation of position uncertainty (m)</td>
<td>0.5</td>
<td>1.5</td>
<td>0.7</td>
</tr>
<tr>
<td>Trajectory mis-closure (estimated) (m)*</td>
<td>0.4</td>
<td>10.9</td>
<td>7.3</td>
</tr>
</tbody>
</table>

Table 6.4 Approximate Positional Results for OSU Supercomputing Trajectory

The average positional uncertainty reflects the relative, frame-to-frame error reported by the algorithm during the indoor trajectory. This includes both IMU and NAVRI solutions. The primary reason for the rather large misclosure in the forward and vertical directions is the result of three distinct issues. First, the Kinect ranging sensor has a limited range; during certain portions of the trajectory the sensor is nearly “blind” due to lack of measurable features within the range. During this period, the algorithm must default to the IMU data, which are known to be suspect, as previously discussed. Secondly, the correlation between SIFT features and range measurement pixels can induce errors, as discussed above. Third, the 3D range positions and the IMU data are not integrated in this demonstration; the range positions are not providing a substitute for the lost GPS signal and the IMU is free to drift. Resolving this final issue would, at a minimum, reduce the IMU drift error and improve the overall solution.
A follow up study conducted at the alternate facility was completed using the same platform and methodology. In this study, a complete traverse was completed indoors forming a “box” or square trajectory, which returned to the original entrance point. A plot of the trajectory results is provided in Figure 6.20. The misclosure is 3.8 meters in the X (right/left) direction, 2.9 meters in the Z (forward) direction, and 22.5 meters in the Y (vertical) axis. The Y axis shows the rather poor contribution of the IMU during periods where 3D image solutions are not available. This is further indicated by Figure 6.21 which shows the uncertainty in the position by frame. Squares indicate the X uncertainty, diamonds the Y uncertainty, and pentagrams the Z uncertainty. The problem of unconstrained IMU drift and poor reliability is obvious.

While similar issues exist with IMU drift (owing to lack of tight integration with the ranging data), a number of problems between the SIFT feature/range pixel correlation portion of the algorithm are evident; note the large “clumps” of data points, where the algorithm struggles to reconcile the motions reported by the coarse (SIFT derived) position and the range derived position.
Figure 6.20 “Box” Trajectory, Alternate Facility, NAVRI solutions based upon 3D Kinect

Figure 6.21 3D Kinect, NAVRI positional uncertainty by frame, Alternate indoor facility data acquisition
The determination of position based upon the combination of SIFT matching and 3D range measurements can be seen to have particular potential to the problem of navigation during periods of operation in GPS denied environments. The experiment demonstrates several salient points of use in the ongoing research activities. First, the effective measurement range of the sensor is paramount; the trivial (but essential) need to acquire data is critical to success. Extending the range of the Kinect sensor to even ten meters (rather than the 3~4 meters in the experimental data) would significantly assist the derivation of position. A major problem was the presence of matching SIFT features but no corresponding range measurement. Secondly, orientation information is just as critical as position; the lack of this information significantly extended the time required to match features (via eigenvector signatures). Third, there is a critical need for the sensor to scan not only forward (along the trajectory) but also right/left and up/down. Obtaining features in all axes would support efforts to minimize IMU drift, particularly in the vertical. Alternatively, a wider field of view could conceivably accomplish the same objective. Finally, the algorithm was not fully integrated as a substitute for GPS positioning and the IMU was free to drift. Since the 3D ranging algorithm cannot guarantee a solution for all epochs, accurate IMU positioning is critical to bridge these outages. Fully integrating the 3D ranging solution with a GPS/IMU/3D schema would significantly reduce positional errors and misclosure.

The study indicates that utilizing 3D ranging images to achieve indoor relative (frame to frame) positioning shows great promise. The utilization of SIFT to match intensity
images was an unfortunate necessity dictated by the quality of data available; the method is technically feasible but our efforts would suggest there are significant drawbacks to this application, both in terms of efficiency and positional accuracy. It would be better to use quality IMU data with orientation solutions to derive the best possible solution, but the capability of the NAVRI algorithm to provide positional updates based upon 3D ranging imagery is clearly successful within the constraints of the experimental setup.
CHAPTER 7

CONCLUSIONS AND OBSERVATIONS

Drawing to a close on this effort, in this chapter we first discuss the developed navigation from 3D ranging imagery (NAVRI) algorithm in terms of its contribution to autonomous navigation, its development, and performance. While the focus of this thesis is on the NAVRI algorithm in a generic sense, in the course of the research, several observations are relevant to the technologies utilized to acquire 3D ranging data, and these are briefly discussed in the second section. Finally, a number of ideas and thoughts for extending this research within the schema of autonomous navigation are offered.

7.1 The NAVRI Algorithm

A number of 3D ranging technologies have become commercially realized in the past decade; research into new methods to exploit the resulting data sets remains relatively scarce with respect to the field of autonomous navigation. As discussed in Chapter 2, a common technique has been to apply existing methods developed for 2D data sets to the 3D instance. As reviewed in Section 2.3.1, a number of research studies have identified
the shortcomings of this approach; yet the development of true 3D algorithms remains limited to a handful of research efforts, as covered in Section 2.3.2.

I have presented an algorithm which utilizes generic 3D ranging data sets to establish the position and orientation of a moving platform on a relative basis between each set of image frames. The algorithm contains three unique, defining features. First, the range data is leveraged to derive statistics which constrain the segmentation process by reducing the dimensionality of the segmentation problem. The ability to utilize the data to define the thresholding characteristics of the segmentation process is unique and eliminates the need for a-priori, hard coded values which may not be appropriate in all environments. Secondly, the algorithm enables the efficient separation of static and non-static elements of two temporally spaced images, thus enabling the static features to be utilized for positioning solutions. Third, the algorithm extends the methodology of matching features by eigenvector signature to the three dimensional instance. By means of these innovations, the derivation of position and orientation is made possible by determining the scale, rotation, and translation function $f(s,R,T)$ which best place the static features (pixels) from the first image into the coordinate frame of the second image such that they match static pixels in the second image. The positional solution is derived from the translation result, and the orientation update from the rotation matrix.

The NAVRI algorithm has been applied to a number of data sets acquired from different 3D ranging technologies. It is generic in the sense that any gridded, three dimensional
dataset can be utilized to derive relative position and orientation between image frames, and adaptable in the sense that the means for providing the initial coarse solution can be leveraged, such as IMU data, or SIFT based features from traditional intensity images. The keys to the successful use of the NAVRI algorithm are:

- Features exist in the acquired imagery
- Sufficient correspondence exists for static features in two temporally spaced images
- Sufficient static features exist to enable determination of position and orientation

This is not to suggest that the algorithm is without limitation. First, the NAVRI algorithm is dependent on a secondary source of information to provide the initial (coarse) functional scale, rotation and translation information \( f(s, R, T) \) to enable the search space to be constrained sufficiently for the algorithm to remain efficient. Without this additional information, the search space for matching eigenvectors must be estimated from likely motions (a-priori) and involved extended searches for matches owing to the large number of translational and rotational possibilities. Secondly, the NAVRI algorithm makes a critical assumption that the number of static features corresponding between the images outnumbers the non-static features. While such safeguards were not incorporated into the programs developed to support this dissertation, implementation in real operations must contain appropriate checks to validate the occurrence of this condition. A likely method is to monitor the magnitude of global error fit between static features on a frame to frame
basis over time and consider large changes to be evidence of a change necessitating alternative approaches. These heuristics will require additional programming and consideration to ensure the case of predominately non-static features is identified and rectified to prevent serious biases in position/orientation determination.

With these caveats in mind, the NAVRI algorithm presented represents a significant contribution to the field of autonomous navigation by demonstrating a method for segmentation, feature extraction and matching. It requires minimal (if any) a-priori information and enables an effective substitute to lost GPS signals within the schema of a GPS/INS system.

7.2 The Technologies

As noted earlier, the primary focus of this research is on the development of a generic algorithm utilizing 3D ranging data. In the course of the research, a few critical observations are worthy of discussion to facilitate future research and development efforts.

First, the trivial (but critical) need to acquire feature data cannot be underestimated. A major problem with LIDAR and/or Kinect data is the lack of dynamic range during the acquisition process. In the instance of Flash LIDAR, the problem of image saturation is a debilitating concern when the camera becomes too close to a wall or other feature. In the
case of Kinect, the lack of range can frequently reduce the availability of data for the
algorithm to utilize. What is truly necessary is an adaptive system which can adjust the
power of the LIDAR emissions on the fly to ensure data acquisition over a greater range
of distances.

In the case of 3D sonar, a major problem is the simple lack of a feature rich environment.
This can likely be extended to any 3D ranging technology – if features are not present,
the algorithm cannot provide positional updates and the system must rely on alternative
methods, such as the INS. If the operational environment is anticipated to possess
stretches where minimal (or no) features are present, appropriate considerations must be
made to the overall system redundancy to ensure required positional maintenance.

Secondly, the need for orientation information is important to the successful
implementation of the algorithm. A critical element of the eigenvector matching process
is the computation of the rotation matrix necessary to place two sets of corresponding
static features in the same coordinate frame. To ensure efficient searches for eigenvector
matches, the orientation information is even more important than positional data. The use
of low grade MEMS which cannot (or do not) provide such feedback is strongly
discouraged; while the algorithm can to an extent compensate, extended search time to
derive the necessary rotation matrix must entail longer computational cycles. The current
NAVRI algorithm, coded in Matlab and executed on a T2370 dual processor, can
compute a position/orientation solution in approximately 4 seconds. It is likely that an
implementation coded within a more sophisticated programming language would generate significantly better results in terms of computational efficiency. The key point is that the determination of the rotation matrix is the most important derivation with respect to computational load; minimizing the search ellipse by means of the best possible initial solutions from the EKF is critical to achieving the best computational efficiency.

Third, a major problem is the lack of data acquisition in directions other than the primary axis (forward). It is strongly recommended that the acquisition platform periodically turn right/left and up/down to ensure the acquisition of features in these directions. The determination of position in the vertical is particularly important if the cooperating technology is an IMU, as the unbounded drift in these devices is well understood.

7.3 Future Research

The capacity of the human mind for creative and innovative thinking remains one of the great mysteries in science, and indeed the problem of autonomous navigation remains intractable in no small part to the inability to emulate the cognitive process in a technological device. There is little doubt that future generations will take this humble contribution into directions that the author could scarcely imagine. In the spirit of providing stimulus to such endeavors, a number of possible research efforts will be discussed. I shall break these topics into two primary thrusts: research on the algorithm itself, and research utilizing the algorithm for various purposes.
On the Algorithm

A primary issue with the algorithm is the necessity for secondary sensor information to provide the initial (coarse) alignment between the image frames. One question would be the ability to track this secondary sensor information and develop dead reckoning estimates based upon the Markovian chain. By doing so, if the secondary sensor became temporarily unavailable, the algorithm could potentially utilize the estimated movement to constrain the search space. This would make the overall system more robust.

An open question is how large the search space can be while still enabling the efficient matching of features via eigenvector comparisons. Ideally, the number of iterations implemented by the algorithms could be tailored to the size of the search space; information on the tradeoff between accuracy and search time would provide the platform developer with critical information relating to the choice of technologies installed onboard.

In the current implementation, the algorithm merges together “like” features and discards them, retaining the edge pixels of features for the purpose of matching. This was done primarily for computational efficiency. However, the edges of features can often provide inexact and transitory relevance within a SFEM schema. One line of research would be to retain and match the surfaces rather than the edges; the impacts to processing time and
position/orientation accuracy could be of interest. A related possibility would be to plane fit merged surfaces and then match planes; such an approach is technically feasible and may represent a powerful and more efficient means of matching features than globally minimizing large sets of pixels via eigenvector matching.

Since both LIDAR and Kinect provide standard intensity images in conjunction with the 3D ranging image, methods to apply this information are certainly worth pursuing. Research thrusts include support for the segmentation process, facilitating the extraction of unique entities for tracking, and enabling improved feature matching.

Finally, the porting of the algorithm from its current Matlab implementation to a more computationally efficient programming language would be beneficial to implementation; moving to C++ or a similar advanced system is highly recommended, particularly as the size (and therefore number) of pixels available for matching increases as 3D ranging technologies mature.

**Algorithm Applications**

The first and most important effort will be the development of a fully integrated, tightly coupled Extended Kalman Filter to enable the direct use of NAVRI solutions in lieu of GPS, or provide a redundant solution when GPS is available to the platform. This is
particularly critical to ensure that IMU drift is constrained; when 3D solutions are not possible, the IMU solution must be available to support interim navigation.

Once such an implementation was accomplished, a number of performance research efforts would become feasible for testing. What is the long term drift of the 3D solution and how long can a desired level of accuracy be maintained? What tradeoffs exist between equipment capability (such as IMU performance) and the related relative positional accuracy? The integration of multiple sensors and the resulting impact to ongoing navigational capability is also of considerable interest.

An interesting extension of the algorithm would be the implementation of parallel algorithms to extract specific entities (such as a moving person) and implement a method for Detection and Tracking of Moving Objects (DATMO). This extension could become useful within the concept of collaborative navigation; a platform could simultaneously establish both its own position (via static features) and the position of a “team member”. The secondary platform could then leverage the position established by its own on board algorithm and the additional position as determined by the first unit. The extraction of specific entities rather than generic features (pixels) is not a trivial exercise and will require considerable innovation and research. This research thrust is a major field of and in itself which could provide many interesting possibilities, such as the tracking of targets, or the search for particular environmental signatures (body heat, chemicals, electro-magnetic signals, etc.).
Another possible research effort would be the use of the 3D ranging solution during landing and/or docking operations. The ability to provide precise ranging information and accurate position/orientation information could become useful in a variety of applications: landing aircraft, docking ships and boats, linking spacecraft modules, matching up parts in robotic assembly operations, etc. The ability to locate and match particular features in a field of view has distinct applications in a number of fields ranging far beyond autonomous navigation.

7.4 *Finis est Principium* (The end is the beginning)

The dream of realizing fully autonomous navigation of a platform remains beyond the present technical capability of mankind. The process of navigation, of how living creatures can navigate, is a topic worthy of deep introspection. During the course of this research, I found myself contemplating simple yet profound questions on a regular basis. How does one “know” that the space between two trees is “open” and the trees are impassible? How does a creature, even a simple being like a squirrel, rapidly assess the velocity and acceleration of an approaching vehicle and determine their capacity to safely cross in front of the said platform? How does one recall landmarks and navigate accordingly, despite the passage of extended time frames? I encourage those who might read this dissertation to sit and think about the wonder of navigation for some time.
The algorithm presented in this research effort provides a numerical approach to apply sensor information and achieve relative position between temporally separated frames. It is not continuous and cannot hope to provide the same depth of sensory information made available by our human integrated sensors to our brain. While unique and innovative, it remains but a poor substitute to the complex cognitive process of navigation accomplished by living creatures on a regular basis.

It can be said that undergraduate studies enable an understanding of knowledge in the simpler instance, such as the trivial physics problems involving no friction and no mass. Graduate studies extend this awareness to the greater complexity of our world, but still within the scope of what is known and understood. True research is to stand on the precipice of our knowledge, contemplate the unknown, and push back the edge of blindness such that future generations may see farther.

In the end, we find ourselves back to the beginning, attempting to understand the cognitive process of navigation, and filled with the desire to push back the darkness, that we might further understand how to achieve the dream of fully autonomous navigation.

- If you are not childlike, you cannot be a scientist.

Dr. C. N. Rao, Eminent Chemist
Graduate School Poem

Arms wide, future embracing
Because the past is behind,
Its lessons have been learned

Growing, learning anew
Skills for exploration,
Methods to explain

Emboldened, moving forward
Encountering a world once known,
Now barely understood

Gazing, everywhere complexity
Restored to childhood wonder,
At the mysteries revealed

Uncertain, seeking answers
Advisor's have no solutions,
Only paths to wander

Stumbling, discovering
Enigma's of missing knowledge,
A knot to untie
Years flash quickly

Arising, nugget of knowledge

Placed upon the altar

Judged worthy, it's placed

With care upon the edifice,

Precipice extended to the unknown

Returning, awareness of

New puzzles to resolve,

Walk alone, arms wide,

Future embracing...

J.N. Markiel, February, 2008
LIST OF REFERENCES


Besl, J. (b) “Surfaces in Early Range Image Understanding”, Doctoral Dissertation, University of Michigan, 1986


Coda Octopus, 2009 “Echoscope”, Technical Brochure, CodaOctopus, URL active as of 28 December 2010
http://www.codaoctopus.com/pdfs/screen/CodaEchoscope_Apr09.pdf


171


http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.7658&rep=rep1&type=pdf


174


Lee, C. and D. Landgrebe. (b) “Feature Extraction Based Upon Decision Boundaries”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 4, April, 1993, pp 388–400


Markiel, J.N. “Separation of Static and Non-Static Features From Three Dimensional Datasets by Implicit Parametric Equations”, Master's Thesis, The Ohio State University, 2007

Markiel, J.N. “Statistical Assessment of the SIFT Feature Matching Algorithm”, SPIN Laboratory Report, The Ohio State University, January 2nd, 2009


179


Sun, H. “The General Design of AIMS-Pro™”, Internal Communication, The Ohio State University, SPIN Laboratory, 2007


181


182


Appendix A

Absolute Orientation – Horn’s Closed Form Solution

Horn [1987] developed a closed form solution to determine the absolute orientation required to transform a set of points in one coordinate frame into a second coordinate frame, given that the locations of the points are known in both coordinate frames. His method is provided in this appendix for completeness of method; to limit confusion the original symbolic system outlined by Horn will be utilized rather than the symbolic system utilized in the main text. The output of the methodology is the scale, rotation, and translation vectors required to exactly complete the desired transformation.

Let there be two non-co-incident, orthogonal coordinate frames $r_l$ and $r_r$, the “$l$” corresponding to left, the “$r$” corresponding to right. Let there be $n \geq 3$ points, present in both coordinate frames, with known coordinates from the origin of the respective coordinate frame.

The first step involves the local transformation of the point origin (the coordinate frame origin) to the centroid of each point set. The number of matching points in each coordinate frame is identical in size.
The new coordinates of the point set $n$ can therefore be expressed as:

$$r_{l,i}' = r_{l,i} - \bar{r}_l, \quad r_{r,i}' = r_{r,i} - \bar{r}_r$$  \hspace{1cm} (A2)

And

$$\sum_{i=1}^{n} r_{l,i}' = 0, \quad \sum_{i=1}^{n} r_{r,i}' = 0$$  \hspace{1cm} (A3)

The error term for the transformation of the left coordinate frame to the right coordinate frame can then be expressed as:

$$e_l = r_{r,i}' - sR(r_{l,i}') - r_0'$$  \hspace{1cm} (A4)

Where the scale factor $s$ and rotation matrix $R$ are unknown, along with the translation vector $r_0'$, reflected as:

$$r_0' = r_0 - \bar{r}_r + sR(\bar{r}_l)$$  \hspace{1cm} (A5)
Where \( r_0 \) is the translation vector between the original coordinate frame origins, and \( r_0' \) is the translation vector between the centroid origins.

Squaring equation A4 provides the sum of squares error to be minimized; the \( || \ || \) indicating the absolute value:

\[
\sum_{i=1}^{n} \left| r_{r,i}' - sR(r_{l,i}') \right|^2 = 2n_0' \sum_{i=1}^{n} \left[ r_{r,i}' - sR(r_{l,i}') \right] + n \left| r_0' \right|^2 \quad (A6)
\]

The middle term must equal zero since all the points are referenced to the centroid. The first term does not depend on the translation vector while the last term cannot be negative. The sum of squares error is minimized when \( r_0' = 0 \); thus

\[
e_i = r_{r,i}' - sR(r_{l,i}') \quad (A7)
\]

And the total error to be minimized is:

\[
\sum_{i=1}^{n} \left| r_{r,i}' - sR(r_{l,i}') \right|^2 \quad (A8)
\]

Equation A8 indicates that if the scale vector and rotation matrix are known, solving for the translation is a trivial exercise in subtracting the rotated and scaled centroid from the first frame from the second centroid.
If the errors in distance to the coordinate frame origin (or centroid origin) are similar, equation A7 can be re-written as:

\[ e_i = \frac{1}{\sqrt{s}} r_{r,i} - \sqrt{s} R \left( r_{r,i} \right) \]  

(A9)

The sum of squared error is therefore:

\[ \frac{1}{s} \sum_{i=1}^{n} \left\| r_{r,i} \right\|^2 - 2 \sum_{i=1}^{n} r_{r,i} \cdot R \left( r_{r,i} \right) + s \sum_{i=1}^{n} \left\| r_{r,i} \right\|^2 \]  

(A10)

Which can be written in simplified manner as shown in equation A11, with \( S_r = \sum_{i=1}^{n} \left\| r_{r,i} \right\|^2 \), \( D = \sum_{i=1}^{n} r_{r,i} \cdot R \left( r_{r,i} \right) \), and \( S_l = s \sum_{i=1}^{n} \left\| r_{r,i} \right\|^2 \).

\[ \frac{1}{s} S_r - 2D + sS_l \]  

(A11)

Completing the square yields:

\[ \left( \sqrt{s} S_l - \frac{1}{\sqrt{s}} S_r \right)^2 + 2 \left( S_l S_r - D \right) \]  

(A12)

Equation A12 is minimized with respect to the scale factor \( s \) when the first term is zero.
\[ s = \left( \frac{\sum_{i=1}^{n} ||r'_{r,i}||^2}{\sum_{i=1}^{n} ||r'_{l,i}||^2} \right)^{1/2} \]  

(A13)

Therefore the scale can be determined without knowing the rotation matrix \( R \); further, the remaining error is minimized when \( D \) is as large as possible. Therefore, to evaluate the rotation it is necessary to seek the rotation which maximizes equation A14.

\[ \sum_{i=1}^{n} r'_{r,i} \cdot R(r'_{l,i}) \]  

(A14)

Transforming the rotation matrix in equation A14 to a quaternion based rotation yields two alternate forms, as reflected in equations A15, with \( q \) indicating the unit quaternion and \( q^* \) it's conjugate.

\[ \sum_{i=1}^{n} (qr'_{l,i}q^*) \cdot r'_{r,i} = \sum_{i=1}^{n} (qr'_{l,i}) \cdot (r'_{r,i}q) \]  

(A15)

Given,

\[ r'_{l,i} = (x'_{l,i}, y'_{l,i}, z'_{l,i})^T, \quad r'_{r,i} = (x'_{r,i}, y'_{r,i}, z'_{r,i})^T \]  

(A16)

Equations A17 and A18 are easily derived.
The matrices \( \bar{R}_{l,i} \) and \( \bar{R}_{r,i} \) are not only skew symmetric, but orthogonal. Substituting equations A17 and A18 into equation A15, it is possible to determine the sum to be maximized in order to derive the best (in a least squares sense) rotational matrix. Equations A19–A20 reflect the noted substitution.

\[
\sum_{i=1}^{n} (\bar{R}_{l,i} q) \cdot (\bar{R}_{r,i} q) = \sum_{i=1}^{n} (q^T \bar{R}_{l,i}) \cdot (\bar{R}_{r,i} q) = q^T (\sum_{i=1}^{n} R_{l,i}^T R_{r,i}) q \tag{A19}
\]

\[
q^T (\sum_{i=1}^{n} N_i) q = q^T N q \tag{A20}
\]

Since each matrix \( N_i \) is symmetric, the final matrix \( N \) is also symmetric.

Now consider a 3x3 matrix, the elements of which are the sums of products of coordinates measured in the left coordinate frame with those in the right coordinate frame. This matrix is reflected in equation A21, and extends to the \( N \) matrix via equations A22 and A23 respectively.
\[ M = \sum_{l=1}^{n} r'_{l,i} (r'_{r,i})^T = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix} \tag{A21} \]

Where

\[ S_{xx} = \sum_{l=1}^{n} x'_{l,i} x'_{r,i} \quad , S_{xy} = \sum_{l=1}^{n} x'_{l,i} y'_{r,i} \quad , \text{etc} ... \tag{A22} \]

\[ N = \begin{bmatrix} (S_{xx} + S_{yy} + S_{zz}) & (S_{yx} - S_{zy}) & (S_{zx} - S_{xz}) & (S_{xy} - S_{yx}) \\ (S_{yx} - S_{zy}) & (S_{xx} - S_{yy} - S_{zz}) & (S_{zx} + S_{xz}) & (S_{xy} + S_{yx}) \\ (S_{zx} - S_{xz}) & (S_{xy} - S_{yx}) & (S_{xx} - S_{yy} - S_{zz}) & (S_{yz} - S_{zy}) \\ (S_{xy} - S_{yx}) & (S_{zx} - S_{xz}) & (S_{yz} + S_{zy}) & (S_{xx} - S_{yy} + S_{zz}) \end{bmatrix} \tag{A23} \]

The ten elements of the real symmetric matrix \( N \), of size 4x4 are expressed by equation A23 as the sums and differences of the nine elements of matrix \( M \), derived from the coordinates of the point set being transformed. The trace of equation A23 sums to zero, which accounts for the 10\(^{th}\) and final degree of freedom.

With respect to the right hand side of equation A20, the rotational unit quaternion which maximizes the expression is the eigenvector corresponding to the largest eigenvalue of the \( N \) matrix. This can be proved as follows.

Let the four real eigenvalues of the symmetric 4x4 matrix \( N \) be \( (\lambda_1, \lambda_2, \lambda_3, \lambda_4) \). A corresponding set of orthogonal unit eigenvectors \( (e_1, e_2, e_3, e_4) \) may then be established enabling the construction of equation A24.
\[ Ne_i = \lambda_i e_i \quad \text{for} \quad i = 1, 2, \ldots 4 \quad (A24) \]

An arbitrary eigenvector can be written as a linear combination: \( q = \alpha_1 e_1 + \alpha_2 e_2 + \alpha_3 e_3 + \alpha_4 e_4 \) and since the eigenvectors are orthogonal \( q \cdot q = \alpha_1^2 + \alpha_2^2 + \alpha_3^2 + \alpha_4^2 = 1 \).

Since \( e_i \) are the eigenvectors of \( N \):

\[ Nq = \alpha_1 \lambda_1 e_1 + \alpha_2 \lambda_2 e_2 + \alpha_3 \lambda_3 e_3 + \alpha_4 \lambda_4 e_4 \quad (A25) \]

And

\[ q^T Nq = q \cdot \left( \begin{array}{cccc}
\lambda_1 & & & \\
& \lambda_2 & & \\
& & \lambda_3 & \\
& & & \lambda_4
\end{array} \right) q = (\alpha_1^2 \lambda_1 + \alpha_2^2 \lambda_2 + \alpha_3^2 \lambda_3 + \alpha_4^2 \lambda_4) \quad (A26) \]

If the eigenvalues are arranged in order of magnitude \( \lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4 \), then the quadratic form cannot be larger than the greatest positive eigenvector, as shown in equation A27.

\[ q^T Nq \leq \left( (\alpha_1^2 \lambda_1 + (\alpha_2^2 \lambda_1 + (\alpha_3^2 \lambda_1 + (\alpha_4^2 \lambda_1)) \right) = \lambda_1 \quad (A27) \]

Therefore, the eigenvector associated with the largest eigenvalue maximizes equation A20 and minimizes the overall sum of squares error expressed in equation A8. The solution to equation A23 is a fourth order polynomial in \( \lambda \) that is obtained via equation A28, with \( I \) as the identity matrix of size 4x4.

191
\[ \det(N - \lambda I) = 0 \]  \hspace{1cm} (A28)

After solving equation A28 to determine the largest eigenvalue \( e_{\text{max}} \), the corresponding eigenvector is determined by solving equation A29.

\[ [N - \lambda_{\text{max}} I]e_{\text{max}} = 0 \]  \hspace{1cm} (A29)