GPS/Optical/Inertial Integration for 3D Navigation and Mapping

Using Multi-copter Platforms

A dissertation presented to

the faculty of

the Russ College of Engineering and Technology of Ohio University

In partial fulfillment

of the requirements for the degree

Doctor of Philosophy

Evan T. Dill

May 2015

©2015 Evan T.Dill. All Rights Reserved.
This dissertation titled
GPS/Optical/Inertial Integration for 3D Navigation and Mapping
Using Multi-copter Platforms

by
EVAN T. DILL

has been approved for
the School of Electrical Engineering and Computer Science
and the Russ College of Engineering and Technology by

Maarten Uijt de Haag
Edmund K. Cheng Professor of Electrical Engineering and Computer Science

Dennis Irwin
Dean, Russ College of Engineering and Technology
ABSTRACT

DILL, EVAN T., Ph.D., May 2015, Electrical Engineering

GPS/Optical/Inertial Integration for 3D Navigation and Mapping Using Multi- copter Platforms

Director of Dissertation: Maarten Uijt de Haag

As the potential use of autonomous unmanned aerial vehicles (UAVs) has become more prevalent in both the public and private sectors, the need for a reliable three-dimensional (3D) positioning, navigation, and mapping (PNM) capability will be required to enable operation of these platforms in challenging environments where the Global Positioning System (GPS) may not necessarily be available. Especially, when the platform’s operational scenario involves motion through different environments like outdoor open-sky, outdoor under foliage, outdoor-urban and indoor, and includes transitions between these environments, there may not be one particular method to solve the PNM problem.

In this dissertation we are not solving the PNM problem for every possible environment, but select a couple of dissimilar sensor technologies to design and implement an integrated navigation and mapping method that can support reliable operation in an outdoor and structured indoor environment. The integrated navigation and mapping design is based on a Global Positioning System (GPS) receiver, an Inertial Measurement Unit (IMU), a monocular digital camera, and three short to medium range laser scanners. To evaluate the developed algorithms a hexacopter was built, equipped with the above sensors, and both hand-carried and flown through the target environments. This dissertation will
show that dm-level relative positioning accuracies can be achieved for operations traversing a building, and that when segments are included where GPS is available, the platform’s trajectory and map will be globally anchored with m-level accuracy.
ACKNOWLEDGMENTS

The completion of this dissertation is the culmination of multiple years of strenuous work. Throughout this process many people have provided support and encouragement, without which, this may not have been possible.

Foremost, I would like to give thanks to my Lord and Savior, Jeremiah 29:11.

Next, I would like to thank many of the engineers that have contributed to this research and my education through various means. My colleagues and friends; Phil Duan, Tim Needham, Kuangmin Li, Rakesh Kashyap, Paul Stocklin, and Adam Schultz have provided research assistance and countless constructive technical conversations. During my fellowships at Wright-Patterson Air Force Base, Dr. Jacob Campbell was a great source of direction and knowledge. Over the past few years, a large portion of my research was done in conjunction with Daniel Serrano and Santi Vilardaga. Working with these two was a joy and a privilege. As I have done my undergraduate, master's and doctoral degrees at Ohio University, I have had the privilege of learning from some of the most intelligent and hardworking engineers I've encountered. Dr. Frank van Graas, Dr. Wouter Pelgrum, and Dr. Doug Lawrence are all immensely talented researchers and teachers that have contributed a great deal to shaping me into the engineer that I've become.

Through this entire process I have been blessed with fantastic support from my family. My in-laws, Mike and Jan Bradford, have been a constant and important source of support. For my entire life my mother, Cindy, has provided unwavering encouragement. My father, Jeff, is an electrical engineering professor and has contributed more to my learning than any other figure in my life. From a young age he taught me how to solve
problems and think critically. Moreover, he taught me how to be a man through his example. It truly is because of him that I am an engineer.

Without a shadow of a doubt this work would not have been possible without my advisor Dr. Maarten Uijt de Haag. I've never seen a professor so respected and adored by all of his students. Maarten's selfless nature, hard work, knowledge and outgoing personality make him one of kind and an unparalleled teacher. I can't thank him enough for the experiences he has afforded me and the knowledge he's given me. I couldn't have asked for a better advisor and friend.

My final thank you is dedicated to my beautiful wife, Jenna. The constant love and encouragement she provides are daily motivators that push me to be the best man that I can be. Her importance to all aspects of my life, this work included, cannot be overstated. I am so blessed to have such an amazing wife to share my life with.

To all that I previously mentioned, a heartfelt thank you.
<table>
<thead>
<tr>
<th>TABLE OF CONTENTS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>3</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>5</td>
</tr>
<tr>
<td>List of Tables</td>
<td>10</td>
</tr>
<tr>
<td>List of Figures</td>
<td>11</td>
</tr>
<tr>
<td>Chapter 1: Introduction</td>
<td>15</td>
</tr>
<tr>
<td>1.1 Research Overview</td>
<td>17</td>
</tr>
<tr>
<td>1.2 Contributions</td>
<td>21</td>
</tr>
<tr>
<td>1.3 Dissertation Structure</td>
<td>23</td>
</tr>
<tr>
<td>Chapter 2: Background</td>
<td>24</td>
</tr>
<tr>
<td>2.1 Global Navigation Satellite Systems</td>
<td>24</td>
</tr>
<tr>
<td>2.2 Inertial Navigation Systems</td>
<td>26</td>
</tr>
<tr>
<td>2.3 GPS/Inertial Integration</td>
<td>29</td>
</tr>
<tr>
<td>2.4 Bayesian Estimators for Sensor Integration</td>
<td>30</td>
</tr>
<tr>
<td>2.5 Simultaneous Localization and Mapping</td>
<td>36</td>
</tr>
<tr>
<td>2.6 Optical Navigation</td>
<td>40</td>
</tr>
<tr>
<td>2.6.1 Laser Based Navigation</td>
<td>40</td>
</tr>
<tr>
<td>2.6.2 Vision Based Pose Estimation</td>
<td>43</td>
</tr>
<tr>
<td>2.6.2.1 Structure From Motion Related Methods.</td>
<td>44</td>
</tr>
<tr>
<td>2.6.2.2 Intrinsic Camera Parameters and Camera Calibration</td>
<td>49</td>
</tr>
<tr>
<td>2.7 Coordinate Transformations</td>
<td>53</td>
</tr>
<tr>
<td>Chapter 3: Laser-Based Estimation Methods</td>
<td>56</td>
</tr>
<tr>
<td>3.1 Dual Laser Altitude and Attitude Estimation</td>
<td>57</td>
</tr>
<tr>
<td>3.1.1 Altitude Estimation</td>
<td>58</td>
</tr>
<tr>
<td>3.1.2 Pitch and Roll Estimation</td>
<td>68</td>
</tr>
<tr>
<td>3.1.3 Results of Laser Based Pitch, Roll and Altitude Estimation</td>
<td>73</td>
</tr>
<tr>
<td>3.1.3.1 Experiment 1: Level Platform at Varying Height</td>
<td>74</td>
</tr>
<tr>
<td>3.1.3.2 Experiment 2: Platform at Varying Heights with Controlled Attitude Changes</td>
<td>76</td>
</tr>
<tr>
<td>3.1.3.3 Experiment 3: Level Platform at a Constant Height Traversing a Ramp</td>
<td>79</td>
</tr>
</tbody>
</table>
6.5 XSENS MTi Inertial Measurement Unit .............................................................. 166
6.6 Arduopilot Mega ............................................................................................. 166
6.7 Riegl LMS-Z360i ........................................................................................... 167
Chapter 7: Results ............................................................................................... 169
Chapter 8: Conclusions ....................................................................................... 191
Chapter 9: Future Work ....................................................................................... 193
References ........................................................................................................... 196
Appendix A: Double Differencing Derivation ....................................................... 207
Appendix B: Triangulation Method ...................................................................... 208
LIST OF TABLES

Table 1. Dominant short term inertial error growth rates. ................................................. 28
Table 2. Hardware specifications. .................................................................................. 162
LIST OF FIGURES

Figure 1. National airspace. [3] ................................................................. 16
Figure 2. Target platform: Commercially available hexacopter (3DR Robotics): modified to accommodate the platform weigh (> 4kg). ........................................................................ 17
Figure 3. Target operational scenario. ................................................................................................... 18
Figure 4. Position, navigation and mapping mechanization. ................................................................. 19
Figure 5. Installation orientations of laser scanners. ........................................................................... 21
Figure 6. Strapdown inertial navigation system mechanization. (adapted from [12]) ... 27
Figure 7. Inertial mechanization with error compensation inputs. (adapted from [9]). ... 29
Figure 8. Occupancy grid. ................................................................................................................ 39
Figure 9. 2D Laser scanner used to provide a 3D point cloud. .............................................................. 42
Figure 10. Epipolar constraint from multi-view geometry. ............................................................... 44
Figure 11. Stereo camera pair. ........................................................................................................ 46
Figure 12. Projective relationship between 2D image frame points and corresponding 3D sensor frame points. ........................................................................................................... 49
Figure 13. Camera calibration. ........................................................................................................... 52
Figure 14. Coordinate transformations for the downward and angled laser scans-(a)Left: sensor frame; (b)Middle: body frame; (c)Right: locally-level frame. .................................................. 55
Figure 15. Laser-based components of proposed mechanization. .................................................... 56
Figure 16. Point cloud 'slices' formed by the intersection of the downward and angle laser. .................. 57
Figure 17. Level floor altitude estimation concept. ........................................................................... 59
Figure 18. Subsets of laser points captured on the environments ground. ....................................... 60
Figure 19. Plane fit between scan points. ......................................................................................... 62
Figure 20. Tilted floor altitude estimation concept. ............................................................................ 63
Figure 21. Similarities between height measurements while traversing a ramp (A), and changing altitude (B). .............................................................................................................. 64
Figure 22. Differences in floor plane normal vectors while traversing a ramp (A), and changing altitude (B). .............................................................................................................. 64
Figure 23. Altitude estimation over tilted floor. ............................................................................... 65
Figure 24. Relationship between angled and vertical scan at different heights. ......................... 67
Figure 25. Functional block diagrams of estimators: (a) altitude using method ii, (b) attitude using lasers scanners only .......................................................................................................... 68
Figure 26. Results of wall identification on downward scan. .............................................................. 69
Figure 27. Results of wall planar surface identification. .................................................................... 70
Figure 28. Altitude Estimation test with platform remaining roughly level (full). ...................... 74
Figure 29. Altitude Estimation test with platform remaining roughly level (~70 seconds). ........ 75
Figure 30. Pitch and roll estimation test with platform remaining roughly level. ............. 76
Figure 31. Altitude estimation test with platform experiencing orientation changes (full). .... 77
Figure 32. Altitude estimation test with platform experiencing orientation changes (~110 seconds).

Figure 33. Pitch and roll estimation test with platform experiencing orientation changes.

Figure 34. Test setup for experiment 3.

Figure 35. Altitude Estimation test with platform traversing a ramp at a constant height.

Figure 36. Attitude Estimation test with platform traversing a ramp at a constant height.

Figure 37. Altitude Estimation test with platform experiencing height changes while traversing a ramp.

Figure 38. Attitude Estimation test with platform experiencing height changes while traversing a ramp.

Figure 39. Altitude estimation from hexacopter flight data (full).

Figure 40. Altitude estimation from hexacopter flight data (during flight).

Figure 41. Roll and pitch estimation from hexacopter flight data.

Figure 42. Scan matching procedure illustrated using real laser scan data.

Figure 43. Scan matching results.

Figure 44. Concept of occupancy grid expansion.

Figure 45. (a) Left: laser scan of a hallway with a level platform; (b) Right: laser scan of a hallway with an un-level platform.

Figure 46. Effect of a tilted platform on distance to wall measurement.

Figure 47. Block diagram of the modified 3DOF SLAM algorithm.

Figure 48. Effect of un-level platform on width of hallway measurement.

Figure 49. Occupancy grid.

Figure 50. Heading results for indoor loop experiment.

Figure 51. Trajectory results for indoor loop experiment.

Figure 52. Occupancy grid results for indoor loop experiment.

Figure 53. Heading results for indoor/outdoor flight experiment.

Figure 54. Occupancy grid results for indoor/outdoor flight experiment.

Figure 55. Trajectory results for indoor/outdoor flight experiment.

Figure 56. Monocular camera components of proposed mechanization.

Figure 57. Original image(left) and scale space representation of image(right).

Figure 58. SIFT feature identification.

Figure 59. SIFT matching results between two consecutive image frames.

Figure 60. Point feature association after inertial based miss-association rejection.

Figure 61. Principle of wavelet decomposition of an image.

Figure 62. Results of wavelet decomposition.

Figure 63. Wavelet vertical decomposition results.

Figure 64. Line feature extraction procedure.

Figure 65. Vertical line extraction results.

Figure 66. Laser/camera intersect calibration procedure setup.

Figure 67. Laser/camera intersect calibration procedure.

Figure 68. 2D vertical line and laser intersections in the image frame.
Figure 69. 2D vertical line and laser intersections in laser scan data .......................... 117
Figure 70. Relationship between camera position, 2D lines, and laser vision point. .... 118
Figure 71. Observed geometric change of line and laser/vision points as ownship position varies. ................................................................................................................ 119
Figure 72. Environment for experiment 1 ................................................................. 122
Figure 73. Visual odometry attitude estimation during controlled attitude changes. .... 123
Figure 74. Environment for experiment 2 ................................................................. 124
Figure 75. Visual odometry attitude estimation as platform experiences controlled motion through a hallway ................................................................. 125
Figure 76. Visual odometry path determination as platform experiences controlled motion through a hallway ................................................................. 126
Figure 77. Environment for experiment 3 ................................................................. 127
Figure 78. Visual odometry attitude estimation during a flight traversing a hallway. 128
Figure 79. Visual odometry path determination during a flight traversing a hallway. 129
Figure 80. Environment for experiment 4 ................................................................. 130
Figure 81. Visual odometry attitude estimation during a flight traversing a hallway. 128
Figure 82. Visual odometry attitude estimation while traveling around an indoor loop. 131
Figure 83. Environment for experiment 5 ................................................................. 132
Figure 84. Visual odometry Attitude estimation during indoor and outdoor flight ...... 133
Figure 85. Correlation between yaw maneuvers and number of features ............... 134
Figure 86. Visual odometry attitude estimation with supplemented inertial in feature scarce situations .......................... 135
Figure 87. Visual odometry path determination during indoor and outdoor flight. ..... 137
Figure 88. Position, navigation and mapping mechanismization .......................... 139
Figure 89. GPS dynamics filter geometry(adapted from [119]) ............................. 144
Figure 90. Laser based mapping from platform motion ................................. 157
Figure 91. Hexacopter platform ............................................................... 159
Figure 92. Hexacopter sensor locations ................................................. 160
Figure 93. Hardware setup ............................................................... 161
Figure 94. Point Grey Firefly .......................................................... 164
Figure 95. Hokuyo UTM-30LX(left) and URG-04LX laser scanners(right) ...... 165
Figure 96. NovAtel OEMStar receiver(left) and NovAtel L1 patch antenna (right) 165
Figure 97. XSENS MTi inertial measurement unit ............................................. 166
Figure 98. Ardupilot Mega .............................................................. 167
Figure 99. (a) Left: Riegl LMS-Z360i; (b) Right: 3D point cloud with overlaid imagery in the basement of Ohio University's Stocker Center ........................................ 168
Figure 100. Availability of sensors during forced outage of the laser ............... 170
Figure 101. GPS challenged outdoor environment ............................................. 170
Figure 102. Starting and ending locations of walking experiment shown on Google Earth™ ............................................................. 171
Figure 103. Pitch and roll results from integrated solution .............................. 172
Figure 104. Differences between pitch and roll estimates from the integrated solution and the standalone IMU. ................................................................. 172
Figure 105. Heading results from integrated solution. ........................................ 173
Figure 106. Differences between heading estimates from the integrated solution and the standalone IMU. ................................................................. 174
Figure 107. Path compared to 2D reference map. .............................................. 175
Figure 108. Enhanced section of path from integrated solution. ....................... 176
Figure 109. Produced flight map compared to truth reference map. ................. 177
Figure 110. Enhanced views of produced 3D map compared to truth reference map. 178
Figure 111. Enhanced views of outdoor sections of 3D map. .......................... 179
Figure 112. Geo-referenced flight path vs. GPS derived positions. ................. 180
Figure 113. (a) Left: EKF produced path; (b) Right: standalone GPS path ......... 180
Figure 114. Geo-referenced 3D environmental map. ..................................... 181
Figure 115. Availability of Sensors during forced outage of the laser .............. 182
Figure 116. Effect of losing GPS and lasers for 20 seconds. ......................... 183
Figure 117. Availability of Sensors during forced outage of the laser and camera . 184
Figure 118. Effect of coasting on the IMU for 20 seconds. ............................ 185
Figure 119. Availability of Sensors during flight test. ..................................... 186
Figure 120. Starting and ending locations of flight experiment shown on Google Earth™. ................................................................. 186
Figure 121. Pitch and roll results during flight test. ......................................... 187
Figure 122. Differences between pitch and roll estimates from the integrated solution and the standalone IMU during flight test. ............................... 188
Figure 123. Heading results during flight test. .............................................. 189
Figure 124. Differences between heading estimates from the integrated solution and the standalone IMU during flight test. ................................. 189
Figure 125. Derived flight path compared to reference map. ......................... 190
CHAPTER 1: INTRODUCTION

In recent years, the availability of unmanned aerial systems (UAS) and their application by non-military users has seen a steady increase. The spectrum of applications has also rapidly expanded to include commercial applications (e.g. surveying and mapping, forestry and agriculture, commercial imaging, news and media support, pipeline inspection), scientific applications (e.g. environmental monitoring and mapping, hyperspectral imaging), homeland security applications (e.g. border patrol, pipeline patrol, surveillance), and other civil government applications (e.g. first responder support, search and rescue, law enforcement, communications, forest fire monitoring).[1]

The expanding utilization of UAS is accompanied by an increased number of operational scenarios and environments. These operational scenarios and environments may impose additional requirements on the platform's capabilities including its positioning, navigation and mapping function. At the current time, drones, as UAS are often referred to, are in the spotlight almost daily. One of the major issues is how to integrate these UAS in the national airspace (NAS). [1] addresses various issues and has sub-divided them into five categories: safety, security, air traffic, regulatory and socio-economic. As most issues are as of yet to be resolved, the Federal Aviation Administration (FAA) has currently only certified a limited amount of UAS (e.g. Aero Environment’s Puma and Insitu’s ScanEagle) under Part 21.25 rules for special purpose operations (aerial surveys in Alaska) mostly in Class G airspace. [2] Class G airspace is uncontrolled airspace with no air traffic controllers. The Puma had its first authorized flight on 8 June 2014.
Since no regulations are in place yet, the FAA only provides guidelines and policy regarding operation in the NAS. Non-commercial users can obtain a FAA Certificate of Authorization (COA) for operation of a single platform in a specific area. However, many UAS operators do operate model aircraft and multi-copters either following the model aircraft guidelines or even operating outside those rules. With respect to the latter category, many platforms are capable of being operated remotely (with or without first point of view equipment), semi-autonomous or fully autonomous.

Especially, current operations in the latter two autonomy categories (semi- and fully autonomous), rely heavily on the Global Positioning System (GPS). However, many environments still exist where GPS is only sparsely available or significantly deteriorated (so-called GPS challenging environments) such as urban environments, indoor environments and under-foliage. In these environments, a GPS positioning capability may be unattainable due to shadowing, signal attenuation, or multipath. A popular augmentation method for GPS in challenging environments is integration with an IMU. This integration method increases the system availability during GPS outages or reduced satellite visibility, and in specific configurations (ultra-tight integration) increases the GPS
receiver’s sensitivity to the point that it is capable of tracking signals in GPS challenging environments. Although a GPS/inertial integration allows for tracking of signals in many more environments, settings still exist where this is not a dependable option. Therefore, another strategy is needed that is not completely dependent on GPS, or any external signals, if navigation in any environment is desired.

1.1 Research Overview

This dissertation presents a solution that focuses on developing and providing a PNM capability that is available continuously as a UAS operates throughout and between outdoor and structured indoor environments. For this work, an indoor environment is considered structured if its walls are vertical and remain approximately parallel, while the floor is either roughly flat or slanted. In this type of environment, GPS will often only be sparsely available. Hence, in our proposed navigation architecture, additional information from laser range scanners and a camera will be used to increase the system PNM availability and accuracy in the GPS challenged indoor environment. Figure 2 shows our target multi-copter, and Figure 3, the operational scenario that will be used.

Figure 2. Target platform: Commercially available hexacopter (3DR Robotics): modified to accommodate the platform weigh (> 4kg).
The methodology implemented in this research is visualized in Figure 4. A more detailed description of the block diagram is provided in Chapter 5. When assessing the capabilities of each of the sensors used in the work, only the inertial produces data that is solely dependent on the motion of the platform and local gravity and is more or less unaffected by its surroundings. Therefore, the inertial is chosen to be the core sensor for this research. Chapter 5 will describe the basic integration mechanizations used to integrate measurements from GPS, the laser scanners and the monocular camera. Where possible, an error formulation through a complementary Kalman Filter (CKF) will be used that estimates the errors in the inertial measurements and feeds them back to the inertial mechanization. For this inertial error estimation method to function properly, preprocessing methods must be implemented that relate the sensors observables to the inertial measurements.
Using methods described by Farrell in [9], GPS carrier phase measurements can be used to yield information regarding the platform dynamics and directly relate to the outputs of the inertial, enabling the estimation of the dynamics of the IMU error states in the dynamics filter shown in Figure 4. Furthermore, when at least four pseudoranges are obtained from GPS satellites, an absolute global position can be obtained. This allows the PNM solution produced in this work to operate in a global coordinate frame instead of a local, relative frame, where possible of course.
As images are captured by the 2D monocular camera, they are processed to extract both point and line features from the observed environment. Using motion estimates from the inertial, these features are tracked between consecutive image frames. The apparent motion of the point and line features is used to drive a visual odometry procedure and, with aiding from one of the lasers, estimate true metric scale for the translational motion. The outputs of these two procedures estimate all of the platform's 6 degrees-of-freedom (6DOF): \((x, y, z, \phi, \theta, \psi)\). These are then mathematically related to the IMU measurements and used for inertial error state estimation in the dynamics filter.

The final type of sensor shown in Figure 4 is a laser range scanner. The mechanization implemented in this work makes use of three such sensors oriented at differing angles. A medium range laser scans in the platform’s xy-plane, and one short range laser scans in the platform’s yz-plane. The third laser, a short range scanner, is oriented at an angle between the other two scanners. This configuration is displayed below in Figure 5. Each of these lasers provides point cloud data representing a 3D 'slice' of the environment. These point clouds are pre-processed and then analyzed. The downward and angled lasers are used to obtain altitude and attitude estimates for the platform. The forward-looking laser is used to perform a Simultaneous Localization and Mapping (SLAM) method to produce estimates for the platforms' 3 degrees-of-freedom (3DOF): \((x, y, \psi)\). Upon combining the estimates produced by all of the laser methods, a 6DOF estimate is observed. This 6DOF estimate can be compared with the IMU's measurements to aid in inertial error state estimation.
Through the use of a wide array of sensors, a dependable estimate for all 6DOF of the platform is obtained. From this precise positioning and attitude information, each collected laser scan is transformed into the navigation frame of the platform and cumulatively accrued. This creates a point cloud that is representative of the environment of operation for the platform. Due to the natural dynamics of a multi-copter, dynamic profiles defined by the flight planning computer, and the differing orientations of the lasers, the created point cloud equates to a 3D map of the environment. Therefore, as the mechanization shown in Figure 4 provides absolute positioning, a 6DOF estimate, and a 3D map, the desired PNM solution is achieved.

1.2 Contributions

The main goal of this dissertation is to provide a multi-copter with a dm-level relative PNM solution in, and between, outdoor and structured indoor environments. To achieve this, multiple novel techniques are derived and implemented that contribute to the current state-of-the-art. This research produces the following core contributions:
1. The design, development and verification of a robust dual laser-based altitude estimator. This method provides consistent cm-level accuracy during platform attitude changes and when non-level terrain is encountered.

2. The design, development and verification of a dual laser-based attitude estimator that requires the monitoring of only two non-parallel planar surfaces to resolve all three attitude states. This method produces attitude estimation capabilities similar to that of a low-cost commercial IMU.

3. The design, development and verification of an extended 3DOF SLAM algorithm that is capable of dealing with larger attitude excursions.

4. The design and analysis of a wavelet line extraction procedure for extracting line features in a 2D image.

5. The design and development of a data association method for point features using triangulation and nearest neighbor search.

6. The design, development and verification of a new visual odometry (VO) technique that uses a laser scanner to estimate the unknown depth and is not dependent on a feature map or extensive filtering techniques. Meter-level translational motion estimates were achieved over limited operational durations.

7. The development of a technique used to calibrate laser/camera point pairs.

8. The design, development and verification of a multi-sensor integration structure to combine inertial, GPS, laser, and camera measurements into a single 6DOF solution. This integration produced dm-level relative positioning and m-level absolute positioning.
9. The design, development and verification of a method that uses the output of (8) to generate a map of the environment.

1.3 Dissertation Structure

The extent of this dissertation is contained in the following 9 chapters. The subsequent chapter, Chapter 2, discusses important background information related to this research. Multiple systems and procedures are explored and the current state of the art of related topics is examined. Chapters 3 and 4 lay out the methods used in this work relating to the monocular camera and the laser scanners. Each technique presented in Chapters 3 and 4 was thoroughly tested and the results are shown. Next, Chapter 5 divulges the full methodology used for sensor integration in this work. Implementation details and practical application issues are discussed. Chapter 6 provides details of the data collection platform and the sensors used in this work. Chapter 7 presents and analyzes experimental results of the PNM capabilities produced by the system described throughout this dissertation. Finally, Chapters 8 and 9 discuss the conclusions attained through this research and suggested future work, respectively.
CHAPTER 2: BACKGROUND

This chapter delves into various aspects of the current state of the art for multiple systems and methods associated with proposed PNM capabilities. Processing methods, implementation issues, strengths and weaknesses are addressed with respect to GPS, inertial measurement units (IMUs), digital cameras, and laser scanners. For both lasers and cameras, techniques associated with SLAM are examined. With respect to sensor integration, Bayesian estimation methods are reviewed. Lastly, internal camera parameters and camera calibrations are discussed with respect to implementing visual sensors in PNM related procedures.

2.1 Global Navigation Satellite Systems

In a Global Navigation Satellite System (GNSS) a user receives radio frequency (RF) signals transmitted from orbiting satellites and uses these signals to make range measurements to the satellites. Together, with the satellites’ orbital parameters, the user can then use these signals to compute a position, velocity and timing solution.

There are currently four GNSS's that are either fully or partially operational. These include China's Compass, Russia's GLONASS, the European Union's Galileo, and the United States' Global Positioning System (GPS). The first of these GNSS systems to be made fully operational was GPS in 1973 after years of research to meet a need for an accurate and reliable navigation system for the US military.[10] The launch of GPS completely revolutionized the field of navigation, as no system before it could come close to providing its global accuracy with such availability. GPS remains the most well
developed and readily available GNSS today, and therefore, will be the GNSS of choice for this research.

Many of the details of GPS can be found in reference texts such as Misra [11]. Within the context of this dissertation it is important to note that GPS basically provides us with orbital parameters of satellites, unambiguous pseudorange measurements with dm-level noise error, and ambiguous carrier-phase range measurements with mm-level noise.

The global navigation capabilities that GPS has provided cannot be overstated, but as laid out by Volpe in [8], there are many situations in which this system will struggle. Similar to all RF based systems, its performance is directly correlated to the strength and quality of the observed signals transmitted by each of its satellites. As these broadcasted signals propagate through certain media, the resultant attenuation can decrease their observed signal to noise ratio (SNR), making them difficult to process and track, or even degrade them to the point of making the signal indistinguishable from the noise around it. Another issue that arises in many environments is multipath, which can cause the receiver to observe multiple copies of the same signal. These signal attenuation and multipath problems can degrade performance or even deny position acquisition capabilities entirely. This is common when signals are forced through and around media such as buildings, dense foliage, water, or the earth's surface. Therefore, although GPS is a viable answer to many navigation related problems, it cannot be relied upon in a standalone manner for navigation in the aforementioned environments. Consequently, other means of navigation must be used to complement GPS as is the topic of this dissertation.
2.2 Inertial Navigation Systems

According to Titterton and Weston [12], an Inertial Navigation System (INS) must be able to accomplish five tasks: (i) determine its relative attitude by measuring angular accelerations, $\omega$, using gyroscopes, (ii) measure the specific forces (or non-gravitational forces), $f$, exerted on the system using a set of accelerometers, (iii) use the attitude estimate to resolve the specific forces into a reference frame, (iv) apply knowledge of the gravitational field, and (v) obtain estimates of the system position and velocity in a navigation frame by integrating the resolved specific forces.

Originally, inertial systems were gimbaled to keep the accelerometers aligned with a chosen reference frame (e.g. the navigation frame). Nowadays, most inertial systems are strapdown systems that perform the accelerometer rotations computationally rather than mechanically. Strapdown inertial systems are smaller and much less mechanically complex than gimbaled systems. Nearly all INS's today implement a strapdown mechanization which follows the basic functions shown in Figure 6. The actual sensor package of an INS is an IMU. An IMU uses 3 orthogonal gyroscopes and accelerometers to measure the necessary angular accelerations and specific forces. Two types of IMU exist; one consisting of rate-gyros and one consisting of rate-integrating gyros. The basic measurements of the former include angular rates, $\omega^b$, and specific forces, $f^b$, expressed in the body frame indicated by superscript ‘b’. The basic measurements of the latter include $\Delta \theta^b$ and $\Delta v^b$ given by:
\[
\Delta v = \int_{t-\Delta t}^{t} f^b \, dt
\]  \hspace{1cm} (2.1)

and angular acceleration, \(\Delta \theta\):

\[
\Delta \theta = \int_{t-\Delta t}^{t} \omega_i^b \, dt
\]  \hspace{1cm} (2.2)

**Figure 6. Strapdown inertial navigation system mechanization. (adapted from [12])**

While high performance INSs can be very accurate, many error sources exist and have been documented in great detail [12]. For lower-cost IMUs, the error model can be simplified to include only the most dominant errors: position displacement error, \(\delta r\), velocity error, \(\delta v\), tilt or miss-orientation, \(\Psi\), gyro bias, \(\delta \omega^b\), and accelerometer bias, \(\delta f^b\).

The dynamic behavior of these errors is given by the following equations [9]:
\[
\delta \dot{\Psi}^n = [\Gamma^n \times] \dot{\Psi} + C^b_b \delta f^b \\
\dot{\Psi} = -\omega^{n}_l \times \Psi - C^b_b \delta \omega^b_{lb}
\] (2.3) (2.4)

If the INS is operating in a standalone manner, a calibration process can be performed to estimate some of these errors, however, since the errors change over time they can never be fully eliminated without integration with other navigation aids. By examining the feedback paths of errors in Figure 6 and Figure 7 short term error growth rates can be calculated and their dominant terms are displayed in Table 1 with 't' representing time. Based on the linear and exponential error growth rates produced by each of the different error sources combined with the knowledge that these errors can never be fully eliminated, it can unequivocally be stated that without aiding from external sensors, the performance of all INSs will degrade over time and, eventually, becomes unavailable for the a vehicles navigation function.

**Table 1. Dominant short term inertial error growth rates.**

<table>
<thead>
<tr>
<th>Error Source</th>
<th>Symbol</th>
<th>Position Error</th>
<th>Orientation Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Position Error</td>
<td>(\delta r_0)</td>
<td>(\delta r_0)</td>
<td>-</td>
</tr>
<tr>
<td>Initial Velocity Error</td>
<td>(\delta v_0)</td>
<td>(t)</td>
<td>-</td>
</tr>
<tr>
<td>Initial Attitude Error</td>
<td>(\delta \theta_0)</td>
<td>(t^2)</td>
<td>(\delta \theta_0)</td>
</tr>
<tr>
<td>Accelerometer Bias</td>
<td>(\delta f)</td>
<td>(t^2)</td>
<td>(t^2)</td>
</tr>
<tr>
<td>Gyroscope Bias</td>
<td>(\delta \omega)</td>
<td>(t^3)</td>
<td>(t)</td>
</tr>
</tbody>
</table>
2.3 GPS/Inertial Integration

Sections 2.1 and 2.2 describe the inner workings, strengths, and weakness of GPS and Inertial based navigation systems. Based on those discussions, the nearly infinite operational continuity and the short term stability and accuracy of an IMU strongly complements the weaknesses of GPS. Therefore, one of the most popular augmentation methods for GPS in challenging environments is integration with an IMU. This integration method increases the systems availability during GPS outages or reduced satellite visibility, and in specific configurations (ultra-tight integration) increases the GPS receiver’s sensitivity to the point of making it capable of tracking signals in some indoor environments. One example of a tight-integration mechanization is laid out by Farrell in [9] for estimation of all 6DOF of a platform. To achieve this, [9] makes use of two separate estimators: a dynamics filter and a position filter.
The dynamics filter uses precise carrier phase measurements from the signals transmitted by each visible GPS satellite to estimate the dynamics of the inertial error states shown in Figure 7, except for the position error. The dynamics filter state vector is, thus, given by:

\[
x_1 = [\delta \psi^n \delta \psi_{nb} \delta \omega^b_{ib} \delta f_b]
\]  \hspace{1cm} (2.5)

After the estimator update cycle, the error estimates are fed back into the inertial mechanization at the appropriate points shown in Figure 7. Furthermore, in this tight integration method, carrier phase measurements from any number of observed satellites can be included to aid in the pose estimation (position and attitude), compared to the conventional standalone GPS pseudorange based least squares estimator that requires a minimum of 4 satellites. Since the number of visible satellites can easily be smaller than 4 in challenging environments, tight integration enables continued use of GPS in a GPS/INS solution.

The second filter is a simple position estimator that uses the velocity estimates from the dynamics estimator as a forcing function to determine position change from the previous position state update.

2.4 Bayesian Estimators for Sensor Integration

Maybeck states in [13] that Bayesian estimators such as Kalman filters and particle filters were developed as optimal data processing algorithms that use stochastic models to estimate the solutions of a set of linear differential equations. These estimators have proven very effective within the field of navigation for combining data from multiple different
sensors to obtain position, velocity, attitude and time solutions with improved navigation performance (e.g. accuracy, integrity, continuity and availability). In a general navigation estimator we define a desired state vector, \( \mathbf{x} \), and use a set of measurements from a navigation aid, \( \mathbf{z} \), to estimate the state vector. Examples of state vectors are the state vector of the dynamics function given by Equation (2.5) in the previous section or the pose of the platform given by \( \mathbf{x} = [x \ y \ z \ \varphi \ \theta \ \psi]^T \). The measurements are related to the state vector through some linear or non-linear function such as:

\[
\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{v}
\]  

(2.6)

Where \( \mathbf{h}(\mathbf{x}) \) represents the deterministic portion of the measurement equation and \( \mathbf{v} \) is the additive measurement noise component with covariance matrix \( \mathbf{R}_k \). Equation (2.6) is referred to as the measurement equation. Stochastically, Equation (2.6) can be expressed by its conditional probability density function, \( p(\mathbf{z}|\mathbf{x}) \). For example, if the noise component is a multi-dimensional Normal distributed random variable or \( \mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \), the likelihood function \( p(\mathbf{z}|\mathbf{x}) \) is given by:

\[
p(\mathbf{z}|\mathbf{x}) = \det(2\pi \mathbf{R})^{-1/2} \exp \left\{ -\frac{1}{2} (\mathbf{z} - \mathbf{h}(\mathbf{x}))^T \mathbf{R}^{-1} (\mathbf{z} - \mathbf{h}(\mathbf{x})) \right\}
\]  

(2.7)

The goal of the estimation process is to find the most likely value of \( \mathbf{x} \) minimizing some cost criterion (e.g. minimum mean squared error). Therefore, one is more interested in the form of the \textit{a posteriori} density function, \( p(\mathbf{x}|\mathbf{z}) \), than the likelihood function. An additional form of the \textit{a posteriori}, often used in robotics, includes control actions or odometry, \( \mathbf{u} \), yielding \( p(\mathbf{x}|\mathbf{z}, \mathbf{u}) \). While the latter is often known, the former must be
derived. When time history is available, the estimation process can be extended to finding:

\[ p(x_k|z_1, z_2, ... z_k, u_1, u_2, ... u_k) \]

or, for short, \( p(x_k|z_{1:k}, u_{1:k}) \). To derive an expression for the \textit{a posteriori} distribution, it is typically assumed that the process is Markov: the current state only depends on the previous state and current control input, and that the current measurements only depend on the current state. Omitting the control input, the basic equation for the \textit{a posteriori} can be derived applying both Bayes theorem and the Markov assumption [14][15]:

\[
p(x_k|z_{1:k}) = \eta p(z_k|x_k) \int p(x_k|x_{k-1}) p(x_{k-1}|z_{1:k-1}) \, dx_{k-1}
\]

(2.8)

where \( \eta = p(z_k|z_{1:k-1}) = \int p(z_k|x_k)p(x_k|z_{1:k-1}) \) is a normalizing constant and \( p(x_k|x_{k-1}) \) is referred to as the state transition probability density function describing the dynamics of the models as well as the randomness of that dynamics model. In the robotics community, the \textit{a posteriori} distribution is often referred to as the belief function:

\[
bel(x_k) = p(x_k|z_{1:k}), \text{ and the integral } \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1}) \, dx_{k-1}
\]

as the prediction term \( \overline{bel}(x_k) \). A typical Bayesian filter now consists of the following two steps, the \textit{prediction} step:

\[
\overline{bel}(x_k) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1}) \, dx_{k-1}
\]

(2.9)

and the \textit{update} step:

\[
bel(x_k) = \eta p(z_k|x_k)\overline{bel}(x_k)
\]

(2.10)
When selecting the MMSE as the cost criterion the optimal solution of the Bayesian filter would be given by the expected value of the belief function, or \( E\{\text{bel}(x_k)\} \).

In case the PDFs are linear multivariate Normal (Gaussian) distributions, the PDFs for the prediction and the estimate in the Bayesian filter are multivariate Normal distributions and completely defined by their mean and covariance matrix. For example, for a linear measurement equation:

\[
z_k = H x_k + v_k \tag{2.11}
\]

where \( v_k \sim \mathcal{N}(0, R_k) \), and a linear state propagation equation:

\[
x_k = \Phi x_{k-1} + w_k \tag{2.12}
\]

where \( w_k \sim \mathcal{N}(0, Q_k) \) and \( Q_k \) is the system noise or process noise covariance matrix. Now, the following PDFs in the Bayesian filter can be defined:

\[
p(z_k | x_k) = \mathcal{N}(H x_k, R_k) \tag{2.13}
\]

And

\[
p(x_k | x_{k-1}) = \mathcal{N}(\Phi x_{k-1}, Q_k) \tag{2.14}
\]

Substituting equations (2.13) and (2.14) in the Bayesian filter, leads to expressions for the prediction and update estimate PDFs. For a detailed derivation the reader is referred to [14] and [16].
\[ \overline{bel}(x_k) = \mathcal{N}(\hat{x}_k, P_k^-) = \mathcal{N}(\Phi x_{k-1}, \Phi P_{k-1} \Phi^T + Q_k) \] (2.15)

\[ bel(x_k) = \mathcal{N}(\hat{x}_k, P_k^-) = \mathcal{N}(\hat{x}_k^- + K_k (z_k - H \hat{x}_k^-), (I - K_k H)P_k^-) \] (2.16)

Where \( K_k = P_k^- H^T (HP_k^- H^T + R_k)^{-1} \) is referred to as the Kalman gain as it regulates the importance of measurements with respect to the prediction.

Equations (2.15) and (2.16) show that the linear Kalman filter (KF) is just a special case of the Bayesian filter, implemented by only considering the mean and covariance rather than the whole distribution as they define the Normal distribution uniquely.

KF prediction stage:

\[ \hat{x}_k^- = \Phi x_{k-1} \] (2.17)

\[ P_k^- = \Phi P_{k-1} \Phi^T + Q_k \] (2.18)

KF update stage:

\[ K_k = P_k^- H^T (HP_k^- H^T + R_k)^{-1} \] (2.19)

\[ \hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-) \] (2.20)

\[ P_k = (I - K_k H)P_k^- \] (2.21)

In addition to the KF, numerous other Kalman filter variations exist such as the Linearized Kalman Filter (LKF), the Extended Kalman Filter (EKF), and the Unscented Kalman Filter (UKF). Brown and Hwang [17] and Veth [18] discuss the differences and advantages between an LKF, EKF and UKF with respect to their use in navigation related operations. In addition to these KF variations, another popular class of filters are Particle
Filters (PF). This class of filters is described in great detail in [15] and [19]. In essence, the PF is a Monte-Carlo simulation technique, where we approximate the PDFs by picking samples or “particles” from a relevant distribution (e.g. the state transition distribution), computing appropriate weights for each sample dependent on the likelihood function, and computing the estimate by calculating the sample mean.

In terms of KFs, the LKF, EKF and UKF are designed to deal with a non-linear measurement and/or dynamics (state transition) model. The non-linear measurement model is shown in Equation (2.22) where \( h(x) \) represents a non-linear measurement. Similarly, a non-linear dynamics model can be setup: \( x_k = g(x_{k-1}) + w_k \). Both the LKF and EKF linearize non-linear measurements and/or state transition functions. However, where the LKF linearizes around a known trajectory, the EKF linearizes the non-linear function around the prediction, or

\[
H_k = \frac{\partial h_k}{\partial x} \bigg|_{x = \hat{x}_k}
\]

(2.22)

The UKF tries to capture the effects of the non-linearities on the PDFs, by approximating the covariance of the distribution by more than the typical covariance matrix.

When comparing these dissimilar mechanization styles of optimal filters, a key performance tradeoff can be established. As the estimated dynamics of a platform become exceedingly non-linear, the PF will perform the best followed by the UKF, EKF and LKF, respectively. This increase in performance for non-linear systems is inversely proportional to the filters computational complexity. Therefore, in a generalized sense, it can be stated
that exceedingly non-linear performance can be improved at the expense of added computational intensity.

When integrating measurements from multiple sensors, the point at which these measurements are combined in the mechanization must be chosen as well. Gebre-Egziabher [20] states that loose integration combines data at the position level, while tight and deep integration combine sensor data at the measurement level. Furthermore, deep integration, which is typically associated with GPS, allows the IMU data to aid in the receivers signal tracking loops. This enhances the receiver’s ability to track signals when it is experiencing high dynamics.

Finally, in integrated navigation systems, one often monitors the term $r_k = (z_k - H \hat{x}_k^\top)$ with associated covariance $S_k = H P_k H^T + R_k$. $r_k$ is referred to as the innovation or residual (although the word residual really refer to the quantity $(z_k - H x_k)$) and shows the difference between the measurements and the synthesized measurements using he filter’s internal models. If everything works correctly these residuals should be ‘white’ (normally distributed). Any non-whiteness can therefore be used to detect cases where the model no longer fits reality and the filter, thus, no longer operates in an optimal or even correct fashion.

2.5 Simultaneous Localization and Mapping

Thrun defines Simultaneous Localization and Mapping or SLAM in [14] as a way to enable operation of a platform at an unknown location in an environment where no $a$ priori knowledge is available for both the pose (position, attitude and heading) of the platform and the map of the environment. In SLAM the typical state vector consists of both
pose and the map. For example, for 2D motion of a ground vehicle and a 2D map with point features, the state vector would consist of \( \mathbf{x} = [x \ y \ \psi \ m_{1x} \ m_{1y} \ \ldots \ m_{Nx} \ m_{Ny}]^T \) where \( [m_{ix}, m_{iy}] \) represents the coordinates of the \( i^{th} \) feature in the map. Note that in this example the state vector grows as the number of features grows, which may be problematic. In terms of a Bayesian estimator, the SLAM problem comes down to finding:

\[
p(x_{1:k}, \mathbf{m}|z_{1:k}, \mathbf{u}_{1:k})
\]

The SLAM problem is solved in either an online SLAM fashion or a full SLAM manner. In the former, only the current pose and map are being estimated, whereas, in the latter, the complete history of poses is estimated (adjusted) in addition to the map. Basically, SLAM estimates of a platform's motion (from the filter’s prediction in combination with control inputs) and uses processed optical/laser data to identify features of interest. Multiple examples regarding the use of optical and laser data for this process will be presented in Section 2.5. These features of interests are then either associated with features in the existing map or added as new features to the map. The incremental accumulation of these observations allows for the creation of a constantly expanding map and, thus, a constantly expanding state vector. Simultaneously to the map creation, the platform localizes itself (estimates its pose) within the produced map relative to observed features. This establishes a correlation between feature positions and predicted platform states as they are used to derive one another.
Unfortunately, as only relative measurements are obtained during the typical SLAM process, errors will accumulate over time indicated by an increased error covariance. This unavoidable issue can be mitigated through loop closure as discussed by Durrant-Whyte in [21]. In loop closure a platform observes a set of features, moves away from them, and at a later time comes back to these features and observes them again. This knowledge can then be used to estimate the drift due to accumulation of errors and results in a significant covariance reduction of not only the pose estimate, but also the map. Statistical adjustments of the whole trajectory can only be made if all previous and current platform states and feature locations are stored and can still be changed.

The literature on the various SLAM implementations is vast, and only a very small subset is mentioned here for reference. The application of an EKF to SLAM was first introduced by Cheeseman in [22] to produce EKF-SLAM where the map consists of features. Further information on EKF-SLAM can be found in Leonard [23]. In implementations of this method, the aforementioned procedure can be difficult due to its computational complexity and potentially large amounts of mapping data. From this, a significant amount of research has been invested into modifying elements of SLAM to be more applicable or computationally efficient. With respect to the constantly expanding state vector, Chekholov [24] explores procedures for efficiently adjusting and re-computing the platforms' state. Hoppe [25] and Clemente [26] investigate methods for handling large scale maps, including breaking them down into smaller, more manageable sub-maps. Another popular 2D SLAM method is FastSLAM [27]. This SLAM method
make used of a so-called Rao-Blackwellized filter [28]. In a Rao-Blackwellized filter, the platform’s path (pose) is separated from the map or

$$p(x_{1:k}, m_{1:M} | z_{1:k}, u_{1:k}) = p(x_{1:k} | z_{1:k}, u_{1:k}) \prod_{i=1}^{M} p(m_i | z_{1:k}, x_{1:k})$$  \hspace{1cm} (2.24)

Now, a PF is implemented using the pose as its state vector and each PF will have M associated small-size EKFs for each landmark.

Rather than features, SLAM methods have been implemented based on occupancy grids by Elfes [29] and Moravec [30]. Occupancy grids take an environment and represent it through evenly spaced grid boxes. These grid elements take small sections of the environment and assign them as unknown, occupied (containing features), or unoccupied (vacant). An example of this is shown in Figure 8.

![Figure 8. Occupancy grid.](image-url)
All spaces begin in the unknown state. As observations of the environment are captured, surveyed grid boxes will be set to indicate whether they are occupied or unoccupied. Based on the efficiency provided through occupancy grids, they are one of the most common map creation techniques used for SLAM. FastSLAM methods have been described and implemented that use 2D occupancy grids to describe the map rather than 2D features in Hahnel [31] and Eliazar [32]. An improved method of [31] was used in the Robotic Operating System in its ‘gmapping’ stack by Grisetti et al [33]. The occupancy grid SLAM methods are used in this dissertation using laser scans.

2.6 Optical Navigation

Aside from inertial and GPS, a third class of sensor technology will be applied to our multi-copter UAV platform: optical navigation. Optical based navigation methods use Electro-Optical (EO) sensors to capture information about the surrounding environment, analyze changes in the observed environment over time and use that information to estimate perceived motion and, potentially, build a map of the environment using SLAM methods. Although there are multiple EO sensors that can be used for navigation, this dissertation focuses on two of the most common, laser range scanners and 2D digital cameras.

2.6.1 Laser Based Navigation

Typical laser range scanners (LRS) measure ranges at known angles in a 2D plane. Using polar coordinates, the resulting measurements can be easily converted to a set of 2D points referred to as a 2D point cloud. The points associated with each scan of the LRS represents a “slice” of the surrounding environment. The use of 2D lasers for 2D
navigation has been explored widely for robotic ground vehicles. 2D navigation requires estimation of 3DOF. Three examples of this can be referenced through the work done by Pfister [34], Bates [35] and Soloviev [36], which monitor features such as lines or points to estimate a platform's 2D position and heading. Grisetti [37], Blösch [38], Shaojie [39], and Slawomir [40] have used 2D laser scanners through multiple approaches to perform 3DOF SLAM. One specific approach is described by Kohlbrecher [41], which performs pose estimation through bilinear interpolation and scan-matching. In scan-matching, the 2D point cloud from the laser range scanner is matched with the map of the environment. Since the 2D point cloud will include points that are already represented in the map as well as new points, the map can be updated. The map of the environment is typically constructed using an occupancy grid. The method described in [41] produces a 2D map with corresponding 3DOF motion estimates and is part of the Hector SLAM package in ROS, a tool that will be used in the implementation described in this work. To expand laser measurements for estimation of 6DOF, utilization of 3D imaging sensors such as the Swissranger, PMD, the Kinect or even the new Occipital sensors were explored by Horn in [42], Venable [43] and Uijt de Haag [44]. These sensors produce 3D point clouds which are processed to extract and monitor 3D features over time to yield a 6DOF estimate. While 6DOF estimation cannot be performed using a 2D laser scanner in a direct manner, research has been conducted to develop methods to turn 2D laser data into 3D data. One way this can be accomplished is by placing the sensor on a motor and physically rotating it or adding an additional rotating mirror, such as in the method described by Nüchter in [45]. Similarly to placing the sensor on a physically rotating motor, Soloviev assumes a structured
environment in [46] and uses a 2D scanner on an aerial platform. The motion of the platform itself yields the capability of observing parts of a 3D environment by combining multiple scans. This process can be visualized in Figure 9.

Figure 9. 2D Laser scanner used to provide a 3D point cloud.

The 3DOF SLAM algorithm has also been expanded in [47] and [48] to yield a 2D map while still producing a 6DOF estimate for the states of an aerial vehicle. To achieve this, IMU's are integrated into the algorithm, and the laser scanners were modified to deflect a small portion of the laser's scan downward using a mirror to get an altitude measurement \((z)\). In [47], Shen obtains 2D map and 3DOF state estimates through a 2D SLAM process using an Iterative Closest Point (ICP) algorithm for scan matching. This leaves the computation of the remaining attitude states of pitch and roll \((\phi, \theta)\) to be derived directly from the inertial measurements. With two states being derived directly from the inertial, the problem is reduced to a 4DOF incremental motion estimation problem. The integration of the inertial with the lasers altitude measurements and baro altimeter measurements allow [47] to offer a mapping capability that spans multiple floors of a
building. With each determined floor change that occurs, [47] maps that floor using a separate 2D occupancy grid. This allows for 2D map creation of a multi-story building and loop-closure between those levels. In a similar way, Grzonka implements a multi-level SLAM approach in [48]. This procedure creates multiple stacked 2D maps differentiated by predetermined discrete altitudes to create a pseudo-3D map.

For this work, the 3DOF SLAM procedure expands off of the work done in [47] and [48] in various ways. To improve pitch and roll estimation, multiple attitude determination algorithms are fused together to attain a single, more stable solution. For altitude estimation, a more robust and more accurate dual laser based estimation is implemented. Finally, both [47] and [48] assume a level platform, which in cases, can be invalid. To mitigate the need for this assumption, a leveling procedure is applied.

2.6.2 Vision Based Pose Estimation

The focus of this dissertation is on operation in an unknown environment; in those environments, 2D digital cameras or vision sensors can be used to estimate the motion or pose of the platform as well as the environment or map. When cameras are used to estimate incremental pose and use this incremental pose to estimate the platforms position and attitude, we refer to the methods as visual odometry. When both pose and map are estimated simultaneously, the methods are referred to as Visual-SLAM methods [49]. This can be accomplished using either optical flow procedures such as Hrabar [50] or feature tracking methods. This work will focus only on feature tracking methods.
2.6.2.1 Structure From Motion Related Methods.

In cases where a single camera (monocular) is used, texts such as Zisserman [51] and Ma [52] show that knowledge of common features between consecutive frames allows the user to estimate incremental rotation and incremental translation from one frame to the next. The inverse of this procedure is often referred to as structure-from-motion (SFM), and is commonly used (e.g. see Beardsley [53]) to obtain 3D maps or imagery of an environment given known motion of a camera. The basic multi-view geometry used for SFM is shown in Figure 10.

![Figure 10. Epipolar constraint from multi-view geometry.](image)

From the geometry in Figure 10 it follows that the following equality must hold:

\[
\mathbf{p}_i^c(t_k) - \frac{C_n^c(t_k)C_b^c(t_{k-1})}{2} \mathbf{p}_i^c(t_k) + \frac{C_n^c(t_k)\Delta \mathbf{r}}{3} = 0
\]

(2.25)

Where the superscript ‘c’ indicates the camera frame which is directly related to the platform’s body frame through the sensor misorientation and lever arm. For each point
feature, \( \mathbf{p}_i \), in the observed scene, the vision sensor measures the projection of this point on the image plane, \( \mathbf{p}^c_{i,i} \). Given two views at different time epochs and the incremental motion (translation and rotation) between both views, \( \Delta \mathbf{r} \) and \( \mathbf{C}^c_{k-1} \), the vectors \( \mathbf{p}^c_{i,i}(t_{k-1}) \), \( \mathbf{p}^c_{i,i}(t_k) \), and \( \Delta \mathbf{r} \) will all lay in the same plane \( V \). Therefore, the following constraint equation must hold in addition to Equation (2.25):

\[
\mathbf{p}^c_{i,i}(t_k) \cdot \mathbf{n}_V = 0 \tag{2.26}
\]

Since the normal vector \( \mathbf{n}_V \) can also be defined as a non-unit vector, \( \mathbf{n}_V = \Delta \mathbf{r} \times \mathbf{C}^c_{t_k-1} \mathbf{e}(t_{k-1}) \), Equation (2.25) can be rewritten as:

\[
\mathbf{p}^c_{i,i}(t_k) \cdot \{ \Delta \mathbf{r} \times \mathbf{C}^c_{t_k-1} \mathbf{p}^c_{i,i}(t_{k-1}) \} = \mathbf{p}^c_{i,i}(t_k)^T \{ \Delta \mathbf{r} \times \mathbf{C}^c_{t_k-1} \} \mathbf{p}^c_{i,i}(t_{k-1}) = 0 \tag{2.27}
\]

where \( \mathbf{E} = \{ \Delta \mathbf{r} \times \mathbf{C}^c_{t_k-1} \} \) is referred to as the essential matrix. Equation (2.27) is referred to as the epipolar constraint equation. Matrix \( \mathbf{E} \) consists of both the translation and the rotation. Equation (2.27) can be rewritten, as shown in [52], using the Kronecker product:

\[
\left[ \mathbf{p}^b_{i,i}(t_k) \otimes \mathbf{p}^b_{i,i}(t_{k-1}) \right]^T \mathbf{E}^s = 0 \tag{2.28}
\]

where \( \mathbf{E}^s \) is a row vector representation of \( \mathbf{E} \), or:

\[
\mathbf{E}^s = [e_{11} \ e_{21} \ e_{31} \ e_{12} \ e_{22} \ e_{32} \ e_{13} \ e_{23} \ e_{33}] \tag{2.29}
\]
Equation (2.29) can be solved using a large variety of algorithms depending on any additional constraints one may have in the environment (e.g. multiple points lay in the same plane [54]). The eight-point algorithm discussed in [52] is one popular algorithm that has been implemented here and discussed in more detail in Chapter 4.

Though a camera can measure the direction, \( e_i \), to a feature in the observed environment, it cannot measure the range, \( \rho_i \), to that feature:

\[
p_i = \rho_i e_i^c
\]  

(2.30)

In SFM for 3D mapping purposes this means that, given sufficient views of the environment, a 3D structure can be created, but with unknown scale. So, since the 2D camera features lack distance information, additional knowledge is required to solve for this depth. Schener [55] and Lucas [56] solve this through implementation of a stereo camera pair rather than a monocular camera. Stereo vision uses two cameras (left and right) that make images of the observed scene at the same time (see the geometry in Figure 11).
Given a known relative position and orientation between the cameras, an estimate can be made regarding the distance to point $p_f^c(t_k)$. Now, the change in location of the observed point features from consecutive frame pairs can be used to estimate the user pose change. Moravec has illustrated this method in his 1980 Ph.D. dissertation for robotic applications [57]. A method similar to the one he described has been used in the Mars rovers since 2004. Other good references on stereo visual odometry are Sinopoli [58] and Nister [59][60].

In scenarios where multiple cameras are not available, monocular solutions have been explored. The authors in [59] discuss a monocular solution that takes the depth as an additional parameter to estimate; note that their proposed method only works when some of the features are co-planar. Soloviev, et al. [61] use a similar method to determine the depth using a vision/laser range scanner integration method. Scaramuzza [62] describes a visual odometry method that uses an omni-directional camera rather than a regular camera with a limited field-of-view.

As mentioned earlier, vision sensor data has also been applied to the SLAM problem. Elinas [63][64] and Mei [65] perform visual-SLAM (VSLAM) using a stereo pair to resolve depth. In [66], Andreasson successfully implements VSLAM with an omni-directional camera. Depth is resolved by Andreasson through multiple feature observations combined with robotic odometry measurements. Using only a monocular camera, Davison [67][68], Eade [69], Klein [70] and Chekhlov [71] have described different methods to implement VSLAM and estimate feature depth. Klein and Murray’s Parallel Tracking and
Mapping (PTAM) method has been incorporated as a ROS package as well. To aid monocular VSLAM techniques, Pinies [72] and Burschaka [73] integrate inertial data into the solution. The inertial data is used to estimate platform odometry between image frames and VSLAM updates. This integration demonstrates improvements in metric scale and feature depth estimation. As most VSLAM implementations operate using point features, the applications of different features, such as lines, have been examined in Lemaire [74], Smith [75] and Jeong [76]. These works indicate that increasing feature diversity improves capabilities. Line features will be an important component in the visual odometry method detailed in Chapter 4.

In [18], Veth introduces an alternative method to perform vision-based 3D motion estimation through feature tracking and IMU integration using an EKF implementation. The feature extraction method used in his work is based on Lowe's [77][78] Scale Invariant Feature Transform (SIFT). In SIFT, each extracted point comes with a 128-element feature vector that can be used for feature association from one frame to the next. Alternative point feature extraction methods that could be implemented are Speeded Up Robust Features (SURF) [79], the Hough transform [80], and Harris [81] and FAST [82][83] corner detectors; but these features do not include the 128-element feature vector, and, therefore require additional methods to perform feature association. In his master's thesis, the author of this dissertation developed a tight integration method involving a 2D camera and IMU. In [84], Dill provides the mathematical structure and methods for an integration and relates the IMU errors to an additional set of constrain equations. Besides SIFT features, [84]
addresses the use of Harris and FAST corner features as well as line features. Elements of this vision/inertial integration method will be re-used in this work.

2.6.2.2 Intrinsic Camera Parameters and Camera Calibration

For the aforementioned vision based pose estimation techniques, it is necessary to establish the geometry between the camera's 2D image frame and a desired 3D navigation reference frame. Both Ma [52] and Zisserman [51] lay out this relationship and highlight the importance of knowing some of the camera's internal parameters. Consider the geometry as laid out in Figure 12.

\[ \begin{align*}
\mathbf{p}_i & \text{ point in the navigation frame given by the 3x1 vector, } p^h_i, \text{ is projected to the corresponding 3D point, } p^c_i, \text{ in the camera's sensor frame. This projection is accomplished via the rotation, } C^c_h, \text{ and translation, } \Delta r^c. 
\end{align*} \]
\[ \mathbf{p}_i^c = \mathbf{C}_n^c \mathbf{p}_i^n + \Delta \mathbf{r}^c \rightarrow \mathbf{p}_i^c = \begin{bmatrix} \mathbf{C}_n^c & \Delta \mathbf{r}^c \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_i^n \\ 1 \end{bmatrix} \] (2.31)

A typical camera observation of point \( \mathbf{p}_i^c \) is its projection on the image frame, \( \mathbf{p}_i^c \). A typical point on the image plane is given by its row \((x)\) and column \((y)\) coordinate as shown in Figure 12. In that case, the vector, \( \mathbf{p}_{i,i}^c \), would be given by:

\[ \mathbf{p}_{i,i}^c = \begin{bmatrix} x_{i,i} \\ y_{i,i} \\ f \end{bmatrix} \] (2.32)

For a pinhole camera, this projection can be described through the following equation:

\[ \mathbf{p}_{i,i}^c = \begin{bmatrix} x_{i,i} \\ y_{i,i} \\ f \end{bmatrix} = \begin{bmatrix} f \\ f \\ f \\ 1 \end{bmatrix} \mathbf{p}_i^c \] (2.33)

Combining Equations (2.31) and (2.33), the following relationship is established:

\[ \lambda \mathbf{p}_i = \lambda \begin{bmatrix} x_{i,i} \\ y_{i,i} \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{C}_n^c & \Delta \mathbf{r}^c \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_i^n \\ 1 \end{bmatrix} \] (2.34)

where \( \Pi_0 \) is referred to as the standard projection matrix and \( \mathbf{K}_f \) as the camera matrix for an ideal perspective camera. Equation (2.34) establishes the projective relationship necessary to perform a transformation between the 3D navigation frame and the 2D image frame assuming an ideal perspective camera. If this is not the case, more of the intrinsic
parameters must be taken into account. The first intrinsic parameter introduced for this case is the principle point (or the center offset). This point \((o_x,o_y)\) is located at the intersection between the camera's sensor frame z-axis and the image plane. The next intrinsic parameters are the scale factors \(s_x\) and \(s_y\). These scale factors are used to describe the horizontal and vertical lengths of the image, respectively. The last intrinsic parameter is the skew-factor, \(s_\theta\), which describes the observed angle between the x and y axes of the image frame. Using these intrinsic parameters, Equation (2.35) can be derived to describe the non-ideal perspective camera matrix, \(K_s\):

\[
K_s = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}
\]  

(2.35)

Combining Equations (2.34) and (2.35), a new matrix \(K\) is derived in Equation (2.36). This new matrix contains all intrinsic camera parameters and is known as the calibration matrix.

\[
K = K_s K_f = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f s_x & f s_\theta & o_x \\ 0 & f s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}
\]  

(2.36)

With the derivation of a calibration matrix that includes the variables necessary to deal with a non-ideal perspective camera, Equation (2.34) can be re-written in the form shown in Equation (2.37). This shows the full form of the mathematical model needed for the projection between the 3D navigation frame and the 2D image frame.
To obtain the intrinsic parameters shown in the calibration matrix, a camera calibration procedure must be accomplished. Moreover, Faugeras [85] demonstrates that a camera's intrinsic parameters can change over time for varying reasons. Therefore, a separate camera calibration was completed at the beginning of each data collection described in this work. The calibration used in this research was completed through the "Camera Calibration Toolbox for Matlab", developed by Jean-Yves Bouguet [86]. Through extracting grid corners present on the calibration checkerboard shown in Figure 13 for multiple images, Bouget obtains the cameras intrinsic parameters. For more information regarding this toolbox the reader is referred to [86].

![Extracted Grid Corners](image)

*Figure 13. Camera calibration.*
2.7 Coordinate Transformations

For multiple procedures in this dissertation, it is required to manipulate data between established coordinate frames. To do so, each coordinate frame must be related by a known, or obtained, rotation and translation. All of the data acquired in this work is collected by sensors that each have their own sensor frame denoted by \( (s) \). Through measured rotations, \( C_s^b \), and translations (lever arms), \( l \), data contained in each sensors' individual frame can be related to an established body frame through:

\[
p^b = C_s^b p^s + l
\]  
(2.38)

where \( p^s \) denotes a 3D data point in the sensor frame. For this work, the body frame, expressed as \( (b) \), is established as the inertial's sensor frame. This indicates that body frame measurements are only relative to the inertial sensor, and therefore, are in no way related to a locally static coordinate frame. All needed sensor to body frame rotations and lever arms for this work are found in Table 2 of Chapter 6.

As a platform traverses an environment, it is important to establish a fixed, local navigation frame \( (n) \). The rotation needed to relate this frame to the aforementioned body frame follows:

\[
C_n^b = \begin{bmatrix}
\cos \psi & -\sin \psi & 0 \\
\sin \psi & \cos \psi & 0 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
\cos \theta & 0 & -\sin \theta \\
0 & 1 & 0 \\
\sin \theta & 0 & \cos \theta
\end{bmatrix} \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \varphi & -\sin \varphi \\
0 & \sin \varphi & \cos \varphi
\end{bmatrix}
\]  
(2.39)

Values for pitch, roll and yaw are obtained through a cumulative attitude computation process. Often, an inertial units gyro measurements are used for this computation.
Furthermore, a navigation frame position, \( r \), is needed. Using these parameters, data in the body frame can be related to an established navigation frame.

\[
p^n = C^b_n p^b + r
\]  \hspace{1cm} (2.40)

For certain cases in this research, a locally-level frame (ll) is desirable. This frame is a version of the previously discussed body frame except it is level with respect to the navigation frame. As the body and locally level frames have a co-located origin, only an attitude correction is needed. Based on the body to navigation matrix of Equation (2.39), a body to locally-level rotation can be established through the product of the latter two matrices.

\[
C^n_b = \begin{bmatrix}
\cos \psi & -\sin \psi & 0 \\
\sin \psi & \cos \psi & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\cos \theta & 0 & -\sin \theta \\
0 & 1 & 0 \\
\sin \theta & 0 & \cos \theta
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \varphi & -\sin \varphi \\
0 & \sin \varphi & \cos \varphi
\end{bmatrix} = C^{ll}_b
\]  \hspace{1cm} (2.41)

This rotation allows data to be converted from the body frame to the locally level frame through:

\[
p^{ll} = C^{ll}_b p^b
\]  \hspace{1cm} (2.42)

For visualization purposes, the transformations established in Equations (2.38) through (2.42) are applied to laser scans from the downward and angled laser scanners. From left to right, the transformations in Figure 14 provide laser points in sensor frame coordinates, followed by the platforms' body frame, and finally the locally level frame. In
this instance the platform was experiencing a positive roll angle and was located in a
hallway. Attitude measurements for Figure 14 were derived from the inertial.

Figure 14. Coordinate transformations for the downward and angled laser scans-(a) Left: sensor frame; (b) Middle: body frame; (c) Right: locally-level frame.
CHAPTER 3: LASER-BASED ESTIMATION METHODS

The first step in the mechanization introduced in Chapter 1, is the pose estimator based solely on laser range scanner measurements and attitude estimates from the inertial system as highlighted in Figure 15.

In Section 3.1, two novel techniques for laser-based altitude and attitude estimation are discussed and evaluated. The chapter's other segment, Section 3.2, addresses the...
implementation of an improved 3DOF SLAM method. Together, these estimators provide a complete laser/inertial-based 3D position and attitude solution in a structured environment. Novel, in this method is that it uses three rather than one laser range scanner as depicted in Figure 15. For ease in this section, the three laser range scanners have been denoted as $F$ for the forward laser, $D$ for the downward laser, and $A$ for the angled or slanted laser.

3.1 Dual Laser Altitude and Attitude Estimation

In a structured environment such as the inside of a building with walls and floors that resemble planar surfaces, the laser point clouds from the $D$ and $A$ lasers, in the body or locally level frame, form two 'slices' through visible planar surfaces as is illustrated in Figure 16. These 'slices' can be used to obtain estimates of the platforms altitude above the observed planar surfaces and the attitude with respect to the navigation frame.

Figure 16. Point cloud 'slices' formed by the intersection of the downward and angle laser.
3.1.1 Altitude Estimation

The use of laser scanners for UAV altitude estimation in indoor environments was introduced by Shen [45] and Grzonka [48]. In their papers, subsets of laser range measurements from a horizontal laser scanner are deflected with a small mirror and used to estimate the height. Specifically, the method used in [45] implements a Kalman filter that uses the variations in the detected laser ranges to determine if the surface below the MAV is not "flat" enough. If this "flatness" criterion is not met, it uses the KF predictions for the height estimates. Using the detected range measurements would be equivalent to using the $N$ center laser ranges from laser scanner $D$. This detection methods will be referred to as method (i) throughout the remainder of this chapter. Define point-cloud points by their locally-level position $p_{li}^l = [x_{li}, y_{li}, z_{li}]^T$ and associated laser range scanner angles $\alpha_{li}$ where 's' indicates laser scanner $A$ or $D$. For laser scanners $A$ and $D$ the angular laser scan range is from $0^\circ$ to $240^\circ$ [87]. So, for method (i) the set of used points is given by $S_D = \{p_{li}^{ll} \}_{i = 1, \ldots, N_D}$ and $120^\circ - \epsilon < \alpha_{li} < 120^\circ + \epsilon$. In the presence of a ground surface which is planar in nature, the point clouds resulting from the intersections of the laser scans with the ground can be used to estimate altitude using two additional methods developed by the author and depicted in Figure 17.
These new methods, referred to as method (ii) and method (iii), first reduce the number of points in each of the slices to the set of points closest to nadir, or $S_s = \{p^l_{i,s} \mid i = 1, \ldots, N_s \text{ and } |y_{i,s}| < \delta y \}$. As it is crucial to only obtain points corresponding to the environments floor, points close to other environmental features (i.e. walls) are rejected. Therefore, $S_D$ and $S_A$ will likely not contain all floor points, however, they will only include points with a high level of integrity with respect to a floor correspondence. These subsets are visualized in red for both the $D$ and $A$ laser scans in Figure 18. Next, a line fit is performed on the points in $S_D$ and $S_A$ to verify if the points in each subset are sufficiently "flat" and to eliminate possible outliers.
Method (ii) uses only points from the $D$ laser and estimates the altitude by taking the average of the $z$-components of set $S_D$, or:

$$\hat{h} = \sum_{z_i \in S_D} z_i$$  \hspace{1cm} (3.1)$$

For method (iii), the fact that the points of $A$ and $D$ are both on the same planar surface $V$ is exploited. Method (iii) first estimates the centroid, $p^l_V$, and the normal vector, $n^l_V$, of planar surface $V$ which is derived from $S_A$ and $S_D$, and then uses the planar surface parameters to compute the shortest distance to the plane, $\rho_V$. The maximum likelihood estimate of $p^l_V$ is given by the point average:

$$p^l_V = \sum_{n_i^l \in S_AS_D} p^l_i$$  \hspace{1cm} (3.2)$$
If the ground is indeed a planar surface, all $N$ points in $S_A$ and $S_D$ should satisfy:

$$\mathbf{d}_i^u \cdot \mathbf{n}_V^u = 0 \quad \forall \mathbf{p}_i^u \in S_A, \quad S_D \tag{3.3}$$

where $\mathbf{d}_i^u = \mathbf{p}_i^u - \mathbf{p}_V^u$. Equation (3.3) is the rewritten in matrix form:

$$[\mathbf{d}_1^u \mathbf{d}_2^u \ldots \mathbf{d}_N^u] \mathbf{n}_V^{u,T} = \mathbf{D} \mathbf{n}_V^{u,T} = 0 \tag{3.4}$$

Through examination of Equation (3.4) it can be seen that the planes’ normal vector lies in the null space of $\mathbf{D}$. To find a solution to this homogeneous equation, we use the Singular Value Decomposition (SVD) of $\mathbf{D} = \mathbf{U} \mathbf{S} \mathbf{V}^T$. The columns of $\mathbf{V}$ that correspond to zero singular values (diagonal elements in $\mathbf{S}$) form an orthonormal basis for the null space of $\mathbf{D}$. Since the points in the point cloud will be noisy, Equation (3.4) will not be exactly homogeneous, and our estimate of the normal vector will equal the column of $\mathbf{V}$ corresponding to the smallest singular value:

$$\mathbf{\hat{n}}_V^u = \mathbf{V}_{min} \tag{3.5}$$

An example result of this procedure is illustrated in Figure 19 for actual laser scans. It is assumed that this plane represents a 'flat' section of the ground directly below the platform. Now, the shortest distance to the plane can be found through:

$$\hat{\rho}_V = \mathbf{\hat{n}}_V^u \cdot \mathbf{\hat{p}}_V^u \tag{3.6}$$
For an approximately level floor, the height above ground level is equal to the shortest distance to the planar surface or \( \hat{h} = \rho_V \). Conversely, when the floor is slanted rather than level as depicted in Figure 20, the height above ground level is no longer equal to the shortest distance to the planar surface. However, if we define a plumb-bob vector \( \mathbf{r}_n = [0 \ 0 \ h]^T \) to the planar surface and substitute this vector into Equation (3.3), the following expression can be derived:

\[
\begin{align*}
\mathbf{r}_n - \hat{\mathbf{p}}^V & \cdot \mathbf{n}_V = 0 \\
\Rightarrow \mathbf{r}_n \cdot \mathbf{n}_V & = \hat{\rho}_V
\end{align*}
\] (3.7)

Defining the z-component of the normal vector as \( z_{nV} \), substitution of \( \mathbf{r}_n = [0; 0; h] \) in Equation (3.7) yields the following expression for the height above ground estimate:

\[
\hat{h} = \frac{\hat{\rho}_V}{z_{nV}}
\] (3.8)
The heights derived in Equations (3.1) and (3.8) are heights with respect to the ground. In navigation one is more interested in the height with respect to a defined reference. In methods (i) and (ii) it will be hard to determine if the platform height varied or the floor/ground underneath the platform changed in height without additional information like the baro-altimeter (only for significantly large changes) or control inputs to the platform. Since method (iii) determines the slope of the floor at each time epoch, it is capable of estimating variations in the ground surface. Of course, if the variations are too rapid advanced methods, such as the ones in Vadlamani [88], must be used to determine this. These issues are illustrated in Figure 21 and Figure 22.
Figure 21. Similarities between height measurements while traversing a ramp (A), and changing altitude (B).

From Figure 22 it is clear that knowledge of the slope of the floor can be exploited to find the absolute height of the platform in a local coordinate frame, $\tilde{h}_a$. To do so, the geometry shown in Figure 23 is exploited.

Figure 22. Differences in floor plane normal vectors while traversing a ramp (A), and changing altitude (B).
First, a planar surface, $W$, can be defined between the platforms translation motion vector, $\Delta \mathbf{r}$, and the $z$ component of the local coordinate frame. The normal vector of the resulting plane is computed through:

$$\hat{\mathbf{n}}_W = \frac{\Delta \mathbf{r}_{2D}}{\|\Delta \mathbf{r}_{2D}\|} \times \mathbf{e}_z \quad (3.9)$$

where $\mathbf{e}_z = [0,0,1]^T$ and $\Delta \mathbf{r}_{2D}$ is the 2D projection (xy-plane) of the platforms full 3D translation. For this work, $\Delta \mathbf{r}_{2D}$, is derived from the 3DOF SLAM procedure discussed in Section 3.2. Using the obtained normal vector, and the normal vector derived from the grounds planar surface, $\mathbf{n}_v$, a vector can is computed that coincides with the intersection of planar surface $W$ and the ground.

$$\ell = \mathbf{n}_W \times \mathbf{n}_v \quad (3.10)$$

This is in turn used to compute the angle between the level floor and the surface to which the platform is traversing:
Through manipulating terms and substituting Equations (3.9) and (3.10) into Equation (3.11), a simplified equation for $\gamma$ is found:

$$\gamma = \cos^{-1}\left[ n_x \cdot \left( \frac{\Delta r_{2D}}{||\Delta r_{2D}||} \right) \right]$$

(3.11)

Once $\gamma$ is obtained, the height change caused by the ramp between laser measurements, $\Delta \delta h_{AGL}$, is calculated:

$$\Delta \delta h_{AGL} = ||\Delta r_{2D}|| \tan(\cos^{-1}(\hat{n}_y \cdot e_z))$$

(3.13)

When multiple height measurements are obtained while over a ramped surface, the individual height change measurements formed in Equation (3.13) are accumulated to form the total height change caused by the ramped surface:

$$\delta h_{AGL}(t_k) = \sum_{m=0}^{N} \Delta \delta h_{AGL}(t_{k-m})$$

(3.14)

where $\delta h_{AGL}$ represents the total effect that floor tilt has injected on height estimates. By removing this accrued value, absolute height can be obtained through:

$$\hat{h}_a(t_k) = \hat{h}(t_k) + \delta h_{total}(t_k)$$

(3.15)
It is important to note that the variance in the normal vector estimate limits the slopes that can be detected. Being on a constant sloped surface for extended periods of time gives you the opportunity to further smooth the normal vector to get a better estimate on those surfaces.

Due to the fact that aforementioned altitude estimation method (iii) is based on reliably performing a plane fit procedure, the success of the method is directly related to the resemblance of the actual ground to a planar surface. In addition, for low altitudes the lines of intersection with the ground plane are in close proximity. When the two scans lie close to each other, very small amounts of noise within the scans, or errors in the lever arm corrections can have a large effect on the accuracy of the orientation estimate of the planar surface. An example of this situation is illustrated in Figure 24 when the platform is at a height of 0.4m. At this 0.4m altitude with a level platform, the two scans lie very nearly on top of each other. In these situations where the average Euclidean distance between the bottom parts of two scans is small (less than 5cm), a different procedure is used. Fortunately, this situation is easy to detect and one could revert back to using method (ii).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure24.png}
\caption{Relationship between angled and vertical scan at different heights.}
\end{figure}
3.1.2 Pitch and Roll Estimation

For the attitude estimator, the point cloud points from laser scanners $D$ and $A$ are not converted to a locally-level frame, but rather, kept in the platform body frame. The steps involved in the attitude estimation process are shown in Figure 25 alongside the process used for the aforementioned method (iii) altitude estimator. The attitude estimation procedure strongly depends on the structure in the environment and assumes that at least two planar surfaces (i.e. wall and floor) are observed by the $D$ and $A$ lasers.

![Figure 25. Functional block diagrams of estimators: (a) altitude using method ii, (b) attitude using lasers scanners only](image)

After conversion of the laser point clouds to the platform body frame, possible floor and wall surfaces must be identified in the laser scans and their planar surface parameters
(i.e. normal vectors) estimated. The identification of the floor follows steps (3), (4), and (5) of altitude estimation method (iii) as shown in Figure 25(a). For the identification of potential walls, a two-step procedure is implemented. First, line extraction methods [89][90] are used to identify subsets of points on each scan that could be associated with a physical wall. Outliers are detected and removed and the residuals of the extracted lines are screened to make sure that the quality (i.e. 'flatness') of the line is sufficient to support steps 4 through 6. The flatness criterion is based off of the proposed lines residual noise and texture. Figure 26 illustrates results of this process by displaying a collected scan in red and showing extracted walls in green and yellow. After identifying candidate subsets of points lying on the walls, planar fits can be accomplished using the SVD discussed in the previous section. Through this procedure, wall normal vectors are obtained and plane fit residuals can be monitored to exclude any surface that is deemed to be unreliable due to noise. Results of the wall planar fit procedure can be observed in Figure 27.

Figure 26. Results of wall identification on downward scan.
After all wall and floor surfaces have been identified in each pair of scans, it is crucial for the attitude determination algorithm to have the capability of associating planar surfaces between observations at different time epochs. This requires that feature monitoring must take place between scans to determine if planar surfaces from two consecutive time epochs illuminate sections of the same physical wall. For this work, a feature association process which is driven by the onboard navigation filter was implemented. Between time epochs, the inertially driven prediction part of the filter is used to project forward data from previously observed planar surfaces to the current time frame. If the normal vector and points contained on a currently observed plane are similar to the characteristics of a plane that was project forward, they are considered to be associated. This process allows for tracking of similar planar surfaces between time epochs.

Through examination of the expected available planar surfaces in a structured indoor environment, it can be seen that up to 3 usable features can be seen at each time

Figure 27. Results of wall planar surface identification.
epoch; two walls and a floor. The relative change in orientation of the planar surfaces will be used to estimate the attitude from laser scanners only. Since the two wall surfaces are likely to be parallel and therefore provide no additional information to the estimator, only one of the walls will be used for the procedures described here. Using the resulting two non-parallel planar surface normal vectors, the optimized TRIAD method described in [91] will be used to obtain an attitude estimate.

Given two non-parallel normal vectors at a known time, a third normalized orthogonal vector can be obtained using their cross-product. This method is displayed in Equation (3.16) utilizing the normal vector of the floor, \( \mathbf{n}_F^b \), and one of the walls, \( \mathbf{n}_W^b \), to create a synthesized normal vector, \( \mathbf{n}_S^b \).

\[
\mathbf{n}_S^b = \frac{\mathbf{n}_F^b \times \mathbf{n}_W^b}{|\mathbf{n}_F^b \times \mathbf{n}_W^b|} \quad (3.16)
\]

If normal vectors are found for one wall and the floor at two different time epochs \( t_k \) and \( t_{k-m} \), the TRIAD method can be implemented. Two TRIADs, \( \mathbf{T}_1 \) and \( \mathbf{T}_2 \), can be formed using Equations (3.17) and (3.18) from the available data.

\[
\mathbf{T}_1 = \mathbf{n}_F^b(t_k)\mathbf{n}_F^{b,T}(t_{k-m}) + \mathbf{n}_S^b(t_k)\mathbf{n}_S^{b,T}(t_{k-m}) + [\mathbf{n}_F^b(t_k)] \\
\times [\mathbf{n}_F^b(t_{k-m}) \times \mathbf{n}_S^b(t_{k-m})]^T \quad (3.17)
\]

\[
\mathbf{T}_2 = \mathbf{n}_W^b(t_k)\mathbf{n}_W^{b,T}(t_{k-m}) + \mathbf{n}_S^b(t_k)\mathbf{n}_S^{b,T}(t_{k-m}) + [\mathbf{n}_W^b(t_k)] \\
\times [\mathbf{n}_W^b(t_{k-m}) \times \mathbf{n}_S^b(t_{k-m})]^T \quad (3.18)
\]
While both $T_1$ and $T_2$ provide a viable estimate of the incremental rotation in the form of a direction cosine matrix, $T_1$ weighs the floor measurement more heavily than the wall, and $T_2$ weighs the wall measurement more heavily than the floor. Since relative measurement quality between the observed floor and walls will likely change over time, a weighting function that combines the two TRIADS is required. Through an extension of the work performed in [92] an optimal TRIAD, $T_{opt}$, can be formed:

$$T_{opt} = w_1 T_1 + w_2 T_2$$

(3.19)

where $w_1$ and $w_2$ allow for individual weighting of the two surfaces and $w_1 + w_2 = 1$. For this work, weights are determined as a function of the each points planar surface fitting residuals, $r_k$, and the number of scan points used to create each planar surface, $N$:

$$w' = \left[ \frac{N}{20} - 1 \right] \frac{1}{N} \sum_{k=1}^{N} r_k$$

(3.20)

Equation (3.20) is performed on the ground plane points to produce $w_1'$, and applied to the wall surface points to yield $w_2'$. As $w_1' + w_2' \neq 1$, they are normalized to produce:

$$w_1 = \frac{w_1'}{w_1' + w_2'}$$

(3.21)

$$w_2 = \frac{w_2'}{w_1' + w_2'}$$

(3.22)

forcing $w_1 + w_2 = 1$. Note that the determination of weights shown in Equations (3.20) through (3.22) is tuned specifically for the sensors and application described in this work.
From the $T_{opt}$ direction cosine matrix, the incremental Euler angles of roll ($\delta \varphi$) and pitch ($\delta \theta$) between two time epochs can be found and used to form the vector $\sigma = [\delta \varphi, \delta \theta, 0]^T$. Yaw information is contained in this matrix as well, but better yaw estimates are typically available through use of the $F$ laser scanner. Now, the incremental angles can be used to update $C^H_b$ according to the attitude update method shown in Equation (3.23). This method is commonly used in inertial systems [12].

$$C^H_b(t_{k+1}) = C^H_b(t_k)e^{\sigma x} \tag{3.23}$$

Since every incremental angle measurement will contain small errors, integration of these angles by accumulation according to Equation (3.23) will introduce a random walk behavior in the attitude estimate. This behavior can be reduced by tracking planar surfaces across multiple time epochs instead of consecutive time epochs, and performing the TRIAD method with respect to the earliest observation of the wall or floor surface.

### 3.1.3 Results of Laser Based Pitch, Roll and Altitude Estimation

To evaluate the altitude and attitude estimators described in Sections 3.1.1 and 3.1.2, various tests were conducted. The platform for the test is described in detail in Chapter 6. In the altitude estimation plots that follow, the red lines represent the results of estimation procedure (i) and the blue lines illustrate the results of procedure (iii). The green line in the altitude plots show the output of the calibrated onboard barometric altimeter. In cases of the attitude plots, the red line shows the results of a free running inertial and the blue line the results of the attitude estimation method presented in Section 3.1.2. As explained in more detail in Chapter 6, the inertial is a commercially available Xsens unit.
whose gyroscopic noise is expected to be 0.05 (deg/s/√Hz). Each experiment and the corresponding results of the described altitude estimation methods are discussed in Sections 3.1.3.1 through 3.1.3.5.

3.1.3.1 Experiment 1: Level Platform at Varying Height

The first experiment was designed to quantify the height estimation capabilities of methods (i), (iii) and the baro-altimeter while the platform remained approximately level. Using a level, a tape measure and a laser range finder, the height of the center of the IMU on the platform was measured accurately at 5 discrete altitudes. The levels were measured at: (0.168m, 0.372m, 0.829m, 1.168m and 1.380m). Next, the platform was held level for about 20-30 seconds at each of these discrete heights. The results for experiment 1 are shown in Figure 28.

![Figure 28. Altitude Estimation test with platform remaining roughly level (full).](image-url)
To better observe the differences between the three height estimates, a small section of Figure 28 has been expanded in Figure 29 for the 0.168m level. In this experiment the baro does not provide a high level of accuracy with up to 70cm of error. The altitude estimate using method (i) provides good accuracy with centimeter level noise. And, finally, method (iii) provides the best results with millimeter-level noise on the height estimate. Note that the accuracies found here far exceed any potential requirement as the flight technical error of the platform more than likely will exceed this number. At this moment altitude accuracy required is not yet defined.

![Altitude Estimates](image)

*Figure 29. Altitude Estimation test with platform remaining roughly level (~70 seconds).*

For experiment 1, the results of the laser based attitude calculation and the free running inertial are shown in Figure 30. It can be seen that both the free running inertial and laser-based attitude determination yield comparable results with similar noise levels
while never diverging more than 3° from one another. As no absolute attitude truth reference was available a relative performance evaluation was deemed sufficient. Ideally, the data should show a platform attitude of approximately 0° apart from the small disturbances introduced while moving the platform to a different level. Figure 30 shows that the pitch and roll estimates derived from the laser based solution drift by 3° and 0.5°, respectively, while the inertial drifts by 5° and 2°, respectively. These drifts over a 350 second span are similar, but the laser based solution slightly outperforms the free running inertial for experiment 1, though insignificantly so.

Figure 30. Pitch and roll estimation test with platform remaining roughly level.

3.1.3.2 Experiment 2: Platform at Varying Heights with Controlled Attitude Changes

The second experiment was similar to the first experiment; the platform was held at discrete heights for short periods of time. However, unlike the previous experiment the
platform was not held level continuously. At each height, the platform was initially held level followed by the introduction of large pitch and roll changes (< 30°). Each run was ended by holding the platform level for a couple of seconds. The results of the altitude estimation can be seen in Figure 31 with measured heights of (0.168m, 0.383m, 0.694m, 1.039m, 1.271m and 1.505m).

![Altitude Estimates](image)

**Figure 31.** Altitude estimation test with platform experiencing orientation changes (full).

Figure 32 shows a section of the results in Figure 31. Similar to the results of the first experiment, the baro-altimeter does not provide a high level of accuracy. With respect to the laser-based altitude estimates, method (iii) produces results similar to method (i), albeit less noisy. However, altitude errors are introduced during the orientation changes for method (i) and not for method (iii), underlining the fact that method (iii) not only provides more accurate results but is also more robust.
Figure 32. Altitude estimation test with platform experiencing orientation changes (~110 seconds).

Figure 33 depicts the results of the laser- and inertial-based pitch and roll estimates for experiment 2. Similarly to experiment 1, the two methods have similar noise and drift levels. With respect to accumulated errors, the 450 second segment encounters no more than 4.5° of drift from either method. Unlike experiment 1, experiment 2 allows for examination of the two attitude determination methods during times in which the platform experienced severe attitude changes. Figure 32 illustrates that even under moderate dynamics, the laser based solution yielded comparable results to the free running inertial.
3.1.3.3 Experiment 3: Level Platform at a Constant Height Traversing a Ramp

Next, the performance of flight over a non-level surface was tested as depicted in Figure 20. To establish a constant altitude while "flying" over the slanted area and controlling the inclination, the sloped floor was moved rather than the platform itself as shown in Figure 34.

Figure 33. Pitch and roll estimation test with platform experiencing orientation changes.

Figure 34. Test setup for experiment 3.
The altitude estimation results can be seen in Figure 35. Initially the UAV was placed on the floor, after which it was picked up (from $t = 96s$ to $t = 103s$) and put on the braces shown in Figure 34($t = 103s$- 160s). From $t = 160s$ to $t = 167s$ the slanted floor is moved to emulate motion of the UAV in the slope direction from $h_{aft} \approx 1.04m$ (afl = above floor level) to $h_{aft} \approx 0.88m$. After the initial movement, the slanted floor was moved two more times: from $t = 176s$ to $t = 181s$ and from $t = 191s$ to $t = 194s$. During these three intervals it appears to a UAV using method (i) that either the platform changed altitude or a change in floor height occurred. Unfortunately, the UAV does not know which of the two situations is correct. However, method (iii) estimates the correct altitude with respect to the initial altitude and is not influenced by the ramp presence. Note that knowledge of both method (ii) and method (iii) outputs can be exploited to determine the behavior of the floor underneath the UAV platform; for example, to determine the inclination angle. It is important to note that the baro-altimeter seems to follow the height above floor level even though the platform in maintaining a constant height. Through analysis of the altimeters pressure data, this behavior is attributed to local pressure changes caused by the ramp sliding beneath the platform.
The attitude determination results for experiment 3 are shown in Figure 36. During this test, the two methods provide similar results while drifting less than 1.5° over the data set with respect to pitch and roll. While the two methods perform in a very comparable manner, a weakness in the laser based estimation can be observed around $t = 106s$. As stated previously, when a transition occurs between planar surfaces being used for attitude determination, uncorrelated estimates are produced. At $t = 106s$ seconds in Figure 36, a wall transition occurs in the tracking algorithm that causes the introduction of a bias of approximately 0.5° into the laser-based solution.
3.1.3.4 Experiment 4: Platform Experiencing Height Changes While Traversing a Ramp.

For further testing of the aforementioned ramp detection and elimination procedure, an indoor data collection was performed that included the platform traversing a ramp in a more dynamic manner than the previous experiment. The UAV began on the ground and was subsequently lifted off and carried down a hallway. At the end of the hall the platform travelled down a ramp with a ~50cm height. Following the decent on the ramp, the platform made several turns and traveled down multiple different corridors before landing. Due to the length of this test, only the portion of the results leading up to and directly after the ramp encounter are discussed in the following plots. The altitude results for this 200 second period are shown below in Figure 37. Between the ~70-80 second segment in Figure 37 the ~50cm height difference caused by traversing the ramp is clearly observed in the dual laser solution while the single laser solution yields no change. As the
floor after the ramp was known to be ~50cm lower than the ground on which the platform started, the dual laser solution shows the desired results.

![Altitude Estimates](image)

*Figure 37. Altitude Estimation test with platform experiencing height changes while traversing a ramp.*

The attitude determination results for experiment 4 are shown in Figure 38. During this test, the roll estimation derived from the two methods slowly drifts to a maximum of 12° of separation. The pitch results display a maximum of 9° of separation, however this discrepancy fluctuates over time. The most likely cause for these larger discrepancies is the complex environment through which the UAV was operating. As there were many offshoot hallways and rooms in this environment, the wall being tracked was constantly changed. To make matters worse, in many situations where only a single wall was available for monitoring, there were objects such as benches, large mailbox compartments and trash cans.
3.1.3.5 Experiment 5: Flight Test with Varying Heights and Attitudes

Unlike the three previous controlled experiments, the final experiment involved actual flight test data. While it is more difficult to evaluate real flight data in an absolute manner, comparisons between the various methods can still be made and advantages identified. The UAV started on ground level in an open-sky environment. Upon takeoff, the platform maneuvered toward the engineering building and entered it through the basement doors. Once inside the structured environment of the basement's hallways, the platform traveled through multiple corridors to the opposite side of the building. Retracing its path, the UAV flew the reverse of the previously described route leading it from inside to outside and landing in a similar place to where it began. Once on the ground, the
hexacopter was kept stationary for multiple seconds before data collection was ceased. Figure 39 shows the altitude results for both method (i) and (iii).

![Altitude Estimates](image)

*Figure 39. Altitude estimation from hexacopter flight data (full).*

Since there are no slanted floors or stairs in the basement of the Stocker engineering building, the results for both methods look very similar. However, method (iii) does provide a lower noise level and is more robust throughout the flight. Specifically, around \( t = 275s \) and \( t = 510s \), method (i) shows severe altitude deviations. For better understanding of these variations, Figure 40 offers an enhanced view. These deviations correspond to sections of the collected data where the platform transitions between the indoor and outdoor environments. These fluctuations could be explained by a large amount of laser outliers due to debris thrown around by the wind gusts from the propellers near the buildings doorways.
The pitch and roll estimate results depicted in Figure 41 show that the laser-based attitude determination method described in this paper can yield results similar to an IMU in a non-controlled environment. During this flight, multiple different planar surfaces were acquired, tracked, and used for attitude determination. Transitions to different wall segments caused some additional error in the laser-based estimates, but neither solution ever deviated from the other by more than 5°. As for the accrued pitch and roll over the full data set, the laser based solution ended at 8° and 17°, respectively. The free running IMU finished at 12° and 15°. As the platform was operating in an uncontrolled environment and did not land in the exact same location, it is probable that the orientation of the ground is different at the conclusion of the flight as opposed to the beginning. Therefore, the estimated attitude values at the end of the flight likely contain an indistinguishable combination of estimation drift error and a difference in ground surface orientation.
3.2 Three Degree-of-Freedom SLAM

In Section 3.1, methods were established to produce laser based estimates for a multi-copters' altitude, pitch and roll states: \((z, \phi, \theta)\). By means of the forward looking laser, a 3DOF SLAM procedure is implemented to estimate the remaining three states: \((x, y, \psi)\). The employed 3DOF SLAM procedure builds upon the work done by Kohlbrecher [41] and Grisetti [33][93], and will be the focus of this section. Both approaches in these papers represent the map of the environment by an occupancy grid (see Section 2.5). Whereas Grisetti et al. use a Rao-Blackwellized particle filter with each particle representing a potential trajectory (sequence of poses) with associated map, Kohlbrecher uses an EKF-based approach. Both methods include the laser range measurements by performing some variation of scan matching.
Laser scan matching or scan-matching methods take a current scan of the laser range scanner and computationally tries to align this scan to earlier scans (e.g. the iterative closest point or ICP method) or to the map (e.g. an occupancy grid as in [41] and [33]) resulting in the likelihood function required to perform the update step of the Bayesian filter (see Equation (2.10)). A simplified explanation of the scan matching procedure is illustrated in Figure 42. Beginning with scan 1, a coordinate frame is established where the platforms heading is set to $\psi = 0^\circ$ and its location is placed at the origin ($x = 0, y = 0$). Once scan 2 is received, the scan matching algorithm aligns the common points between scan 1 and scan 2. The overlay of scan 2 on top of scan 1 produces a larger map of the environment containing points from both scans. This is designated below as the match between scans 1 and 2. The alignment procedure essentially evaluates the likelihood function: given the rotation and translation that most likely result in the observed laser scan. In this case, application of the rotation and translation results in the new pose estimate: $x = 1, y = 0, \psi = 9^\circ$. When scan 3 is obtained, it is compared and aligned with the map created through the combination of scan 1 and scan 2. Through the same procedure used to align scan 2 with scan 1, the results of aligning all 3 scans are shown below. With respect to the frame established from scan 1, the platforms’ state during the collection of scan 3 is determined to be: $x = 2.5, y = -1, \psi = 86^\circ$. It can be observed that this procedure yields the necessary geometric observability to determine the platforms relative 3DOF states while building a constantly expanding map. The results of the map building process for the three scans used in Figure 42 are displayed in Figure 43.
Figure 42. Scan matching procedure illustrated using real laser scan data.

Figure 43. Scan matching results.
The method by which scan matching is accomplished in the example is referred to as the Iterative Closest Point (ICP) algorithm. ICP was initially introduced and explored by Besl [94], Chen [95] and Zhang [96]. Given two point clouds at time epochs $k$ and $k-1$ with at least some corresponding points, $S_k = \{P_{k,1}, P_{k,2}, \ldots, P_{k,N_k}\}$ and $S_{k-1} = \{P_{k-1,1}, P_{k-1,2}, \ldots, P_{k-1,N_{k-1}}\}$, ICP was designed with the intent of determining the rotation, $C$, and translation, $r$, between the two clouds. If the two sets would contain uncorrupted data and all point associations between the two scans are known, an unambiguous solution for $C$ and $r$ can be calculated in a closed form. Examples of these are given by Arun [97], Horn [98] and Walker [99]. Unfortunately, associated points connecting the two sets are typically unknown and the laser measurements are corrupted by noise or variations due to, for example, the effects of texture. Thus, a perfect solution for rotation and translation is unattainable. Instead, ICP attempts to find the values for $C$ and $r$ that minimize the sum of the squared error:

$$E(C, r) = \sum_{i=1}^{N_k} \sum_{j=1}^{N_{k-1}} w_{i,j} \| P_{k,i} - (C P_{j,k-1} + r) \|^2$$

This process is an iterative process and assumes in each iteration step that the “closest” point corresponds. For many ICP methods this means that the starting position must be close enough. As exhaustively finding matching points is computationally expensive, ICP begins by choosing points within each scan for comparison. This can be effectively accomplished a multitude of different ways depending on the type of data being used and the physical structure of the point clouds. Methods for choosing points of interest
including the use of all points, normal-space sampling and feature sampling are discussed in detail by Rusinkiewicz [100] and Gelfrand [101]. Through comparison among the two subsets of sampled points, data association is performed. Matched points are identified and outliers are rejected through procedures such as Trimmed ICP [102] and Picky ICP [103].

So far, the scan-matching involved two point clouds, however, in the SLAM methods that are used as a basis for the method proposed in this dissertation, the received laser scans are matched with an occupancy grid. Typically, this process involves using the predicted pose, external odometry or inertial data as the beginning of a search within the occupancy grid. This basic concept is illustrated in Figure 44.

Both pictures show an small excerpt of a 2D occupancy grid, the predicted platform pose, $\mathbf{x}_k^-$, and two possible pose updates, $\mathbf{x}_{k,1}$ and $\mathbf{x}_{k,2}$. The vectors emitting from the possible poses are the range vectors form the laser scanner. In this example, it can be seen...
that choosing $x_{k,2}$ results in a better fit of the measurements to the map, in other words, the likelihood $p(z_k | x_{k,1}, m_k) < p(z_k | x_{k,2}, m_k)$. The literature on how to implement this search is vast as are the available implementations of these 2D occupancy grid-based scan-matching algorithms. We considered three different implementations available under the Robotic Operating System (ROS): GMAPPING based on the work in [33], Hector SLAM [41] and the ethzasl ICP mapper [94].

For most 3DOF SLAM methods, including those in the GMAPPING and Hector SLAM packages, an approximately level platform assumption is made. This is typically a valid assumption for ground vehicles, and may be reasonable for limited motion, but may be unreasonable for multi-copter platforms. The dynamics of a multi-copter in, especially, smaller spaces may inherently contain variations in the platforms roll and pitch. The level platform assumption will result in a deformation of the point cloud as illustrated in Figure 45 where the level platform on the left captures an accurate slice of the environment, while the platform on the right obtains a stretched slice of the environment due to platform roll. This effect is further visualized in Figure 46. In this figure, the measured distance to a surface, $\rho_m$, relates to the actual distance, $\rho_a$, through the platforms roll angle, $\theta$. This relationship is captured in Equation (3.25). The true measurement distortion caused by a change to the pitch or roll angles is apparent. The magnitude of measured environmental deformation caused by these attitude changes can make the level platform assumption invalid with respect to the scan-matching procedure.

$$\rho_a = \rho_m \cos \theta$$  \hspace{1cm} (3.25)
Figure 45. (a) Left: laser scan of a hallway with a level platform; (b) Right: laser scan of a hallway with an un-level platform.

Figure 46. Effect of a tilted platform on distance to wall measurement.

So, the innovation that is proposed here is the compensation of the laser scans before they go in the “standard” 2D SLAM algorithms. The block diagram of this improvement is shown in Figure 47. First, the laser scans are converted to the platform frame (step 1) and leveled (step 2) through the coordinate transformation described in Section 2.7 by Equation (2.41) where $\mathbf{C}_b^{ll}$ is derived from the laser based attitude estimation procedure described in Section 3.1.2.
Since the 2D SLAM algorithm acts on 2D point clouds or 2D laser scan measurements, the method relies heavily on the structured environment and only considers the 2D projection of the leveled point cloud by taking only the x- and y-coordinates (step 3). For the SLAM methods that require the input to be a laser scan, the transformed point cloud is converted back to a laser scan (ranges and angles). Since, the new laser scan measurements no longer have a uniform distribution in angles (which is required for many of the methods available), the range measurements are interpolated, introduced artifacts are identified (i.e. interpolation between two points that are far away may cause an artificial point in between), and removed. Next, ground points are removed (step 4). These ground points are introduced due to the fact that even small changes in pitch or roll could result in the laser scanner measuring the ground. At this point, a “clean” point cloud is available for the 3DOF SLAM (or 2D SLAM) method. Additionally, this same point cloud could be
used to extract line features and use those for navigation as in Bates [35] and Soloviev [46] (create additional odometry) (step 6).

The improvements due to steps (1) and (2) can be observed in Figure 48 by comparing two scans collected by the multi-copter platform described in Chapter 6. The red scan shows measurements collected by a relatively level platform and the blue scan depicts a slice of the same environment obtained from a non-level platform. The environmental warping caused by the normal dynamics of a multi-copter will cause the scan-matching algorithm to struggle and possibly create false walls in the occupancy grid as these two scans should match.

Figure 48. Effect of un-level platform on width of hallway measurement.

An example of an occupancy grid built by our system can be seen in Figure 49.
To evaluate the described 3DOF SLAM procedure, multiple experiments were conducted. To assess the heading estimates obtained from SLAM, the following plots depict a comparison between the acquired results to that of a calibrated inertial. As previously stated, the path estimation and map creation from the SLAM algorithm are directly correlated. Therefore, evaluation of one inherently quantifies the quality of the other. For the experiment described in Section 3.3.1.1 the traveled path was measured using laser range finders and a tape measure. The path derived through SLAM is evaluated against this measured path. Conversely, a truth reference map obtained by the Riegl LMS-Z360i is used to assess the map created through SLAM for the flight test described in Section 3.3.1.2.

3.2.1 SLAM Experiment 1: Platform Carried Around a Loop.

Experiment 1 was conducted with the intent of assessing the described SLAM procedure while walking around an indoor loop of structured hallways. During this test,
the platform was carried around a rectangular loop of corridors at a roughly constant height.
To test the derived leveling method, large changes in the platforms roll angle were
introduced at known locations.

The heading results of experiment 1 are displayed in Figure 50. Upon comparing
the heading results of the calibrated inertial to those of the SLAM procedure, multiple
observations can be made. By contrasting the development of heading versus time, both
processes derive heading estimates that follow the same trend and yield similar results. It
can also be seen that the two heading curves slowly divert from each other as time
progresses. After 250 seconds, the two heading solutions are $5^\circ$ apart.

![Figure 50. Heading results for indoor loop experiment.](image-url)
Figure 51 shows a comparison between the calculated SLAM trajectory and the measured path. This plot indicates that the SLAM trajectory solution is operating with at least sub-meter accuracy and a high level of stability. For most of the trajectory, the SLAM path remains within a decimeter of the measured path. Further evidence of this precision can be observed through the occupancy grid of Figure 52. Upon plotting the measured path within the created occupancy grid, it can be observed to always remain within the maps hallways. Additionally, the map does not warp in the locations where large roll changes were known to occur. This indicates that the leveling procedure is working properly. These results point to the implemented SLAM procedure working well for both localization and mapping.

![Estimated SLAM Trajectory (xy)](image)

*Figure 51. Trajectory results for indoor loop experiment.*
3.2.2 SLAM Experiment 2: Flight Test through Outdoor and Structured Indoor Environments.

The second experiment to assess the described SLAM procedure involved traversing indoor and outdoor environments. The UAV began in an outdoor environment with some structure. Upon takeoff, the platform maneuvered alongside a building and entered it through the basement doors. Once inside, the platform traveled through multiple corridors and exited the opposite end of building. The UAV then traced a path around the outside of the building and landed in a similar place to where it began.

The heading results produced by the implemented SLAM procedure can be seen in Figure 53. When compared to the inertial heading, similar trends are observed as the solutions developed over time. Over 385 seconds, the two solutions never divert by more...
than 10° and finish with a 4° difference. In this experiment, the SLAM and inertial solutions provide comparable heading estimates.

![Figure 53. Heading results for indoor/outdoor flight experiment.](image)

For the indoor portion travelled during experiment 2, a "truth reference" map was made available through the Riegl LMS-Z360i laser scanner. This allowed for evaluation of the map creation and path estimation results. Figure 54 shows a comparison between the truth map and the occupancy grid produced through SLAM. Upon examination, the produced occupancy grid matches the truth data to within a decimeter at all overlapping locations. This level of accuracy can also be observed in Figure 55, where the SLAM trajectory is plotted against the truth map. In this figure, it is observed that the estimated path of the platform remains within the hallways of the truth map and follows the course the UAV is known to have taken. These results show that the implemented SLAM
procedure is performing well in both indoor and outdoor environments that contain structure.

Figure 54. Occupancy grid results for indoor/outdoor flight experiment.

Figure 55. Trajectory results for indoor/outdoor flight experiment.
CHAPTER 4: 2D MONOCULAR CAMERA METHODS

In Chapter 2 we discussed some of the basic principles of visual odometry. In this chapter we will extend that discussion and address our specific proposed method, which is summarized by the highlighted blocks in Figure 56.

Figure 56. Monocular camera components of proposed mechanization.

In this method we perform feature detection and tracking of both point features and line features. Specifically, elements from Lowe's Scale Invariant Feature Transforms (SIFT) [77] will be used to track point features. SIFT is discussed in Section 4.1. Section
4.3 discusses the use of these SIFT features to obtain estimates of a camera's rotational and un-scaled translational motion using SFM-based methods. To resolve the ambiguous scale factor, Section 4.4 discusses a novel scale estimation technique that uses the platform's horizontally scanning laser with the 2D line tracking procedure described in Section 4.2. Finally, Section 4.5 discusses the results of the described odometry on varying tests.

4.1 SIFT Point Feature Extraction and Association

With the intent of using features to determine camera motion, Lowe [77] developed SIFT as a way of identifying local features that are invariant to translation, rotation, and image scaling. This yields 2D point features that are unique to their surroundings, and therefore, can be easily identified and associated between camera images.

For feature identification, SIFT begins by creating a multi-scale representation of an image through the generation of numerous image copies and convolving these copies with a Gaussian kernel with increasing standard deviation. For increasing standard deviation values, scale information is successively suppressed. This multi-scale representation can be seen in Figure 57 and is referred to as the scale space.

Figure 57. Original image(left) and scale space representation of image(right).
Next, SIFT identifies key locations in an image by computing a difference of Gaussian (DoG) between the images of the nearby scales in the scale space. Through analysis of this function, key locations are identified to be at the maxima or minima of the DoG function, which are inherently located at places in the image with high variations at each scale. The larger these variations, the better the features can be identified in future images while undergoing camera motion. Through the processes laid out in [77], each key location and its surroundings are analyzed to yield a descriptive 128 element feature vector, known as a SIFT key. Example results of the SIFT key identification procedure can be seen in Figure 58.

![Figure 58. SIFT feature identification.](image)

Given the results of the SIFT key feature extraction process on two image frames, feature association between these frames is performed using the 128-element vector. For
this work, a two-step procedure is implemented. First, SIFT keys are associated using the matching procedure in the "Sift for Matlab" code developed by Vedalidi [104]. Vedalidi implements the matching process described by Beis [105] to identify nearest-neighbors in an image to previously seen SIFT keys. For each estimated nearest neighbor, a probability of correctness is associated with it to quantify the quality of the match. These probabilities are then used to weigh the estimated matches from the previous image to the current image for determination of the optimal set of matches between the two image frames. Results of this process for two frames of multi-copter flight data are shown in Figure 59.

It can be seen in Figure 59 that misassociations can occur when applying the procedure implemented in [104]. To eliminate these incorrect matches, inertial measurements are utilized to ensure the correctness of the associations. Using the triangulation method described by Zisserman in [51], prospective associations are used to crudely estimate each feature’s 3D position with respect to the previous frame is obtained.
(*Further explanation of the triangulation method can be found in Appendix B.*) While this triangulation method yields 3D data, it is of poor quality, and is therefore only used to obtain rough approximations, sufficient for association purposes, insufficient for navigation purposes. Once in 3D, the projected distances of each feature are compared with one another. Prospective associations that produce significantly different depths than surrounding points are eliminated. Through analysis of the same images used to generate Figure 59, Figure 60 shows the association results after the elimination of possible incorrect associations. The described two step approach of attaining matches assures feature association.

![Figure 60. Point feature association after inertial based miss-association rejection.](image)

4.2 Wavelet Line Feature Extraction and Association

For the proposed method, it is also necessary to identify and track line features within each collected image. For line detection, many methods were explored and tested to obtain the desired line features including classic methods such as the ones described by
Canny[106] and Marr[107]. Even though these methods are well-understood, a more robust and better method with respect to this work has been developed that uses wavelets to extract lines.

Wavelets have proven useful in a number of practical applications such as signal de-noising, data compression, and signal modulation. In general, a wavelet transform (WT) is a multi-scale transformation which produces a time-frequency representation of a signal using “basis functions” which are scaled and shifted copies of a “mother wavelet”. When applied to a 2D image, a WT more or less performs filtering in the x and y directions (here referred to as channels) of an image with of high-pass(H) and low-pass(L) filters with different bandwidth. Specifications of these filters are defined by the type of wavelet employed. By applying either a high- or low-pass filter to both of an images channels, 4 sub-images can be created and are designated as the image approximation (a), the horizontal details (h), vertical details (v) and diagonal details (d). The sub-images and corresponding filter permutations can be seen on the left of Figure 61. This filtering process can be iteratively performed N times where the number of iterations is referred to as the level. For each level past the first, the image approximation created at the previous level is decomposed into 4 new sub-images. This is illustrated on the right side of Figure 61 where the results of the level 2 decomposition are produced through filtering of the level 1 approximation. For more details on wavelet transformation theory the reader is referred to [108].
Figure 61. Principle of wavelet decomposition of an image.

For this work a level 1 bi-orthogonal 1.3 wavelet was used to decompose each image. Details on the bi-orthogonal family of wavelets can be found in [108] and [109]. An example of the 4 sub-images produced by this wavelet can be seen on the right half of Figure 62.

Figure 62. Results of wavelet decomposition.
The (h), (v), and (d) decompositions shown in Figure 62, clearly emphasize line features for different orientations. While information pertaining to lines at different orientations is produced by the WT, only lines that are roughly vertical w.r.t each captured image are taken into account. Figure 63 shows an example of the results produced by the vertical wavelet decomposition. Segments of the sub-image possessing vertical lines are visibly apparent.

![Wavelet vertical decomposition results.](image)

Through further processing of the vertical decomposition results, strong line features are found through the process explained in Figure 64. By inspecting the illuminated elements along the vertical channels of the decomposed image, clusters of adjacent pixels are identified. Next, any of these clusters that do not contain a significant number of pixels(<100 for this work) are eliminated. A 2D line fit is then applied to the reaming groups to obtain a measure of average residual noise. Pixel collections with low
residuals (< 3 pixels for this work) are considered to be line features. Results of this line extraction process are shown in Figure 65.

Figure 64. Line feature extraction procedure.

Figure 65. Vertical line extraction results.

For association purposes, lines cannot be compared solely based on location. This is due to the fact that similar line features between image frames do not necessarily possess the same endpoint, and, therefore, can be of varying lengths. However, corresponding lines will possess many common points and similar orientations if they are projected into the
same frame. Using the inertial, each lines orientation, \( \hat{\mathbf{l}}_t \), can be transformed across image frames:

\[
\hat{\mathbf{l}}_t(t_k) = \hat{\mathbf{c}}_{t_{k-1}}^{t_k} \hat{\mathbf{l}}_t(t_{k-1})
\] (4.1)

Lines between frames that contain multiple similar points and have comparable orientations are considered associated.

4.3 Projective Visual Odometry and Epipolar Geometry

The basic concept of visual odometry and associated calibration procedures have been introduced in Sections 2.6.2.1 and 2.6.2.2, respectively. The basic measurement produced through analysis of a 2D vision sensor are the locations of the point features on the image plane, \( \mathbf{p}_t \) (see Equation (2.34)). Given the relationships discussed in Section 2.6.2.1, associated point features at two time epochs obey the epipolar constraint, or:

\[
\mathbf{p}_t(t_{k-1})^T \mathbf{F}_t \mathbf{p}_t(t_k) = 0
\] (4.2)

where \( \mathbf{F} \) is referred to as the fundamental matrix and related to the essential matrix as follows:

\[
\mathbf{E} = \mathbf{K}^T \mathbf{F} \mathbf{K}
\] (4.3)

Define \( \mathbf{F} \) by its elements as:

\[
\mathbf{F} = \begin{bmatrix}
F_{11} & F_{12} & F_{13} \\
F_{21} & F_{22} & F_{23} \\
F_{31} & F_{32} & F_{33}
\end{bmatrix}
\] (4.4)
Now, define $F^s$ as the row vector representation of $F$:

$$F^s = [F_{11} \ F_{12} \ F_{13} \ F_{21} \ F_{22} \ F_{23} \ F_{31} \ F_{32} \ F_{33}]^T \quad (4.5)$$

Given $N$ associated point features, we can now define the $N \times 9$ matrix $A$ by:

$$A = \begin{bmatrix}
[p_1(t_{k-1}) \otimes p_1(t_k)]^T \\
[p_2(t_{k-1}) \otimes p_2(t_k)]^T \\
\vdots \\
[p_N(t_{k-1}) \otimes p_N(t_k)]^T
\end{bmatrix} \quad (4.6)$$

Now, the epipolar constraint equation can be rewritten using Equations (4.2) and (4.6):

$$A \ F^s = 0 \quad (4.7)$$

This relationship is used to solve for $F$ using the SVD of $A = USV^T$. The column of $V$ which corresponds to zero singular values in the diagonal elements in $S$ contains a solution for $F^s$ (or, in the presence of noise, the column of $V$ corresponding the smallest singular value). Through the relationship between Equations (4.4) and (4.5), $F^s$ is converted back into $F$ to form the fundamental matrix. For this solution to be attained, there must be as many rows of $A$ as there are unknown elements of $F$. While $F$ appears to have 9 unknowns, one of its elements is guaranteed to be uniquely 0. Therefore, 8 sets of matching points are required to find a unique solution to $F$. For this reason, this process is often known as the eight-point algorithm.

Once the eight-point algorithm is completed, the intrinsic parameters of the camera are taken into account. These parameters are contained in the matrix $K$, and are infused
into the fundamental matrix to form the essential matrix, $E$. Through application of the SVD on $E = USV^T$, two possible solutions for the rotation can found as follows:

$$C_{t_{k+1}} = R_1 = U WV^T \quad (4.8)$$

$$C_{t_{k+1}} = R_2 = U W^TV^T \quad (4.9)$$

Where

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.10)$$

Two possible translations can be found as well:

$$[\Delta \hat{r} \times] = Y_1 = U WSU^T \quad (4.11)$$

$$[\Delta \hat{r} \times] = Y_2 = U W^TSU^T \quad (4.12)$$

where $\Delta \hat{r}$ is unit-length and, thus, only indicates the direction in which the camera is travelling. $\Delta \hat{r}$ is related to the translation $\Delta r$ through a scale factor, $m$:

$$\Delta r = m \Delta \hat{r} \quad (4.13)$$

Since two possible solutions exist for both the rotation and the translation, there are four possible solutions. By taking each of these four valid geometric solutions and using the translation and rotation contained in each to project associated point features into 3D space, the correct rotation and translation combination can be identified. When these four
solutions are applied for projective purposes, only one of them yields a 3D point that would be physically observed by the camera.

4.4 Resolution of True Metric Scale

Using the visual odometry methods discussed in Section 4.3, consecutive image frames provide an estimate of incremental rotation and incremental un-scaled translation, $\hat{\mathbf{C}}_{k-1}^k, \Delta \tilde{\mathbf{r}}$. As the un-scaled translation estimate is a unit vector, $\|\Delta \tilde{\mathbf{r}}\| = 1$, it only indicates the most likely direction of motion for the sensor. To obtain the actual translational motion, the scale factor, $m$, is required to relate the un-scaled parameter $\Delta \tilde{\mathbf{r}}$ to the absolute translation $\Delta \mathbf{r}$. Determination of this scale factor is the focus of this section.

It can be stated that without \textit{a priori} knowledge of the operational environment or the use of measurements from other sensors, scale cannot be established. Unknown scale has been estimated using many approaches, all using additional information. Engel obtains the scale estimate through a Kalman filter with input velocities from multiple barometric and sonar altimeters in [110] and [111]. Through an EKF, Kneip [112] and Soloviev [61] estimate scale by integrating inertial measurements with visual data. Aside from the use of extra sensors, Kitt [113] and Song [114] obtain scale for a camera on a ground vehicle by capitalizing on a camera's known height and features on the ground plane. Using features tracked over multiple images (at least three), Montiel [115], Munguia [116] and Civera [117] perform inverse feature depth estimation to circumvent the need for a scale estimation. While each of these methods has merit, our platform’s sensor suite is not suitable for any of these methods. Therefore, a new method is formulated that uses the horizontally scanning laser to aid the visual odometry.
The proposed method estimates the scale in an image by identifying points in the environment that are simultaneously observed by the camera and the forward-looking laser range scanner. To enable this estimation method we must identify what pixels (each defined by a direction unit vector $e^{b}_{c,x,y}$ corresponding to the row (x) and column (y) pixel) correspond to what laser scanner measurements (each defined by direction unit vector $e^{b}_{l,\alpha}$).

A calibration procedure has been developed to establish this correspondence. The calibration procedure was accomplished by suspending a very thin, horizontally-oriented board in front of the camera in such a way that the board's edges corresponded to the minima and maxima of the image frames x coordinates. This setup is shown in Figure 66 where the board is highlighted with a blue line.

![Figure 66. Laser/camera intersect calibration procedure setup.](image)

Simultaneously, the board is oriented in such a way that it is entirely in the laser's scan plane. The board's occlusion of the laser scan is indicated by the green scan points in Figure 67. Through this procedure, a mapping is established between the blue highlighted
pixels in the image frame of Figure 66 and the green scan points of Figure 67, in other
words, within the tuple \( \{x_i, y_i, \alpha_j | i = 1: N_c M_c, j = 1: N_l \} \) unit vectors are approximately
equal:

\[
e_{c,x,y}^b \approx e_{l,\alpha}^b
\]  

(4.14)

This process was repeated at multiple distances to assure accuracy of the calibration.

![Figure 67. Laser/camera intersect calibration procedure.](image)

Given the corresponding laser ranges, 2D features located on the scan/pixel
intersection can be scaled up to 3D points. Unfortunately, extracted 2D point features are
rarely illuminated by the laser scan in two consecutive frames. This is resolved by
considering the intersection of 2D line features instead of point features that intersect the
laser scan. As the laser intersects the camera frame at the same location regardless of
platform motion, and the platform does not make extensive roll and pitch maneuvers,
vertical line features in the image frame are preferred as they will be relatively orthogonal
to the laser scan plane. Using the vertical line extraction procedure from Section 4.2, Figure
Figure 68 shows the physical points illuminated by the laser as a blue line and the vertical line features as green lines. Furthermore, the multiple intersections between the 2D vertical lines and the laser scan are calculated and identified in Figure 68 as red points.

![Image of a corridor with laser scan data]

**Figure 68. 2D vertical line and laser intersections in the image frame.**

Inversely, Figure 69 depicts the location of all laser scan points in green, all laser points located in the image frame in blue, and the intersection points in red.

![Graph showing associated laser and camera feature extraction]

**Figure 69. 2D vertical line and laser intersections in laser scan data.**
For scale factor calculation purposes, it is necessary to track the motion of these 3D laser/vision intersection points, \( p_{LV}(t_k) \), between camera image frames. As each intersection point uniquely belongs to a line feature in the 2D image frame, it can be stated that if two lines are associated, their corresponding intersection points are consequently associated. Using the rotation obtained from the visual odometry process of Section 4.3, the line association method described in Section 4.2 has been implemented, resulting in associations between laser/vision intersection points across frames.

To calculate the desired scale factor through these associated laser/vision points, geometric relationships are established. First, as shown in Figure 70, \( N \) unit vectors, \( e_i \), can be established from the camera center, \( f \), to \( N \) points located on a 2D line. From this, two of these unit vectors are used to identify the line's normal vector, \( n \):

\[
\textbf{n} = e_1 \times e_N
\]  

(4.15)

![Figure 70. Relationship between camera position, 2D lines, and laser vision point.](image)
The relative geometry to the monitored features in two frames is shown in Figure 71. Intersection point $p_{LV}^*$ at $t_k$ can be projected backward using the VO-derived translation and orientation as follows:

$$p_{LV}^*(t_k) = m \Delta \tilde{r} + C_{k+1}^k p_{LV}(t_{k+1})$$  \hspace{1cm} (4.16)

where the superscript ‘*’ indicates it is a back-projected quantity. This point must lay on the same line and thus also in the same planar surface defined by $p_{LV}(t_k)$ and $(t_k)$. Hence, the scale factor $m$ can be obtained through:

$$n \cdot p_{LV}(t_k) = n \cdot p_{LV}^*(t_k) = 0 \Rightarrow$$
$$n \cdot p_{LV}(t_k) = n \cdot \left( m \Delta \tilde{r} + C_{k+1}^k p_{LV}(t_k) \right) \Rightarrow$$
$$mn \cdot \Delta \tilde{r} = n \cdot p_{LV}(t_k) - n \cdot C_{k+1}^k p_{LV}(t_{k+1}) \Rightarrow$$
$$m = \frac{n \cdot p_{LV}(t_k) - n \cdot C_{k+1}^k p_{LV}(t_{k+1})}{n \cdot \Delta \tilde{r}}$$  \hspace{1cm} (4.17)

*Figure 71. Observed geometric change of line and laser/vision points as ownship position varies.*
In the case of one associated laser/vision point, Equation (4.17) yields an acceptable solution as long as the camera’s motion is not in the direction of the 2D line. As this is a possibility, it is preferable to perform this calculation using as many associated features as possible. Through further manipulation, Equation (4.17) can be extended to include $M$ associated features in the estimation of scale:

$$m = \frac{n \cdot p_{LV}(t_k) - n \cdot c_{k+1}^k p_{LV}(t_{k+1})}{n \cdot \Delta \tilde{r}} \Rightarrow \begin{bmatrix} n_1 \cdot \Delta \tilde{r} \\ \vdots \\ n_M \cdot \Delta \tilde{r} \end{bmatrix} m = \begin{bmatrix} n_1 \cdot p_{LV,1}(t_k) - n_1 \cdot c_{k+1}^k p_{LV,1}(t_{k+1}) \\ \vdots \\ n_M \cdot p_{LV,M}(t_k) - n_M \cdot c_{k+1}^k p_{LV,M}(t_{k+1}) \end{bmatrix}$$ (4.18)

The ordinary least squares approach is then used to solve for unknown scale:

$$m = (H^T H)^{-1} H^T y$$ (4.19)

Through experimentation, this method has been shown to produce a reliable and accurate scale estimate for motion in the x-y plane. Unfortunately, when motion is parallel (typically up and down) to the extracted vertical lines, the scale becomes unobservable as $n \cdot \Delta \tilde{r} = 0$. Therefore, this method does not estimate vertical motion very well. Fortunately, the altitude estimator described in Section 3.1.1 provides highly accurate estimates of the platform's altitude changes. The magnitude of these height changes is
added into the estimated scale factor to supplement its lack of observability in the vertical direction.

4.5 Monocular Odometry Results

To assess the capabilities of the visual odometry processes laid out in Sections 4.1 through 4.4, multiple experiments were conducted. During each test, the produced visual odometry results for rotation, shown in blue, were easily evaluated through comparison with the inertial's measured rotation, displayed in red. The rotational results for each sensor were broken down into the Euler angles of pitch, roll and yaw with respect to an established navigation frame. Unfortunately, the inertial cannot be used to evaluate the translation results of visual odometry due to its large drift. As no independent measurements were available to evaluate translation with high precision, the paths taken during each flight were carefully measured and modeled through various means including laser range finders, tape measures, and the Riegl LMS-Z360i. Due to the different data produced by these devices (i.e. distance measurements vs. environmental maps), the comparative plots for translation appear slightly different for each experiment and are explained for each of the subsequent sections. In these plots, the blue path is derived from the visual odometry results while the red elements describe "truth" data. Each experiment and its corresponding results are described and discussed in Sections 4.5.1 through 4.5.5.

4.5.1 Experiment 1: Controlled Attitude Changes in a Feature Rich Environment

Given that all 6DOF for visual odometry are extracted from the essential matrix discussed in Section 4.3, it was desired to evaluate the procedures’ ability to distinguish between rotation and translation; experiment 1 was designed for this purpose. During this
test, the sensor platform incurs controlled attitude changes in a feature rich environment while translation changes were kept to an absolute minimum. As roughly no translational motion is encountered or measured during this test, only rotation is evaluated. Figure 72 offers a sample image and extracted features for the operational environment of this experiment.

*Figure 72. Environment for experiment 1.*

Figure 73 displays the rotation estimate results for experiment 1. It can be seen that in a feature rich setting with controlled motion, visual odometry provides comparable attitude and yaw estimates to the calibrated inertial used in this work. For this experiment, the roll and yaw obtained from the camera follow the same trends as the inertial's attitude and never divert by more than 4°. The visual pitch computation matches the inertial slightly worse than the other angles, but it still follows the same trend.
4.5.2 Experiment 2: Controlled Motion Traversing a Feature Rich Hallway

To further test the estimation capabilities of the described visual odometry process, experiment 2 was conducted in the feature rich environment depicted in Figure 74. For this trial, the platform started on the ground to allow the inertial a sufficient time for leveling and bias estimation. Subsequently, the platform was manually lifted to a nominal height of ~1.5 meters, rotated around each of the inertial's axes, and eventually returned to a level state. Remaining approximately level, the platform was translated forward (x direction) 5.2 meters. Then more attitude changes were induced on the platform, followed by it returning to level ground.
The attitude results for experiment 2 are displayed below in Figure 75. In a similar fashion to the outcomes seen in Figure 73 for experiment 1, all three Euler angles estimates are comparable to the IMU outputs. The cumulative difference for roll and yaw never exceeds \(4^\circ\). However, although the pitch estimate performs well, it appears to experience more noise than the other two angles and diverts by as much as \(8^\circ\) from the inertial solution. It can be observed that the pitch estimates for experiment 1 and 2 contain more noise and cumulative error than the estimates of roll and yaw. Through analysis of these two situations, it can be seen that the observed geometry between point features is worse when the platform experiences changes in pitch. This is due to the increased view of the floor and ceiling of the environment when pitch is incurred, as these surfaces have few trackable features. In cases where measurement quality needs to be assessed this issue must weigh into the assignment of measurement confidence.
Figure 75. Visual odometry attitude estimation as platform experiences controlled motion through a hallway.

Figure 76 illustrates an evaluation of the visual odometry path calculation for experiment 2. Through comparison with the measured path, it can be seen that the visual odometry solution never exceeds more than 30 centimeters of error in the y-direction, and the solution finishes 20 centimeters past the known landing position. As the platforms true motion was approximately constrained to the x-axis, it can be stated the cross track error is produced by the eight-point algorithm and the amplified magnitude of the flight is caused by erroneous scale estimation.
4.5.3 Experiment 3: Flight Down a Feature Rich Hallway

Experiment 3 was accomplished with the intent of testing the effects of a flying sensor platform against a manually carried platform. This experiment operates in the same environment as experiment 2 and the same path was flown. In the feature dense environment displayed in Figure 77, the platform took off, flew 5.2 meters forward, and then landed. As opposed to experiment 2, no controlled attitude changes were conducted. All attitude variations were due to the platforms natural indoor dynamics and the pilot.
Through inspection of Figure 78, the attitude estimates produced by experiment 3 can be evaluated. During the 113 second flight, pitch and yaw never divert from the inertial's solution by more than 3°, while roll remains within 5°. Based on the similarities between experiments 2 and 3, comparisons can be made to evaluate the results of a flying platform verses a carried platform with regard to attitude. First, it can be seen that the attitude variations observed in experiment 3 were comparable to those observed in experiment 2. This illustrates that the visual odometry process is not adversely effected during normal flying scenarios. Second, the pitch estimate was no longer noticeably worse than the other two estimates. When inspecting the point feature geometry observed during this test, the small changes in pitch angle during experiment 3 never force the cameras view to capture a significant chunk of the feature barren floor or ceiling. The accuracy of the
pitch produced by this test gives more credence to the aforementioned hypothesis regarding the effect of point geometry on each of the individual Euler angles.

![Graph of Pitch, Roll, and Yaw over Time]

*Figure 78. Visual odometry attitude estimation during a flight traversing a hallway.*

The path produced by visual odometry during experiment 3’s flight is shown in Figure 79. The constructed path never diverts away from the measured path by more than 25 centimeters in the platforms cross-track, and the path finishes 40 centimeters before the true landing position. As the platform in this experiment was flown, as opposed to being carried in experiment 2, perturbations in the proposed flight path were inevitable. Therefore, the path deviations shown in Figure 79 cannot be unambiguously diagnosed as error, but could be flight technical errors as well. However, the flight distance was precisely measured, meaning errors were present that caused the paths magnitude to be deficient.
4.5.4 Experiment 4: Controlled Motion around a Feature Rich Indoor Loop

Experiment 4 was conducted with the intent of assessing the implemented visual odometry process while walking around an indoor hallway loop. As opposed to experiments 1 through 3, experiment 4 contained translation in multiple dimensions, large heading changes, as well as an increase in duration. During this test, the platform was carried around a rectangular loop of corridors on the third floor of the Stocker engineering building shown in Figure 80. The platform was kept a roughly constant height.
Figure 81 depicts the visual odometry and inertial attitude results obtained during experiment 4. Throughout the data collection, the maximum separation between the IMU and vision based attitude estimators for pitch, roll and yaw is 9°, 19°, and 14° respectively. Upon comparison with the maximum attitude errors present in experiments 1 through 3, experiment 4 is more erroneous. There are multiple reasons for this increase. First, the duration of experiment 4 was greater than the previous 3 trials. Errors accumulate over time, so increasing flight duration will increase cumulative error. Next, the looping path observed throughout this test cause the eight-point algorithm and scale estimation procedures to quickly adapt to differing scenery. Drastic scene changes (i.e. turning a corner) increase the difficulty of feature association between frames. This directly effects the procedures used for visual odometry in an adverse manner. Finally, unlike the first three
experiments, there are situations in experiment 4 where features are sparse. In general, a decrease in features will cause a decrease in the estimation capabilities of visual odometry.

![Figure 81](image.png)

*Figure 81. Visual odometry attitude estimation while traveling around an indoor loop.*

The visual odometry path calculation for experiment 4 can be observed in Figure 82. In this figure, it can be seen that the estimated length of each of the 4 straight legs of the rectangular loop matches to within 2 meters of the measured hallway lengths. This implies that the scale estimation technique is working reasonably well. As for the estimated translational directionality produced by the eight point algorithm, the first two legs of the loop never divert from the measured path by more than 2 meters. Unfortunately, the third leg diverts by 5 meters. This is most likely due to a lack of well dispersed features in that specific hallway. The cumulative error contained in the third linear leg of the loop also makes evaluation of the final leg difficult. However, if previous errors are removed, the
The final leg appears to match the measured path well. In total, the landing position calculated through visual odometry is 6.5 meters away from the measured end of the trial.

![Path Taken](image)

*Figure 82. Visual odometry path determination while traveling around an indoor loop.*

### 4.5.5 Experiment 5: Indoor/Outdoor Flight through a Feature Sparse Environment

The final trial to assess the described visual odometry procedure involved both indoor and outdoor flight. The UAV started on ground pointing toward a pair of buildings. Upon takeoff, the platform maneuvered toward one of the buildings and entered it through the basement doors. Once inside the environment shown in Figure 83, the platform traveled through multiple feature sparse corridors to the opposite side of the building. Retracing its path, the UAV flew the reverse of the previously described route leading it from inside to outside and landing in a similar place to where it began. Upon landing, the hexacopter was kept stationary for multiple seconds before data collection was ceased.
The camera based attitude determination results for experiment 5 can be seen in Figure 84. These results show large deviations between the inertial attitude and the attitude produced by visual odometry. Despite these considerable cumulative discrepancies, the same trends appear to exist thought the development of each Euler angle. Rationale for this is explored in the following analysis.

Figure 83. Environment for experiment 5.

Figure 84. Visual odometry Attitude estimation during indoor and outdoor flight.
Through inspection of the eight point algorithm laid out in Section 4.3, it can be determined that a lack of quality features will cause considerable errors in the attitude computation. Hence, time epochs were identified in which 10 or fewer associated features were available. Upon examination, the identified situations occurred throughout 3% of the flight and correlated directly with large changes in the platform's yaw. This can be observed in Figure 85. As most occurrences of considerable heading change occurred as the platform was turning a corner, this decrease in available features is predictable. Before pivoting into a new hallway, the platform's camera is typically very close to a flat wall with little pixel variation in the observable seen.

Figure 85. Correlation between yaw maneuvers and number of features
To demonstrate the adverse effect produced by the instances identified in Figure 85, a new attitude estimate was calculated. During time epochs yielding less than 10 trackable features, the camera based change in attitude estimate was supplemented by the inertial. The results of this procedure can be seen in Figure 86. Comparing the previous results shown in Figure 84 to the adjusted results shown in Figure 86, a substantial improvement is observed. Throughout the entire 670 second flight, the inertial and visual odometry never diverge by more than 10° in any of the Euler angles. The considerable effect of replacing only 3% of the visual odometry attitude with the inertial attitude highlights the 8-point algorithms dependence on numerous quality features. Also, after the platform completes its flight and lands (~540-670 seconds), drift can clearly be observed in the inertial attitude estimates while the camera offers stable attitude estimation. This lack of sensor drift is advantageous during long flights.

Figure 86. Visual odometry attitude estimation with supplemented inertial in feature scarce situations.
The flight path produced during experiment 5 can be seen in Figure 87. For comparison purposes, the indoor section of the flight is compared to a detailed 2D truth map obtained using the Riegl LMS-Z360i. The first leg of the flight, displayed in green in Figure 87, was flown outdoors facing a building. This allowed the camera and laser to simultaneously observe a structured outdoor environment. While no empirical way of evaluating the outdoor sections of this flight are available, it can be stated that the first leg exhibits the expected magnitude and directionality of the observed flight. The Second leg of the flight passes through the basement's indoor environment. During this period, shown in dark blue in Figure 87, the path correlates to the travelled hallway with up to 5 meters of error. These indoor path generation results are proportionate to those achieved in experiment 4. The final leg of the flight path, displayed in light blue, begins with a transition from indoor to outdoor. During this period, the camera and laser simultaneously observe a non-structured outdoor environment. The visually generated path does not remotely resemble the expected path. This final leg should return the platform to its starting location. While observing the non-structured outdoor environment, the metric scale estimation technique described in Section 4.4 breaks down, meaning scale is improperly estimated. The incorrect path calculation produced during the final leg of the flight shown in Figure 87 can significantly be attributed to invalid scale estimates. In conclusion, this experiment demonstrates that the visual odometry process described in Sections 4.1 through 4.4 provides reasonable path estimation in both structured outdoor and structured indoor environments, but fails in unstructured areas.
Figure 87. Visual odometry path determination during indoor and outdoor flight.
CHAPTER 5: METHODOLOGY

In the previous two chapters, estimators were introduced for computing the pose of a multi-copter using only laser range scanners and using a combination of vision imagery and laser range scanner data as part of the mechanization depicted in Figure 88. Both methods provide adequate relative navigation at the dm-level for laser-based navigation and meter-level for camera-based, but generated trajectories are not referenced to a global navigation frame. Since the operational scenario includes outside flight initially, GPS will be used to tie the trajectories down to an absolute global reference frame. Since, in our outside environment, 4 satellites are not always guaranteed, integration with an inertial is performed using an already existing method described by Farrell [9]. This chapter will first describe Farrell’s method and its role within the filter structure and then address the incorporation of both the laser-based and vision-based solutions into the integration filter.
5.1 Inertial Data Processing

Two of the major error sources in an IMU are the gyro bias, $\delta \omega$, and the accelerometer bias, $\delta f$, as discussed in Section 2.2. While these two error sources will be part of the state vector in the integration filter, they can initially be estimated using a so-called Zero-Velocity Update (ZUPD). When the platform is completely stationary the inertial should measure nothing except for gravity and the centripetal force associated with a rotating earth. At this time the initial attitude ($\theta$ and $\phi$) can be estimated from the orientation of the observed gravity vector by the accelerometers as seen in Groves [118]. In addition, the average of $N$ gyro-measurements is used to obtain the gyro bias:
Similarly, the accelerometer bias can be obtained as follows:

\[
\delta \mathbf{f}_b = \frac{1}{N} \sum_{k=0}^{N} \mathbf{f}_b(t_k)
\]

(5.2)

However, the local gravity, expressed in the body-frame, must be subtracted from this estimate. For this work, the platform is kept stationary for \(N=20-60\) seconds at the beginning of each data set and biases are estimated during this period. Once biases are calculated they are subtracted from every measurement made by the inertial.

In addition to bias estimation and leveling, we must be able to express the inertial outputs at the time-tags associated with the measurements of the other sensors. For a typical implementation of an inertial mechanization, the update rate of the inertial must be high enough to capture the platform’s true motion. At these high update rates, it is often assumed that the motion between two consecutive epochs is linear, allowing for linear interpolation of the IMU's angular rates and accelerations to obtain measurements that coincide with the time stamps associated with each laser point from all three lasers, the 2D camera, and the GPS receiver. Having these interpolated values available, the mechanization is capable of computing the inertial pose at the exact time of the other measurement.

5.2 Dynamics and Error Estimator

To combine the data from all of the different sensors discussed in this work, an extension of the filtering mechanization designed by Farrell [9] is implemented. Farrell
establishes a structure for GPS/inertial integration that consists of two filters: a dynamics filter and a position filter. Adapting and extending the dynamics filter to include monocular camera and laser data within its mechanization is the focus of the following section.

The dynamics filter is a Complementary Kalman Filter (CKF) designed to estimate the dynamics of the inertial error states: velocity error in the NED frame, $\delta v^n$, misorientation (including tilt error), $\delta \psi_{nb}$, gyro bias error, $\delta \omega^b$, and specific force or accelerometer bias error, $\delta f_b$. This yields the following state vector:

$$
\mathbf{x}_1 = [\delta v^n \; \delta \psi_{nb} \; \delta \omega^b \; \delta f_b]
$$

(5.3)

Within our approach the inertial is established as the mechanizations core sensor as it is not dependent on any external infrastructure. This allows for consistent performance in any environment.

Given the state vector shown in Equation (5.3), and the dynamics equation defined in equations (2.3) and (2.4), we can define the following dynamics equation for the CKF:

$$
\mathbf{x} = \mathbf{Fx} + \mathbf{w}
$$

(5.4)

Where $\mathbf{F}$ is the state transition matrix

$$
\mathbf{F} = \begin{bmatrix}
0 & -\Delta t (\mathbf{f}_n \times) & 0 & \Delta t \mathbf{C}^n_b \\
0 & 0 & \Delta t \mathbf{C}^n_b & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
$$

(5.5)
and the process noise matrix, $W$, is defined by equation (5.6). The noise elements of this matrix are defined as the velocity error process noise, $\sigma_{p,v}^2$, the misorientation process noise, $\sigma_{p,\psi}^2$, the gyro bias process noise, $\sigma_{p,\omega}^2$, and the accelerometer bias process noise, $\sigma_{p,f}^2$. Values for these parameters are tuned with respect to the amount of uncertainty we expect from the dynamics model. For the remaining elements of $W$, $I$ represents a 3x3 identity matrix and $\theta$ represent a 3x3 matrix of all 0's.

$$W = \begin{bmatrix}
\sigma_{p,v}^2 I & 0 & 0 & 0 \\
0 & \sigma_{p,\psi}^2 I & 0 & 0 \\
0 & 0 & \sigma_{p,\omega}^2 I & 0 \\
0 & 0 & 0 & \sigma_{p,f}^2 I
\end{bmatrix} \quad (5.6)$$

Next, the discrete-time transition matrix, $\Phi_k$, and the discrete-time system noise covariance matrix, $Q$, are established following the derivation in [17]. By defining $G$ as a 12x12 identity matrix, Equations (5.7) through (5.9) can be completed to calculate $\Phi$ and $Q$:

$$A = \begin{bmatrix}
-F & GWG' \\
0_{12\times12} & F'
\end{bmatrix} \Delta t \quad (5.7)$$

$$B = e^A = \begin{bmatrix}
B_{\text{upper left}} & B_{\text{upper right}} \\
B_{\text{lower left}} & B_{\text{lower right}}
\end{bmatrix} \quad (5.8)$$

$$\Phi = B_{\text{lower right}}^T \quad (5.9)$$
For the continuous-time dynamics equation (3.2), this process yields the following state transition matrix:

\[
\Phi = \begin{bmatrix}
1 & -\Delta t (f^n \times) & \frac{-\Delta t^2}{2} (f^n \times) C^n_b & \Delta t C^n_b \\
0 & 1 & \Delta t C^n_b & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\tag{5.10}
\]

which in turn is used to obtain \( Q \):

\[
Q = \Phi B_{\text{upper right}}
\tag{5.11}
\]

Both \( Q \) and \( \Phi \) are re-calculated for every Kalman filter iteration since they are dependent on values that are being estimated.

The remaining undefined Kalman filter parameters involve the obtained measurements from the GPS receiver, the pose from the monocular camera, and the pose derived from the laser scanners. For each of these systems, the parameters include the observation matrix, \( H \), the measurement vector, \( z \), and the measurement error covariance matrix, \( R \). Each of these variables are uniquely calculated to relate each type of secondary sensor measurement to measurements from the inertial. Therefore, Sections 5.2.1 through 5.2.3 establish these parameters for each type of sensor.

5.2.1 GPS Kalman Filter Equations

To relate measurements from the GPS receiver to those from the IMU, the GPS satellite and receiver geometry must be examined. This geometry is shown in Figure 89 for one satellite and one receiver at consecutive time epochs \( t = t_{k-1} \) and \( t = t_k \).
Following Farrell [9] and Uijt de Haag and van Graas [119], single differences can be formed with respect to time (a.k.a. sequential differences) by examining the relative change in geometry between satellite 'j' and the receiver across two time epochs.

\[
\Delta R_j = e^T_j(t_k)R_{SV_j}(t_k) - e^T_j(t_{k-1})R_{SV_j}(t_{k-1}) - [ e^T_j(t_{k-1})e^T_j(t_k) ]r(t_{k-1}) - e^T_j(t_k)\Delta r
\]

Given \( M+1 \) observed satellites, Equation (5.12) can be used to produce \( M+1 \) single differences; one per satellite. Through various metrics which are typically used to identify the most reliable satellite, a key satellite 'k' is chosen. Using the single difference associated with the key satellite, differencing between the key satellite and each other satellite is done to yield \( M \) double differences:

\[
\nabla \Delta R_{jk} = \Delta R_j - \Delta R_k \Rightarrow
\]

Figure 89. GPS dynamics filter geometry (adapted from [119])
\[ \nabla \Delta R_{jk} = e^T_j(t_k)R_{SV_j}(t_k) - e^T_j(t_{k-1})R_{SV_j}(t_{k-1}) - [e^T_j(t_{k-1})e^T_j(t_k)]r(t_{k-1}) \]

\[ - e^T_j(t_k)\Delta r - e^T_k(t_k)R_{SV_k}(t_k) - e^T_k(t_{k-1})R_{SV_k}(t_{k-1}) \]

\[ - (e^T_k(t_{k-1})e^T_k(t_k))r(t_{k-1}) - e^T_k(t_k)\Delta r \]

By defining \( h_{kl}(t_k) = e^T_k(t_k) - e^T_j(t_k) \), Equation (5.13) can be manipulated (derivation found in Appendix A):

\[ \nabla \Delta R_{jk} = h_{kl}(t_k)\Delta r + a_{jk} + b_{jk} \text{ compensation terms} \quad (5.14) \]

where \( a_{jk} \) and \( b_{jk} \) are compensation terms that can be computed and removed. The range measurements used to form Equation (5.15) are carrier phase measurements rather than pseudoranges since the noise levels on these measurements are much lower (mm-level rather than dm-level). Defining \( N \) as an integer number of carrier phase cycles and \( \lambda \) as the GPS signals wavelength, errors in carrier phase measurements can be obtained:

\[ \phi_j(t_k) = R_j + \delta t_{rcvr} + N\lambda + \epsilon_{tropo} + \epsilon_{iono} + \epsilon_{mp,cp} + \epsilon_{noise,cp} + \epsilon_{sv} \quad (5.15) \]

Substituting these measurement in equation (5.12) through (5.14) and exploiting the fact that many of the error sources are either spatially or temporally correlated and, thus, get cancelled by the difference operation:

\[ \Delta \phi_j = \phi_j(t_k) - \phi_j(t_{k-1}) = \Delta R_j + \Delta \delta t_{rcvr} + \Delta \epsilon_j \quad (5.16) \]
where $\Delta t_{rcvr}$ represents the receiver clock frequency bias and $\Delta \varepsilon_j$ contains residual (uncorrelated) atmospheric, orbit, and noise errors. Subsequently, double differences can be taken w.r.t. a key satellite:

$$
\nabla \Delta \phi_{jk} = \Delta \phi_j - \Delta \phi_k = \frac{\Delta R_j - \Delta R_k}{0} + \Delta \varepsilon_j - \Delta \varepsilon_k + \frac{\Delta \delta t_{rcvr} - \Delta \delta t_{rcvr}}{\Delta \varepsilon_{jk}} \Rightarrow
$$

$$
\nabla \Delta \phi_{jk} = h_{kj}(t_k)\Delta \mathbf{R} + a_{jk} + b_{jk} + \Delta \varepsilon_{jk}
$$

(5.17)

The double differenced carrier phase measurement errors of Equation (5.17) are formed over the time epoch from $t_{k-1}$ and $t_k$. Over this same time epoch, inertial measurements are collected and analyzed to form a position change measurement, $\Delta \mathbf{r}_{INS}$. This inertial-derived position change is substituted into the $h_{kj}(t_m)\Delta \mathbf{r}$ term of Equation (5.17) and related to the inertial error measurements of the state vector of Equation (5.3).

$$
h_{kj}(t_m)\Delta \mathbf{r}_{INS} = h_{kj}(t_m) \int_{t_{m-1}}^{t_m} v^n_e \, dt + h_{kj}(t_m) \left[ \dot{c}_b^n(t_m) - \ddot{c}_b^n(t_{m-1}) \right] \Rightarrow
$$

$$
= h_{kj}(t_m) \int_{t_{m-1}}^{t_m} v^n_e \, dt + h_{kj}(t_m) \left[ \Psi \times C_b^n(t_m) - \Psi \times C_b^n(t_{m-1}) \right]
$$

(5.18)

Subtracting Equation (5.17) from (5.18), yields a complementary measurement that compares the inertial and GPS carrier phase measurements.

$$
z_{jk} = h_{kj}(t_m)\Delta \mathbf{r}_{INS} - \nabla \Delta \phi_{jk} + a_{jk} + b_{jk}
$$

(5.19)

For $M+1$ satellites, Equation (5.19) will contribute $M$ measurements to the following measurement vector:
Based on the linear measurement equation for an EKF, $z = Hx + v_k$, the measurement vector of Equation (5.20) is related to the state vector in Equation (5.21) through the observation matrix:

$$
\begin{bmatrix}
Z_{1,k} \\
Z_{2,k} \\
\vdots \\
Z_{M,k}
\end{bmatrix}
$$

(5.20)

$$
H = \begin{bmatrix}
  h_{k1}(t_m) [I_{3x3} & 0_{3x9}] & \left[ \int_{t_{m-1}}^{t_m} \Phi(r, t_{m-1}) \, dr \right] \Phi^{-1}(t_m, t_{m-1}) \\
  h_{kM}(t_m) [I_{3x3} & 0_{3x9}] & \left[ \int_{t_{m-1}}^{t_m} \Phi(r, t_{m-1}) \, dr \right] \Phi^{-1}(t_m, t_{m-1}) \\

+ \\
\begin{bmatrix}
  0_{3x3} & [\delta C^0_b(t_m) l \times h_{k1}(t_m)] & [(t_m - t_{m-1}) C^n_b(t_{m-1}) l \times h_{k1}^T(t_m)] & 0_{3x3} \\
  \vdots & \vdots & \vdots & \vdots \\
  0_{3x3} & [\delta C^0_b(t_m) l \times h_{kM}(t_m)] & [(t_m - t_{m-1}) C^n_b(t_{m-1}) l \times h_{kM}^T(t_m)] & 0_{3x3}
\end{bmatrix}
$$

(5.21)

where $l$ is the lever arm between the GPS receiver and the inertial. For further explanation of Equations (5.12) through (5.21) and a derivation of Equation (5.21), the reader is referred to [9] and [119]. The measurement covariance matrix $R$ is determined by the expected error covariance of the double difference measurements in Equation (5.17). However, due to the sequential difference operation and the difference operation with respect to a key satellite, correlations have been introduced causing the off-diagonal elements to be non-zero, and, thus, the noise to be non-white. The effect of sequential differencing results in negatively correlated errors due to overlapping carrier-phase measurements. This effect can be easily mitigated by whitening the measurement. This procedure is quite simple and provided by [9]. The correlation introduced by differencing w.r.t. a common key satellite is positive and can be mitigated by state augmentation.[120]
5.2.2 Laser Scanner Kalman Filter Equations

As laser scans are obtained, they are processed by means of the procedures discussed in Chapter 3. Between scans at \( t_{k-1} \) and \( t_k \), laser based estimates for position and attitude changes are defined as:

\[
\Delta r_{\text{laser}} = \begin{bmatrix} \Delta x^n \\ \Delta y^n \\ \Delta z^n \end{bmatrix}
\]  
(5.22)

and:

\[
\Delta \Psi_{\text{laser}} = \begin{bmatrix} \Delta \theta^n \\ \Delta \phi^n \\ \Delta \psi^n \end{bmatrix}
\]  
(5.23)

for the time epoch \( \Delta t = t_k - t_{k-1} \). Over the same period \( \Delta t \), IMU measurements are mechanized to produce an inertial-derived position change:

\[
\Delta r_{\text{INS}} = \begin{bmatrix} \Delta x^n \\ \Delta y^n \\ \Delta z^n \end{bmatrix}
\]  
(5.24)

and an inertial-derived rotation change:

\[
\Delta \Psi_{\text{INS}} = \begin{bmatrix} \Delta \theta^n \\ \Delta \phi^n \\ \Delta \psi^n \end{bmatrix}
\]  
(5.25)

Through comparison of the position and rotation changes estimated by the lasers and the inertial, complementary measurements can be formed.
\[ \delta r = \Delta r_{INS} - \Delta r_{laser} \]  
(5.26)

\[ \delta \psi = \Delta \theta_{INS} - \Delta \theta_{laser} \]  
(5.27)

Further manipulation allows these complementary measurements to relate to elements of the state vector in Equation (5.12).

\[ \delta r = \int_{t_{k-1}}^{t_k} \delta v \approx \Delta t \, \delta v^n \]  
(5.28)

\[ \delta \psi \approx \delta \psi_{nb} \]  
(5.29)

Given the form of the linear measurement equation, \( z = Hx + v_k \), Equations (5.28) and (5.29) can be written as

\[
\begin{bmatrix}
\Delta r_{INS} - \Delta r_{laser} \\
\Delta \psi_{INS} - \Delta \psi_{laser} \\
z
\end{bmatrix} =
\begin{bmatrix}
\Delta t & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\delta v^n \\
\delta \psi_{nb} \\
\delta \omega_{b}^b
\end{bmatrix} + v_k 
\]  
(5.30)

Equation (5.30) lays out the measurement vector and observation matrix necessary to include laser data in the derived EKF.

To form the corresponding error covariance matrix associated with the \( z \) vector in Equation (5.30), the laser-based estimation methods of Chapter 3 are inspected. With respect to the 3DOF SLAM method, uncertainty estimates for \( (x, y, \psi) \) are obtained by quantifying the quality of the scan matching procedure used in SLAM. Kohlbrecher[41] and Burguera [121] accomplish this by examining how well each collected scan point...
matches the accumulated map as a function of platform state estimates. Within the SLAM
procedure, an observation matrix, $H_{\text{SLAM}}$, is produced that relates each scan measurement
to the 3DOF state vector. This observation matrix can then be used to obtain the SLAM
error covariance matrix:

$$R_{\text{SLAM}} = \sigma_{\text{L,SLAM}}^2 H_{\text{SLAM}}^{-1}$$ (5.31)

where $\sigma_{\text{L,SLAM}}^2$ represents the expected measurement noise of the physical laser.

For the laser based altitude ($z$) and attitude ($\varphi, \theta$) measurements, advanced methods
were not used to obtain uncertainty metrics. Instead, the error covariance matrix is
populated based on expected noise values for altitude estimation, $\sigma_{\text{L,alt}}^2$, and attitude
estimation, $\sigma_{\text{L,att}}^2$. Therefore, the error covariance matrix for laser-based estimates is
defined as:

$$R = \begin{bmatrix}
R_{11}^{\text{SLAM}} & R_{12}^{\text{SLAM}} & 0 & 0 & 0 & R_{13}^{\text{SLAM}} \\
R_{21}^{\text{SLAM}} & R_{22}^{\text{SLAM}} & 0 & 0 & 0 & R_{23}^{\text{SLAM}} \\
0 & 0 & \sigma_{\text{L,alt}}^2 & 0 & 0 & 0 \\
0 & 0 & 0 & \sigma_{\text{L,att}}^2 & 0 & 0 \\
0 & 0 & 0 & 0 & \sigma_{\text{L,att}}^2 & 0 \\
R_{31}^{\text{SLAM}} & R_{32}^{\text{SLAM}} & 0 & 0 & 0 & R_{33}^{\text{SLAM}}
\end{bmatrix}$$ (5.32)

where the superscripts on $R_{\text{SLAM}}$ indicate matrix element indices.
5.2.3 Monocular Camera Kalman Filter Equations

Before use in the integrated solution, the monocular camera data is processed through the visual odometry procedure laid out in Chapter 4. Given a set of images at \( t_{k-1} \) and \( t_k \), the method yields the following estimated position changes:

\[
\Delta r_{\text{camera}} = \begin{bmatrix} \Delta x^n \\ \Delta y^n \\ \Delta z^n \end{bmatrix}
\]  

(5.33)

and:

\[
\Delta \psi_{\text{camera}} = \begin{bmatrix} \Delta \theta^n \\ \Delta \phi^n \\ \Delta \psi^n \end{bmatrix}
\]  

(5.34)

for the period \( \Delta t = t_k - t_{k-1} \). Through comparison of the visual odometry measurements found in Equations (5.33) and (5.34) with the laser measurement Equations in (5.22) and (5.23) a similarity is observed. The units and the reference frame of measurements from both the camera and lasers are identical as they are input into the filter mechanization. Therefore, the laser measurement relationships established in Equations (5.24) through (5.29) correspondingly hold true for camera measurements. This produces the following linear measurement equation:

\[
\begin{bmatrix} \Delta r_{\text{INS}} - \Delta r_{\text{camera}} \\ \Delta \theta_{\text{INS}} - \Delta \theta_{\text{camera}} \end{bmatrix}_x = \begin{bmatrix} 1 \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}_H \begin{bmatrix} \delta v^n \\ \delta \psi_{nb}^b \\ \delta \omega_{ib}^b \\ \delta f_b^x \end{bmatrix} + v_k
\]  

(5.35)
Equation (5.35) establishes the measurement vector and observation matrix for processed camera data.

To form the error covariance matrix associated with the camera pose estimates, the visual odometry methods described in Chapter 4 are examined. As shown in Chapter 4 Equation (4.3), the pose estimate is derived from the essential matrix, $E$, which is derived from the normalized pixel coordinates. To relate the covariance of the pose estimate to the pixel uncertainty from the feature extraction method, we must take into account the internal structure of the 8-point algorithm. This issue has been addressed in Sur [122] and Papadopoulo [123]. On the input end we have the pixel uncertainty in $x$- and $y$-components of feature vector $p_i$ (see Equation (2.32)). Csurka [124] and Sur [122] have established methods to find the covariance of the fundamental matrix $R_F$ as a function of the pixel covariance, $R_{pix}$. Given that the essential matrix is a similarity transformation of the fundamental matrix given by Equation (4.3) the covariance matrix of $E$ can be approximated as:

$$R_E = \frac{\partial (K^TFK)}{\partial F} R_F \frac{\partial (K^TFK)^T}{\partial F}$$

(5.36)

Following the steps described in Section 4.3, the rotation matrix uncertainty and translation uncertainty can be found as follows [123]:

$$R_c = \frac{\partial (UWV^T)}{\partial E} R_E \frac{\partial (UWV^T)^T}{\partial E}$$

(5.37)

and:
The $3 \times 3$ translation covariance matrix obtained through Equation (5.38) relates directly to the three translation states of the platform. Unfortunately, the rotational uncertainty calculation yields a $9 \times 9$ matrix whose form is unusable. Extracting the covariance matrix of the attitude and yaw estimates from this $9 \times 9$ covariance matrix falls outside the scope of this dissertation, but should be addressed in future work. Therefore, a rotational uncertainty term $\sigma_R^2$ is calculated based on the number of extracted features and the geometric distribution of those features thought the image frame. This uncertainty is reduced as the number of features increases and geometric distribution of features broadens. The corresponding error-covariance matrix is defined as:

$$
R = \begin{bmatrix}
    R_{\Delta r} & 0_{3 \times 3} \\
    0_{3 \times 3} & \sigma_R^2 I_{3 \times 3}
\end{bmatrix}
$$

(5.39)

5.3 Position Estimator

The position estimator for this work is very simple KF implementation that uses the outputs of the corrected inertial from the previously discussed dynamics filter and satellite pseudoranges. The state vector for this filter is shown in Equation (5.40) where each state is a coordinate in a north-east-down (NED) frame.

$$
x = [x \ y \ z]^T
$$

(5.40)
Given the output of the aforementioned dynamics filter, a velocity estimate, $\dot{v}(t_k)$, is obtained from the calibrated inertial mechanization. As the IMU operates at a much faster update rate than the GPS receiver, there are multiple inertial measurements between GPS measurements. Between two GPS updates there will be 'N' inertial measurements with corresponding integration periods, $\Delta t$. This allows for the formation of Equation (5.41) to obtain a position displacement vector, $\Delta r^n_{\text{dynamics}}$, from the dynamics filter outputs.

$$\Delta r^n_{\text{dynamics}}(t_{k+N}) = \sum_{k=m}^{m+N} \dot{v}(t_k)\Delta t$$  \hspace{1cm} (5.41)

Using Equation (5.41), a basic relationship for calculating the desired position state is shown in Equation (5.42). By adding the platform's position displacement to its previous position, $r^n(t_{k-1})$, the current position, $r^n(t_k)$, can be obtained:

$$r^n(t_k) = r^n(t_{k-1}) + \Delta r^n_{\text{dynamics}}(t_{k+N})$$  \hspace{1cm} (5.42)

Elements of Equation (5.42) can then be related to the position filter state vector:

$$r^n(t_k) = x(t_k) = [x \ y \ z]^T$$  \hspace{1cm} (5.43)

Substituting Equation (5.43) into Equation (5.42), the position filter's state transition equation can be formed:

$$x(t_{k+1}) = \Phi(t_{k+1}, t_k) x(t_k) + \Delta r^n_{\text{dynamics}}(t_{k+N}) + w(t_{k+1})$$  \hspace{1cm} (5.44)
The state transition matrix for Equation (5.44) is an identity matrix due to the direct projection of the state vector:

\[
\Phi(t_{k+1}, t_k) = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]  

When pseudorange measurements are available, they are used to form the measurement vector, \( \mathbf{z} \), for the KF. Given \( N \) number of satellites, \((N-1)\) single differences can be formed between each obtained satellite \((j)\) and a key satellite \((k)\). The single difference calculation between two satellites is \( \mathbf{z} \):

\[
\mathbf{z} = R_j - R_k
\]  

These single differences create a vector of length \( N - 1 \). Through manipulation, \( \mathbf{z} \) is established in terms of a geometry vector, \( \mathbf{h}_{kj} \), and the state vector:

\[
\mathbf{z} = R_j - R_k \Rightarrow
\mathbf{z} = \begin{bmatrix} e_j^T(t_k)R_j(t_k) - e_j^T(t_k)x(t_k) - e_k^T(t_k)R_k(t_k) + e_k^T(t_k)x(t_k) \end{bmatrix} \Rightarrow
\mathbf{z} = \begin{bmatrix} e_k^T(t_k) - e_j^T(t_k) \end{bmatrix} x(t_k) + \begin{bmatrix} e_j^T(t_k)R_j(t_k) - e_k^T(t_k)R_k(t_k) \end{bmatrix} \Rightarrow
\mathbf{z} = \mathbf{h}_{kj}x(t_k) + a_{kj}
\]  

Using \( N \) measurements, \((N-1)\) geometry vectors are produced to form an \((N-1)\times3\) observation matrix, \( \mathbf{H} \):
As previously stated, this filter initializes at the origin of an established local navigation frame. However, the local frame is relative and an absolute reference frame is desired. Given the above mechanization, if four or more satellite pseudoranges are available, an absolute position calculation is made. When an absolute position is obtained, the reference frame for the filter is converted from a local NED frame to an absolute earth centered earth fixed (ECEF) frame. Once conversion to this frame is made, the filter proceeds to output absolute positions regardless of the number of pseudorange measurements. Essentially, if an absolute position is ever obtained, then all of the filters outputs will correspondingly be absolute.

5.4 Mapping

The 3D mapping capability for this work is based off of Slawomir's findings in [40]. Slawomir observed that the natural dynamics of a multi-copter forces onboard laser scanners to physically observe differing sections of its surroundings. Soloviev et al. [46] took this one step further and proposed that the platform would also introduce intentional motion to better observe its environment. If the platforms motion is known, individual laser scans can be transformed into a navigation frame and concatenated to form a 3D point cloud. The advantageous relationship between platform state and laser scan plane is illustrated through Figure 90.

\[
H = \begin{bmatrix}
    e^T_k(t_k) - e^T_i(t_k) \\
    e^T_k(t_k) - e^T_j(t_k) \\
    \vdots \\
    e^T_k(t_k) - e^T_{N-1}(t_k)
\end{bmatrix}
\]  

(5.48)
To use laser data within the described mapping procedure, a conversion must be performed on scan points to transform them from a sensor frame, \( p^s_i \), to an established navigation frame, \( p^b_i \).

\[
p^n_i = r^n + C^n_b (C^b_s p^s_i + I)
\]  

(5.49)

The sensor to body rotation matrix, \( C^b_s \), and the sensors lever arm, \( I \), parameters are derived through a platform calibration. For this work, values for these attributes are found in Table 2 of Section 6.1. The body to navigation rotation matrix, \( C^n_b \), and the navigation frame position, \( r^n \), are obtained from the outputs of the previously discussed dynamics filter (Section 5.2) and position filter (Section 5.3).

As previously stated, the hexacopter platform used in this research is equipped with three onboard laser scanners. Point cloud data from all three scanners is transformed to the navigation frame using Equation (5.49).

Many robotic applications have investigated methods to more efficiently store this point cloud data using, for example, voxel grids and octomaps. Both these methods divide
the 3D space into small cubic volumes referred to as voxels. These voxels can then be subdivided in smaller voxels if there is reason to do so. Efficient methods have been implemented to store and access all these voxel (e.g. for ray tracing purposes), such as the octree.[125] Although, these methods are available under ROS as part of the Point Cloud Library (PCL) and the OctoMap libraries, these efficient storage methods were not used. Future work may include these for mapping, as they can be used when implementing a 3D occupancy grid.
CHAPTER 6: TEST SETUP

In order to evaluate the proposed algorithms, data was collected through multiple flights and walk-throughs of a hexacopter, described in Section 6.1, through a structured indoor and outdoor environment and including transitions between these two environments. The UAV was equipped with multiple sensors which are discussed in Sections 6.2 through 6.6. Apart from the data collections with the hexacopter, truth reference maps were created for the indoor operational environment. These maps were rendered using a Riegl laser scanner discussed in Section 6.7.

6.1 Hexacopter Platform

To collect the data required for this study, the hexacopter shown flying in Figure 91 was designed, built, and tested. The basic frame of this platform was purchased as a kit and then modified to accommodate the needs of this research. The designed hexacopter has been shown to be capable of flying both indoor and outdoor while collecting laser, GPS, inertial, barometric, and digital imagery data.
The most significant modification made to the base hexacopter platform was the addition of each of the sensors. These sensors include (i) a Point Grey FireFly MV 2D camera, (ii) a Hokuyo UTM-30LX and two Hokuyo URG-04LX laser range scanners, (iii) a NovAtel OEMSTAR and corresponding L1 Antenna to obtain GPS measurements, (iv) an Xsens Mti IMU, and (v) an Ardupilot Mega(APM) 2.5 to perform the platform control and provide barometric measurements. Figure 92 shows the location of each of these sensors onboard the Hexacopter.

In Figure 93, the interconnections of the entire data collection system can be seen. Using either RS232 or USB for communication, each of the sensors is connected to a FitPC, which is mounted just under the main sensor stack on the hexacopter. The FitPC is a small-size Intel Atom Z530-based embedded computer and runs robotic operating system (ROS).
on top of Ubuntu. ROS is an open-source software framework that is intended for the
development of software for robot applications. ROS provides “hardware abstraction,
device drivers, libraries, visualizers, message-passing, package management, and more
[126].” Through ROS, the FitPC can communicate with each of the sensors and interpret
and analyze their messages. For data diagnostic purposes and communication with each of
the sensors during flight, an 802.11b wireless connection is established between the FitPC
and an external ground station. During the data collections, all of the desired sensor data is
recorded and time stamped on the FitPC in ROS-specific data files, referred to as "rosbags",
and then offloaded to the ground station where ROS and Matlab are used to analyze the
data. “Rosbags” allow for easy playback of data.

To enable operation of this platform with the sensor payload, an analysis of weight
and power consumption was performed; the results are displayed in Table 2. Given the
weight of the platform with a full payload being over 4kg, motors were chosen that could produce a maximum of 1.38kg of thrust each. These motors yield a combined possible thrust of 7.3kg if all 6 propellers are operating at their upper limit. This excess amount of thrust allows the pilot to have reserve power for the purpose of correcting for external sources of interference such as wind gusts during flight. Once these motors were installed on the hexacopter, their power consumption was tested. For normal flight they drew between 50 and 60 amps combined. It can be seen in Table 2 that most of the power consumed by this platform is drawn from the motors. Therefore, it was determined that two batteries would be used to isolate the sensors from operating on the same power source as the motors.

Table 2. Hardware specifications.

<table>
<thead>
<tr>
<th>Components</th>
<th>Weight(g)</th>
<th>Operation Voltage(V)</th>
<th>Estimated Power Consumption (W)</th>
<th>Lever Arm(cm)</th>
<th>Rotation (deg) [roll, pitch, yaw]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xsens</td>
<td>50</td>
<td>5</td>
<td>.35</td>
<td>[0,0,0]</td>
<td>[0,0,0]</td>
</tr>
<tr>
<td>Firefly</td>
<td>35</td>
<td>5</td>
<td>1</td>
<td>[3.5,0,8.3]</td>
<td>[0,0,0]</td>
</tr>
<tr>
<td>UTM-30LX(F)</td>
<td>245</td>
<td>12</td>
<td>9</td>
<td>[0,0,11.4]</td>
<td>[0,0,0]</td>
</tr>
<tr>
<td>URG-04LX(D)</td>
<td>160</td>
<td>5</td>
<td>2.5</td>
<td>[11.4,0,-8.3]</td>
<td>[0,90,180]</td>
</tr>
<tr>
<td>URG-04LX(A)</td>
<td>160</td>
<td>5</td>
<td>2.5</td>
<td>[-11.4,0,-8.3]</td>
<td>[0,135,180]</td>
</tr>
<tr>
<td>APM</td>
<td>35</td>
<td>5</td>
<td>1</td>
<td>[0,0,-1.9]</td>
<td>[0,0,0]</td>
</tr>
<tr>
<td>OEMStar</td>
<td>135</td>
<td>5</td>
<td>.5</td>
<td>[-3.8,0,17.8]</td>
<td>[0,0,0]</td>
</tr>
<tr>
<td>FitPC</td>
<td>250</td>
<td>12</td>
<td>7</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Motors/Frame</td>
<td>3225</td>
<td>12</td>
<td>600-700</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Total</td>
<td>4295</td>
<td>N/A</td>
<td>623.85-723.85</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
To accurately fuse the sensor information gathered on this platform, relative lever arms had to be measured and these can be found in Table 2 as well. All lever arms were measured relative to the center of the inertial unit (see Section 2.7 on coordinate frames). Using the specified origin, a coordinate frame was designated for the platform by the x-axis out of the nose of the platform, the y-axis out of the left side, and the z axis pointing upward.

6.2 Point Grey Firefly

To obtain the necessary 2D digital imagery for this research, the Point Grey Firefly MV was chosen [127]. The Firefly, which can be seen in Figure 94, is a compact monochrome camera which operates using less than 1 W of power and weighs roughly 35g. At its limits, it can capture data at up to 60 frames per second (FPS) with 752x480 active pixels on a Micron MT9V022 1/3" progressive scan complementary metal oxide semi-conductor (CMOS) imaging sensor.

Taking into consideration the target application of the 2D camera within the scope of this research, the firefly meets all requirements. Its compact size and limited power consumption are desirable with both attributes being at a premium on the hexacopter platform. For this study the standard resolution of 640x480 pixels was used at a frame rate of 20Hz. As for the scan type for the imaging sensor, the Firefly implements a progressive scan instead of an interlaced scan. According to [128], this scan type is advantageous in scenarios, such as ours, in which relative motion exists between consecutive image frames.
6.3 Hokuyo UTM-30LX and URG-04LX Laser Scanners

Three lasers at different orientations were mounted on the hexacopter platform (see Figure 5 in Chapter 1). The data from these 2D laser scanners provide "slices" of the environment in which they are operating. With respect to the body frame of the hexacopter, the Hokuyo UTM-30LX is mounted in such a way that it scans in parallel to the rotor deck (plane though the propellers). The laser range scanner has an angular range of 270° and a linear range of 30 meters. The UTM-30LX [129] weighs 245g. Using a \( \sim 0.28^\circ \) angular resolution, 960 scan points are captured per scan at a rate of 30Hz.

The other two lasers, Hokuyo URG-04LX's [87], are mounted below the hexacopters propellers and weigh 160g a piece. These short range lasers have a 4 meter range with a 240° scanning angle and a \( \sim 0.35^\circ \) angular resolution. This provides each scan with 682 distance measurements. The UTM-30LX and the URG-04LX are depicted in Figure 95.
6.4 NovAtel OEMStar GPS Receiver

To acquire GPS signals, a NovAtel OEMStar GPS receiver and corresponding NovAtel L1 patch antenna were installed on the UAV platform. The OEMStar receiver can provide position, velocity, and timing (PVT) information by tracking both L1 and SBAS signals. It supports up to 14 channels and allows for carrier phase tracking. While consuming less than 500mW, the OEMStar can provide PVT information at a rate equal to 10Hz. For this study, a 1Hz. update rate was determined to be adequate. The receiver and antenna have a combined weight (including antenna cabling) of 135g and are shown below in Figure 96.

Figure 95. Hokuyo UTM-30LX(left) and URG-04LX laser scanners(right).

Figure 96. NovAtel OEMStar receiver(left) and NovAtel L1 patch antenna (right).
6.5 XSENS MTi Inertial Measurement Unit

The inertial unit chosen for this research was the XSENS MTi IMU [131] shown in Figure 97. It weighs 50g, and is a Micro Electromechanical System (MEMS) based unit that is commercially available. It is comprised of 3 orthogonal gyroscopes, accelerometers and magnetometers whose respective expected noise levels are .05 (deg/s/√Hz), .002 (deg/√s/√Hz), and .5 mGauss. Each of these sensors can collect data at a rates as high as 512Hz. For this research, a 100Hz. data rate was deemed sufficient due to vehicle dynamics not being exceptionally high.

Figure 97. XSENS MTi inertial measurement unit.

6.6 Arduopilot Mega

The Ardupilot Mega (APM) board performs multiple functions related to the platforms flight control and data acquisition. It is equipped with RC receiver input ports, pulse width modulation (PWM) output ports, a magnetometer, an inertial unit, and a barometric altimeter. For stable flight, the APM requires a calibration of its onboard sensors with the platform it is mounted to. After calibration, the APM can receive RC
commands from a pilot and issue motor control operations to the platforms electronic speed controllers (ESC), which control the platforms motors. The interpretation of the RC signals combined with the outputs of the APM's sensor data allow the onboard Ardupilot software to issue commands to the UAV's motors while keeping it in a stable mode. For this study, only the APM's barometric data was recorded for analysis.

![Ardupilot Mega](image)

*Figure 98. Ardupilot Mega.*

### 6.7 Riegl LMS-Z360i

To evaluate the 3D mapping and navigation capability a truth reference database has been generated using a professional terrestrial laser scanner, the Riegl LMS-Z360i [132]. The Riegl LMS-Z360i, shown on the left side of Figure 99, consists of a terrestrial laser scanner with a 90° by 360° field of view and a Nikon D100 digital camera. The laser scanner can accurately measure ranges of up to 200m with a range accuracy of up to 6mm and can be used to create dense point clouds of most constrained environments. While the laser is collecting point cloud data, the digital camera can be used to simultaneously collect calibrated imagery data. Since the data from the two sensors is precisely calibrated, triangulation and texturing algorithms can be used to accurately overlay the imagery data.
onto the 3D point cloud. This creates 3D maps, similar to the one seen on the right of Figure 99, which shows imagery data in which each pixel has an associated depth. In some complex environments, it is desirable to take multiple scans from different positions to observe everything of interest. These scans can be stitched together using Riegl's RiSCAN Pro software. This feature is of specific importance to this research as a full map of the indoor data collection environment was created.

Figure 99. (a) Left: Riegl LMS-Z360i; (b) Right: 3D point cloud with overlaid imagery in the basement of Ohio University's Stocker Center.
CHAPTER 7: RESULTS

To fully evaluate the GPS/inertial/optical mechanization, data sets were collected in a structured indoor and outdoor environment. These data set also included transitions from the outdoor environment to the indoor environment and back. The first data collect was performed by carrying the platform as to control its motion. The platforms starting location was behind Stocker Center in an open area with multiple GPS satellites visible. Subsequently, it was lifted off and maneuvered alongside the Stocker engineering building and eventually entered the basement through the rear basement doors. During this transition the number of satellites dropped from 5 to 0 in about 7 seconds as shown in Figure 100. Once inside, the availability of any GPS signals was scarce, and in places, fully denied. After travelling through corridors of the GPS challenged indoor environment, the platform transitioned back to an outdoor environment by exiting to the opposite end of building. The area to which the UAV exited is surrounded by buildings and has walkways overhead, images of which are shown in Figure 101. This created another GPS challenging environment due to multi-path and signal attenuation. While receiving sporadic GPS measurements, the platform traced an outdoor path between multiple buildings, and eventually, landed in a similar position to where it started its trajectory. The starting and ending locations for this walkthrough can be observed in Figure 102 using Google Earth™.
Figure 100. Availability of sensors during forced outage of the laser.

Figure 101. GPS challenged outdoor environment.
The attitude results of the described sensor integration procedure for this test are shown in Figure 103 along with the results of the standalone inertial. For further analysis, Figure 104 displays the differences between the two solutions shown in Figure 103. Through examination of these two figures it can be seen that the EKF and IMU provide highly similar results that typically contain sub-degree level differences. This indicates that only small corrections were made to the inertial roll and pitch states. This was to be expected since attitude performance is quite stable even for lower-cost sensors like the Xsens.
Figure 103. Pitch and roll results from integrated solution.

Figure 104. Differences between pitch and roll estimates from the integrated solution and the standalone IMU.
A comparison between the EKF and the standalone inertial heading results are displayed in Figure 105 and their differences are shown in Figure 106. As the two solutions develop over time, similar trends can be seen with a few isolated deviations contributing to most of the error. However, the EKF and IMU conclude the data collection with an 11.4° difference in heading. This points to larger corrections being made to the platforms heading as opposed to the other two attitude states.

Figure 105. Heading results from integrated solution.
In the indoor environment, the mapping function is engaged in addition to the navigation function. Hence, for the indoor portion of the trajectory, we made use of a "truth reference" map which was made available through the Riegl LMS-Z360i laser scanner. As no external Vicon system [133], high-end inertial or other truth reference system was available, the truth reference of the environment is compared to the flight path obtained from the EKF. It can be seen that the flight path constantly remains in hallways present in the truth data, indicating sub-meter level performance. Based on the previous results in Chapter 3 and 4, this performance can likely be almost completely attributed to the laser-based navigator, and based on the accuracy of the map, the navigation accuracy is likely to be at the same level as the mapping accuracy.

*Figure 106. Differences between heading estimates from the integrated solution and the standalone IMU.*
For further analysis of the flight trajectory produced by the EKF, Figure 108 shows an enhanced view of a small section of the multi-rotor’s path in 3D. During the data collection, the platform was periodically made to perform up and down maneuvers which are highly visible in the 3D plot. These maneuvers cause the horizontal laser range scanner to scan the hallway in front of the multi-rotor at various altitudes increasing the extent of the locally generated map. These height changes are clearly present in the z-axis of Figure 108.
Figure 108. Enhanced section of path from integrated solution.

Through the mechanization discussed in this work, a 3D map of the environment was produced. Comparison between the indoor section of the produced 3D map and the Riegl's truth reference data is shown in Figure 109. A strong correlation between the two maps can be observed. Upon examination, it has been determined that the produced 3D map matches the truth data to within dm-level precision. As previously stated in Section 5.4, the map building procedure for this work is derived based on the platform’s 6DOF state estimation, and therefore, evaluation of one inherently evaluates the other. Thus, the dm-level precision found through the comparison displayed in Figure 109 verifies the dm-level precision of the produced flight path shown in Figure 107.
For better visualization of the produced 3D map, multiple enhanced views of Figure 109 are shown in Figure 110. In these different views, the high quality of the produced map can clearly be observed as it correlates heavily with the truth reference scan. Also apparent in Figure 110 is the increased vertical extent of the horizontal laser scanner by having the multi-copter go up and down. This confirms that at a normal flight level (1-2 meters), the laser scanners observe enough of the environment to yield dense 3D maps.
As a 3D point cloud is generated for the full data collection, a 3D map of the trajectories outdoor section is available. Enhanced views of various outdoor segments of the 3D map are shown in Figure 111. In this figure, points from the forward and downward laser scanners were plotted in red while points from the angled laser scanner were plotted in blue. This is for visualization purposes only as varying colors aid in observing the structure of the point cloud.
During portions of the described data collection, there was enough visibility (＞3 satellites) to calculate a GPS position. The availability of GPS measurements to the position estimation portion of the filter allowed for geo-referencing of the produced flight path and 3D map. Figure 112 displays the geo-referenced flight path based on the integration filter as well as the locations of calculated standalone GPS positions based on pseudoranges only. For further visualization, the geo-referenced flight path is plotted using Google Earth™ on the left of Figure 113, while the standalone GPS solution is plotted on the right. The geo-referenced path correctly displays the platform passing through Stocker Center, the Ohio University engineering building.
Figure 112. Geo-referenced flight path vs. GPS derived positions.

Figure 113. (a) Left: EKF produced path; (b) Right: standalone GPS path
Along with a geo-referenced flight path, a fully geo-referenced 3D map was created. This can be observed in Figure 114. Each laser point in the subsequent plot has individual latitude, longitude and height coordinates.

![Figure 114. Geo-referenced 3D environmental map.](image)

Based on the previously discussed results, it is observed that when the platform is indoors, the solution produced by the EKF is dominated by laser measurements. Therefore, the contributions made by the monocular camera are nearly unobservable. To demonstrate the contributions of the monocular camera to the integration method, laser measurements
were removed from the solution for a 20 second period. This forced sensor outage can be seen in Figure 115.

![Figure 115. Availability of Sensors during forced outage of the laser.](image)

During the 20 second removal of laser data, the system is forced to operate on integration between visual odometry measurements and the IMU. The cumulative effect caused by this situation can be observed in Figure 116. After coasting on an IMU/camera solution for 20 seconds, the path is subsequently altered by 3 meters, as opposed to the solution with all sensors.
To further emphasize the contribution of the camera-based VO component, both the laser and camera were removed from the integration for the same 20 second period. The effect of this outage can be observed in Figure 117. During this time period ($t = 180$ through $t = 200$ seconds) the EKF is forced to coast on calibrated inertial measurements.

*Figure 116. Effect of losing GPS and lasers for 20 seconds.*
The effect of losing all secondary sensors for a 20 second period is shown in Figure 118. During the forced sensor outage, a 45 meter cumulative difference is introduced between the path using all sensors and the path with denied sensors. Through comparison of the results shown in Figure 116 and Figure 118. The contribution of monocular camera data can be isolated. When the EKF was forced to operate for 20 seconds using an IMU/camera solution, 3 meters of error were introduced. This is significantly smaller than the 45 meters of error observed when using only the inertial for the same period. Thus, the camera is shown to provide stability to the EKF when neither the laser nor GPS are available.
Figure 118. Effect of coasting on the IMU for 20 seconds.

For evaluation of the sensor integration scheme during flight conditions, a different data collection was performed. The UAV started on the ground in an open-sky environment. Upon takeoff, the platform maneuvered toward the engineering building and entered it through the basement doors. During this transition the number of satellites dropped from 7 to 0 in about 8 seconds as shown in Figure 119. Once inside, the platform traveled through multiple corridors to the opposite side of the building. During this time, the laser based SLAM solution became unavailable. This effect is shown in Figure 119. Retracing its path, the UAV flew the reverse of the previously described route leading it from inside to outside and landing in a similar place to where it began. Using Google Earth™, the starting and ending locations of this test are displayed in Figure 120.
Figure 119. Availability of Sensors during flight test.

Figure 120. Starting and ending locations of flight experiment shown on Google Earth™.
Attitude estimates produced during the flight test are shown in Figure 121 for both the integrated solution and the stand alone IMU. To better visualize the comparison of the two solutions displayed in Figure 121, the differences between the two solutions are shown in Figure 122. For most of the data collection, the EKF based attitude results are comparable to those of the Xsens IMU. For the first 540 seconds of the flight, the pitch estimate for the two solutions diverges by 2.3° and the roll diverges by 3.5°. However, the two solutions begin to divert after the platform lands. With the platform stationary in an outdoor environment, the vision-based attitude estimates remain stable and contain very little drift error. This can be shown to aid in correcting the inertial errors for the final 140 seconds of the experiment. As no laser-based attitude solution was available during this period due to a lack of features, this reduction in drift is attributed to the camera.

Figure 121. Pitch and roll results during flight test.
Figure 122. Differences between pitch and roll estimates from the integrated solution and the standalone IMU during flight test.

A comparison between the EKF and the standalone inertial heading results are shown in Figure 123 and their corresponding differences are displayed in Figure 124. With regards to their development over time, it can be seen that the EKF and inertial solutions slowly divert from one another. The rate of divergence between the two solutions begins to accelerate after about 250 seconds of operation. It is at this instance that the buildings structure is detected and the laser-based SLAM solution becomes viable. Furthermore, in a similar fashion to the previously displayed pitch and roll results, the heading solution from the EKF becomes more stable than the free running inertial when the platform lands. This reduction in drift error for the EKF solution is attributed to the stability of cameras attitude when the platform is static.
Figure 123. Heading results during flight test.

Figure 124. Differences between heading estimates from the integrated solution and the standalone IMU during flight test.

The path produced through the aforementioned flight is compared to the indoor truth reference map in Figure 125. During the initial leg of the flight, GPS or a SLAM
solution was available at all times. The results produced during this period are comparable, albeit slightly worse, than the results produced when the platform was carried. About a 1 meter error is accrued over the first 410 second leg. At the beginning of the return leg of the flight, the laser based SLAM solution became unavailable. Furthermore, GPS signals were not reacquired until 550 seconds into the flight. Therefore, nearly all motion derived during the second leg of the flight was produced through the IMU/camera integration. While relying on the IMU and camera, the EKF produced good results when operating in the indoor environment. Before transitioning to outdoors, 4 meters of accumulated error were accrued. Once outside, the scale estimation procedure used for visual odometry began to show difficulties (as previously observed in Figure 87) due to a lack of observable features. From the time between exiting the building and landing, the IMU/camera solution introduced a 24 meter error.

![Figure 125. Derived flight path compared to reference map.](image)
CHAPTER 8: CONCLUSIONS

This dissertation has introduced a novel position, navigation and mapping capability for a multi-copter platform through sensor integration. Present trends regarding applications of multi-copters in structured indoor and outdoor environments justify the demand for such a system. Taking into account the current state-of-the-art of PNM technologies, a mechanization was devised that combines data from a GPS receiver, an IMU, a monocular camera, and multiple laser scanners. Along with the development of a high-level GPS/Inertial/Optical filtering architecture, many other novel procedures were derived and implemented. To test the integration mechanization and many of its subsystems, a hexacopter platform was built and used to collect data in the target environment. This platform allowed for a variety of data collection test flights and environment walk-throughs, which have been used to evaluate the performance of proposed algorithms. From the results produced during these test flights, the following conclusions have been drawn with respect to specific categories:

GPS/IMU/Optical integration mechanization:

- In both structured indoor and outdoor environments, the discussed mechanization provides a high quality 6DOF platform state estimate as well as a dense 3D map of the environment. For most cases, a relative position accuracy on the order of dm-level was attained. The attitude performance was comparable to a low-cost IMU.

- Through integration of attitude estimates produced by the inertial, lasers, and camera, a highly robust attitude solution was obtained that mitigates drift and can deal with short term failures in a single sensor.
In cases where a global position can be obtained through GPS, both the platform trajectory and created map are anchored to a global frame with m-level accuracy.

Laser-based methods:

- The designed dual laser-based altitude estimator provides centimeter level height accuracy that is robust to platform attitude changes and mildly uneven terrain.
- Using only two planar surfaces, the novel dual laser-based attitude estimator can resolve all three platform attitude states and provided accuracy similar to the Xsens inertial.

Monocular camera methods:

- When a camera and laser scanner observe common features, the magnitude of a platform's motion can be resolved using the created scale estimation process. This allows visual odometry to operate at true scale with no environmental assumptions.
- The attitude produced through visual odometry can operate in environments without structure to allow for attitude corrections in a wider array of environments.
- In cases where neither GPS carrier phase nor a structured environment are available, the implemented visual odometry process yields m-level accuracy over limited amounts of time. This allows the system to coast between short term outages from the aforementioned sensors.

In conclusion, the integration and methods discussed in this dissertation were shown to produce a high quality 3D PNM solution for a multi-copter platform.
CHAPTER 9: FUTURE WORK

Upon examining the outcomes of this dissertation, many areas for improvement and future development can be identified. The areas that possess the largest room for future research and development are hardware modifications, a conversion to real time processing, the inclusion of visual-SLAM, and the migration to unstructured environments.

There are many possible hardware modifications that could be performed on the developed multi-copter to improve its performance. One such modification would be the addition of circuitry to synchronize sensors at the hardware level. Currently, each sensor collects data that is time tagged by the FitPC using the ROS time-tagging mechanism; a software clock that is polled by the various ROS device drivers. By performing synchronization through software instead of hardware, excess latency is injected into the system and additional computational resources are required.

The optical sensor payload in this research consisted of multiple laser range scanners and a single camera. Other sensors should be considered that resolve some of the issues raised in the current setup. For example, a stereo camera pair or a multi-view camera systems could be used to (a) resolve the depth uncertainty for which we were required to include a laser scanner in the vision solution, and (b) to increase the field-of-view so more features will potentially be visible and extracted features may be visible for a longer period of time. Other sensors such as an infrared camera would enable vision-aided operation in low-light conditions and advanced sensors such as 3D imagers and RGB-D sensors could potentially replace the low-range laser range scanners for altitude and attitude estimation and mapping. While many different types of sensors were used in this work, the focus of
this dissertation was not on finding the optimal combination of sensors. Therefore, much research can still be performed to choose the optimum sensor package.

A significant amount of the results presented in this dissertation were obtained through the post processing of data. As most applications including autonomous flight of a multi-copter require reliable real-time performance, post processing of data is unacceptable. On the current platform, only the bagging of data and basic analysis are performed on board the platform. To run all necessary processes discussed in this dissertation during flight, the current computational platform (i.e. the FitPC) must be re-evaluated and possibly upgraded. In addition to potentially increasing the platforms processing capabilities, algorithm optimization could take place as code was composed in this research for post-processing. As little concern was given to algorithmic computational consumption, massive strides could be made to make code more efficient and/or parallelized.

Along with code optimization and an evaluation of the platforms computational power, another necessary step to achieve real time results would be to convert away from Matlab®. Matlab® is an interpreted language as opposed to compiled languages such as C, C++, or Python. While Matlab® is a powerful data evaluation tool, it inherently performs much slower on certain processes than a compiled language. As the current platform is running ROS, a conversion of algorithms to C++ or python within ROS would be desirable. For example, the 3D SLAM algorithm is currently already run on a PC under ROS (in more or less real-time) as these occupancy grid-based methods have already been highly optimized.
Another area of future development is the addition of VSLAM into the integration solution. While the visual odometry procedure described in this dissertation offered good results and proved indispensable in certain situations, it can be observed that they are often not of the same motion estimation quality as other sensors. It is believed that the odometry estimates produced by the monocular camera could be significantly improved through the introduction of VSLAM, preferably based on a stereo camera pair. Based on the results presented by Davison [67][68] and others, VSLAM techniques would be a promising addition to the described mechanization.

The final area of future work that would be recommended is significantly more open ended then the aforementioned suggestions. The overarching goal of this field of research is to discover a method by which a precise PNM capability can be obtained anytime, in any environment, and can be used indefinitely. While this work focuses on providing PNM capabilities for a multi-copter operating in and between structured indoor and outdoor environments, it ignores an array of other scenarios. Research could (and should) be done to expand many of the discussed algorithms to operate in different environments such as unstructured environments or environments with natural structure (e.g. trees).

The research performed in this dissertation produced promising results, but a true solution to the overarching problem will require a vast amount of additional research. Hopefully, the work accomplished in this dissertation is one of many stepping stones that may eventually lead to a system that can provide PNM capabilities anytime, anywhere.
REFERENCES


APPENDIX A: DOUBLE DIFFERENCING DERIVATION

GPS double differencing derivation based on Farrell [9] and Uijt de Haag and van Graas [119].

\[
\nabla \Delta R_{jk} = h_{kj}(t_m)\Delta r + [h_{kj}(t_m) - h_{kj}(t_{m-1})] + e^T_j (t_m)R_{SV_j}(t_m) - e^T_j (t_{m-1})R_{SV_j}(t_{m-1}) - e^T_k (t_m)R_{SV_k}(t_m) + e^T_k (t_{m-1})R_{SV_k}(t_{m-1})
\]

\[
= h_{kj}(t_m)\Delta r + [h_{kj}(t_m) - h_{kj}(t_{m-1})] + e^T_j (t_m)R_{SV_j}(t_m) - e^T_j (t_{m-1})R_{SV_j}(t_{m-1}) - e^T_k (t_m)R_{SV_k}(t_m) + e^T_k (t_{m-1})R_{SV_k}(t_{m-1})
\]

\[
= h_{kj}(t_m)\Delta r + [h_{kj}(t_m) - h_{kj}(t_{m-1})] + e^T_j (t_m)R_{SV_j}(t_m) - e^T_j (t_{m-1})R_{SV_j}(t_{m-1}) - e^T_k (t_m)R_{SV_k}(t_m) + e^T_k (t_{m-1})R_{SV_k}(t_{m-1})
\]

\[
= h_{kj}(t_m)\Delta r + [h_{kj}(t_m) - h_{kj}(t_{m-1})] + \left[ e^T_j (t_m) - e^T_j (t_{m-1}) \right] R_{SV_j}(t_m)
\]

\[
= h_{kj}(t_m)\Delta r + [h_{kj}(t_m) - h_{kj}(t_{m-1})] + \left[ e^T_j (t_m) - e^T_j (t_{m-1}) \right] R_{SV_j}(t_m)
\]

\[
= h_{kj}(t_m)\Delta r + [h_{kj}(t_m) - h_{kj}(t_{m-1})] + \left[ e^T_j (t_m) - e^T_j (t_{m-1}) \right] R_{SV_j}(t_m)
\]

\[
= h_{kj}(t_m)\Delta r + [h_{kj}(t_m) - h_{kj}(t_{m-1})] + \left[ e^T_j (t_m) - e^T_j (t_{m-1}) \right] R_{SV_j}(t_m)
\]

\[
\nabla \Delta R_{jk} = h_{kj}(t_m)\Delta r + [h_{kj}(t_m) - h_{kj}(t_{m-1})] + \frac{\Delta e^T_j R_{SV_j}(t_m)}{\Delta R_{SV_j}} + e^T_j (t_m)\Delta R_{SV_j} + \frac{\Delta e^T_k R_{SV_k}(t_m)}{\Delta R_{SV_k}} + e^T_k (t_{m-1})\Delta R_{SV_k}
\]

\[
\nabla \Delta R_{jk} = h_{kj}(t_m)\Delta r + a_{jk} + b_{jk}
\]

(A.1)
APPENDIX B: TRIANGULATION METHOD

Multiple instances in this work require the projection of 2D features into a 3D environment. This is accomplished through applying Zisserman's [51] Direct Linear Transform (DLT) triangulation method to roughly estimate a feature's depth. To perform this process, multiple observations of the same feature are required. It is also necessary to have estimates for rotation and translation between camera frames. These values are typically derived for external sensors such as an inertial. Note, in cases where the estimated rotation and translation parameters contain error, this method only yields a rough approximation. Variables are defined as follows:

- $C_{t_{k-1}}$: estimated rotation between camera images (3x3)
- $\Delta r$: estimated translation between camera images (3x1)
- $p^c(t_{k-1})$: normalized 2D image frame coordinates at time $t_{k-1}$ (3x1)
- $p^c(t_k)$: normalized 2D image frame coordinates at time $t_k$ (3x1)
- $K$: camera calibration matrix (3x3)
- $p^n(t_{k-1})$: 3D projection of point $p^c(t_{k-1})$ (3x1)
- $p^n(t_k)$: 3D projection of point $p^c(t_k)$ (3x1)

To begin the process, the intrinsic parameters contained in the camera calibration matrix are applied to each of the 2D image points:

$$\hat{p}^c(t_{k-1}) = K^{-1}p^c(t_{k-1}) = \begin{bmatrix} \hat{p}_1^c(t_{k-1}) \\ \hat{p}_2^c(t_{k-1}) \\ \hat{p}_3^c(t_{k-1}) \end{bmatrix} \quad (B.1)$$

$$\hat{p}^c(t_k) = K^{-1}p^c(t_k) = \begin{bmatrix} \hat{p}_1^c(t_k) \\ \hat{p}_2^c(t_k) \\ \hat{p}_3^c(t_k) \end{bmatrix} \quad (B.2)$$
This allows for the formation of the $A$ matrix:

$$A_1 = [0 \ 0 \ 1 \ 0] \cdot \hat{p}^c_1(t_k) - [1 \ 0 \ 0 \ 0] \quad (B.3)$$

$$A_2 = [0 \ 0 \ 1 \ 0] \cdot \hat{p}^c_2(t_k) - [0 \ 1 \ 0 \ 0] \quad (B.4)$$

$$A_3 = M_3 \circ \hat{p}^c_1(t_k) - M_1 \quad (B.5)$$

$$A_4 = M_3 \circ \hat{p}^c_2(t_k) - M_2 \quad (B.6)$$

where $M$ is defined as:

$$M = \begin{bmatrix} c_{tk}^f & \Delta r \\ \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & \Delta r_1 \\ c_{21} & c_{22} & c_{23} & \Delta r_2 \\ c_{31} & c_{32} & c_{33} & \Delta r_3 \end{bmatrix} = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} \quad (B.7)$$

Subsequently, each row of the calculated $A$ matrix is normalized:

$$A = \begin{bmatrix} \frac{A_1}{|A_1|} \\ \frac{A_2}{|A_2|} \\ \frac{A_3}{|A_3|} \\ \frac{A_4}{|A_4|} \end{bmatrix} \quad (B.8)$$

Through application of a SVD on $A = U S V^T$, elements of the resultant $V$ matrix are defined as:

$$V = \begin{bmatrix} v_{11} & v_{12} & v_{13} & v_{14} \\ v_{21} & v_{22} & v_{23} & v_{24} \\ v_{31} & v_{32} & v_{33} & v_{34} \\ v_{41} & v_{42} & v_{43} & v_{44} \end{bmatrix} \quad (B.10)$$

which yields a 3D estimate for $p^c(t_{k-1})$:

$$p^n(t_{k-1}) = [v_{14} \ v_{24} \ v_{34}]^T \quad (B.11)$$

The estimated rotation can then be applied to estimate a 3D location for $p^c(t_k)$:

$$p^n(t_k) = C_{tk-1}^t p^n(t_{k-1}) \quad (B.12)$$