INTENSE NAVIGATION

USING ACTIVE SENSOR INTENSITY OBSERVATIONS TO IMPROVE LOCALIZATION AND MAPPING

by

ROBERT A. HEWITT

A thesis submitted to the
Department of Electrical and Computer Engineering
in conformity with the requirements for
the degree of Doctor of Philosophy

Queen’s University
Kingston, Ontario, Canada
March 2018

Copyright © Robert A. Hewitt, 2018
Abstract

Where am I? This question continues to be one of the fundamental challenges posed in robotics research. The ability of a robot to localize itself and map its environment has proven to be a difficult and rich research problem. While significant progress has been made it still remains a difficult task to perform in dynamic, 3D environments, over long distances. Stereo cameras are a proven workhorse for the task of Visual Odometry (VO) and three-dimensional Simultaneous Localization and Mapping (SLAM), but they require reliable lighting conditions and matching regions in both images. Light Detection And Ranging (LiDAR) sensors provide an alternative; they are lighting-invariant, provide dense depth information directly, and intensity information that resembles grayscale camera images. In many cases where lighting is unavailable or inconsistent, such as underground mining or planetary exploration, LiDAR is particularly suited for the task of localization.

Both VO and SLAM can use a type of nonlinear optimization called bundle adjustment (BA) to solve for the optimal sensor pose and landmark positions given a set of matched observations at two or more separate poses. This thesis develops a version of BA, called IntenseBA. The algorithm estimates a map of landmarks, augmenting the standard three-dimensional point landmark with surface normal and reflectivity states. Because LiDAR intensity observations are dependent on the sensor
pose and these landmark states, it is able to use observations of the landmarks to probabilistically determine the most likely estimate of the sensor pose and landmarks. The problem is shown to be observable in all states and an analysis of its sensitivity to noise in each observation is done through a simulation. A calibration procedure and analysis of modern keypoint algorithms is presented which allows the theoretical model to be applied to real data from a SwissRanger SR4000 Time-of-Flight (ToF) camera. Experiments were conducted in the European Space Agency’s Planetary Utilisation Testbed, which emulates a Martian terrain. These experiments tested the IntenseBA algorithm (used to perform VO) and show the algorithm can accurately map all state estimates and improve upon accuracy compared to traditional and state-of-the-art approaches by incorporating these additional observations.
For my mother, Candace Leslie Hewitt.
Acknowledgments

I owe the work and the experiences from my time as a PhD student to many individuals. First my supervisor, Joshua Marshall, for allowing me to follow my interests wherever they took me and unwaveringly providing advice, support, and his technical expertise. It’s impossible to count how many times I’ve been thankful to have him as my supervisor.

I’d also like to thank my supervisor at the European Space Agency, Gianfranco Visentin, for the ideas, advice, and opportunities he provided me throughout my year in the Netherlands. The experience of working in such a diverse group, in a different country, on space robotics, is about as much as I could ask for.

I’ve also shared the time, workload, and fun with my colleagues. In particular, I’d like to thank Evangelos Boukas, Marc Gallant, Marco Pagnamenta, Brian Lynch, Curtis Watson, Jakub Tomášek and Martin Azkarate for the whiteboard sessions, field trials, space discussions, Ultimate Frisbee, video games, and the occasional pint.

To my partner, Olivia Moir, thanks for helping me see the light at the end of the tunnel and always supporting me.

Finally, I’d like to thank my family; my parents for always pushing and believing in me, my brother for his opinions and friendship, and the rest of my family for their love and support.
Contents

Abstract i
Acknowledgments iv
Contents v
List of Tables viii
List of Figures ix
List of Symbols xii

Chapter 1: Introduction 1
  1.1 Organization of Thesis ........................................ 5

I Theory 7

Chapter 2: Visual Odometry 8
  2.1 Image Distortion Correction .................................. 12
  2.2 Keypoint Detection ............................................ 13
  2.3 Keypoint Tracking ............................................. 15
  2.4 Outlier Rejection .............................................. 17
  2.5 Nonlinear Optimization ....................................... 18
  2.6 Summary ....................................................... 20

Chapter 3: State Parameterization 21
  3.1 Rotations ...................................................... 22
    3.1.1 Rotation Matrices ........................................ 22
    3.1.2 Lie Groups, Lie Algebras and Manifolds ................ 24
    3.1.3 Perturbing Expressions that Use Rotation Matrices ..... 27
  3.2 Poses ......................................................... 29
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.1</td>
<td>Transformation Matrices</td>
<td>31</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Perturbing Expressions that Use Transformation Matrices</td>
<td>32</td>
</tr>
<tr>
<td>3.3</td>
<td>Surface Normals</td>
<td>33</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Perturbing Expressions With Surface Normals</td>
<td>37</td>
</tr>
<tr>
<td>3.4</td>
<td>Homogeneous Points and Reflectivity</td>
<td>39</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Homogeneous Points</td>
<td>39</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Reflectivity</td>
<td>41</td>
</tr>
</tbody>
</table>

**Chapter 4: Intense Bundle Adjustment**

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Related Work</td>
<td>43</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Surface Normals</td>
<td>44</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Intensity</td>
<td>45</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Surface Normals and Intensity</td>
<td>46</td>
</tr>
<tr>
<td>4.2</td>
<td>Physical Principles of LiDAR</td>
<td>47</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Range Measurements</td>
<td>48</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Intensity Measurements</td>
<td>54</td>
</tr>
<tr>
<td>4.3</td>
<td>Batch Nonlinear Least-Squares Optimization</td>
<td>56</td>
</tr>
<tr>
<td>4.4</td>
<td>Gauss-Newton Optimization</td>
<td>60</td>
</tr>
<tr>
<td>4.5</td>
<td>Intense Bundle Adjustment</td>
<td>63</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Problem Definition</td>
<td>64</td>
</tr>
<tr>
<td>4.5.2</td>
<td>IntenseBA Observation Model</td>
<td>66</td>
</tr>
<tr>
<td>4.5.3</td>
<td>Log-likelihood of IntenseBA Observations</td>
<td>72</td>
</tr>
<tr>
<td>4.5.4</td>
<td>Linearizing IntenseBA Observation Errors</td>
<td>74</td>
</tr>
<tr>
<td>4.6</td>
<td>Sparse Bundle Adjustment</td>
<td>77</td>
</tr>
</tbody>
</table>

**II Applications**

**Chapter 5: Observability and Sensitivity**

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Observability Analysis</td>
<td>84</td>
</tr>
<tr>
<td>5.2</td>
<td>Simulation and Sensitivity Analysis</td>
<td>94</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Simulation Setup</td>
<td>94</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Pose and Landmark Initialization</td>
<td>96</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Sensitivity Analysis</td>
<td>98</td>
</tr>
</tbody>
</table>

**Chapter 6: Image Processing**

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>ToF Camera Model Calibration</td>
<td>107</td>
</tr>
<tr>
<td>6.2</td>
<td>Keypoint Detection and Description in ToF Intensity Images</td>
<td>110</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Removal of Viewing Direction and Range Dependence</td>
<td>114</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Determining Recall, 1-Precision, and Repeatability</td>
<td>115</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Evaluation and Comparison of Keypoints on ToF Dataset</td>
<td>118</td>
</tr>
</tbody>
</table>
6.3 Outlier Detection ........................................ 120
   6.3.1 RANSAC ........................................... 123
6.4 Intensity Calibration .................................... 125
   6.4.1 Calibration Results ................................. 129

Chapter 7: Experiments ....................................... 133
   7.1 Experimental Setup ................................... 133
   7.2 Intense Visual Odometry ............................... 135
   7.3 Results and Discussion ................................. 139

Chapter 8: Conclusion ......................................... 149
   8.1 Contributions ........................................... 150
   8.2 Future Work ............................................. 151

Appendix A: Computing Surface Normal Measurements from Three-
              Dimensional Data .................................. 168
# List of Tables

5.1 Minimum landmarks to construct observability matrix . . . . . . . . 89  
5.2 Rank deficiency with minimum number of landmarks . . . . . . . . 89  
5.3 Noise magnitudes tested in simulation . . . . . . . . . . . . . . . . 98  

6.1 SR4000 Specifications . . . . . . . . . . . . . . . . . . . . . . . . . . 107  
6.2 Distortion parameters for SR4000 ToF camera . . . . . . . . . . . . 109  
6.3 Keypoint combinations tested . . . . . . . . . . . . . . . . . . . . . 112  

7.1 Experiment noise magnitudes . . . . . . . . . . . . . . . . . . . . . 136
# List of Figures

1.1 Examples of dark environments where LiDAR is advantageous . . . . 3

2.1 VO pipeline . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
2.2 Mesa SR4000 ToF camera images corrected for lens distortion . . . 14
2.3 KAZE keypoints detected in ToF intensity image . . . . . . . . . . . 14
2.4 Keypoint outlier removal . . . . . . . . . . . . . . . . . . . . . . . . . 16

3.1 Vector expressed in two frames . . . . . . . . . . . . . . . . . . . 23
3.2 Perturbation in SO(3) . . . . . . . . . . . . . . . . . . . . . . . . . . 28
3.3 A point expressed in a sensor frame and a global frame . . . . . . . 30
3.4 Surface normal generation examples . . . . . . . . . . . . . . . . . . 33
3.5 Surface normal in $S^2$ . . . . . . . . . . . . . . . . . . . . . . . . . 34

4.1 Scanning LiDAR examples . . . . . . . . . . . . . . . . . . . . . . . 49
4.2 Optical signal in a CW-LiDAR . . . . . . . . . . . . . . . . . . . . . 51
4.3 SwissRanger 4030 ToF camera . . . . . . . . . . . . . . . . . . . . . 53
4.4 Bundle Adjustment . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65
4.5 Pinhole camera model . . . . . . . . . . . . . . . . . . . . . . . . . . 68
4.6 Intensity observation model . . . . . . . . . . . . . . . . . . . . . . 70

5.2 Rank deficiency for surface normal BA . . . . . . . . . . . . . . . . 91
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3</td>
<td>Rank deficiency for intensity BA.</td>
<td>92</td>
</tr>
<tr>
<td>5.4</td>
<td>Rank deficiency for BA using all measurements</td>
<td>93</td>
</tr>
<tr>
<td>5.5</td>
<td>Simulation Parameters</td>
<td>95</td>
</tr>
<tr>
<td>5.6</td>
<td>Sensor position error for IntenseBA</td>
<td>100</td>
</tr>
<tr>
<td>5.7</td>
<td>Sensor orientation error for IntenseBA</td>
<td>101</td>
</tr>
<tr>
<td>5.8</td>
<td>Landmark position error for IntenseBA</td>
<td>102</td>
</tr>
<tr>
<td>5.9</td>
<td>Landmark surface normal error for IntenseBA</td>
<td>103</td>
</tr>
<tr>
<td>5.10</td>
<td>Landmark reflectivity error for IntenseBA</td>
<td>104</td>
</tr>
<tr>
<td>6.1</td>
<td>NIR LED placement on SR4000</td>
<td>110</td>
</tr>
<tr>
<td>6.2</td>
<td>Depiction of inaccurate range measurement due to field of view</td>
<td>111</td>
</tr>
<tr>
<td>6.3</td>
<td>Types of image processing done on ToF camera</td>
<td>113</td>
</tr>
<tr>
<td>6.4</td>
<td>Keypoint homography between images</td>
<td>117</td>
</tr>
<tr>
<td>6.5</td>
<td>Selected image sequences from Planetary Utilization Testbed dataset</td>
<td>119</td>
</tr>
<tr>
<td>6.6</td>
<td>Detector repeatability</td>
<td>121</td>
</tr>
<tr>
<td>6.7</td>
<td>Recall vs 1-precision graphs for keypoint detection, description and matching</td>
<td>122</td>
</tr>
<tr>
<td>6.8</td>
<td>Minimum number of RANSAC iterations for values of ( w, p, ) and ( n )</td>
<td>124</td>
</tr>
<tr>
<td>6.9</td>
<td>Underexposed histogram of intensity</td>
<td>126</td>
</tr>
<tr>
<td>6.10</td>
<td>Overexposed histogram of intensity</td>
<td>127</td>
</tr>
<tr>
<td>6.11</td>
<td>Calibration experimental setup</td>
<td>130</td>
</tr>
<tr>
<td>6.12</td>
<td>Filtered image segments at different exposures</td>
<td>131</td>
</tr>
<tr>
<td>6.13</td>
<td>Response curves</td>
<td>132</td>
</tr>
<tr>
<td>7.1</td>
<td>ExoTer rover in the Planetary Utilization Testbed</td>
<td>134</td>
</tr>
<tr>
<td>7.2</td>
<td>Perspective view of first trajectory.</td>
<td>140</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>7.3</td>
<td>Perspective view of second trajectory</td>
<td>141</td>
</tr>
<tr>
<td>7.4</td>
<td>Perspective view of third trajectory</td>
<td>142</td>
</tr>
<tr>
<td>7.5</td>
<td>Perspective view of fourth trajectory</td>
<td>143</td>
</tr>
<tr>
<td>7.6</td>
<td>Error in first trajectory</td>
<td>144</td>
</tr>
<tr>
<td>7.7</td>
<td>Error in second trajectory</td>
<td>145</td>
</tr>
<tr>
<td>7.8</td>
<td>Error in third trajectory</td>
<td>146</td>
</tr>
<tr>
<td>7.9</td>
<td>Error in fourth trajectory</td>
<td>147</td>
</tr>
<tr>
<td>A.1</td>
<td>Surface normal estimation methods</td>
<td>168</td>
</tr>
</tbody>
</table>
## List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0$</td>
<td>Zero matrix</td>
</tr>
<tr>
<td>$1$</td>
<td>Identity matrix</td>
</tr>
<tr>
<td>$a^T$</td>
<td>Transpose of $a$</td>
</tr>
<tr>
<td>$a^\wedge$</td>
<td>Cross product matrix of $a$</td>
</tr>
<tr>
<td>$a^+$</td>
<td>Compound operator acting on $a$</td>
</tr>
<tr>
<td>$a^-$</td>
<td>Inverse compound operator acting on $a$</td>
</tr>
<tr>
<td>$a^{-1}$</td>
<td>Inverse of $a$</td>
</tr>
<tr>
<td>$|a|$</td>
<td>Vector norm of $a$</td>
</tr>
<tr>
<td>$\bar{a}$</td>
<td>True value of $a$</td>
</tr>
<tr>
<td>$\hat{a}$</td>
<td>Estimate of $a$</td>
</tr>
<tr>
<td>$\check{a}$</td>
<td>Estimate of $a$</td>
</tr>
<tr>
<td>$\hat{a}^*$</td>
<td>Optimal estimate of $a$</td>
</tr>
<tr>
<td>$\Box$</td>
<td>Manifold subtraction operator</td>
</tr>
</tbody>
</table>
Manifold addition operator

Bundle Adjustment

Rotation matrix that takes points in frame 2 to frame 1

Dilation matrix that converts points to homogeneous points

ToF camera range image at timestep \( k \)

Global coordinate frame

Sensor coordinate frame at timestep \( k \)

Gaussian (or normal) distribution with mean \( a \) and covariance \( A \)

Global Navigation Satellite System

ToF camera intensity image at timestep \( k \)

Inertial Measurement Unit

Light Detection And Ranging

Identity surface normal

Vector from point \( a \) to point \( b \), expressed in reference frame \( F_c \)

Topological space of unconstrained design parameter, \( x \)

Euclidean space in \( n \) dimensions

Topological space of three-dimensional surface normal vectors

Topological space of three-dimensional transformation vectors
\( \mathfrak{so}(3) \) Topological space of three-dimensional rotation vectors

\( S \) Topological space of design parameter, \( x \)

\( S^2 \) Group that describes three-dimensional directions that exist on the unit 2-Sphere

\( SE(3) \) Special Euclidean group that describes three-dimensional transformations

SLAM Simultaneous Localization and Mapping

\( SO(3) \) Special orthogonal group that describes three-dimensional rotations

\( T_{12} \) Transformation matrix that takes points in frame 2 to frame 1

\( \mathcal{U}(a, b) \) Uniform distribution between values \( a \) and \( b \)

ToF Time-of-Flight

VO Visual Odometry

\( y_{jk} \) Observation of landmark \( j \) from pose \( k \)
Chapter 1

Introduction

Mars and the Moon remain the most important and feasible targets for space exploration in the near future. While NASA still claims the only successful rover missions to the Red Planet, the latest visitor to the lunar surface is the Chinese rover *Yutu*. At the time of writing, India is poised to land a rover on the Moon in the upcoming Chandrayaan-2 mission and the European Space Agency has plans to land the Exo-Mars rover on the surface of Mars in 2020. NASA continues to close in on a sample return mission from Mars, with its Mars 2020 rover aiming to cache samples on the surface for later return. After decades of slow progress, the nations of the world are starting to truly explore these alien worlds.

Why the sudden uptake? One might point to the huge technology developments of the last few decades, particularly in robotics. Many of these developments have their roots in space exploration, but have been continued and perfected in robotics here on Earth. Autonomous localization and mapping is one such technology. Communication delays to distant targets inherently make human tele-operation difficult or impossible. Light takes 1.3 seconds to travel from Earth to the Moon, and up to 21 minutes to Mars. NASA’s rovers communicate to Earth directly via X-Band
(maximum 32 kb/s data rate) or via orbiting artificial satellites like the Mars Reconnaissance Orbiter (MRO) that communicate to Earth over UHF (maximum 2 mb/s) which typically only pass overhead once per day. These kind of communication restrictions mean that any rover mission to Mars must be at least somewhat autonomous to achieve scientific objectives in the short time scales the machines are expected to operate in such harsh environments.

Key developments in autonomous three-dimensional localization and mapping using stereo cameras occurred in the 1980s, and it is no coincidence that in the next decade the first successful rover mission to Mars occurred with NASA’s Pathfinder mission and the Sojourner rover. The follow up missions, Mars Exploration Rovers (MER) in 2004 and the Mars Science Laboratory (MSL) in 2012, were built on this heritage and now exist in a world where autonomous localization and mapping is a dynamic area of growth and is poised to have a huge impact on the global economy with the advent of autonomous cars. One of the key ways robotics has improved over the last few decades is through probabilistically estimating the location of the robot and map, considering each sensor’s noise properties when determining how to incorporate its latest observation. One early example is the localization algorithm that went on to run on MER, developed as an extension to stereo camera localization that incorporated the geometry of the sensor and the resulting noise properties of its observations to improve its estimate of motion. This thesis aims to introduce its own extension to localization algorithms by incorporating more information from sensor observations than is typical.

Active sensors like Light Detection and Ranging (LiDAR) differ from passive sensors such as stereo cameras in that they do not rely on external lighting to observe the
Figure 1.1: Three examples of dark environments where LiDAR sensors are more suitable for localization and mapping: Shackelton Crater in the lunar south polar region is permanently shadowed (top), collapsed lava tubes on Mars and the Moon (bottom left), underground mining tunnels on Earth (bottom right)
environment. This makes them particularly useful in environments that have variable or no external lighting (e.g., permanently shadowed craters on the Moon, lava tubes on Mars, and underground mines on Earth shown in Figure 1.1). While stereo cameras are the de facto standard for vehicle localization (on Earth, the Moon, and Mars), there is continued interest in developing LiDAR technologies that are complementary or replace stereo vision sensors when appropriate. Localization algorithms that use LiDAR typically use the sensor’s range observations and accompanying direction observation (e.g., a laser’s elevation and azimuth directions, or an imaging sensor’s pixel location). However, most LiDAR sensors also report a signal strength observation, sometimes called *intensity*, that is proportional to the power of the returned signal. The intensity observation, like the range and direction observations, is dependent on properties of the landmark and the sensor pose. This thesis introduces an extension to localization and mapping algorithms that use LiDAR by augmenting the standard range and direction observations with observations of the surface normal and intensity. The goal here is to: (a) improve localization accuracy of the sensor pose and landmarks, (b) add new information to the map (i.e., surface reflectivity and shape), and (c) increase the speed of outlier detection which constitutes a significant portion of the total time spent performing localization. This thesis describes how this can be achieved with an algorithm named *IntenseBA*, and includes the results of experiments performed at the European Space Agency that show its feasibility.
1.1. ORGANIZATION OF THESIS

1.1 Organization of Thesis

This thesis is divided into two parts: theory (Part I), and applications (Part II). In Part I, a high level overview of the visual odometry (VO) pipeline is provided, including related work upon which this thesis is built (Chapter 2). Next, the parameterization of the states used to estimate the robot pose and landmarks in the environment is developed, covering three-dimensional rotations, transformations, positions, surface normal directions, and reflectivity (Chapter 3). These parameterizations are then used to develop the Intense Bundle Adjustment (IntenseBA) algorithm (Chapter 4), a major contribution of this thesis. This chapter starts with an overview of related work in the use of non-traditional observations (i.e., surface normals and intensity) to improve localization and mapping, and the physical principles of different types of LiDAR sensors. Finally, a formal description of the IntenseBA algorithm is given.

Part II begins by verifying the validity of IntenseBA (Chapter 5) with an observability analysis to show the problem is solvable for a minimal set of observations, and a simulation that demonstrates the sensitivity of IntenseBA compared to traditional forms of localization when varying the different types of measurement noise. Next, the image processing needed to take the idealized models described in Chapter 4 and apply them to real data from a SR4000 ToF camera (Chapter 6) is described. This includes distortion calibration, camera response calibration, an analysis of keypoint extraction and tracking, and a description of the form of outlier detection used, and analysis of its improved performance over traditional implementations. Finally, the processed data in Chapter 6 is fed to the IntenseBA algorithm to perform visual odometry (Chapter 7), named *IntenseVO*. These real-world tests demonstrate the
1.1. ORGANIZATION OF THESIS

feasibility of IntenseBA to improve the localization estimate and add new information to the map for a robot outfitted with a LiDAR sensor performing any type of localization and mapping that uses nonlinear least-squares optimization and is a major contribution of this thesis. Finally, a summary of the contributions of this thesis and a discussion of future work and challenges going forward is given (Chapter 8).
Part I

Theory
Chapter 2

Visual Odometry

This thesis seeks to introduce new measurements into Simultaneous Localization and Mapping (SLAM) algorithms that utilize active\textsuperscript{1} exteroceptive sensors that return a measurement of signal strength, sometimes referred to as an intensity measurement, along with three-dimensional range measurements. These sensors can be divided into two categories, scanning Light Detection And Ranging (LiDAR) and Time-of-Flight (ToF) cameras. These measurements are used to perform a fundamental, and ubiquitous form of SLAM in the field of mobile robotics: visual odometry (VO). VO is a form of dead-reckoning which estimates the transformation between reference frames associated with a sensor at different times in a traverse. By compounding successive transformations, an estimate of the robot pose at any time along its traverse can be attained. The format of this chapter is borrowed from [1], which describes a visual odometry pipeline for a passive stereo camera setup. Here we present in a similar fashion the pipeline for active sensors. This chapter reviews related work and describes each of the major computing blocks in a sparse VO pipeline for active sensors.

\textsuperscript{1}In the context of this thesis a passive sensor is one which relies on external lighting in the environment to measure the scene, whereas active sensors project light onto the environment and measure its reflection.
sensors that return both range and intensity measurements.

Before describing VO algorithms, it is important to note that VO differs from batch SLAM algorithms, such as pose-graph SLAM [2], which estimate the entire robot trajectory within a much larger optimization problem. It is also possible to introduce additional measurements from other sensors, such as an inertial measurement unit (IMU), to improve the solution. All types of SLAM can benefit from including intensity measurements in the optimization problem that they share, however visual odometry with no additional sensor measurements provides the best way to quantify the improvement. While more complicated forms of SLAM and additional sensors would likely improve estimation accuracy, it would be less obvious how intensity measurements contribute to improving the solution accuracy, and would not add to the core contribution of this thesis (improving sensor localization) beyond what is demonstrated with visual odometry.

The most common sensors used for visual odometry are passive cameras. A monocular camera (i.e., single camera) can be used to estimate motion but there is no way to recover the metric scale of this trajectory without additional constraints [3, 4]. In some examples, this is done by fixing both the first and second pose, with the transformation between them defined by another sensor such as wheel odometry or an IMU [5, 6]. More commonly, stereo cameras are used to simplify the motion estimation problem with a known transformation between the two cameras. This allows the 3D structure of the scene to be estimated from a single image pair. In addition to the three-dimensional position representation of each measurement, the surface shape can also be observed with the approximation that each measurement lies on a locally planar surface [7] to recover more complete scene geometry and have
an additional way to observe the rotation of the sensor between frames.

Algorithms for estimating motion from an image sequence can be split into two categories: dense and sparse. Dense algorithms estimate the complete scene’s 3D structure as well as the camera motion, whereas sparse methods extract a sparse set of interest points, or keypoints, from each image, and estimate the camera motion and landmark position. Sparse methods exhibit higher computational efficiency, owed to the reduced size of the optimization problem and feasibility of jointly estimating both pose and scene keypoints, however they do not retain as much information about the scene as dense methods and require texture in the scene in order to describe the keypoints. While sparse methods have dominated in the past due to their performance, dense and semi-dense methods have gained popularity in recent years [8], due to increased computational power available (including powerful graphical processing units (GPUs) that are particularly suitable for running parallel operations), algorithms that decouple estimation of motion and scene reconstruction (known as alternating dense methods), and new methods that minimize photometric error, in addition to depth error, under an assumption that image intensity will not change over small pose changes with consistent lighting. Hybrids and modifications to dense methods also exist, known as semi-dense methods; Engel’s semi-dense VO [9] discards pixels with low image gradient to reduce the number of points included in the optimization problem. Forster et al. [10] utilizes dense methods for initial alignment and sparse methods for pose and structure refinement.

Despite these improvements, sparse methods remain on equal footing with dense methods [8], and work continues in both domains. Sparse VO is used in this thesis
Figure 2.1: VO pipeline for active sensors. Raw images enter the pipeline where they are corrected for lens distortion. Next keypoints are detected and described, and stored so they can be compared to the next image. Keypoint matching is done between sequential images and outlier detection rejects bad keypoint matches. The associated measurements and the previous pose estimate are then passed to a nonlinear optimization algorithm that estimates the current pose.

primarily due to the computational power of computers used in the envisioned applications of this work (vehicles such as planetary rovers) which are typically orders of magnitude less powerful than the processors used in commercial PCs and mobile phones [11]. Many dense methods also rely on an assumption that image intensity will be invariant to small changes in camera pose (assuming consistent external lighting), however this is not the case with active sensors, where the scene is illuminated by the sensor itself. In this case the measurement error is a function of landmark properties such as surface shape, reflectivity, and distance. Traditional cameras are a much more mature technology than active sensors such as scanning LiDAR and ToF cameras. Accordingly, techniques for describing and tracking individual keypoints produced from passive camera images for sparse methods are well developed in the computer vision field. However, the same techniques developed for passive camera images have been applied successfully to active sensors. To do this, the intensity
2.1. IMAGE DISTORTION CORRECTION

measurements of active sensors are processed to form a 2D image of the scene similar to the one cameras produce. Correspondences between measurements from different viewpoints are then made just as they are with cameras. By applying these techniques to sensors that already produce 3D measurements from a single image, the image rectification and stereo matching step can be removed from the visual odometry pipeline for scanning LiDAR [12, 13] and ToF cameras [14–16]. The basic outline of the VO pipeline for active sensors is shown in Figure 2.1.

An early example of VO was demonstrated by Moravec in his PhD thesis [17], laying the groundwork for each of the computing blocks in the VO pipeline. This was further developed by Matthies [18, 19] with an application to stereo imagery and probabilistic modelling of the sensor error which became the VO implemented on NASA’s Mars Exploration Rovers, Spirit and Opportunity [20, 21]. The rest of this chapter outlines each of the computing blocks in the active sensor VO pipeline.

2.1 Image Distortion Correction

Both scanning LiDAR and ToF cameras exhibit distortion in their measurements. Scanning LiDAR sensors are modelled as a spherical camera outlined in Section 4.5.2, and are distorted by misalignment and manufacturing tolerances when the sensor is built which can be modelled as additive biases. A calibration method for scanning LiDAR is presented in [22] where additive biases are included in the observation model. A SLAM problem is constructed with known landmarks extracted from a checker-board calibration target using the LiDAR intensity measurements to estimate the pose of the LiDAR (with respect to the checker-board) and the bias values.
2.2. KEYPPOINT DETECTION

ToF cameras exhibit the same type of distortion as passive cameras. This distortion is due to the lenses that light passes through before being absorbed by the camera sensor. In order to use an idealized pinhole camera model for a ToF camera to project 3D keypoints into 2D pixel locations, and vice-versa, these distortions must be corrected. There are two forms of distortion, tangential and radial; tangential distortion occurs due to the imaging plane and the lens not being perfectly parallel, whereas radial distortion happens due to the curvature of the lens itself. Typically, as the focal length decreases, the distortion increases [3]. These distortions are constant values, and can be determined through calibration in a similar fashion to the biases in a scanning LiDAR using a checker-board target. Once determined, each new image can be corrected by using these calibrated values and landmarks can be properly projected on 2D image coordinates. The mathematical model of the idealized pinhole camera is given in Section 4.5.2 and details on the distortion calibration for a SwissRanger ToF camera are given in Section 6.1. An example of the SwissRanger ToF camera distortion is shown in Figure 2.2, where the radial distortion is quite apparent when comparing the center of the images to the outer edges.

2.2 Keypoint Detection

After correcting the incoming images, a keypoint detection algorithm is used to find regions of interest in the image and a keypoint description algorithm extracts a keypoint vector that describes the region. While some algorithms encompasses both parts of this process, it is possible to apply one algorithm for detection and another for description. An evaluation of the performance of some common detection and description algorithms is presented in Section 6.2. The results—a contribution of this
2.2. KEYPOINT DETECTION

Figure 2.2: ToF camera images are corrected for lens distortion so that they correspond to images captured by an idealized pinhole camera. Horizontal lines across the image show the significant radial distortion in the raw image.

Figure 2.3: KAZE keypoints detected in a SR4000 ToF camera intensity image. The radius of each circle represents the scale of the keypoint region.
2.3. KEYPOINT TRACKING

thesis—show that out of the algorithms tested, KAZE keypoints [23] (used for both detection and description) tends to perform the best on various types of imagery encountered by ToF cameras. Therefore, KAZE was chosen as the keypoint detection and description method for the research presented in this thesis.

The KAZE algorithm describes keypoints using a 64-dimensional vector, $d \in \mathbb{R}^{64}$, that has a norm of 1 (i.e., a unit vector). The similarity of two keypoints $d_1$ and $d_2$ is calculated as $s_{12} = d_1^T d_2$, with a higher score meaning the two keypoints are likely from the same landmark.

Each keypoint pixel location is modelled as a true measurement, $\bar{p}$, and zero mean, Gaussian noise, $\delta p$, with covariance $R_p$:

$$p = \bar{p} + \delta p, \quad \delta p \sim \mathcal{N}(0, R_p). \quad (2.1)$$

The choice of $R_p$ is left ambiguous in most of the literature, however for single scale keypoints it may be appropriate to use $R_p = 1$. In the case of multi-scale keypoints, such as KAZE, Furgale [1] assigned $R_p = \sigma^2 1$ where $\sigma = 2^\nu$ pixels and $\nu$ is the octave the keypoint is detected in. In the case of the 0th octave this is equivalent to a single scale keypoint, where the scale is the entire image. In practice, we also found better results using $R_p = \sigma^2 1$, and adopted it for this work. An example of KAZE keypoints detected in a time-of-flight image is shown in Figure 2.3.

2.3 Keypoint Tracking

Once keypoints have been detected and described, they are tracked in consecutive frames by searching for keypoint correspondences. To avoid searching the entire image for each keypoint, the measurements associated with each keypoint can be used to
2.3. KEYPOINT TRACKING

Figure 2.4: Keypoint pixel coordinates are plotted and connected to their corresponding matched keypoints in an image taken at a later timestep. After outlier removal, all matches correspond to a single motion.

produce a map of point landmarks in a defined frame (e.g., the first camera frame). Then, using an initial motion estimate, the predicted location of these landmarks can be determined in the new camera image, and each keypoint can be searched for in a window around this location. The initial motion estimate can be obtained through inertial or other odometric sensors, a constant velocity assumption or an assumption of no motion between frames.

These keypoint tracks still tend to contain outliers as they are based only on image appearance and proximity. Moving objects within the scene are a major cause for tracks that are not consistent with the robot motion. Even in a rigid scene with no moving objects there are still mismatches due to similar descriptors in similar regions of consecutive images but originate from different landmarks. An example of these feature tracks is shown in Figure 2.4a. To correct for these outliers, typically an additional outlier rejection step is included after keypoint tracking in the visual
2.4. OUTLIER REJECTION

As shown in Figure 2.4a, outliers are inevitably present in keypoint matches. These outliers must be removed before the associated landmarks are passed on to the non-linear optimization processing block. Outlier rejection ends up being a task of determining a model that will explain most of the data, and then removing the data that does not fit with this model.

This problem can be solved using RAndom Sample And Concensus (RANSAC) [24]. RANSAC repeatedly samples the minimum data points needed to generate a model hypothesis, and then tests this hypothesis on the rest of the data. Data-points are labeled as inliers if they fit the hypothesis with an error less than a predetermined threshold. Once all data is checked, a new hypothesis is generated from all of the potential inliers, and the residual error serves as a way to evaluate the hypothesis. This repeats for a number of iterations, and the inliers from the hypothesis with the smallest residual error are selected to continue on to the nonlinear optimization processing block. An example of RANSAC removing a set of outliers is shown in Figure 2.4b.

RANSAC can only guarantee that it will correctly reject all outliers with a certain probability, that increases as more hypothesis are tested. How many iterations of hypothesis generation and testing are needed? One way to approach this, discussed by Fischler and Bolles [24], is to assume each measurement is selected independently, and has a probability \( w \) of being an inlier. We can then say that the probability, \( p \),
of success given $k$ iterations is:

$$1 - p = (1 - w^n)^k$$  \hspace{1cm} (2.2)

where $n$ is the minimum number of data-points needed to compute a hypothesis. Solving for $k$ we obtain

$$k = \frac{\ln(1 - p)}{\ln(1 - w^n)}.$$  \hspace{1cm} (2.3)

## 2.5 Nonlinear Optimization

Once outliers have been rejected, the inlier data-points are sent to the nonlinear optimization block. This block refines the estimate of all state variables in order to best explain the camera measurements. In traditional VO, this includes the change in position and orientation of the sensor from the first image to the second image, as well as the three-dimensional position of all landmarks tracked across images and that remain after outlier rejection. In this thesis intensity measurements are included in the optimization problem, and to this end an additional set of state variables, surface normal and reflectivity are appended to each landmark.

The observation model for the ToF camera used in this thesis is nonlinear, and therefore solving for these state variables is done by using nonlinear least-squares estimation. This section provides an overview of related work. The solution used for this thesis is detailed in Chapter 4.

Moravec’s [17] motion estimation was an early instance of a mobile robot estimating its pose and three-dimensional landmark positions. In this case two point clouds were aligned using scalar-weighted nonlinear least squares optimization. Matthies and Shafer [18] introduced a probabilistic approach to this problem by transforming
the uncertainties in image space to the three-dimensional space of the triangulated points through linearized covariance propagation. These transformed covariance matrices could then be used in a matrix-weighted nonlinear least squares optimization for point cloud alignment. More recently, VO has been improved through the introduction of Sparse Bundle Adjustment (SBA) [25, 26] as the method of pose and landmark estimation. Bundle Adjustment changes the minimization problem from one of point cloud alignment from already triangulated points, to one that remains in the original space of the measurements of the camera by minimizing the error between predicted measurements (generated from estimated state variables) and actual measurements. SBA is an extension to bundle adjustment that takes advantage of a typical scenario; a camera moving through an environment where the number of landmarks observed and tracked by the sensor through consecutive frames outweighs the number of camera frames, forming a sparsity pattern in the equations that update the state parameters being estimated that can be taken advantage to significantly speed up computing a solution. SBA is described in more detail in Section 4.6.

Strasdat et al. [27] detailed the clear advantage of SBA pose-graph over filtering methods, with the key insight that to increase the accuracy of visual SLAM it is usually more advantageous to increase the number of features over increasing the number of frames. This reasoning helps to explain the shift in SLAM research from methods making use of filtering techniques (i.e., Information-, Kalman-, and Particle-Filters) [7, 28, 29], to batch methods that solve the problem across part, or all of, the sensor’s trajectory [30, 31]. In this way, VO and SLAM have converged on a similar solution, with VO being a limited case of SLAM over a window (as small as just two
2.6. SUMMARY

frames) of the sensor’s trajectory.

In this thesis, the optimization problem has been applied to a VO problem, but it would be possible to extend to a full SLAM implementation over the entire trajectory, the remaining work amounting to a bookkeeping problem of storing the information about correspondences between frames.

2.6 Summary

This chapter provides a relatively high level overview of the VO pipeline. It is important to have a firm grasp on the steps of this framework as all of the work presented hereafter is built upon it. The contributions of this thesis do not change the underlying structure of the pipeline. Instead, they propose changes to some of the aforementioned processing blocks which are needed when introducing intensity measurements from active sensors (such as ToF cameras). An analysis of several keypoint detection and description algorithms for intensity images is shown in detail in Section 6.2. In Section 5.1, the observability of the problem is discussed, with the key result and contribution that less landmarks are needed to compute a valid transformation than in traditional VO. This contribution has implications when performing RANSAC, and leads to an analysis of RANSAC performance when intensity measurements are included in Section 6.3. Finally, the main contribution of this thesis is to enhance the nonlinear optimization block of the VO pipeline, by including additional measurements and landmark state variables in the problem and is discussed in Section 4.5 with results from real-world experiments shown in Section 7.2.
Chapter 3

State Parameterization

This chapter reviews the parameterizations and associated operations that are used to represent the 3D pose of the sensor as well as the parameterization of landmarks in its map for use in navigation algorithms, like the visual odometry pipeline discussed in Chapter 2. These algorithms are most commonly based on a form of nonlinear least squares optimization, a technique that minimizes a cost function by iteratively adjusting the state estimates until the cost function is sufficiently small. These techniques require the nonlinear error terms in the cost function to be linearized. This chapter discusses each of the state parameterization classes that are used, and derives the operations needed to linearize them. The notation used follows that of [32] with additions to accommodate the extension to landmarks with added parameters of surface normal directions and reflectivity. The following state parameter classes are discussed:

1. rotation matrices: $3 \times 3$ matrices that represent elements of $SO(3)$;
2. transformation matrices: $4 \times 4$ matrices that represent elements of $SE(3)$;
3. unit axes: unit length $3 \times 1$ columns that represent elements of $S^2$;
4. homogenous points and reflectivity: Euclidean values that are elements of $\mathbb{R}^3$ and $\mathbb{R}$ respectively.

3.1 Rotations

Three-dimensional rotations are used in this thesis to represent the relative orientation of a vehicle or sensor frame (with respect to a global frame) with three degrees of freedom. However, all three-parameter representations of rotations contain singularities. To avoid these, over-parameterized representations with constraints are used. This section introduces rotation matrices which over-parameterize three-dimensional rotations with nine parameters and six constraints. Rotation matrices are represented with the variable $C$. The constraints for a rotation matrix can be written as a single matrix orthogonality constraint: $CC^T = 1$ where $(\cdot)^T$ represents the transpose of a matrix. The set of orthogonal matrices with determinant 1 are known as the special orthogonal group $SO(n)$, and the group that describes three-dimensional rotations is denoted $SO(3)$.

This section reviews the $SO(3)$ group, its operations, and how they are used here to represent rotations as random variables that can be perturbed from a nominal value. The explanation here is derived from [1] and [32].

3.1.1 Rotation Matrices

Consider two frames, $\mathcal{F}_A$ and $\mathcal{F}_B$. A vector $\mathbf{r}$ can be expressed in either frame (denoted $\mathbf{r}_A$ and $\mathbf{r}_B$ respectively). The rotation matrix $C_{AB}$ relates the vector $\mathbf{r}_B$ in $\mathcal{F}_B$ to $\mathbf{r}_A$ in $\mathcal{F}_A$ through

$$\mathbf{r}_A = C^{-1}_{BA} \mathbf{r}_B = C_{AB} \mathbf{r}_B,$$  

(3.1)
3.1. ROTATIONS

Figure 3.1: There are two frames, $\mathcal{F}_A$ and $\mathcal{F}_B$, which vector $r$ can be expressed in. The rotation matrix $C_{BA}$ can be used to rotate $r$ expressed in $\mathcal{F}_A$ ($r_A$), to $r$ expressed in $\mathcal{F}_B$ ($r_B$).

where in this thesis bold lower case letters are used to represent vectors and subscripts denote the reference frame a vector is expressed in.

In addition, rotation matrices inverses are their transpose:

$$C_{AB} = C_{BA}^{-1} = C_{BA}^T.$$  \hfill (3.2)

Using Euler’s theorem, we can write a rotation matrix, $C$ as a rotation about a unit-length axis, $a$ through an angle $\phi$:

$$C(a, \phi) = \cos \phi 1 + (1 - \cos \phi)aa^T - \sin \phi a^\wedge,$$ \hfill (3.3)

where $1$ is the identity matrix and we have adopted the notation for a matrix cross product operation used in [32], that is:

$$\phi^\wedge = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} \in \mathbb{R}^{3 \times 3}, \ \phi \in \mathbb{R}^3.$$ \hfill (3.4)
3.1. ROTATIONS

It is also possible to represent the rotation between two frames as a sequence of three principal rotation (an Euler sequence). There are many possible Euler sequences that can be used. In aerospace applications, the ‘roll-pitch-yaw’ sequence is typically used, also known as the 1-2-3 attitude sequence, which is as follows:

1. A rotation $\theta_1$ about the original 1-axis (i.e., ‘roll’ axis);
2. A rotation $\theta_2$ about the intermediate 2-axis (i.e., ‘pitch’ axis);
3. A rotation $\theta_3$ about the final 3-axis (i.e., ‘yaw’ axis).

Expressed as a rotation matrix from $\mathcal{F}_A$ to $\mathcal{F}_B$ this is given by:

$$
C_{BA}(\theta_3, \theta_2, \theta_1) = C_3(\theta_3)C_2(\theta_2)C_1(\theta_1),
$$

$$
= \begin{bmatrix}
    c_2c_3 & c_1s_3 + s_1s_2c_3 & s_1s_3 - c_1s_2c_3 \\
    -c_2s_3 & c_1c_3 - s_1s_2s_3 & s_1c_3 + c_1s_2s_3 \\
    s_2 & -s_1c_2 & c_1c_2
\end{bmatrix},
$$

where $s_i = \sin \theta_i, c_i = \cos \theta_i$. In all cases, Euler sequences have singularities. In this example, a singularity occurs when $\theta_2 = \frac{\pi}{2}$, where $\theta_1$ and $\theta_3$ are associated with the same degree of freedom and cannot be uniquely determined. However, this is only a problem if we want to extract the rotation angles from the rotation matrix to parameterize the rotation. The rotation matrix itself is free of singularities.

3.1.2 Lie Groups, Lie Algebras and Manifolds

As discussed in the previous section, rotations are represented by the $SO(3)$ group. One important consideration is the $SO(3)$ group is not a vector space, but is a non-commutative group. For instance, adding two rotation matrices does not result in a
3.1. ROTATIONS

new rotation matrix (i.e., it is not closed under addition). A zero matrix is also not a valid rotation matrix. Without these properties, $SO(3)$ is not a vector space. This is important because state estimation algorithms are essentially nonlinear least square problems, and estimates are continually updated through computed perturbations that must lie in a vector space.

It would seem that one is forced between choosing an over parameterization that is free of singularities (e.g., rotation matrices or unit quaternions) or one that can be operated on in a vector space (e.g., Euler angles). Most commonly, global representations are used, with an additional step that enforces constraints after a perturbation in vector space is applied. One example is a normalization step for a quaternion that has been perturbed and is now not a unit quaternion. Another is to treat associated constraints as an additional pseudo-measurement with very low measurement noise. In all cases, it adds complexity and potential inaccuracies to the update steps in state estimation algorithms.

Instead, it is possible to use both types of parameterizations and still avoid singularities while also enforcing the constraints of the global representation. It turns out the key to this problem is that the $SO(3)$ group is something called a matrix Lie group, and that there exists homeomorphisms (i.e., a continuous, invertable mapping between topological spaces) between $SO(3)$ and the vector space $\mathbb{R}^3$ that allow us to locally transform the parameterization between the two spaces.

As mentioned, the $SO(3)$ group is a matrix Lie group. A group is a set and an operation that combines any two of the set elements to form another element of the set, that also satisfies four conditions; closure, associativity, identity, and invertibility. A proof is not given here (this is well documented in other sources [32]), but the $SO(3)$
3.1. ROTATIONS

group satisfies all of these conditions. A Lie group is simply a group that is also a differential manifold that is smooth (i.e., if we change the inputs to a group operation a small amount, the output will only change a small amount). Finally, a matrix Lie group specifies that the elements of the group are all matrices, the combination operation is matrix multiplication, and the inversion operation is matrix inversion.

Along with a Lie group, there is an associated Lie algebra. Lie algebras consist of a vector space that is tangent to the associated Lie group at the identity element of the group, and contains the local structure of the group. In the context of rotations, the matrix Lie group describes the global representation of rotations in $SO(3)$ while the Lie algebra describes a local parameterization of rotations in a vector space. The strategy here is to hold our representation of any given rotation in $SO(3)$ but to update our representations when needed in the tangent vector space that we denote $\mathfrak{so}(3)$, defined as:

$$\mathfrak{so}(3) = \{ \Phi = \phi \wedge \in \mathbb{R}^{3 \times 3} | \phi \in \mathbb{R}^{3} \}.$$  \hfill (3.7)

This is simply a definition of the skew-symmetric matrix operator, however this notation is used to easily denote the reverse operation; $(\cdot)^\vee$. A depiction of the relation between $SO(3)$ and $\mathfrak{so}(3)$ is shown in Figure 3.2. What’s needed is a way to transform an element of $\mathfrak{so}(3)$ to $SO(3)$ and vice versa. For a matrix Lie group, we use the matrix exponential to relate the matrix Lie group to its Lie algebra:

$$\exp(A) = 1 + A + \frac{1}{2!}A^2 + \frac{1}{3!}A^3 + \cdots = \sum_{n=0}^{\infty} \frac{1}{n!}A^n,$$  \hfill (3.8)

where $A \in \mathbb{R}^{3 \times 3}$ is any square matrix. Going the other direction, there is also the
3.1. ROTATIONS

matrix logarithm:

\[
\ln(A) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n}(A - 1)^n. 
\]  
(3.9)

In the case of rotations, we can relate elements of \( SO(3) \) to elements of \( \mathfrak{so}(3) \):

\[
C = \exp(\phi^\wedge) = \sum_{n=0}^{\infty} \frac{1}{n!}(\phi^\wedge)^n, 
\]  
(3.10)

The inverse is achieved through the matrix logarithm:

\[
\phi = \ln(C)^\vee, 
\]  
(3.11)

noting that while this does not provide a unique solution for \( \phi \), there is only one solution that has \( |\phi| < \pi \).

3.1.3 Perturbing Expressions that Use Rotation Matrices

If we have a function \( f(x) \), then we can perturb \( x \) from its nominal value, \( \bar{x} \), by \( \delta x \), to change the function’s value. This can be expressed as a Taylor series expansion of \( f \) about \( \bar{x} \):

\[
f(\bar{x} + \delta x) = f(\bar{x}) + \left. \frac{\partial f(x)}{\partial x} \right|_{\bar{x}} \delta x + \text{h.o.t.} 
\]  
(3.12)

However, this expression assumes that \( \delta x \) is not constrained, which for most representations of rotations is not true. The exception are three parameter representations, like the Lie algebra rotation vectors. If we consider a rotation matrix, \( C \), rotating a vector, \( v \), as a function of the Lie algebra vector perturbation and a nominal rotation, \( \bar{C} \), we have:

\[
\bar{C} \boxplus \phi := \exp(\phi^\wedge)\bar{C} = C. 
\]  
(3.13)
3.1. ROTATIONS

Figure 3.2: A perturbation of $\bar{C} \in SO(3)$ by $\phi \in \mathfrak{so}(3)$ results in $C$. This is accomplished through the exponential mapping of the Lie algebra $\mathfrak{so}(3)$ to the Lie group $SO(3)$.

where the $\oslash$ ("box plus") operator is used to denote the operation that maps $\phi$ to $SO(3)$ and applies it to $\bar{C}$ as depicted in Figure 3.2. Similarly, we can define a $\ominus$ ("box minus") operator that computes the difference between two rotations, $C_1$ and $C_2$, and maps this to $\mathfrak{so}(3)$; i.e.,

$$C_2 \ominus C_1 := \ln (C_2^T C_1)^\vee = \phi.$$  \hspace{1cm} (3.14)

There are two obvious uses for perturbations like this in the case of state estimation; the first is in nonlinear least-squares optimization, where we compute a small correction to our current estimate in an iterative way based on a computed measurement error and the second is to represent random rotations. Parameterizing a three
3.2. POSES

dimensional rotation as a random variable with a rotation matrix can be difficult because the rotation has three degrees ($SO(3)$) of freedom but is represented with a $3 \times 3$ matrix with the constraint that the determinant is one. However, we can represent the three-dimensional rotation by its mean $\hat{C} \in SO(3)$ and covariance matrix $P_{\delta\phi} \in \mathbb{R}^{3 \times 3}$ such that

$$\hat{C} = \bar{C} \boxplus \delta\phi, \quad \delta\phi \sim \mathcal{N}(0, P_{\delta\phi}),$$

(3.15)

where the error $\delta\phi \in \mathfrak{so}(3)$ is a perturbation of the true rotation $\bar{C} \in SO(3)$. Note that random rotations cannot be exactly represented by a Gaussian distribution because the space errors exist in $(\mathfrak{so}(3))$ is bounded, representing angles between $[-\pi, \pi]$ along each degree of freedom. However, as discussed by Gallant on p. 39 in [33], this is a sufficient approximation for our application where in one degree of freedom $3\sigma_{\delta\phi} < \pi$ will contain the vast majority of the probability within $\mathfrak{so}(3)$.

3.2 Poses

So far we have discussed rotations and the parameterization of rotations using rotation matrices. Together with translation, these quantities can be used to describe the pose of a moving body. Estimating the pose of a moving body is useful if we are interested in transforming a point in a moving frame (e.g., camera sensor frame) to a stationary frame (often called the *global* frame).

For example, let the frame attached to a moving camera sensor be denoted $\mathcal{F}_s$ and the stationary global frame $\mathcal{F}_g$ as shown in Figure 3.3. Let a point $P$ be an arbitrary point in $\mathbb{R}^3$, and the origin of $\mathcal{F}_s$ be another arbitrary point in $\mathbb{R}^3$. We can relate the
3.2. POSES

Figure 3.3: A point $P$ is observed by a camera sensor which has a frame attached to it called $\mathcal{F}_s$. A global frame that is stationary is denoted $\mathcal{F}_g$. Vector superscripts represent the base and end of the vectors from right to left. Here we have vectors between each of the frames and the point $P$.

two points in the stationary frame as:

$$\mathbf{r}^{pg}_g = \mathbf{r}^{ps}_g + \mathbf{r}^{sg}_g,$$  \hspace{1cm} (3.16) 

where a vector superscript denotes the base and the end of the vector from right to left. If we know the position of point $P$ in $\mathcal{F}_s$ (e.g., a measurement from a 3D camera), we can determine the coordinates in $\mathcal{F}_g$ through the relationship:

$$\mathbf{r}^{pg}_g = \mathbf{C}_{gs} \mathbf{r}^{ps}_s + \mathbf{r}^{sg}_g,$$  \hspace{1cm} (3.17) 

where together the rotation between the two frames, $\mathbf{C}_{gs}$, and the translation, $\mathbf{r}^{sg}_g$, form the pose of the camera.
3.2. POSES

3.2.1 Transformation Matrices

The relationship in (3.17) can be expressed in the form of a $4 \times 4$ transformation matrix, $T_{gs}$:

$$
\begin{bmatrix}
\mathbf{r}_g^p \\
1
\end{bmatrix} =
\begin{bmatrix}
\mathbf{C}_{gs} & \mathbf{r}_g^s \\
\mathbf{0}^T & 1
\end{bmatrix}
\begin{bmatrix}
\mathbf{r}_s^p \\
1
\end{bmatrix}.
$$

(3.18)

Note that to use a transformation matrix the point coordinates must be appended with a 1, known as a homogeneous point representation [3].

Just as was shown with a rotation matrix, we can transform the points back the other way using the inverse of the transformation matrix:

$$
\begin{bmatrix}
\mathbf{r}_s^p \\
1
\end{bmatrix} = T_{gs}^{-1}
\begin{bmatrix}
\mathbf{r}_g^p \\
1
\end{bmatrix},
$$

(3.19)

where

$$
T_{gs}^{-1} =
\begin{bmatrix}
\mathbf{C}_{gs} & \mathbf{r}_g^s \\
\mathbf{0}^T & 1
\end{bmatrix}^{-1} =
\begin{bmatrix}
\mathbf{C}_{gs}^T & -\mathbf{C}_{gs}^T \mathbf{r}_g^s \\
\mathbf{0}^T & 1
\end{bmatrix} =
\begin{bmatrix}
\mathbf{C}_{gs} & -\mathbf{r}_s^s \\
\mathbf{0}^T & 1
\end{bmatrix} =
\begin{bmatrix}
\mathbf{C}_{gs} & \mathbf{r}_s^g \\
\mathbf{0}^T & 1
\end{bmatrix} = T_{sg}.
$$

(3.20)

Again, as with rotation matrices, it is possible to compound transformations. This is especially useful in visual odometry where we are often estimating successive transformations from one frame to the next, and arriving at a final pose estimate is accomplished through compounding each transformation.

Transformation matrices are also a matrix Lie group, called the Special Euclidean Group, denoted $SE(3)$. It also has an associated Lie algebra, $\mathfrak{se}(3)$, that allows us to
perturb matrices in a vector space tangent to the Lie group, defined as:

$$\mathfrak{se}(3) = \{ \Xi = \xi^\wedge \in \mathbb{R}^{4 \times 4} | \xi \in \mathbb{R}^6 \},$$

(3.21)

where we have adopted the overloaded notation used in [32], that is:

$$\xi^\wedge = \begin{bmatrix} \nu \\ \phi \end{bmatrix}^\wedge = \begin{bmatrix} \phi^\wedge & \nu \\ 0^T & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4}, \ \nu, \phi \in \mathbb{R}^3,$$

(3.22)

where $\nu$ is the translation component and $\phi$ is the rotation component of the perturbation.

### 3.2.2 Perturbing Expressions that Use Transformation Matrices

The same concepts applied in Section 3.1.3 can be used to perturb transformations. Elements of $\mathfrak{se}(3)$ can be transformed to $SO(3)$ through an exponential mapping:

$$T = \exp(\xi^\wedge) = \sum_{n=0}^{\infty} \frac{1}{n!} (\xi^\wedge)^n,$$

(3.23)

where $T \in SE(3), \ \xi \in \mathbb{R}^6, \xi^\wedge \in \mathfrak{se}(3)$. In the other direction (and like rotations, not uniquely):

$$\xi = \ln(T)^\vee.$$

(3.24)

Through the same logic as rotations, we can arrive at perturbation operations for transformation matrices; i.e.,

$$\bar{T} \oplus \xi := \exp(\xi^\wedge)\bar{T} = T,$$

(3.25)
3.3. SURFACE Normals

Directions are a common way to describe everyday phenomena. Examples include the directions measured by a compass, the direction of gravitational acceleration, the direction of the sun relative to a position on Earth, or simply the heading of a vehicle. In the case of the state estimation problem at hand, they can also describe the orientation of a plane. Commonly, point clouds generated by 3D imagery have planes associated with neighboring points to describe the surface that is imaged. Grouping is usually done through Delaunay meshes [34, 35] or k nearest neighbours [36, 37] with planes associated with each tetrahedral, or for each set of nearest neighbours as shown in Figure 3.4. One way to describe the plane is a point and normal vector.

\[
T_2 \Box T_1 := \ln (T_2 T_1^{-1})^\vee = \xi. \tag{3.26}
\]

\[\Box\] operator is used to denote the operation that maps a perturbation \(\xi\) to SE(3) and applies it to a nominal transformation \(\bar{T}\). Similarly, we can define a \(\Box\) operator that computes the difference between two transformations, \(T_1\) and \(T_2\), and maps this to se(3); i.e.,

\[
T_2 \Box T_1 := \ln (T_2 T_1^{-1})^\vee = \xi.
\]
3.3. SURFACE NORMALS

Figure 3.5: A surface normal vector $m \in S^2$ has a vector part, $\kappa$, and a scalar part, $\psi$.

Vectors are unit vectors that lie in one of two possible directions on an axis that is normal to the plane. While an axis is distinct from a normal vector in that it has two possible directions that equally describe the plane, in the case of a camera sensor imaging a surface it is a reasonable assumption that any observed normal vector is pointed towards the sensor, eliminating the other option as a possibility.

The parameterization of normal vectors in this thesis is borrowed from the approach presented by Hertzberg et al. in Appendix B.2 of [38] and expanded upon by Gallant in Chapter 2 of [33]. In this parameterization each normal vector in $\mathbb{R}^3$ is described by a point in $S^2$. 
3.3. SURFACE NORMALS

Normal Vectors

A normal vector can be parameterized as the orientation of a plane in $\mathbb{R}^3$ with two degrees of freedom. This is an over-parameterization (three parameters, one constraint) that has no singularities. A normal vector $m$ is a $3 \times 1$ column:

$$m := \begin{bmatrix} \kappa \\ \lambda \end{bmatrix},$$

(3.27)

where $\lambda \in \mathbb{R}$ and $\kappa \in \mathbb{R}^2$ are scalar and vector components respectively. Normal vectors are constrained to the unit sphere $S^2$, and therefore $\lambda^2 + \kappa^T \kappa = 1$. We can also relate the surface normal to the inclination $\psi \in [0, \pi]$ and the direction $r \in S^1$ through

$$\psi := \cos^{-1}(\lambda), \quad r := \frac{\kappa}{\|\kappa\|}.$$

(3.28)

The scalar and vector components, along with the inclination and direction, are shown in Figure 3.5.

The composition of normal vectors $m, n \in S^2$ is done using the normal product:

$$m \otimes n := m^+ n$$

(3.29)

where $m^+ o = m$ and $m^+ \in SO(3)$ is the compound operator on $m$, and $o \in S^2$ is the identity normal vector:

$$o := \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

(3.30)

The compound operator, $m^+$ is derived by considering the inclination angle, $\psi$, ...
3.3. SURFACE NORMALS

between \( \mathbf{m} \) and \( \mathbf{o} \). If \( \psi = 0 \), then \( \mathbf{m}^+ = \mathbf{1} \), the identity rotation matrix. If \( \psi \in (0, \pi] \), \( \mathbf{m}^+ \) can be constructed from a rotation \( \psi \) about the axis, \( \mathbf{a} \), that is perpendicular to both \( \mathbf{m} \) and \( \mathbf{o} \). Therefore we can compute \( \mathbf{a} \) from the cross product

\[
\mathbf{a} = \frac{\mathbf{1}}{\sin(\psi)} \mathbf{o} \times \mathbf{m} = \frac{\mathbf{1}}{\sin(\psi)} \begin{bmatrix} \mathbf{u} \\ 0 \end{bmatrix}.
\]

Letting \( \mathbf{m} = \begin{bmatrix} \kappa^T & \lambda^T \end{bmatrix}^+ \in S^2 \), \( \kappa = [\kappa_1 \: \kappa_2]^T \), and \( \mathbf{u} = [-\kappa_2 \: \kappa_1]^T \) and rearranging

\[
\mathbf{m}^+ = \cos(\psi) \mathbf{1} + (1 - \cos(\psi)) \mathbf{a} \mathbf{a}^T - \sin(\psi) \mathbf{a}^\wedge,
\]

substituting (3.32) into (3.33) we have

\[
\mathbf{m}^+ = \lambda \begin{bmatrix} \kappa & 0 \\ 0 & \lambda \end{bmatrix} + \frac{1 - \cos(\psi)}{\sin^2(\psi)} \begin{bmatrix} \mathbf{u} \\ 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ 0 \end{bmatrix}^T + \begin{bmatrix} \mathbf{u} \\ 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ 0 \end{bmatrix}^\wedge,
\]

(3.34)

\[
\mathbf{m}^+ = \left[ \begin{array}{cc}
\lambda \mathbf{1} + \frac{1}{\lambda + 1} \mathbf{u} \mathbf{u}^T \\
-\kappa^T \\
\end{array} \right] 
\]

(3.35)

The inverse compound operation has \( \mathbf{m}^+ \mathbf{m}^{-1} = \mathbf{o} \), and since \( \mathbf{m}^+ \) is an element of
3.3. SURFACE NORMALS

SO(3) we know its inverse is its transpose, therefore

\[ m^{-1} = (m^+)^T o \]  \hspace{1cm} (3.36)

\[ \begin{bmatrix} \lambda 1 + \frac{1}{\lambda + 1} uu^T & -\kappa \\ \kappa^T & \lambda \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \]  \hspace{1cm} (3.37)

\[ = \begin{bmatrix} -\kappa \\ \lambda \end{bmatrix} , \]  \hspace{1cm} (3.38)

where one can define an inverse compound operator \( m^- \)

\[ m^- := (m^{-1})^+. \]  \hspace{1cm} (3.39)

3.3.1 Perturbing Expressions With Surface Normals

Like rotations and transformations, surface normals are over parameterized, and prevent direct application of traditional state estimation algorithms which expect operations on the state and measurements to be in a vector space. Hertzberg et al. [38] addressed this problem in a similar way to rotation and transformation matrices, where a homeomorphism is defined that locally transforms the surface normal that exists in \( S^2 \) to a manifold in \( \mathbb{R}^2 \) where an algorithm can apply operations while maintaining a singularity-free global parameterization in \( S^2 \). However, unlike rotations and transformations, the unit sphere \( S^2 \) is not a differentiable manifold, it must have at least one point whose tangent is the null vector [39]. This can be resolved by defining a homeomorphism between \( S^2 \) and \( \mathbb{R}^2 \) that addresses the singularity. Note that \( \psi \) can be operated on the manifold if it exists in a subspace of \( \mathbb{R}^2 \), where for
3.3. SURFACE NORMALS

\( \psi \in \mathbb{R}^2, \)

\[
s^2 := \{ \psi : \| \psi \| < \pi \} \cup \{(\pi, 0)\}.
\] (3.40)

Letting \( \mathbf{m} \in S^2 \) and \( \psi \in \mathbb{R}^2 \) the surface normal exponential and logarithm are defined as

\[
\mathbf{m} = \exp(\psi) := \begin{bmatrix} \cos(\|\psi\|) \\ \text{sinc}(\|\psi\|) \psi \end{bmatrix},
\]

(3.41)

\[
\psi = \ln(\mathbf{m}) := \begin{cases} 0, & \text{for } \kappa = 0 \\ \text{atan2}(\|\kappa\|, \lambda), & \text{otherwise} \end{cases}
\]

(3.42)

where the unnormalized sinc function that has sinc(0) := 1 is used.

Using the compound operator defined in (3.35) and the homeomorphism defined in (3.42) we can define our usual manifold operator that perturbs a nominal surface normal, \( \bar{\mathbf{m}} \); i.e.,

\[
\bar{\mathbf{m}} \oplus \psi := \bar{\mathbf{m}}^+ \exp(\psi).
\] (3.43)

Similarly, we can define the difference between two surface normals, \( \mathbf{m}_2 \) and \( \mathbf{m}_1 \), mapped to \( s^2 \). Letting \( a = \mathbf{m}_2^T \mathbf{m}_1 \) we have

\[
\psi = \mathbf{m}_1 \ominus \mathbf{m}_2 := \begin{cases} \ln(\mathbf{m}_2^{-1} \mathbf{m}_1) & \text{if } a > -1, \\ (\pi, 0) & \text{otherwise}. \end{cases}
\]

(3.44)

where the cases in (3.44) ensure that \( \psi \in s^2 \) and is unique.

Just as with rotations, parameterizing a random surface normal with a normal vector would be difficult as it represents two degrees of freedom with three parameters.
and a unit constraint. If we were to do so, the covariance matrix representing the uncertainty in these parameters would be singular. We address this issue in the same way, by parameterizing the uncertainty using the normal vector parameterization that has the same number of parameters as degrees of freedom. Assuming a Gaussian distribution represents a random surface normal by its mean $\hat{m} \in S^2$ and covariance matrix $P_{\delta\psi}$ we have

$$\hat{m} = \bar{m} \oplus \delta\psi, \quad \delta\psi \sim N(0, P_{\delta\psi}),$$

where the error $\delta\psi \in s^2$ is a perturbation of the true value of the normal vector $\bar{m} \in S^2$. It is important to note that random surface normals, like rotations, cannot be exactly represented by a Gaussian distribution because $s^2$ (the space errors exist in) is bounded. However, as discussed by Gallant [33] on p. 28, for random axes whose individual variances are such that $3\sigma_{\delta\psi} < \pi$, a Gaussian distribution will enclose the vast majority of the probability within $s^2$ and is a sufficient approximation.

### 3.4 Homogeneous Points and Reflectivity

The two final types of parameterizations used in this thesis are homogeneous points and reflectivity. These quantities exist in $\mathbb{R}^3$ and the unit interval $[0,1]$ (a subset of $\mathbb{R}^1$) respectively, and are perturbed by quantities in these spaces. Thus, they can be perturbed through standard vector addition. However, there are some additional properties that are discussed in the following section.

#### 3.4.1 Homogeneous Points

Points in $\mathbb{R}^n$ can be parameterized by homogeneous coordinates [3], where the coordinates are scaled by a factor $s$ that is appended as an extra parameter. For example,
3.4. HOMOGENEOUS POINTS AND REFLECTIVITY

points in $\mathbb{R}^3$ are

$$ p = \begin{bmatrix} sx \\ sy \\ sz \\ s \end{bmatrix}. $$

(3.46)

Homogeneous coordinates are useful in describing landmarks whether they are near or far away, with no singularities or scaling issues [40]; no matter the scale, dividing the first three coordinates by the fourth returns the original point. They are also easily transformed from one frame to another by using transformation matrices, as was demonstrated in Section 3.2.2.

Homogeneous coordinates are perturbed by $\zeta \in \mathbb{R}^3$ by using a dilation matrix, $E$, such that

$$ \bar{p} + E\zeta = p $$

(3.47)

where

$$ E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}. $$

(3.48)

Assuming a Gaussian distribution represents a random homogeneous point by its mean $\tilde{p} \in \mathbb{R}^3$ and covariance matrix $P_\delta \zeta$

$$ \tilde{p} = \tilde{p} + E\delta \zeta, \quad \delta \zeta \sim \mathcal{N}(0, P_\delta \zeta) $$

(3.49)

where the error $\delta \zeta \in \mathbb{R}^3$ is a perturbation of the true value of the point $\tilde{p} \in \mathbb{R}^3$.  

40
3.4. HOMOGENEOUS POINTS AND REFLECTIVITY

3.4.2 Reflectivity

Reflectivity is a material property of a surface; it describes the effectiveness of a surface in reflecting incident light across the electromagnetic spectrum. In this thesis, reflectance describes the fraction of incident light that is reflected by the surface at the specific wavelength the LiDAR sensor being used emits at. Typically LiDAR sensors used on mobile robots operate in the near infrared region of the electromagnetic spectrum (e.g., the SwissRanger 4000 ToF camera used in Chapters 5-7 emits at 850 nm).

Because the fraction cannot be greater than 1, a reflectivity value exists in a subset of \( \mathbb{R}^1 \), the unit interval \([0, 1]\). A nominal reflectivity \( \bar{\varrho} \in [0, 1] \) can be perturbed by \( \gamma \in [-1, 1] \) such that

\[
\bar{\varrho} + \gamma = \varrho
\]  

(3.50)

and assuming a Gaussian distribution represents a random reflectivity by its mean \( \hat{\varrho} \in [0, 1] \) and variance \( \sigma_\gamma^2 \) we have

\[
\hat{\varrho} = \bar{\varrho} + \delta\gamma, \quad \delta\gamma \sim \mathcal{N}(0, \sigma_\gamma^2)
\]  

(3.51)

where the error \( \delta\gamma \in [-1, 1] \) is a perturbation of the true reflectivity \( \bar{\varrho} \in [0, 1] \). Note that like random rotations and axes, a Gaussian cannot exactly represent a random reflectivity because the interval \([-1, 1]\) on which the errors exist is bounded. However, for random reflectivities whose variances are such that \( 3\sigma_\delta > 1 \), a Gaussian distribution will enclose the vast majority of the probability within \([-1, 1]\) and is a sufficient approximation.
Chapter 4

Intense Bundle Adjustment

This chapter introduces Intense Bundle Adjustment (IntenseBA), a state estimation algorithm that works by augmenting each landmark with a directional state (represented by a surface normal) and a reflectivity state (represented by a one dimensional parameter between 0 and 1). This chapter, which describes some of the primary contributions of this thesis, shows that by including additional measurements into the standard 3D point landmark state, it is possible to improve the estimation accuracy and precision of a sensor’s trajectory (a discrete sequence of poses) and increase the information that is in the final map of landmarks. The path and map are estimated by using an active sensor that reports the intensity of the returned signal, as well as the distance to objects in the robot’s surroundings. These measurements are used to form a large optimization problem over the space of the robot pose and landmark states, where the solution maximizes the likelihood of all measurements. While the contributions of this thesis improve any form of SLAM, pure visual odometry, (i.e., no other sensors, such as an inertial measurement unit, are included in the optimization problem) is utilized to demonstrate the improvements so that a direct comparison between traditional methods of solving the optimization problem can be made.
4.1. RELATED WORK

This chapter begins with an overview of previous work related to the use of surface normal and intensity measurements for estimation problems in robotics and computer vision and an explanation of the physics behind both range and intensity measurements. Next, the generic batch optimization problem, specifically the construction of the objective function and the method used to maximize it are explained. The chapter then formally states the problem solved by IntenseBA and describes all steps completed during optimization. This starts with the development of observation models for active sensors that are a function of the landmark and sensor pose parameters used for estimation and concludes with an explanation of Sparse Bundle Adjustment (SBA) which exploits the inherent sparsity of the problem.

4.1 Related Work

An often overlooked and unused feature of active sensors is the additional measurement of irradiance that typically accompanies each range measurement. This is usually reported as intensity, and is proportional to the fraction of power returned to the sensor compared to what was emitted. In this thesis, the observation of intensity is modelled as a function of parameters describing the environment (position, surface normal and reflectivity of landmarks) and the sensor’s pose, and used directly in nonlinear optimization to improve the results of VO or SLAM algorithms. There is a significant amount of previous work that use planes as an additional part of the landmark state (Section 4.1.1), and in the use of intensity measurements to improve visual odometry and SLAM in various ways (Section 4.1.2). There are even a few examples that use both measurements (Section 4.1.3).
4.1. RELATED WORK

4.1.1 Surface Normals

An early example of using surface normals for localization is demonstrated in [41] where sonar sensors are used to model an indoor environment in terms of two-dimensional line segments. These line segments are then used to constrain the real-time pose estimate. The idea of using line segments to represent an environment has continued with the advent of modern SLAM algorithms. One version of EKF SLAM [42] extracts line segments and their associated covariance from LiDAR range data points showing improved accuracy and reduced complexity in the map (the main driver of complexity in SLAM). Surface normal estimation in two dimensions has also been used to improve estimated noise characteristics of LiDAR range estimates by modelling the range noise as dependent on the incident angle between the LiDAR viewing direction and the surface normal [43].

The use of surface normals has extended to three dimensional EKF-SLAM [44], and IntenseBA paramterizes surface normals and points in a very similar way. The same authors also represent plane uncertainty by weighting points by a scalar value related to the uncertainty in the point measurements [45]. The normal measurement in IntenseBA is similar, however it uses the sensor measurements in the sensor space to generate more accurate uncertainties for surface normal measurements, described in Appendix A.

Estimation of local surface normals was successfully applied to a monocular camera SLAM problem [7], where they were used to improve feature correspondences during extreme motions with large rotation. Pathak et al. [46] applied a similar technique to a pose graph SLAM problem through extraction of large surface patches (versus smaller local ones). Even with large patches they were able to show that plane
4.1. RELATED WORK

registration is accurate enough at estimating rotations that only translation must be optimized across the pose-graph, greatly reducing computational complexity.

Gallant et al. [47] extracted surface normals with a ToF camera to build an axis map, a representation of the dominant orientations of planar joints in rock outcrops. This version of pose-graph SLAM is unique in that it does away with estimation of position entirely, only estimating the sensor orientation and rock joint orientations. The parameterization and perturbation operations introduced for axis vectors is based on the work by Hertzberg et al. [38] and is used in the IntenseBA algorithm.

4.1.2 Intensity

There is relatively little work about the use of intensity in VO and SLAM. One early experiment was performed by Guivant et al. [48] where reflective beacons were placed in the environment and used as landmarks. In this way, they were able to avoid more complicated keypoint detection and matching schemes to determine correspondences between successive measurements. Yoshitaka et al. [49] took this a step farther, incorporating intensity to determine point correspondences in an iterative closest point algorithm [50] to estimate transformations between successive frames.

Intensity measurements have also been used to form images on which 2D keypoint detection, description, and matching algorithms are run to provide landmark correspondences between successive frames for visual odometry for scanning LiDAR [12, 13] and ToF cameras [14–16].

Levinson [51] performed a correction of scanning LiDAR intensity based on differing biases between a LiDAR’s set of 64 lasers. These biases result in one laser reporting a higher or lower intensity value than another when observing the same
4.1. RELATED WORK

surface. By noting the intensity returned by each laser when measuring the same location, a mean intensity across all lasers is determined that is used to correct future measurements. This is done for all possible values of intensity to build up a look-up table of corrections for each laser and possible intensity value. The result is more consistent shading for visualization of the maps built in a SLAM algorithm.

4.1.3 Surface Normals and Intensity

There has also been an effort in the literature to correct intensity data according to reflection models and estimate the reflective properties of the measured surfaces that are independent of the sensor location. The models used for correction are dependent on the incident direction of the light with the reflecting surface and must include an estimation of surface normal along with an intensity measurement.

A Lambertian model is often assumed [52], dependent on range and the cosine of the incident angle. However, it has been noted that areas with wet, smooth or highly reflective surfaces result in specular reflections that are not explained by a Lambertian model. Ding et al. [53] apply a filter using the Phong reflection model [54] that includes specular reflections.

The most relevant work with respect to this thesis is presented by Sheraz Khan et al. [55]. Here, they apply a data-driven approach to modelling laser intensity of two scanning LiDAR models, fitting curves to intensity as a function of range and incident angle through a calibration procedure, instead of using a Lambertian or more complicated reflectivity model. The reflectivities are also normalized to arbitrary reference reflectivity, in their case normal white printing paper. They apply their measurement model as an extension to Hector SLAM [56] that includes the observation of intensity
4.2. PHYSICAL PRINCIPLES OF LIDAR

directly in the optimization problem, and show improved map consistency along with estimated reflectivities added to their map.

IntenseBA in contrast applies a physical model of reflectivity which is corrected through a calibration procedure (described in Chapter 6.4). A surface of known reflectivity is used during calibration, so the reflectivities estimated are not referenced to any surface. The model is applied to ToF camera intensity observations, and provides improvements to three-dimensional pose and landmark estimation. Non-Lambertian reflections are handled through RANSAC outlier removal, and do not impact the performance of the system.

4.2 Physical Principles of LiDAR

The common attribute of all LiDAR sensors is that they use an active lighting source as opposed to a passive system, such as a typical camera, which relies on an external lighting source such as the sun to light the scene. A LiDAR sensor measures the returning light that is reflected off target objects and determines the range between the target and the sensor. This is accomplished by determining the light’s time-of-flight to the target and back. The azimuth and elevation of the lighting source is also estimated through a combination of calibration, encoders and/or an IMU. The sensor also typically measures the power or amplitude of the returning light, and compares this with what the power or amplitude at which it was emitted. The ratio between the two is often called Intensity by LiDAR manufacturers and is usually reported alongside range measurements by LiDAR sensors.
4.2. PHYSICAL PRINCIPLES OF LIDAR

4.2.1 Range Measurements

To determine the time-of-flight, and consequently the range, there are two competing technologies [57] and variants among these: (1) pulse LiDAR which produces a light pulse and measures the time-of-flight between the emitted and received pulse, and (2) Continuous-wave LiDAR (CW-LiDAR) which modulates a continuous light emission’s amplitude (or in some cases, frequency [58]) with time. The light is emitted by either a LED or a laser and the sensor measures the phase shift of the received light to determine the time-of-flight. In all cases the time-of-flight, \( t_f \), is simply:

\[
\frac{4r}{c},
\]

where \( r \) is the range between the sensor and the target surface and \( c \) is the speed of light.

Going forward it is important to mention coordinate systems and how measurements from the LiDAR are reported. In most cases, 2D LiDAR reports a range and an azimuth value. This azimuth value is with respect to the LiDAR reference frame, and is the rotation about the \( z \)-axis which is depicted in Figure 4.1a. When discussing 3D LiDAR, an additional value of elevation is reported, which is with respect to the \( y \)-axis of the sensor reference frame (two examples are depicted in Figure 4.1b and Figure 4.1c).

Pulse LiDAR

Pulse LiDAR is by far the more common of the two types of LiDAR, implementing timing circuitry [57, 58] to directly measure the time-of-flight that elapses between the time of a LiDAR pulse emission and when it is received. Depending on the accuracy,
4.2. PHYSICAL PRINCIPLES OF LIDAR

(a) A scanning 2D LiDAR

(b) Dual mirror scanning 3D LiDAR

(c) Velodyne 64-E 3D LiDAR with multiple scanning lasers. Credit: Velodyne Acoustics, Inc. [59].

Figure 4.1: Types of scanning LiDAR

range, time and dimensional (2D or 3D) requirements, different laser sources and scanning mechanisms are used. Pulse LiDAR that produce 2D scans generally have a single LiDAR that is pointed towards a mirror that spins and sweeps across the field of view of the sensor as shown in Figure 4.1a. In this case, the azimuth of the mirror is tracked using optical encoders. Various authors have mounted 2D LiDARs on tilt units or turntables [60] that vary the sensor’s elevation or azimuth angle with respect to the vehicle to produce 3D measurements.

Pulse LiDAR that produce 3D scans without these methods fall into two categories: (1) Scanning LiDAR that use an additional mirror in a fashion similar to 2D
4.2. PHYSICAL PRINCIPLES OF LIDAR

sensors and (2) multiple lasers organised in arrays that are mounted in a housing that is spun about the LiDAR’s z-axis. One example has a hexagonal mirror that determines the azimuth angle of the laser, and a nodding rectangular mirror determines its elevation as shown in Figure 4.1b. Surveying LiDAR used to map rock faces and mines also use some combination of two mirrors. An example of the latter type is the Velodyne LiDAR systems [59] shown in Figure 4.1c. Depending on the model, the Velodyne housing contains between 16 and 128 lasers and corresponding photodiodes which are oriented at equal increments in elevation and azimuth across its field of view. The housing is then able to rotate a full 360° about the sensor’s z-axis. Pulse LiDAR systems vary widely in the number of points per second they provide, their maximum range and their density. The main trade-off is between the density of the point cloud (as well as the scan pattern that possibly leaves gaps in coverage) and the acquisition rate of a full scan. Power and weight also have an influence, as the more powerful the laser signal is the larger the maximum range.

**Continuous-wave LiDAR**

CW-LiDAR is a less common form of LiDAR, and does not measure the light’s time-of-flight directly. Instead, the sensor modulates the intensity of continuously emitted light at a known modulation frequency, \( f \). The sensor then samples the returning light and determines the phase difference between the emitted and received light, \( \varphi \). The main advantage of CW-LiDAR is high measurement frame rates, typically 2 orders of magnitude greater than pulse LiDAR [58]. The maximum measurable phase difference is \( 2\pi \), which corresponds to the maximum unambiguous range measurement.
4.2. PHYSICAL PRINCIPLES OF LIDAR

Figure 4.2: Optical signal in a CW-LiDAR system based on sinusoidal modulation. Credit: Schwider et al. [61].

for the sensor. The maximum range, \( r_{\text{max}} \), is given by:

\[
r_{\text{max}} = \frac{c}{2f}, \tag{4.2}
\]

The relationship between the emitted power of the modulated signal, \( p_e \), and the returning signal, \( p_r \), is shown in Figure 4.2 and correspond to the following equations [61]

\[
p_e = p_a \cdot [1 + \cos (2\pi ft)], \tag{4.3}
\]

\[
p_r = p_b + k \cdot p_a \cdot [1 + \cos (2\pi f(t - \tau))], \tag{4.4}
\]

where \( p_a \) is the amplitude of the transmitted wave, \( p_b \) is background noise, \( \tau \) is the transit time and the phase, \( \phi \), between the returning signal and the transmitted signal is: \( \phi = 2\pi f(t - \tau) \).

To calculate the phase, the sensor samples the returning signal for a specified
4.2. PHYSICAL PRINCIPLES OF LIDAR

exposure time, $t_e$, at four $90^\circ$ intervals. During the integration time, the sensor is accumulating photons and converting them to stored electrons. This is called a photo current and the sensor is actually sampling this photo current over the integration time. For every watt of power received the sensor generates a corresponding photo current that obeys the responsivity law:

$$r_e = \frac{\eta_q \lambda}{1.23985} \text{ A/W,}$$  \hspace{1cm} (4.5)

where $\eta_q$ is the quantum efficiency of the sensor, and $\lambda$ is the wavelength of the light (typically 750 - 1400 nm). Sampled over $t_e$, this corresponds to charge carriers (i.e., electrons), which the sensor then converts to a digital number. The full equation is then:

$$a = br_e p_t t_e,$$  \hspace{1cm} (4.6)

where $a$ is the digital amplitude (often reported as intensity), and $b$ is the digital conversion factor of the sensor in units of inverse Coulombs ($A^{-1}s^{-1}$). This sampling occurs four times, resulting in four amplitude measurements, which are needed to compute the phase and amplitude [62]:

$$\varphi = \text{atan} \left[ \frac{a_3 - a_1}{a_0 - a_2} \right],$$  \hspace{1cm} (4.7)

$$i = \sqrt{(a_3 - a_1)^2 + (a_0 - a_2)^2}.$$  \hspace{1cm} (4.8)

Range is then easily computed as a fraction of the maximum range:

$$r = r_{\text{max}} \cdot \frac{\varphi}{2\pi}.$$  \hspace{1cm} (4.9)
Unfortunately, the range accuracy and precision are inversely proportional to $r_{\text{max}}$ [57] and a trade off must be made between the modulation frequency that allows the desired performance. Alternatively, multiple modulation frequencies can be used. A relatively low modulation frequency provides a coarse, long range measurement, and a high modulation frequency provides an accurate measurement whose range ambiguity is solved by the low frequency measurement. If more frequencies are used, the trade-off can be improved, however the sensor frame-rate is divided by the number of frequencies used [58]. Laser-based CW-LiDAR has existed for over a decade in aerial and survey scanning [63] and uses the same mechanisms to achieve 3D scanning (e.g., mirrors) as pulse LiDAR. However, a new type of CW-LiDAR that is LED-based has emerged in the last decade. This sensor trades a laser light source for an array of infrared LEDs, and a photo-diode light sensor with CMOS/CCD camera pixels [64, 65]. This provides a completely solid-state solution that requires no moving parts,
less power, less volume, and can produce measurements at the same moment across the entire field of view at much higher frame rates (up to 100 Hz, versus anywhere from 0.001 to 5 Hz for scanning LiDAR [63]).

In this thesis this type of sensor is referred to as a ToF camera and a typical example is the MESA SwissRanger 4030 shown in Figure 4.3. Point clouds are produced by using the range measurements at each pixel location according to a standard pin-hole camera model. Measuring range across the entire field of view at the same moment is particularly suited to mobile robotics because, in contrast, scanning-based LiDAR are known to introduce motion-blur [66] as the LiDAR scans across the sensor’s moving field-of-view.

However, ToF cameras introduce their own disadvantages; they have problems working in outdoor environments because the pixels measurements are noisier or even completely saturated due to ambient infrared light, and the specialised lock-in CCD/CMOS pixel arrays [65] are typically lower resolution compared to RGB cameras or scanning LiDAR. Active research in filtering ambient light [67] and the sensor’s foundation in semiconductor technology will hopefully address these limitations in the future.

4.2.2 Intensity Measurements

As we have seen, LiDAR and ToF cameras emit light (from a laser or LEDs) and measure the returning light that is reflected from surrounding objects to determine the range between the target and the sensor. Additionally, the sensor measures the power or amplitude of the returning light, and compares this with the power or amplitude at which it was emitted. The ratio between the two is often called intensity
4.2. PHYSICAL PRINCIPLES OF LIDAR

by active sensor manufacturers and is usually reported alongside range measurements.

The physical principles that determine the power of the returning signal are the same as microwave radar [52]. The radar range equation comprises the three main factors: (a) the sensor, (b) the target, and (c) atmospheric parameters. An assumption here is the target surface is Lambertian, which means that light is scattered evenly in all directions when reflected. For the emitted signal with power $p_e$,

$$ p_r = \frac{p_e a_r^2 \rho}{4r^2} \eta_{sys} \eta_{atm} \cos \alpha, $$

(4.10)

where $p_r$ is the returning power of the signal, $a_r$ is the receiver aperture diameter, $\rho$ is the target reflectivity, $r$ is the range to the target, $\eta_{sys}$ is the system transmission factor, $\eta_{atm}$ is the atmospheric transmission factor and $\alpha$ is the incident angle. The intensity measurement is linearly proportional to $p_r$ by some constant parameters of the LiDAR sensor, $\eta_{\text{LiDAR}}$, that can be grouped with other constant parameters $\eta_{sys}$, $\eta_{atm}$, $a_r$, and $p_e$ to form $\eta$;

$$ i = \eta_{\text{LiDAR}} p_r $$

(4.11)

$$ = \eta_{\text{LiDAR}} \frac{p_e a_r^2 \rho}{4r^2} \eta_{sys} \eta_{atm} \cos \alpha $$

(4.12)

$$ = \eta \frac{\rho \cos \alpha}{r^2}. $$

(4.13)

Looking at (4.10), we can see that $p_r$ is a function of the range, which itself is a function of the target position and the sensor pose, the incidence angle which is determined by the sensor orientation and the surface normal, and on $\rho$ (the target’s reflectivity), which depends on the material of the target.
4.3 Batch Nonlinear Least-Squares Optimization

The task of state estimation is finding the estimate of all state parameters, $x$, that best explains the prior input information, $u$, and the measurements (i.e., observations), $y$ [32]. In discrete time, with index $k$, we can define the sensor motion and observation models as:

\[ x_k = f(x_{k-1}, u_k) + w_k \]  \hspace{1cm} (4.14)

\[ y_k = g_k(x_k) + v_k \]  \hspace{1cm} (4.15)

where $w_k \sim \mathcal{N}(0, Q_k)$ and $v_k \sim \mathcal{N}(0, R_k)$ are the Gaussian noise variables for $u_k$ and $y_k$ respectively and $f(\cdot)$ and $g(\cdot)$ are the (potentially nonlinear) transition and observation functions. More formally, we are solving the maximum a posteriori problem:

\[ \hat{x} = \arg\max_x p(x|u, y), \]  \hspace{1cm} (4.16)

where we use $\hat{\cdot}$ to denote posterior estimates of $x$. This estimate maximizes the probability density function $p(x)$ given the inputs $u$ and the measurements $y$ and we have

\[ x = (x_0, ..., x_K), \]  \hspace{1cm} (4.17)

\[ u = (u_0, u_1, ..., u_K), \]  \hspace{1cm} (4.18)

\[ y = (y_0, ..., y_K). \]  \hspace{1cm} (4.19)
4.3. BATCH NONLINEAR LEAST-SQUARES OPTIMIZATION

where we use $\hat{\cdot}$ to denote prior estimates. By using Bayes’ rule, (4.16) can be rewritten as:

$$
\hat{x} = \arg\max_{x} p(x|u, y) = \arg\max_{x} \frac{p(y|x, u)p(x|u)}{p(y|u)} = \arg\max_{x} p(y|x)p(x|u),
$$

(4.20)

The denominator is dropped because it does not depend on $x$ and the $u$ in $p(y|x, u)$ as it does not affect $y$, the measurement, if the value of $x$ is known. We make the assumption that all noise variables are uncorrelated, allowing $p(y|x)$ to be factored as:

$$
p(y|x) = \prod_{k=0}^{K} p(y_k|x_k),
$$

(4.21)

and $p(x|v)$ as

$$
p(x|v) = \prod_{k=1}^{K} p(x_k|x_{k-1}, u_k).
$$

(4.22)

Each of these Gaussian probability density functions are given by:

$$
p(y_k|x_k) = \frac{1}{\sqrt{(2\pi)^K \det R_k}} \exp \left( -\frac{1}{2} \left( y_k - g_k(x_k) \right)^T R_k^{-1} \left( y_k - g_k(x_k) \right) \right),
$$

(4.23)

$$
p(x_k|x_{k-1}, u_k) = \frac{1}{\sqrt{(2\pi)^K \det Q_k}} \exp \left( -\frac{1}{2} \left( x_k - f_k(x_{k-1}, u_k) \right)^T Q_k^{-1} \left( x_k - f_k(x_{k-1}, u_k) \right) \right).
$$

(4.24)

To further simplify the logarithm of both sides is taken. Since a logarithm is a monotonically increasing function, the states that maximize the original problem also
4.3. BATCH NONLINEAR LEAST-SQUARES OPTIMIZATION

maximize its logarithm:

\[
\ln (p(y_k|x_k)p(x_k|x_{k-1})) = \sum_{k=1}^{K} \ln p(x_k|x_{k-1}, u_k) + \sum_{k=0}^{K} \ln p(y_k|x_k)
\]

(4.25)

where

\[
\ln (p(x_k|x_{k-1}, u_k)) = -\frac{1}{2} (x_k - f_k(x_{k-1}, u_k))^T Q_k^{-1} (x_k - f_k(x_{k-1}, u_k)) - \frac{1}{2} \ln ((2\pi)^K \det Q_k)
\]

(4.26)

\[
\ln (p(y_k|x_k)) = -\frac{1}{2} (y_k - g_k(x_k))^T R_k^{-1} (y_k - g_k(x_k)) - \frac{1}{2} \ln ((2\pi)^K \det R_k)
\]

(4.27)

Finally, the second terms on the right hand side are dropped as they are not dependent on \(x\). With these two terms, it is now possible to build an objective function. The typical objective function for optimization is a scalar squared-error function \(J\):

\[
J(x) = \frac{1}{2} \sum_{k=1}^{K} e_k(x)^T W_k e_k(x),
\]

(4.28)

where \(e_k\) is one of the \(K\) error terms weighted by matrix \(W_k\). We can use the negatives of the log-likelihoods in (4.26) and (4.27) as our terms, stacking them together to form \(e_k\) and \(W_k\),

\[
e_k = \begin{bmatrix} x_k - f_k(x_{k-1}, u_k) \\ y_k - g_k(x_k) \end{bmatrix}, \quad W_k = \text{diag}\{Q_k^{-1}, R_k^{-1}\},
\]

(4.29)
4.3. BATCH NONLINEAR LEAST-SQUARES OPTIMIZATION

and finding the design parameter, \( \mathbf{x}^* \), that minimizes the objective function

\[
\mathbf{x}^* = \arg\min_{\mathbf{x}} J(\mathbf{x}), \quad (4.30)
\]

is equivalent to maximizing the joint likelihood of the state given all measurements. If we stack our error terms in a vector, and our measurement noise covariance matrices along a diagonal, we can set up the cost function in matrix form,

\[
\mathbf{e}(\mathbf{x}) := \begin{bmatrix}
\mathbf{e}_1(\mathbf{x}) \\
\vdots \\
\mathbf{e}_K(\mathbf{x})
\end{bmatrix}, \quad \mathbf{W} := \text{diag} \{ \mathbf{Q}_1, \ldots, \mathbf{Q}_K, \mathbf{R}_1, \ldots, \mathbf{R}_K \} \quad (4.31)
\]

\[
J(\mathbf{x}) = \frac{1}{2} \mathbf{e}(\mathbf{x})^T \mathbf{W}^{-1} \mathbf{e}(\mathbf{x}) \quad (4.32)
\]

which is convenient if \( \mathbf{e} \) is a linear function, as \( J \) is quadratic in \( \mathbf{x} \) and the minimum can be found by setting \( \frac{\partial J^T}{\partial \mathbf{x}} = \mathbf{0} \) and solving the system of equations. Unfortunately, often our measurement models are nonlinear and we must use nonlinear optimization techniques, such as Gauss-Newton optimization, to iteratively find the values of \( \mathbf{x} \) that minimize \( J \). The next section gives a description of Gauss-Newton optimization and going forward the motion prior terms from the cost function in (4.3) are dropped so that our problem is purely dependent on the sensor measurements. In this way the problem simplifies from one of Simultaneous Localization and Mapping (SLAM) to Bundle Adjustment (BA) [32].
4.4. GAUSS-NEWTON OPTIMIZATION

4.4 Gauss-Newton Optimization

The Gauss-Newton algorithm is a version of Newton's method, where a nonlinear objective function is approximated iteratively. At each iteration the objective function is approximated through a second-order Taylor series expansion about the current operating point, \( \bar{x} \), and a perturbation that moves the design parameter towards the minimum, i.e.,

\[
J(\bar{x} + \delta x) \approx J(\bar{x}) + \left( \frac{\partial J(x)}{\partial x} \bigg|_{\bar{x}} \right) \delta x + \frac{1}{2} \delta x^T \left( \frac{\partial^2 J(x)}{\partial x \partial x^T} \bigg|_{\bar{x}} \right) \delta x.
\] (4.33)

To find the \( \delta x \) that minimizes this approximation, we take the derivative with respect to \( \delta x \) and set to 0,

\[
\frac{\partial J(\bar{x} + \delta x)}{\partial \delta x} = \left( \frac{\partial J(x)}{\partial x} \bigg|_{\bar{x}} \right) + \delta x^T \left( \frac{\partial^2 J(x)}{\partial x \partial x^T} \bigg|_{\bar{x}} \right) = 0
\] (4.34)

\[
\Rightarrow \left( \frac{\partial^2 J(x)}{\partial x \partial x^T} \bigg|_{\bar{x}} \right) \delta x^* = - \left( \frac{\partial J(x)}{\partial x} \bigg|_{\bar{x}} \right)^T.
\] (4.35)

This is repeated until the perturbation is sufficiently small. Gauss-Newton modifies this by asserting that near the minimum, the Hessian itself can be approximated as

\[
\frac{\partial^2 J(x)}{\partial x \partial x^T} \bigg|_{\bar{x}} \approx \left( \frac{\partial J(x)}{\partial x} \bigg|_{\bar{x}} \right)^T \left( \frac{\partial J(x)}{\partial x} \bigg|_{\bar{x}} \right)
\] (4.36)

due to the error being relatively small. This has the advantage over Newton’s method of not having to compute the many partial derivative terms that make up the Hessian. As long as the initial error is small, the same benefits of Newton’s algorithm over a simple gradient descent (guaranteed convergence when the initial error is small
and quadratic rate of convergence) hold. To apply this to a BA problem (assuming no motion prior), we begin with the following error term for each observation of a landmark from a pose:

\[ e(x_{jk}) = y_{jk} - g(x_{jk}), \]  

where \( y_{jk} \) is a measurement of the \( j \)-th landmark from the \( k \)-th pose. The objective function that must be minimized is then

\[ J(x) = \frac{1}{2} \sum_{j,k} e(x_{jk})^T R_{jk}^{-1} e(x_{jk}). \]  

Because \( e(\cdot) \) is nonlinear, \( J \) is not a quadratic function that allows us to solve for \( x \) directly. However, \( J \) can be approximated as a quadratic function of \( \delta x \) (a small change, or *perturbation*, towards the minimum of \( J \)), evaluated at an initial guess for \( x \), denoted \( \bar{x} \).

\[ J(\bar{x} + \delta x) = \frac{1}{2} \sum_{jk} e(\bar{x}_{jk} + \delta x_{jk})^T R_{jk}^{-1} e(\bar{x}_{jk} + \delta x_{jk}) \]  

\[ \approx \frac{1}{2} \sum_{jk} (e(\bar{x}_{jk}) + G_{jk} \delta x_{jk})^T R_{jk}^{-1} (e(\bar{x}_{jk}) + G_{jk} \delta x_{jk}) \]  

\[ \approx \frac{1}{2} \sum_{jk} (e(\bar{x}_{jk})^T R_{jk}^{-1} e(\bar{x}_{jk}) + 2 \delta x_{jk}^T G_{jk}^T R_{jk}^{-1} e(\bar{x}_{jk}) + \delta x_{jk}^T G_{jk}^T R_{jk}^{-1} G_{jk} \delta x_{jk}) \]

where

\[ G_{jk} = \left. \frac{\partial e(x)}{\partial x} \right|_{x_{jk}}, \quad \delta x_{jk} = \begin{bmatrix} \delta \xi_k \\ \delta \ell_j \end{bmatrix} \]

where \( \delta \xi_k \) and \( \delta \ell_j \) are the perturbations associated with the observation from the \( k \)-th pose of the \( j \)-th landmark respectively. From here the value of \( \delta x \) needed to
minimize this approximation

\[ \delta \mathbf{x}^* = \arg\min_{\delta \mathbf{x}} J(\bar{\mathbf{x}} + \delta \mathbf{x}), \quad (4.41) \]

is found by taking the derivative of \( J(\bar{\mathbf{x}} + \delta \mathbf{x}) \) with respect to \( \delta \mathbf{x} \) and setting to zero:

\[ \frac{\partial J(\bar{\mathbf{x}} + \delta \mathbf{x})}{\partial \delta \mathbf{x}} \approx \sum_{jk} \left( \mathbf{G}^T_{jk} \mathbf{R}_{jk}^{-1} \mathbf{e}(\bar{x}_{jk}) + \mathbf{G}^T_{jk} \mathbf{R}_{jk}^{-1} \mathbf{G}_{jk} \delta \mathbf{x}_{jk} \right) = 0, \quad (4.42) \]

\[ \Rightarrow \mathbf{H}\delta \mathbf{x} = -\mathbf{b}, \quad (4.43) \]

where

\[ \mathbf{H} = \sum_{jk} \mathbf{P}^T_{jk} \mathbf{G}^T_{jk} \mathbf{R}_{jk}^{-1} \mathbf{G}_{jk} \mathbf{P}_{jk}, \quad (4.44) \]

\[ \mathbf{b} = \sum_{jk} \mathbf{P}^T_{jk} \mathbf{G}^T_{jk} \mathbf{R}_{jk}^{-1} \mathbf{e}(\bar{x}_{jk}), \quad (4.45) \]

\[ \delta \mathbf{x} = \begin{bmatrix} \delta \xi_1 \\ \vdots \\ \delta \xi_K \\ \delta \ell_1 \\ \vdots \\ \delta \ell_M \end{bmatrix}, \quad (4.46) \]

where the pose perturbations are in the top section of \( \delta \mathbf{x} \) and the landmark perturbations are in the bottom section. \( \mathbf{P}_{jk} \) is a matrix that picks off the \( jk \)-th perturbations when multiplied by \( \delta \mathbf{x} \). This is a linear system of equations that can be solved as long as the approximate Hessian \( \mathbf{H} \) is invertible. However, due to the structure of \( \mathbf{H} \)
there are usually methods for solving the linear system of equations faster than direct inversion of $H$, which is discussed in Section 4.6. Once the optimal perturbation $\delta \mathbf{x}^*$ is solved, $\bar{x}$ is updated:

$$\bar{x} \leftarrow \bar{x} + \delta \mathbf{x}^*, \quad (4.47)$$

and this is repeated until some convergence criteria is met (i.e., $\delta \mathbf{x}$ is “small”). In the case of IntenseBA, the perturbations for rotations and surface normals are not Euclidean, so special care has to be taken using the perturbation schemes discussed in Section 3.1 and Section 3.3 respectively to calculate Jacobian and Hessian values with respect to the minimum representation. To that end, the next section addresses the problem of bundle adjustment by using a Time-of-Flight camera.

4.5 Intense Bundle Adjustment

Bundle adjustment (BA) is an application of nonlinear least squares optimization to the problem of 3D reconstruction. Measurements from a set of images of the same environment are used to derive information about the environment and the camera’s motion. If the error in the measurements is zero-mean Gaussian, then BA can be shown to be the maximum likelihood estimator. BA was first introduced in the 1950s [68], and led to the automation of photogrammetry, the process of stitching aerial images to build maps. Over time this extended to its use on satellite imagery, and with the advent of modern keypoint detection and description algorithms [69] has become common place in computer vision applications. For our problem we seek to apply this technique to a mobile robot equipped with a LiDAR sensor that solves for the optimal pose of the sensor at discrete times where LiDAR measurements are received as well as the position, reflectivity and surface normal of landmarks that are
recognised in the robot’s environment.

In the robotics literature, a distinction between SLAM and BA is usually made; if a motion prior generated from interoceptive sensors (e.g., an IMU) is included in the objective function along with exteroceptive observations (e.g., a camera) it is considered a SLAM algorithm. If only exteroceptive sensors are utilized, it is considered BA. Because this thesis’ contribution is limited to the sensor model of the camera, we focus on BA. However the same improvements this thesis presents could also be included in any SLAM algorithm.

4.5.1 Problem Definition

IntenseBA determines an estimate $\mathbf{x}$ that consists of the path of the LiDAR sensor (a set of discrete poses represented as transformations) and a landmark map (a discrete set of points that are augmented with associated surface normals and reflectivities). It does this through observations of landmarks at each discrete pose, depicted in Figure 4.4. The variables of interest (and some simplified notation) are:

\[ \mathbf{T} = (T_1, ..., T_K) : \text{Transformation path} \]

\[ T_k = T_{s_kg} : \text{Transformation matrix that represents the pose of the sensor at time-step } k, \text{ parameterized by the transformation between the global frame and the } k\text{-th sensor frame } (T_{s_kg}) \]

\[ \ell = (\ell_1, ..., \ell_M) : \text{Landmark map} \]

\[ \ell_j = \begin{bmatrix} p_{jg}^T & \varrho_j & n_{jg}^T \end{bmatrix}^T : \text{Landmark representing the position, reflectivity, and surface normal of } j\text{-th landmark in the global frame} \]
4.5. INTENSE BUNDLE ADJUSTMENT

Figure 4.4: The bundle adjustment problem estimates the transformation between the moving frame and the global frame across all timesteps, $T_{s_{1:k}g}$, and all the landmarks, $\ell_{j_{1:M}}$ from the observations $y_{1:MK}$.

$x = (T, \ell)$: Transformation path and landmark map

$x_{jk} = (T_k, \ell_j)$: Transformation at time step $k$ and landmark $j$

$y_{jk}$: Observation of the $j$-th landmark from the $k$-th pose.

$R_{jk}$: Covariance of $y_{jk}$

and we use $x_{jk} = \{T_k, \ell_j\}$ to indicate the subset of the state vector that includes the $k$-th pose and the $j$-th landmark. Each of the entries in $x$ is modelled as a random variable, represented by $\hat{x} = (\hat{T}, \hat{\ell})$. IntenseBA determines the estimate $\hat{x}$ that maximizes the likelihood of all of the observations. Note that the initial pose, $T_0$, is not included as a state to be estimated because the system is unobservable if this is not defined beforehand, a more in depth analysis of which is included in Section 65.
4.5. INTENSE BUNDLE ADJUSTMENT

5.1. Additionally an initial estimate of \( \hat{x} \) is needed, which unless otherwise stated will have \( \hat{T} \) initialized to the defined initial pose \( T_0 \) and \( \hat{\ell} \) initialized by projecting the first set of measurements from pose \( T_0 \).

4.5.2 IntenseBA Observation Model

The problem at hand is one of probabilistic estimation, where we must pass our uncertain estimated parameters through a nonlinear time-of-flight camera model to make measurement predictions that can be used in an optimization problem (e.g., Bundle Adjustment). The observation model is of the form

\[
y_{jk} = g(x_{jk}) + v_k
\]  

where \( g(\cdot) \) is the nonlinear observation model and \( v_k \sim (0, R_{jk}) \) is additive Gaussian noise. In this section, we derive \( g(\cdot) \) for LiDAR sensors with an extension to landmarks that include surface normal and reflectivity parameters, following the style presented by Barfoot in [32].

In the case of a LiDAR sensor, we can separate the observation model into a composition of two functions. The first transforms the landmark parameters from the global coordinate frame to the sensor coordinate frame where for the \( k \)-th pose and the \( j \)-th landmark we have

\[
z(x_{jk}) = M(T_k)\ell_j = 
\begin{bmatrix}
    p_{k}^{jg} \\
    \varrho_j \\
    n_{n_j}^i
\end{bmatrix}
\]  

(4.49)
4.5. INTENSE BUNDLE ADJUSTMENT

where

\[
M(T) = \begin{bmatrix}
T & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & C
\end{bmatrix},
\]

(4.50)

where \( C \) is the rotation matrix embedded in the top left corner of \( T \) and allows us to transform all landmark parameters (including reflectivity and surface normal) to the sensor frame. Note the point and surface normal subscripts have changed to \( k \) from \( g \) as they are now transformed to the sensor frame at time-step \( k \). The second function, \( s(\cdot) \), is transforming the landmarks in the camera frame to camera measurements. Together we have:

\[
g(x_{jk}) = s(z(x_{jk})).
\]

(4.51)

The second function, \( s(\cdot) \), depends on the sensor used and which additional measurements are included (e.g., surface normals and/or intensity), so it is designed in a modular fashion and presented in the following subsections.

3D Measurements from Scanning LiDAR

Scanning LiDAR measurements are usually obtained by using a photo-diode to directly measure the time of flight of a laser pulse and its amplitude. Depending on how the laser and photo-diode are mounted on the sensor, their angle of elevation is usually constant and their angle of azimuth is measured with an encoder. With the measured time of flight, the distance the light travelled can be computed and the sensor typically reports the return as an azimuth, elevation, and range.

Scanning LiDAR observe a landmark represented by a homogeneous point \( p \in \mathbb{R}^3 \)
4.5. INTENSE BUNDLE ADJUSTMENT

Figure 4.5: A landmark is measured by a ToF camera with an attached reference frame $F_{sk}$ and a pinhole camera image plane $F_{ck}$ according to:

$$s_L(\ell) = \begin{bmatrix} \alpha \\ \beta \\ r \end{bmatrix} = \begin{bmatrix} \text{atan}(p_y, p_x) \\ \text{atan} \left( p_z, \sqrt{p_x^2 + p_y^2} \right) \\ \|E^T \mathbf{p}\| \end{bmatrix}$$ (4.52)

3D Measurements from ToF Camera

ToF camera data are typically reported as range values and 2D pixel locations as shown in Figure 4.5. As long as the image has been corrected for any spherical or tangential distortions, the pixel locations in a 2D image are equivalent to the elevation and bearing angles in scanning LiDAR measurements. ToF cameras take an entire image at once, whereas scanning LiDAR performs more like a rolling shutter camera, taking each column in the image sequentially. This can produce artifacts like motion blur. The trade-off usually becomes one of higher accuracy but low density images (scanning LiDAR) versus lower accuracy but dense images (ToF cameras). In either case, (4.13) can be restated as a function of the sensor and landmark states, forming an observation model. The measurements of a time-of-flight camera are taken with a CCD image sensor that has been adapted to additionally compute the time-of-flight...
4.5. INTENSE BUNDLE ADJUSTMENT

of each signal a pixel receives as discussed in Section 4.2.1. Therefore the idealized pinhole camera model is used with the addition of a range measurement at each pixel location. A homogeneous point \( p \in \mathbb{R}^3 \) can be transformed to pixel measurements by first computing the normalized image coordinates

\[
x_n = \frac{p_x}{p_z}, \tag{4.53}
\]

\[
y_n = \frac{p_y}{p_z}. \tag{4.54}
\]

These coordinates correspond to an idealized pinhole camera, with unit focal length and an origin that is intersected by the camera optical axis. Because actual cameras do not have a unit focal length and by convention image origin is typically located in the top left corner of the image plane, we can find the corresponding camera image coordinates, \([u, v]^T\), through

\[
\begin{bmatrix}
u \\
v \\cdot 1
\end{bmatrix} = \begin{bmatrix}
f_x & 0 & c_x \\
0 & f_y & c_y \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
x_n \\
y_n \\
1
\end{bmatrix}, \tag{4.55}
\]

where \( f_x \) and \( f_y \) are the focal length expressed in pixel units in the horizontal and vertical directions respectively, and \( c_u \) and \( c_v \) are the offset of the image origin from the intersection of the optical axis, also in horizontal and vertical pixel units. All together, these parameters are known as the intrinsic parameters of the camera and the matrix \( K \) is called the intrinsic parameter matrix.
4.5. INTENSE BUNDLE ADJUSTMENT

![Diagram of intensity measurements](image)

Figure 4.6: Intensity measurements are a function of landmark parameters. Note that the viewing direction is the unit vector pointing from the landmark towards the sensor. This vector can be computed from the landmark point coordinates in the sensor frame.

The range observation model is simply the norm of $p$, and combining them together we have

$$
\begin{bmatrix}
u \\
v \\
r
\end{bmatrix} = s_T(\ell) = \begin{bmatrix} \frac{1}{p_z} KE^T p \\ \sqrt{\parallel E^T p \parallel} \end{bmatrix}. \tag{4.56}
$$

Note that the parameters in $K$ are either provided by the manufacturer or are determined through calibration. In addition, typically camera images exhibit some type of lens distortion that should also be corrected before using this model. The calibration procedure to determine all intrinsic parameters is discussed in Section 6.1.

Intensity and Surface Normal Measurements

For both scanning LiDAR and ToF cameras, it is possible to incorporate additional measurements in the observation model. The intensity model described in (4.13) can be reformulated in terms of the parameters we wish to estimate, as seen in Figure 4.6. Revisiting the model for intensity,
4.5. INTENSE BUNDLE ADJUSTMENT

\[ i = \eta \frac{\cos \alpha}{r^2}, \]  

(4.57)

where \( \alpha \) is the angle of incidence between the surface normal, \( \mathbf{n} \), and the viewing direction of the sensor. The scaling parameter, \( \eta \), includes the conversion between irradiance received by the sensor and the final intensity measurement reported by the sensor which is determined through calibration\(^1\).

The dot product of two unit vectors is the cosine of the angle between them and the distance between a point and the origin is the norm of its coordinates, therefore we can determine both quantities:

\[
\cos \alpha = \mathbf{n}^T \left( -\frac{\mathbf{E}^T \mathbf{p}}{\| \mathbf{E}^T \mathbf{p} \|} \right), \quad r = \| \mathbf{E}^T \mathbf{p} \| 
\]  

(4.58)

Substituting (4.58) into (4.57), a sensor that observes the landmark \( \mathbf{\ell} \in \mathbb{R}^3 \times [0 \ 1] \times S^2 \) has an intensity observation model such that

\[
i = s_I(\mathbf{\ell}) = -\eta \frac{\mathbf{n}^T (\mathbf{E}^T \mathbf{p})}{\| \mathbf{E}^T \mathbf{p} \|^3}.
\]  

(4.59)

It is also possible to include surface normal observations in our estimation problem. Surface normal observations are computed from a landmark and its neighbouring points. A full explanation of this processing step is included in Appendix A. These are observations of the surface normal in the sensor frame, denoted \( \mathbf{\hat{n}} \), and therefore the observation model is straightforward:

\[
\mathbf{\hat{n}} = s_N(\mathbf{\ell}) = \mathbf{n} 
\]  

(4.60)

\(^1\)A description of the calibration procedure as well as results for a SwissRanger 4000 ToF camera are given in Chapter 6.4.
4.5. INTENSE BUNDLE ADJUSTMENT

4.5.3 Log-likelihood of IntenseBA Observations

With each observation model defined, the observation errors can be constructed according to (4.37) as well as the objective function to be minimized, shown in (4.38). The optimal estimate of $x$ is the one that minimizes the objective function. The possible observation errors for the $k$-th pose and the $j$-th landmark are then

\[ e_{L}(x_{jk}) = y_{Ljk} - s_{L}(z(x_{jk})), \]  
\[ e_{T}(x_{jk}) = y_{Tjk} - s_{T}(z(x_{jk})), \]  
\[ e_{I}(x_{jk}) = y_{Ijk} - s_{I}(z(x_{jk})), \]  
\[ e_{N}(x_{jk}) = y_{Njk} \boxplus s_{N}(z(x_{jk})). \]

(4.61a)  
(4.61b)  
(4.61c)  
(4.61d)

Each error in (4.61) is the difference between the observation and the expected observation given $x_{jk}$. Note that (4.61d) is the difference between two surface normals $e_{Njk}(x) \in h^2$ and thus the $\boxplus$ operation defined in Section 3.3.1 is used to compute it.

Depending on the sensor used, either $e_{Ljk}$ (scanning LiDAR) or $e_{Tjk}$ (ToF camera) are used. The corresponding log-likelihoods computed according to (4.32) for the current estimate $\hat{x}_{jk}$, are

\[ \ln \left( f \left( y_{Ljk} | \hat{x}_{jk}, R_{Ljk} \right) \right) = -\frac{1}{2} e_{Ljk}(\hat{x}_{jk})^{T} R_{Ljk}^{-1} e_{Ljk}(\hat{x}_{jk}), \]  
\[ \ln \left( f \left( y_{Tjk} | \hat{x}_{jk}, R_{Tjk} \right) \right) = -\frac{1}{2} e_{Tjk}(\hat{x}_{jk})^{T} R_{Tjk}^{-1} e_{Tjk}(\hat{x}_{jk}), \]  
\[ \ln \left( f \left( y_{Ijk} | \hat{x}_{jk}, R_{Ijk} \right) \right) = -\frac{1}{2} e_{Ijk}(\hat{x}_{jk})^{T} R_{Ijk}^{-1} e_{Ijk}(\hat{x}_{jk}), \]  
\[ \ln \left( f \left( y_{Njk} | \hat{x}_{jk}, R_{Njk} \right) \right) = -\frac{1}{2} e_{Njk}(\hat{x}_{jk})^{T} R_{Njk}^{-1} e_{Njk}(\hat{x}_{jk}). \]

(4.62a)  
(4.62b)  
(4.62c)  
(4.62d)
4.5. INTENSE BUNDLE ADJUSTMENT

The overall objective function in (4.32) can now be constructed. Because each observation is considered conditionally independent, the overall joint likelihood is simply the product of each observation type’s joint likelihood. Therefore the overall objective function across all intenseBA observations using scanning LiDAR \((J_L)\) or ToF cameras \((J_T)\) is

\[
J_L(x) = \frac{1}{2} \sum_j \sum_k (e_{Ljk}(x_{jk}))^T R_{Ljk}^{-1} e_{Ljk}(x_{jk}) + e_{Ijk}(x_{jk})^T R_{Ijk}^{-1} e_{Ijk}(x_{jk}) \tag{4.63}
\]

\[
J_T(x) = \frac{1}{2} \sum_j \sum_k (e_{Tjk}(x_{jk}))^T R_{Tjk}^{-1} e_{Tjk}(x_{jk}) + e_{Ijk}(x_{jk})^T R_{Ijk}^{-1} e_{Ijk}(x_{jk}) \tag{4.64}
\]

Moving forward, we drop consideration of the scanning LiDAR observation model as it is treated in an identical fashion as the ToF camera.

Because each of our estimates is uncertain, and represented by a Gaussian, we must take care when propagating them through our nonlinear camera model to compute the errors in the objection function. Usually this is done through linearization, most commonly through a first-order Taylor series expansion. Other methods include sigma-point transformations and further expansion of the Taylor series to include additional higher order terms. In this thesis we use a Taylor series expansion of the observation model to first-order. The observation model is of the form

\[
y_{jk} = g(x_{jk}) + v_{jk}, \tag{4.65}
\]
where $g(\cdot)$ is the nonlinear model and $v_{jk} \sim \mathcal{N}(0, R_{jk})$ is Gaussian noise.

### 4.5.4 Linearizing IntenseBA Observation Errors

As discussed in Section 4.3, the minimization of the objective function requires the observation error to be differentiated with respect to the design parameter, $x$. However, because $x$ has more parameters than degrees of freedom, the differentiation must be done with respect to changes in the degrees of freedom to avoid a singular Hessian in our objective function. The topological space of $x$ is

$$
S := SE(3) \times \cdots \times SE(3) \times \mathbb{R}^3 \times [0 \ 1] \times S^2 \cdots \mathbb{R}^3 \times [0 \ 1] \times S^2.
$$

(4.66)

Recalling in Section 3.2.2 and Section 3.3, the unconstrained parameterization of $x$ is determined with the corresponding $\ln(\cdot)$ operation for transformations and surface normals respectively. Using these parameterizations, we define a new topological space of $\mathcal{R} \subset \mathbb{R}^{KM}$

$$
\mathcal{R} := se(3) \times \cdots \times se(3) \times \mathbb{R}^3 \times [0 \ 1] \times h^2 \times \cdots \times \mathbb{R}^3 \times [0 \ 1] \times h^2
$$

(4.67)

This is the topological space we take the partial derivative of $e(x_{jk})$ with respect to, and a perturbation to $x_{jk}$ exists in a subset of this space. Analytically calculating the partial derivatives of the observation errors with respect to $x$ is not always practical, for this reason the partial derivatives are calculated numerically. To do this, we must perturb the current estimate $\hat{x}_{jk}$ by a small amount in each degree of freedom. Let
the perturbation of the $j$-th pose and the $k$-th landmark be

$$
\delta x_{jk} = \begin{bmatrix}
\delta \nu_k \\
\delta \phi_k \\
\delta \zeta_j \\
\delta \gamma_j \\
\delta \psi_j
\end{bmatrix}
\in \mathfrak{se}(3) \times \mathbb{R}^3 \times [0, 1] \times h^2.
$$

(4.68)

Given an nominal value of $x_{jk}$, it can be perturbed by $\delta x_{jk}$ using the $\boxplus$ operator:

$$
x_{jk} = x_{jk} \boxplus \delta x_{jk},
$$

(4.69)

which applies each element of the perturbation corresponding to the topological space it exists in and the definitions in Chapter 2.

In addition, while most of the observation errors are of the form: $e(x) = y - s(z(x))$ and exist in $\mathbb{R}^n$, the error function of surface normals, $e_N(\cdot) = y \boxminus s_N$, exists in the topological space of $h^2$.

The Jacobian of $e(x)$ with respect to $x_{jk}$ is constructed with twelve columns because there are six degrees of freedom in the pose and six degrees of freedom in the landmark i.e.,

$$
G_{jk} = \left. \frac{\partial e(x_{jk})}{\partial x_{jk}} \right|_x \approx [a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ b_1 \ b_2 \ b_3 \ b_4 \ b_5 \ b_6].
$$

(4.70)

To compute each element, a small perturbation $\epsilon \in \mathbb{R}$ is applied in each degree of freedom separately where we have denoted the unconstrained parameterization of a
transformation \( \xi = (\nu_1, \nu_2, \nu_3, \phi_1, \phi_2, \phi_3)^T \), and the unconstrained parameterization of a landmark

\[
\delta \ell = [\zeta_1, \zeta_2, \zeta_3, \gamma, \psi_1, \psi_2]^T,
\]

i.e.,

\[
\begin{align*}
a_1 &= \frac{\partial e(x_{jk})}{\partial \nu_1} \bigg|_{x_{jk}} = \frac{e(x_{jk} \oplus (\epsilon J)_1) - e(x_{jk} \oplus (-\epsilon J)_1)}{2\epsilon} \\
a_2 &= \frac{\partial e(x_{jk})}{\partial \nu_2} \bigg|_{x_{jk}} = \frac{e(x_{jk} \oplus (\epsilon J)_2) - e(x_{jk} \oplus (-\epsilon J)_2)}{2\epsilon} \\
a_3 &= \frac{\partial e(x_{jk})}{\partial \nu_3} \bigg|_{x_{jk}} = \frac{e(x_{jk} \oplus (\epsilon J)_3) - e(x_{jk} \oplus (-\epsilon J)_3)}{2\epsilon} \\
a_4 &= \frac{\partial e(x_{jk})}{\partial \phi_1} \bigg|_{x_{jk}} = \frac{e(x_{jk} \oplus (\epsilon J)_4) - e(x_{jk} \oplus (-\epsilon J)_4)}{2\epsilon} \\
a_5 &= \frac{\partial e(x_{jk})}{\partial \phi_2} \bigg|_{x_{jk}} = \frac{e(x_{jk} \oplus (\epsilon J)_5) - e(x_{jk} \oplus (-\epsilon J)_5)}{2\epsilon} \\
a_6 &= \frac{\partial e(x_{jk})}{\partial \phi_3} \bigg|_{x_{jk}} = \frac{e(x_{jk} \oplus (\epsilon J)_6) - e(x_{jk} \oplus (-\epsilon J)_6)}{2\epsilon} \\
b_1 &= \frac{\partial e(x_{jk})}{\partial \zeta_1} \bigg|_{x_{jk}} = \frac{e(x_{jk} \oplus (\epsilon J)_7) - e(x_{jk} \oplus (-\epsilon J)_7)}{2\epsilon} \\
b_2 &= \frac{\partial e(x_{jk})}{\partial \zeta_2} \bigg|_{x_{jk}} = \frac{e(x_{jk} \oplus (\epsilon J)_8) - e(x_{jk} \oplus (-\epsilon J)_8)}{2\epsilon} \\
b_3 &= \frac{\partial e(x_{jk})}{\partial \zeta_3} \bigg|_{x_{jk}} = \frac{e(x_{jk} \oplus (\epsilon J)_9) - e(x_{jk} \oplus (-\epsilon J)_9)}{2\epsilon} \\
b_4 &= \frac{\partial e(x_{jk})}{\partial \gamma} \bigg|_{x_{jk}} = \frac{e(x_{jk} \oplus (\epsilon J)_{10}) - e(x_{jk} \oplus (-\epsilon J)_{10})}{2\epsilon} \\
b_5 &= \frac{\partial e(x_{jk})}{\partial \psi_1} \bigg|_{x_{jk}} = \frac{e(x_{jk} \oplus (\epsilon J)_{11}) - e(x_{jk} \oplus (-\epsilon J)_{11})}{2\epsilon} \\
b_6 &= \frac{\partial e(x_{jk})}{\partial \psi_2} \bigg|_{x_{jk}} = \frac{e(x_{jk} \oplus (\epsilon J)_{12}) - e(x_{jk} \oplus (-\epsilon J)_{12})}{2\epsilon}
\end{align*}
\]

where \((J)_n\) is the \(n\)th column of the identity matrix. With the Jacobian \(G_{jk}\) constructed from these columns it is possible to linearize our observation error through
4.6. SPARSE BUNDLE ADJUSTMENT

A first-order Taylor series expansion

\[ e(\hat{x} \oplus \delta x) \approx e(\hat{x}) + G_{jk} \delta x \]  

which is linear with respect to \( \delta x \). This linearization is used to apply the perturbations generated in the Gauss-Newton algorithm discussed in Section 4.4. The next section reviews this and discusses some practical adjustments to the algorithm that result in solving the problem more quickly.

4.6 Sparse Bundle Adjustment

A version of BA named Sparse Bundle Adjustment (SBA) is used to carry out the Gauss-Newton algorithm described in Section 4.4 to solve for the perturbed quantities, \( \delta x \), but with some additions that allow it to run more efficiently. First, Levenberg-Marquardt (LM) is used [70] to converge quickly even when the initial guess for the estimated parameters is far from the solution. Second, it can be shown that solving for \( \delta x \) has complexity \( O((K + M)^3) \) in the number of unknown parameters [40]. It is apparent from this that one of the drawbacks of BA is that the computation of a solution grows very quickly as the number of landmarks in the map grows. SBA reduces the impact of this by taking advantage of the sparse structure of the problem.

Recalling the derivation of the Gauss-Newton algorithm in Section 4.4, to minimize the linearized objective function \( J(\hat{x} \oplus \delta x) \) its derivative is taken and set equal to zero. Rearranging this we end up with

\[ H\delta x^* = -b. \]  (4.74)
where $\delta \mathbf{x}^*$ is the optimal perturbation of the linearized objective function and

$$\mathbf{H} = \sum_{jk} \mathbf{P}_{jk}^T \mathbf{G}_{jk}^T \mathbf{R}_{jk}^{-1} \mathbf{G}_{jk} \mathbf{P}_{jk}, \quad \mathbf{b} = \sum_{jk} \mathbf{P}_{jk}^T \mathbf{G}_{jk}^T \mathbf{R}_{jk}^{-1} \mathbf{e}(\bar{\mathbf{x}}_{jk}) \quad (4.75)$$

LM's key innovation is solving a slightly different set of equations that introduce a damping parameter, $\mu > 0$

$$(\mathbf{H} + \mu \mathbf{1}) \delta \mathbf{x}^* = -\mathbf{b}. \quad (4.76)$$

If the updated value for $\hat{\mathbf{x}}$ leads to a reduction in $\mathbf{e}(\hat{\mathbf{x}})^T \mathbf{e}(\hat{\mathbf{x}})$, the update is accepted, $\mu$ is decreased, and the algorithm moves onto the next iteration. On the other hand if the error increases, the update is not accepted, $\mu$ is increased, and the algorithm repeats until a value for $\delta \mathbf{x}$ that leads to a reduction in error is found. It is apparent that if $\mu$ is large the solution is found along the steepest descent direction and if $\mu$ is small, LM approximates Gauss-Newton minimization. This allows LM to adaptively alternate between a slow descent approach when far from the minimum and fast convergence when near the minimum. The convergence conditions and the amount $\mu$ is changed follow the heuristics introduced by Neilsen [71].

SBA takes advantage of the regular sparse structure of the equations in (4.74). Recall that the Jacobian $\mathbf{G}_{jk}$ can be separated into two parts, one with respect to the sensor pose parameters and one with respect to the landmark parameters as shown in (4.70). Let

$$\mathbf{A}_{jk} = [a_{1,jk}, a_{2,jk}, a_{3,jk}, a_{4,jk}, a_{5,jk}, a_{6,jk}], \quad (4.77)$$

$$\mathbf{B}_{jk} = [b_{1,jk}, b_{2,jk}, b_{3,jk}, b_{4,jk}, b_{5,jk}, b_{6,jk}] \quad (4.78)$$
4.6. SPARSE BUNDLE ADJUSTMENT

such that \( G_{jk} = [A_{jk} | B_{jk}] \). Looking at the complete Jacobian \( G \) structure we see

\[
G = \begin{bmatrix}
A_{11} & & & & \allowbreak & \allowbreak & B_{10} \\
& A_{21} & & & \allowbreak & \allowbreak & B_{20} \\
& & \ddots & & \allowbreak & \allowbreak & \ddots \\
& & & \ddots & \allowbreak & \allowbreak & \ddots \\
& & & & A_{1K} & \allowbreak & B_{1K} \\
& & & & & \ddots & \ddots \\
& & & & & & A_{MK} & B_{MK}
\end{bmatrix}
\]

where empty spaces represent empty elements of the matrix and shows the sparsity of \( G \). We can exploit this structure by multiplying out the left side of (4.74), \( H = \sum_{jk} P_{jk}^T G_{jk}^T R_{jk}^{-1} G_{jk} P_{jk} \), and grouping elements in the matrix according to

\[
H = \begin{bmatrix}
U & W \\
W^T & V
\end{bmatrix}
\]
where

\[ U_i \equiv \sum_{j=1}^{M} A^T_{jk} R^{-1}_{jk} A_{jk}, \]  
\[ V_j \equiv \sum_{k=0}^{K} B^T_{jk} R^{-1}_{jk} A_{jk}, \]  
\[ W_{jk} \equiv A^T_{jk} R^{-1}_{jk} B_{jk}, \]

which allows us to express \( H \) as

\[
H = \begin{bmatrix}
U_1 & W_{11} & W_{12} & \cdots & W_{1K} \\
U_2 & W_{21} & W_{22} & \cdots & W_{2K} \\
& \ddots & \ddots & \cdots & \ddots \\
U_K & W_{1K} & W_{2K} & \cdots & W_{MK} \\
W^T_{11} & W^T_{12} & \cdots & W^T_{1K} & V_1 \\
W^T_{21} & W^T_{22} & \cdots & W^T_{2K} & V_2 \\
& \ddots & \ddots & \cdots & \ddots \\
W^T_{M1} & W^T_{M2} & \cdots & W^T_{MK} & V_K
\end{bmatrix},
\]

that reveals the special sparsity pattern of the Hessian of the objective function, sometimes referred to as an *arrowhead* matrix. Note that \( U \) and \( V \) are block diagonal, due to each observation involving just one landmark and one pose. This sparsity can
be used to solve $\delta x^*$ very efficiently. Let

$$\alpha_k \equiv \sum_{j=1}^{M} A_{jk}^T R_{jk}^{-1} e(\hat{x}_{jk}), \quad (4.85)$$

$$\beta_j \equiv \sum_{k=1}^{K} B_{jk}^T R_{jk}^{-1} e(\hat{x}_{jk}), \quad (4.86)$$

then $b$ can be expressed as

$$b = \begin{bmatrix} \alpha_1 \\
\vdots \\
\alpha_K \\
\beta_1 \\
\vdots \\
\beta_M \end{bmatrix} = \begin{bmatrix} \alpha \\
\beta \end{bmatrix}. \quad (4.88)$$

By augmenting the diagonal elements of $U$ and $V$ with the damping factor, (4.76) is compactly written as

$$\begin{bmatrix} U & W \\
W^T & V \end{bmatrix} \delta x^* = \begin{bmatrix} U & W \\
W^T & V \end{bmatrix} \begin{bmatrix} \delta \xi \\
\delta \ell \end{bmatrix} = - \begin{bmatrix} \alpha \\
\beta \end{bmatrix}. \quad (4.89)$$

The Schur complement can be used to manipulate (4.89) into a more easily solved form. By multiplying both sides by

$$\begin{bmatrix} 1 & -WV^{-1} \\
0 & 1 \end{bmatrix}$$
we have
\[
\begin{bmatrix}
U - \mathbf{WV}^{-1}\mathbf{W}^T & 0 \\
\mathbf{W}^T & \mathbf{V}
\end{bmatrix}
\begin{bmatrix}
\delta \boldsymbol{\xi} \\
\delta \ell
\end{bmatrix} =
\begin{bmatrix}
\alpha - \mathbf{WV}^{-1}\beta \\
\beta
\end{bmatrix}.
\] (4.90)

Here we can see that the dependence of the sensor parameters on the landmark parameters has been removed, and that they can be determined from the top half of (4.90) quickly since \( \mathbf{V} \) is block diagonal and therefore \( \mathbf{V}^{-1} \) is easy to compute. Substituting the solution into the lower half allows the landmark parameters to be solved quickly, again due to \( \mathbf{V} \) being block diagonal. SBA has a complexity of \( O(K^3 + K^2 M) \) for each solution [26], a large improvement over the \( O((K+M)^3) \) of traditional BA. The pseudo-code for the IntenseBA algorithm is provided in Algorithm 1.

**Algorithm 1:** A summary of the Intense Bundle Adjustment algorithm.

**Require:** Initial estimate \( \hat{\mathbf{x}} \), observations \( \{ \mathbf{y}, \mathbf{R} \} \)

**output:** Optimal estimate \( \hat{\mathbf{x}}^* \)

1. for all observations \( \{ \mathbf{y}_{jk}, \mathbf{R}_{jk} \} \) do
2. Compute observation error \( \mathbf{e}_{jk} = e(\hat{\mathbf{x}}_{jk}) \)
3. Compute Jacobian \( \mathbf{G}_{jk} = \frac{\partial e(\mathbf{x})}{\partial \mathbf{x}} \bigg|_{\hat{\mathbf{x}}_{jk}} \)
4. end
5. Compute \( \mathbf{H} = \sum_{jk} \mathbf{P}_{jk}^T \mathbf{G}_{jk}^T \mathbf{R}_{jk}^{-1} \mathbf{G}_{jk} \mathbf{P}_{jk} \)
6. Compute \( \mathbf{b} = \sum_{jk} \mathbf{P}_{jk}^T \mathbf{G}_{jk}^T \mathbf{R}_{jk}^{-1} \mathbf{e}_{jk} \)
7. while \( \hat{\mathbf{x}} \) not converged according to [71] do
8. Solve \( (\mathbf{H} + \mu \mathbf{I})\delta \mathbf{x}^* = -\mathbf{b} \) for \( \delta \mathbf{x}^* \)
9. if \( J(\hat{\mathbf{x}} \oplus \delta \mathbf{x}^*) < J(\hat{\mathbf{x}}) \) then
10. \( \hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} \oplus \delta \mathbf{x}^* \)
11. Recompute \( \mathbf{e}_{jk} \) and \( \mathbf{G}_{jk} \) for all observations
12. Recompute \( \mathbf{b} \) and \( \mathbf{H} \)
13. end
14. adjust \( \mu \) according to [71]
15. end
16. \( \hat{\mathbf{x}}^* = \hat{\mathbf{x}} \)
Part II

Applications
Chapter 5

Observability and Sensitivity

This chapter takes the IntenseBA algorithm discussed in the previous chapter and determines if the problem is observable (i.e., is there enough information to estimate each of the states given a minimal set of observations) and how sensitive it is to each of the observation types and their noise magnitudes. To do this an observability analysis is conducted that suggests the problem can be solved mathematically and a simulation is presented that verifies this and analyzes the sensitivity of the problem to different noise magnitudes.

5.1 Observability Analysis

This section provides an observability analysis of IntenseBA with a ToF camera that measures range and intensity at each pixel location, and computes a surface normal measurement for each landmark. This is a contribution of the thesis and was originally published in [72]. First, the system’s observability when incorporating only pixel coordinates and range is analyzed, with intensity and surface normal measurements added to investigate how this affects observability.

A sufficient condition of local weak observability [73] is that the observability
5.1. OBSERVABILITY ANALYSIS

rank condition is satisfied. Local observability is a stronger condition than global observability [74]. The system is locally weakly observable if it is “possible to instantaneously distinguish (the states) from their neighbors (in the local state space) for all possible states”. IntenseBA satisfies this condition if any of the observability matrices are of rank $6m + 3n$ for traditional BA, and rank $6m + 6n$ for IntenseBA which includes the reflectivity and surface normal measurements discussed in Section 4.5. The traditional BA problem analysis includes only the position of landmarks and is estimated with the measurements in (4.61b). A similar analysis, using the augmented states and errors in (4.61), is conducted and the observability of both are compared. The approach used here has been extended from [74], which is about 2D SLAM and from [75] which extends these concepts to 3D SLAM.

For a nonlinear function, such as $g(x)$, the observability matrix is formed such that the elements in this matrix are any combination of the repeated Lie derivatives of the components of the measurement model $g(x)$, and the process model $f(x, u)$ that describes how the system states change with time given an input $u$. An observability matrix is composed by using any combination of successive Lie derivatives of any of the components of the measurement model. The Lie derivatives are defined recursively as

\[
L^0_f dg(x) = \frac{\partial g(x)}{\partial x},
\]

\[
L^1_f dg(x) = L^0_f dg(x) \frac{\partial f(x, u)}{\partial x} + \left[ \frac{\partial}{\partial x} \left( L^0_f dg(x) \right)^T f(x, u) \right]^T.
\]

Because BA drops the motion model, $f = 0_{(6m+3n) \times 1}$, only the first Lie derivative
5.1. OBSERVABILITY ANALYSIS

(i.e., the Jacobian of the observation model) is relevant. However for a system that incorporates measurements of motion (e.g., an IMU), further Lie derivatives would be computed. For traditional SLAM, the observability matrix is simply any combination of $6m + 3n$ rows that are taken from any of the successive Lie derivatives. If any of these possible observability matrices can be shown to be full rank, then the system is locally weakly observable.

For example, a system with only one landmark ($M = 1$) and one sensor pose ($K = 1$) has a state vector with 9 degrees of freedom and 3 observations. A potential observability matrix for this system is

$$O = \begin{bmatrix}
L_F^0 dg_1(x) \\
L_F^0 dg_2(x) \\
L_F^0 dg_3(x) \\
\vdots \\
L_F^2 dg_1(x) \\
L_F^2 dg_2(x) \\
L_F^2 dg_3(x) \\
L_F^3 dg_1(x)
\end{bmatrix},$$

where $g_n$ refers to the $n$-th row of the observation model output. However, because only the first Lie derivative is non-trivial, we can see right away that this system is unobservable and has at most a rank equal to 3. To determine the minimum number of estimated sensor poses and landmarks needed to construct an observability matrix
5.1. OBSERVABILITY ANALYSIS

we can set up the inequality

\[ 6K + 3M \leq 3KM, \]  

(5.4)

which shows that this always requires at least two sensor poses. For two sensor poses, we require at least four landmarks to satisfy inequality (5.4) and for any number of poses greater than two we require at least three landmarks. With two sensor poses and four landmarks, only one observability matrix exists that includes all possible Lie derivatives

\[ O = (O_{1,1}, O_{1,2}, \ldots, O_{2,4}), \]  

(5.5)

where

\[ O_{jk} = \begin{bmatrix} L_{F}^{0}dg_{1}(x_{jk}) \\ L_{F}^{0}dg_{2}(x_{jk}) \\ L_{F}^{0}dg_{3}(x_{jk}) \end{bmatrix}, k = 0, \ldots, 1 \text{ and } j = 1, \ldots, 4. \]  

(5.6)

We determined the rank of \( O \) by using MATLAB’s Symbolic Toolbox, which implements Gaussian elimination. In the case of two robot poses (\( K = 2 \)) and four landmarks (\( M = 4 \)), \( O \) must have a rank of \( 6K + 3M = 24 \). However, carrying out the analysis it can be shown that \( \text{rank}(O) = 18 \) which is a rank deficiency of six. This is a well known problem in the SLAM community [74], and it can be shown that no matter how many landmarks or sensor poses there are, \( O \) is always rank deficient. This is due to the inability to observe the global frame, \( F_{g} \). To alleviate this, the initial sensor pose can be fixed at the origin of \( F_{g} \). This reduces the minimum rank
requirement to $6(K - 1) + 3M$ and carrying out the same operations reveals that the problem is indeed locally weakly observable.

By using the augmented observation model that includes intensity and surface normal measurements, there are now additional landmark states to estimate that have three degrees of freedom ($\varrho \in [0, 1]$ and $\mathbf{n} \in S^2$). Here we devise three types of “augmented” BA that all still make use of the standard ToF camera measurements (pixel coordinates and range) but also use augmented measurements: (1) one that uses only the first augmented measurement $i$ and estimates each landmark’s associated $\mathbf{n}$ and $\varrho$; (2) one that only uses the second augmented measurement, $\mathbf{\tilde{n}}$, and estimates $\mathbf{n}$; and (3) one that uses both augmented measurements and estimates $\mathbf{n}$ and $\varrho$. The Lie derivatives are computed as before to construct the observability matrix. We can determine the minimum number of estimated sensor poses and landmarks needed to construct an observability matrix by the inequality

$$6K + 6M \leq 6MK. \quad (5.7)$$

Thus, at minimum the problem requires two sensor poses and two landmarks. Immediately we see a difference from traditional BA, with only half the landmarks needed to construct the observability matrix. The rank deficiency for this problem can be shown to be three, a reduction from six in traditional BA with four landmarks. If the initial robot pose is fixed to the origin of $\mathcal{F}_g$, the problem becomes fully observable. This has some encouraging properties over traditional BA. We are now estimating the surface normal and reflectivity of the landmarks in our map, useful properties for scientific analysis and landmark recognition. Additionally, the problem is observable with fewer landmarks, which means it may perform better when the number of visible
Table 5.1
Minimum landmarks and views to construct $O$, brackets indicate value when first frame is fixed

<table>
<thead>
<tr>
<th>Measurements</th>
<th>Dim. of $\ell$</th>
<th>$M$ for 2 views</th>
<th>$M$ for 3 views</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g = (u, v, r)$</td>
<td>$3M$</td>
<td>4(2)</td>
<td>3(2)</td>
</tr>
<tr>
<td>$g = (u, v, r, i)$</td>
<td>$6M$</td>
<td>6(3)</td>
<td>3(2)</td>
</tr>
<tr>
<td>$g = (u, v, r, \bar{\mathbf{n}})$</td>
<td>$5M$</td>
<td>3(2)</td>
<td>2(2)</td>
</tr>
<tr>
<td>$g = (u, v, r, i, \bar{\mathbf{n}})$</td>
<td>$6M$</td>
<td>2(1)</td>
<td>2(1)</td>
</tr>
</tbody>
</table>

Table 5.2
Rank deficiency with minimum number of landmarks, brackets indicate value when first frame is fixed

<table>
<thead>
<tr>
<th>Measurements</th>
<th>2 views</th>
<th>3 views</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g = (u, v, r)$</td>
<td>6(1)</td>
<td>6(2)</td>
</tr>
<tr>
<td>$g = (u, v, r, i)$</td>
<td>12(3)</td>
<td>6(2)</td>
</tr>
<tr>
<td>$g = (u, v, r, \bar{\mathbf{n}})$</td>
<td>6(0)</td>
<td>6(0)</td>
</tr>
<tr>
<td>$g = (u, v, r, i, \bar{\mathbf{n}})$</td>
<td>6(1)</td>
<td>6(2)</td>
</tr>
</tbody>
</table>

landmarks is reduced, which is typical for low resolution sensors like scanning LiDAR and ToF cameras.

The conditions needed to construct an observability matrix with two or three separate view points for the sensor are shown in Table 5.1 for all types of BA discussed previously. The rank deficiency of the problem is shown for the same set of circumstances in Table 5.2.

Interestingly, the minimum number of landmarks needed for the different types of measurement vectors is the last case, which includes all measurements. By fixing the first frame, an observability matrix with just one landmark can be constructed.

As shown in Table 5.2, the BA problem is only full rank when fixing the first frame for the nearest neighbor surface normal BA (third row). The other problems
5.1. OBSERVABILITY ANALYSIS

Figure 5.1: Rank deficiency for different number of poses and landmarks for traditional BA.
5.1. OBSERVABILITY ANALYSIS

Figure 5.2: Rank deficiency for different number of poses and landmarks for nearest neighbor surface normal augmented BA
5.1. OBSERVABILITY ANALYSIS

Figure 5.3: Rank deficiency for different number of poses and landmarks for intensity augmented BA
5.1. OBSERVABILITY ANALYSIS

Figure 5.4: Rank deficiency for different number of poses and landmarks for all measurements
become full rank as we add viewpoints or landmarks to the problem. We can see where the problem becomes full rank in Figures 5.1-5.4. In the case of traditional BA, this corresponds to 3 landmarks at 2 viewpoints, for intensity augmented BA, this corresponds to 3 landmarks at 3 viewpoints, and when using all types of measurements it will take 2 landmarks at 2 viewpoints. In summary, by estimating properties of the surface normals and reflectivity, BA is observable under less strict requirements and we gain additional useful information about the surface normal and infrared reflectivity of each landmark compared to traditional BA.

5.2 Simulation and Sensitivity Analysis

To verify the validity of the IntenseBA and test its sensitivity to observation noise a simulation was implemented. The goal of this simulation is to test the ability of the algorithm to incorporate these additional measurements in the bundle adjustment problem (i.e., verify the problem is observable), but also to analyze its sensitivity to sensor noise. For example, there may be little benefit over traditional methods to the pose estimation problem if there are many landmarks and the sensor used has low range measurement noise, however through simulation we can possibly estimate at what point it becomes beneficial to incorporate these measurements by varying parameters such as measurement noise and the number of estimated landmarks. This section describes the simulation and presents the results of a sensitivity analysis.

5.2.1 Simulation Setup

The simulation assumes known data association and for each sensor pose all landmarks are visible and measured. A ground plane is set in the global frame, \( \mathcal{F}_g \), at \( z = 0 \)
Figure 5.5: The simulated rover is placed on a ground plane with boundaries \((x_{\text{min}}, x_{\text{max}})\) and \((y_{\text{min}}, y_{\text{max}})\). The sensor is mounted 1.2 metres off the ground plane and tilted 18° below the horizontal. Random landmarks are generated according to the distributions shown.

and the camera is mounted in a similar fashion to rovers used in later experiments: approximately 1.2 metres above the ground plane and tilted 18° towards the ground as shown in Figure 5.5. The simulation starts by determining the true poses and landmark states through the steps below:

1. Initialize the first camera position so it is located directly above the origin of \(F_g\).

2. Given a sequence of linear and angular velocity commands, determine the sensor’s pose, \(T_k\) at each time step \(k\).

3. Sample points uniformly from the region set by user-defined parameters \((x_{\text{min}}, x_{\text{max}})\) and \((y_{\text{min}}, y_{\text{max}})\) in the \(x\) and \(y\) directions (i.e., \(p_x \sim \mathcal{U}(x_{\text{min}}, x_{\text{max}})\), \(p_y \sim \mathcal{U}(y_{\text{min}}, y_{\text{max}})\)). Sample the height of each point according to \(p_z \sim \mathcal{N}(0, \sigma_{pz})\), where \(\sigma_{pz}\) is a user-defined variance parameter.
4. Sample each landmark’s reflectivity from a standard uniform distribution \( \varrho \sim U(0, 1) \).

5. Sample each landmarks surface normal from a Gaussian distribution with mean equal to the ground plane normal \( n \sim N\left([0, 0, 1]^T, \begin{bmatrix} (20^\circ)^2 & 0 \\ 0 & (20^\circ)^2 \end{bmatrix}\right) \).

6. Determine the union of the frustrum volumes from each sensor pose in \( F_g \) and remove any landmarks that do not appear in this region, or whose surface normals do not satisfy \(-n^T(T_k p) > 0\) for any of the sensor poses (i.e., incident angles less than \( \pi/2 \)). Only landmarks viewed from all sensor poses remain.

With the true sensor poses and landmarks generated, noisy observations for each time-step \( k \) are generated using the observation model, \( y_{jk} = g(x_{jk}) + v \), where \( v \sim N(0, R) \) is additive Gaussian noise and \( R \) is a user-defined covariance matrix that should be based on the specifications of the sensor that is being simulated.

### 5.2.2 Pose and Landmark Initialization

To carry out Gauss-Newton optimization an initial estimate for \( \hat{x} \) is needed. In this simulation the sensor pose estimates, \( \hat{T}_k \), are initialized at the same location as the first fixed pose, \( T_0 \), and the \( j \)-th landmark estimate, \( \hat{\ell}_j \), is initialized using the observations from the fixed first frame (i.e., \( y_{j0} = [u_{j0}, v_{j0}, r_{j0}, i_{j0}, \hat{n}_{j0}]^T \)). To do this, the observation model is inverted such that

\[
\ell_j = g^{-1}(T_0, y_{j0}) = z^{-1}(T_0, s^{-1}(y_{j0})).
\] (5.8)
5.2. SIMULATION AND SENSITIVITY ANALYSIS

where

$$z^{-1}(T_0, s^{-1}(y_{j0})) = M^{-1}(s^{-1}(y_{j0})) = \begin{bmatrix} T_0^{-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & C_0^{-1} \end{bmatrix} (s^{-1}(y_{j0})), \quad (5.9)$$

and $s^{-1}(\cdot)$ computes a landmark in the sensor frame from an observation. Just as with $s(\cdot)$, we present the separate components of $s^{-1}(\cdot) = [s_T^{-1}(\cdot), s_I^{-1}(\cdot), s_N^{-1}(\cdot)]^T$ that determine the landmark position, $p$, reflectivity, $\varrho$, and surface normal, $n$, respectively. For the ToF camera being simulated, the inverted observation model for $p$ is

$$p_z = \sqrt{r^2 x_n^2 + y_n^2 + 1}, \quad \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad p = s_T^{-1}(y) = \begin{bmatrix} xnp_z \\ ynp_z \\ p_z \\ 1 \end{bmatrix}, \quad (5.10)$$

the surface normal $n$ is

$$n = s_N^{-1}(y) = \tilde{n}, \quad (5.11)$$

and the reflectivity $\varrho$ is

$$\varrho = s_I^{-1}(y) = \frac{1}{\eta} \left( \frac{ir^2}{n^T \frac{-p}{||p||}} \right), \quad (5.12)$$
5.2. SIMULATION AND SENSITIVITY ANALYSIS

Table 5.3
Noise magnitudes tested in simulation

<table>
<thead>
<tr>
<th>Observation type</th>
<th>Magnitudes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range ($\sigma_r$)</td>
<td>0.02 m</td>
</tr>
<tr>
<td>Surface Normal ($\sigma_n$)</td>
<td>${0.1, 1, 2, 5, 10, 20}^\circ$</td>
</tr>
<tr>
<td>Intensity ($\sigma_i$)</td>
<td>${1, 10, 20, 50, 100}$</td>
</tr>
</tbody>
</table>

5.2.3 Sensitivity Analysis

In the previous sections we have determined a set of true $K$ poses and $M$ landmarks, $x$, and an initial estimate, $\hat{x}$, generated from the simulated noisy observations of $\ell$ taken at the fixed first pose, $T_0$. This covers all of the inputs to IntenseBA shown in Algorithm 1, and we can now test how sensitive our solution $\hat{x}$ is to the noise in each of our observations\footnote{Note that the value for the scaling parameter, $\eta$, used in the intensity observation model here depends on a calibration procedure described in Section 6.4.} by changing the associated variances in the observation covariance matrix $R$, running the simulation many times (we used 1000 in this thesis), and computing the root mean squared error (RMSE) in each of the degrees of freedom of the sensor pose and landmark states. The noise magnitudes tested for each type of observation are shown in Table 5.3.

The simulation’s adjustable parameters are the number of poses, number of landmarks, and the measurement noise properties. After each simulation we then analyze how the error changes in each part of the state vector. This results in a large number of possible configurations but here we present the results for $\sigma_r = 0.02$ m for two poses as this corresponds to the noise magnitude of the SR4000 ToF camera [76] in a typical frame-to-frame VO problem that is tested in the next section. From these
5.2. SIMULATION AND SENSITIVITY ANALYSIS

results we can determine when it becomes useful to include surface normals and intensity measurements into our BA problem (in all instances we require the pixel and range observations) for the SR4000 ToF camera.

The primary goal in VO is estimating the pose change, so we examine these results first (shown in Figure 5.6 and Figure 5.7). The problem is set up with $K = 2$ poses and $M = 20$ landmarks which is a typical amount of correspondences seen in images from the dataset presented in the experiments described in Section 7.2. Here we see that there is a significant improvement to both orientation and position estimates when surface normal observations are added (as we expect, given results in the literature that have demonstrated this in real-world examples, discussed in Section 4.1.1). When intensity observations with a standard deviation of less than or equal to 50 are included, we see a strong improvement to both estimates when surface normal noise is also low (less than 10°) between ~4% – 40%. When surface normal noise grows beyond this, we see that intensity can actually make the estimate worse. This intuitively makes sense, because of the dot product dependence on surface normal estimates and the fact that the problem is unobservable for two poses with intensity if the surface normal observation is not included. Therefore, if we want to use intensity observations, our surface normal observations must be relatively accurate.

We are also interested in how well our landmark states are estimated. There is a little improvement in landmark position estimates with intensity when surface normal observation noise is low (shown in Figure 5.8). However surface normal landmark estimates are improved with intensity (shown in Figure 5.9). Interestingly, this improvement is proportional to the magnitude of the surface normal observation noise magnitude. This makes sense, as intensity observations are providing an additional,
Figure 5.6: Sensor position error for $\sigma_r = 0.02m$, $K = 2$ poses, $M = 20$ landmarks for different magnitudes of surface normal and intensity noise.
5.2. SIMULATION AND SENSITIVITY ANALYSIS

Figure 5.7: Sensor orientation error for $\sigma_r = 0.02m$, $K = 2$ poses, $M = 20$ landmarks for different magnitudes of surface normal and intensity noise.
5.2. SIMULATION AND SENSITIVITY ANALYSIS

Figure 5.8: Landmark position error for $\sigma_r = 0.02m$, $K = 2$ poses, $M = 20$ landmarks for different magnitudes of surface normal and intensity noise.
5.2. SIMULATION AND SENSITIVITY ANALYSIS

Figure 5.9: Landmark orientation error for $\sigma_r = 0.02m$, $K = 2$ poses, $M = 20$ landmarks for different magnitudes of surface normal and intensity noise.
Figure 5.10: Landmark reflectivity error for $\sigma_r = 0.02 \text{m}$, $K = 2$ poses, $M = 20$ landmarks for different magnitudes of surface normal and intensity noise.
independent observation of the surface normal. Finally, the reflectivity estimates generated from IntenseBA are quite accurate when surface normal observation error is low, at less than 5% error in most cases (shown in Figure 5.10).

To summarize, adding intensity observations improves both sensor pose and orientation estimates. In addition, while it shows little improvement to landmark position estimates, it does show a significant improvement to landmark surface normal estimates and has been demonstrated to successfully augment states with the additional surface normal and reflectivity values. In all cases it is important to note that for large surface normal observation noise (i.e., $> 10^\circ$) intensity can be detrimental to the estimate, so it isn’t advisable to include intensity observations in such scenarios.
Chapter 6

Image Processing

The VO pipeline discussed in Chapter 2 contains multiple processing blocks. In order to apply the IntenseBA algorithm to real data, several changes were required. For this thesis, a commercial ToF camera called the SwissRanger 4000 (specifications are listed in Table 6.1) was chosen for its relatively low cost, high resolution, and ability to control directly through a provided application programming interface (e.g., exposure time can be controlled). This chapter outlines changes and improvements that resulted from adapting the VO pipeline to accept ToF camera measurements and produce a final set of measurements as input to the IntenseBA algorithm described in Chapter 4.

To begin, a brief overview is given of the lens distortion correction (a well-understood problem for all cameras) used for the ToF camera. Next, an analysis of keypoint detection and matching performance on the ToF camera intensity data is done. This is not well described in the literature, with only a few notable examples of 2D keypoint detection and description algorithms used on intensity data and represents a contribution of this thesis. Finally, a contribution to outlier detection is discussed, where the inclusion of surface normal measurements and intensity is shown

106
6.1. TOF CAMERA MODEL CALIBRATION

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions (mm)</td>
<td>$65 \times 65 \times 90$</td>
</tr>
<tr>
<td>Mass (kg)</td>
<td>0.47</td>
</tr>
<tr>
<td>Resolution (pixels)</td>
<td>$176 \times 144$</td>
</tr>
<tr>
<td>Min. range (m)</td>
<td>0.3</td>
</tr>
<tr>
<td>Max. range (m)</td>
<td>4.977</td>
</tr>
<tr>
<td>Reported range std (m)</td>
<td>0.01</td>
</tr>
<tr>
<td>Horiz. field of view (deg)</td>
<td>43.6</td>
</tr>
<tr>
<td>Vert. field of view (deg)</td>
<td>34.6</td>
</tr>
<tr>
<td>Max. frequency (frames per second)</td>
<td>up to 54</td>
</tr>
<tr>
<td>Power usage (W)</td>
<td>9.6</td>
</tr>
<tr>
<td>Illumination wavelength (nm)</td>
<td>850</td>
</tr>
</tbody>
</table>

Table 6.1
SR4000 Specifications

to improve the performance of the RANSAC algorithm when applied to VO.

6.1 ToF Camera Model Calibration

Like passive cameras, ToF cameras can have significant distortion in the images they produce. The distortion comes in two forms, tangential and radial. Tangential distortion occurs due to the imaging plane and the lens not being perfectly parallel. Radial distortion is due to distortions in the lens itself, and is dependent on focal distance and viewing direction. This distortion can often be partially corrected by using Brown’s distortion model [77]. Given a point $p \in \mathbb{R}^3$ in the sensor frame $F_s$, we saw in Section 4.5.2 that the normalized image coordinates associated with an idealized pinhole camera model are

$$x_n = p_x/p_z, \quad (6.1)$$

$$y_n = p_y/p_z.$$
6.1. TOF CAMERA MODEL CALIBRATION

The undistorted normalized coordinates \((x_b, y_b)\) are then computed using the even ordered polynomial

\[
x_b = x_n \left(1 + k_1 d^2 + k_2 d^4 + k_3 d^6\right) + 2 k_4 x_n y_n + k_5 \left(d^2 + 2x_n^2\right),
\]

(6.2)

\[
y_b = y_n \left(1 + k_1 d^2 + k_2 d^4 + k_3 d^6\right) + 2 k_5 x_n y_n + k_4 \left(d^2 + 2y_n^2\right),
\]

(6.2)

where \(d = \sqrt{x_n^2 + y_n^2}\). The undistorted pixel coordinates are found by multiplying the intrinsic parameter matrix by the undistorted normalized coordinates:

\[
\begin{bmatrix}
u \\
v \\
1
\end{bmatrix} = K \begin{bmatrix}
x_b \\
y_b \\
1
\end{bmatrix}.
\]

(6.3)

To determine the intrinsic parameter matrix and the radial and tangential distortion parameters, a calibration procedure is done, usually using a checker-board pattern displayed on a flat surface with known square sizes. By tracking the corners of the squares which are a known distance apart, an optimization problem [3] can be set up to compute these 9 parameters. Using the intensity imagery of the ToF camera, the same procedure was applied using MATLAB®'s Computer Vision Toolbox [78]. The parameters computed for the Swissranger 4000 are listed in Table 6.2.

The undistorted normalized coordinates in (6.2) are related to the angle between the viewing direction of the pixel and the camera optical axis. By taking the dot product of the homogeneous point \(p = [x_b, y_b, 1]^T\) with the unit vector pointed along the camera optical axis, \(u_3 = [0 \ 0 \ 1]^T\), we can determine the relation between pixel
6.1. TOF CAMERA MODEL CALIBRATION

Table 6.2
Distortion parameters for SR4000 ToF camera

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Calibrated values $\ell$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_1$</td>
<td>-0.8789</td>
</tr>
<tr>
<td>$k_2$</td>
<td>0.5821</td>
</tr>
<tr>
<td>$k_3$</td>
<td>0.2059</td>
</tr>
<tr>
<td>$k_4$</td>
<td>$-2.1488 \times 10^{-5}$</td>
</tr>
<tr>
<td>$k_5$</td>
<td>$-7.1747 \times 10^{-6}$</td>
</tr>
<tr>
<td>$f_x$</td>
<td>250.8570</td>
</tr>
<tr>
<td>$f_y$</td>
<td>250.8673</td>
</tr>
<tr>
<td>$c_x$</td>
<td>86.5003</td>
</tr>
<tr>
<td>$c_y$</td>
<td>68.0005</td>
</tr>
</tbody>
</table>

location and pixel viewing direction:

$$\sum_{i=1}^{n} \frac{p \cdot u_3}{\|p\|\|u_3\|} = \cos \theta_p,$$

where $\theta_p$ is the angle between the two vectors. Rearranging we see that $\cos \theta_p = \frac{1}{\|p\|}$.

This quantity can be used to help improve image brightness around the edges of the image. The SR4000 has a set of near-infrared light emitting diodes (NIR LEDs) placed around the imaging sensor that simulate a single point source that lies on the optical axis (shown in Figure 6.1). This helps avoid occlusions and makes illumination symmetric. However, because each LED emits according to a roughly cosine distribution along its vertical axis (i.e., it emits more strongly in the direction of the optical axis), the scene is still illuminated unevenly. The manufacturer has included a filter with its software package that corrects for this effect [76]. To determine the correction so that it can be applied before being used as an input to IntenseBA, the image with the filter applied was divided by the image without the filter, and a curve
6.2 Keypoint Detection and Description in ToF Intensity Images

LiDAR and ToF intensity imagery differs from camera images in a few ways that make keypoint detection and description difficult; the images are dependent on squared range (i.e., areas in the foreground are much brighter than areas in the background) and the angle of incidence between the sensor and the surface. While this thesis proposes that these effects are useful when estimating pose changes, they are detrimental when detecting/describing keypoints. Figure 6.3 shows raw imagery from an area observed by both a ToF camera and a traditional camera where it is clear that some type...
6.2. KEYPOINT DETECTION AND DESCRIPTION IN TOF INTENSITY IMAGES

![Diagram](image)

SwissRanger ToF

Figure 6.2: Depiction of inaccurate range measurement return due to pixel field of view encompassing both a part of the foreground and background.

of image processing is needed to be able to use the intensity images as an input to a keypoint detection/description algorithm. Range data from ToF and LiDAR sensors is also not well suited for the class of keypoint detection and description algorithms that identify and describe corners in images. Keypoint descriptors like Harris corners are often used for VO [79], however the range measurements associated with these keypoints tend to be much noisier than measurements associated with region-based keypoints. This is due to the nature of range measurements across boundaries, where the return is essentially a weighted average of the range at the boundary and behind the boundary as shown in Figure 6.2. For this reason only region-based keypoints were used for the purpose of VO.
6.2. KEYPOINT DETECTION AND DESCRIPTION IN TOF INTENSITY IMAGES

Table 6.3
Keypoint Detector and Descriptor combinations tested on intensity images

<table>
<thead>
<tr>
<th>Detector</th>
<th>Descriptor</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT</td>
<td>SIFT</td>
</tr>
<tr>
<td>SURF</td>
<td>(SURF, FREAK)</td>
</tr>
<tr>
<td>BRISK</td>
<td>(BRISK, FREAK)</td>
</tr>
<tr>
<td>KAZE</td>
<td>(KAZE, FREAK)</td>
</tr>
</tbody>
</table>

Despite these issues, there are some examples in the literature of keypoint detection and description algorithms being used successfully on active sensor intensity images. The SIFT [69] algorithm has been demonstrated to operate on ToF intensity images successfully [14, 15] and McManus et al. [12] showed good performance from SURF [80] operating on scanning LiDAR intensity images. However in these examples the authors do not provide a comparison to some of the other widely used keypoint detection and description algorithms and did not discuss in detail why they chose SIFT and SURF respectively. Since those publications, new keypoint detection (e.g., BRISK [81], KAZE [23]) and description (e.g., FREAK [82]) algorithms have been introduced that have been shown to outperform SIFT and SURF under certain conditions.

This section details what processing was done on the ToF camera imagery, as well as a comparison of various keypoint detection and matching algorithms running on processed imagery from a ToF camera. Image pairs were chosen to highlight the robustness of each algorithm to challenging conditions such as increasingly large translation or rotation changes and blur. A table of all combinations tested is shown in Table 6.3.
6.2. KEYPOINT DETECTION AND DESCRIPTION IN TOF INTENSITY IMAGES

Figure 6.3: Image processing performed on a SR4000 ToF image of the Planetary Utilization Testbed: (a) reference image of the terrain taken by a passive BumbleBee2 camera, (b) the ToF range image received from the sensor, brighter intensities represent larger range, (c) the raw ToF intensity image, (d) an image that has been normalized by dividing each pixel intensity by the brightest pixel, (e) Adaptive histogram equalization applied to the normalized image, (f) multiplication of each pixel intensity by the corresponding range pixel, (g) multiplication of each pixel intensity by the corresponding range pixel squared, (h) multiplication of each pixel intensity by the corresponding range pixel and the illumination correction value in V.
6.2. KEYPOINT DETECTION AND DESCRIPTION IN TOF INTENSITY IMAGES

6.2.1 Removal of Viewing Direction and Range Dependence

Before passing the image pairs to keypoint detection and description algorithms the images must be processed to remove the dependence on viewing direction and range from the sensor as much as possible. To accomplish this, there are a few options; McManus first reported results in [66] using adaptive histogram equalization, however this technique does not take into consideration the range dependence in the images. In followup work [12], McManus improved these results by multiplying the intensity image element-wise by the corresponding range image. Interestingly, McManus reported that multiplying by the square of the range image produced worse results.

For this thesis, all three methods were attempted, as well as a fourth that includes a correction for range squared as well as a correction for the angle between the pixel and the ToF camera optical axis as shown in Figure 6.3. The dot product of the viewing direction and unit vector along the optical axis for each pixel is determined beforehand during camera calibration according to (6.4) to form an illumination correction lookup table, $V$, described in Section 6.1.

The overall corrected image, $I_c$, is computed as

$$I_{cuv} = \frac{I_{uv}D_{uv}^2}{V_{uv}} \quad \forall \quad u = 1, 2, \ldots, 176 \text{ and } v = 1, 2, \ldots, 144,$$

(6.5)

where $I$ is the raw intensity image, $D$ is the raw range image, and $u$ and $v$ are the pixel coordinates in the SR4000 ToF camera image plane.
6.2.2 Determining Recall, 1-Precision, and Repeatability

With the corrected images, it is possible to evaluate different combinations of keypoint detection and description algorithms. Keypoint descriptors (e.g., SIFT, SURF, KAZE) described in real space are matched by L2 norm distance, and binary descriptors (BRISK) are matched by the Hamming distance [83].

The most common way to evaluate the performance of keypoint detection and extraction algorithms is by plotting their associated recall/1-precision curves and their repeatability [84]. Recall and 1-precision are defined as:

\[
\text{recall} = \frac{\text{correct match}}{\text{correspondences}}, \quad (6.6) \\
\text{1-precision} = \frac{\text{false matches}}{\text{correct matches} + \text{false matches}}, \quad (6.7)
\]

where a correct match is a true positive match, and correspondences are the total possible correct matches. The 1-precision is simply the percentage of matches that are false positive. To generate a curve, the distance metric used to determine a match for each algorithm is varied. A theoretical perfect detector/descriptor/matching algorithm would have recall = 1 for all values of 1-precision, and curves that are taller correspond to better performance.

Repeatability is measured from the region that is viewable in both images. The total number of keypoints from this region in each image is determined, and the repeatability is defined as the ratio between the total number of correspondences and the lower of the two keypoint totals.

Some type of ground truth is needed to determine whether algorithm-supplied matches are false or correct, and the total number of correspondences. Ground truth
correspondences are usually determined from images taken on a plane [85], with a large amount of texture. Because the keypoints lie on a plane, a homography can be established that transforms the keypoints in one image to where they would lie in a second image given a known pose change of the camera. In the case of intensity images there is no equivalent dataset available to use, however accurate pose ground-truth is available for the sensor using a VICON measurement system [86], as well as accurate range measurements associated with each pixel of the sensor. With this information, the points in one frame can be projected to another quite accurately.

For example, let there be two images of the same scene taken from different vantage points, $I_A$, and $I_B$, shown in Figure 6.4. Each has two frames associated with it: one that defines the camera image plane (e.g., $\mathcal{F}_{CA}$ and $\mathcal{F}_{CB}$) and one attached to the sensor (e.g., $\mathcal{F}_A$ and $\mathcal{F}_B$). The latter two are related through a transformation $T_{AB}$ that is measured using the VICON external measurement system. For each keypoint computed in the two images there is a center pixel coordinate and region around that coordinate that exist in the associated camera image plane. The points within the region form a set of measurements, $\mathbf{y}$:

$$
\mathbf{y}_j = \begin{bmatrix} u_j \\ v_j \\ r_j \end{bmatrix} \quad \forall j = 1\ldots J,
$$

(6.8)

where $J$ is the number of points in the keypoint region. We can transform these measurements to points, $\mathbf{p}_j \in \mathbb{R}^3$ in the reference frame attached to the associated camera and vice versa through the ToF camera observation model, $s_T(\cdot)$, and its inverse, discussed in Chapter 4. If we take $I_A$ as the reference image we can project
6.2. KEYPONTE DETECTION AND DESCRIPTION IN TOF INTENSITY IMAGES

Figure 6.4: Two images of the same scene, $I_A$ and $I_B$. Each image contains a keypoint region represented as an ellipse that are a potential match. The points in the region of $I_B$ are transformed through a homography (computed using the camera measurement model and VICON measured transformation $T_{AB}$) to measurements in $I_A$ where the overlap error, $\epsilon_S$, determines whether they are a match.
6.2. KEYPOINT DETECTION AND DESCRIPTION IN TOF INTENSITY IMAGES

a keypoint region \( y_B \) in \( I_B \) to a set of measurements \( y_A \) in \( I_A \) through:

\[
y_A = s_T(T_{AB}s_T^{-1}(y_B)).
\]  

By projecting all 3D points in the keypoint region to pixel locations in the reference frame, we can determine the overlap between the two regions of associated keypoints. The overlap error, \( \epsilon_S \), measures how well the regions of the two keypoints correspond over a transformation, defined by the ratio between the intersection and union of the two regions:

\[
\epsilon_S = 1 - \frac{(a \cap s_T(T_{AB}s_T^{-1}(b)))}{(a \cup s_T(T_{AB}s_T^{-1}(b)))},
\]  

where \( a \) and \( b \) are the two sets of image measurements respectively within each keypoint region. With this metric we determine correct matches provided by the algorithm if \( \epsilon_S \) is below a set threshold\(^1\).

Correspondences are determined for an image pair by computing the overlap error for every possible pairing between the two sets of images. This overlap error is treated as a cost, and used with the Hungarian assignment algorithm [87] to assign the optimal matches in the image pair, with the cost of not assigning a keypoint set to the threshold for \( \epsilon_S \).

6.2.3 Evaluation and Comparison of Keypoints on ToF Dataset

Selected image sequences, shown in Figure 6.5, were chosen from the Planetary Utilization Testbed dataset (more details about the Testbed are described in Section 7.2) to test the performance of different keypoint algorithm combinations listed in

\(^1\)In the literature the overlap error threshold is usually set between 20 – 50%, and for this work a value of 20% was used.
6.2. KEYPONT DETECTION AND DESCRIPTION IN TOF INTENSITY IMAGES

Figure 6.5: Selected image sequences from Planetary Utilization Testbed dataset
6.3. OUTLIER DETECTION

Table 6.3. These image pairs test the performance of keypoint algorithms with varying magnitudes of translation and rotation change and with different levels of apparent texture and blur. The implementation of SIFT used here is from the VLFeat open source library [88], all other algorithms used are from Matlab®’s Computer Vision Toolbox [78].

The repeatability for each image sequence measures the performance of the keypoint detector and is shown in Figure 6.6. Here we see that KAZE outperforms the other detectors in most cases and appears to have the most consistent performance. SURF and SIFT perform similarly, and BRISK performs the most inconsistently. KAZE likely exhibits the best performance due to its strength with detection of keypoints in blurry images [23] which are common in the ToF camera data due to its high exposure time.

The recall/1-precision curve measures the joint performance of detection, description and matching for each of the combinations in Table 6.3. Curves were generated for select image pairs and are shown in Figure 6.7. From these comparisons, we can see that SIFT-SIFT performs the best. However, KAZE-KAZE has comparable performance, a higher repeatability score, and generally runs in less time than SIFT [89]. For this reason KAZE-KAZE description/detection was used for the rest of the experiments presented in this thesis.

6.3 Outlier Detection

The primary difference between how outlier detection is handled in this thesis and traditional VO is in the number of points necessary for hypothesis generation. As we saw in Chapter 2, we can reduce the number of iterations needed when the number
6.3. OUTLIER DETECTION

Figure 6.6: Detector repeatability for 20% overlap between image #1 and #2 – 6
6.3. OUTLIER DETECTION

Figure 6.7: Recall vs 1-precision graphs for selected image pairs
of points required to produce a hypothesis is reduced. Section 5.1 details why, in the case of IntenseBA, the number of landmarks required is reduced from 3 to 2. There is also a dependence on the number of inliers present in the tracked keypoints, and the reduction of iterations compared to traditional VO is more pronounced when this number is low. In the case of ToF cameras, we expect more outliers compared to traditional stereo camera imagery because the keypoints are generated from processed, low resolution intensity images. For these reasons RANSAC could be expected to perform better with these additional measurements than if it was run on the range data alone.

### 6.3.1 RANSAC

The outlier detection technique used in this thesis is RANSAC [24]. The fundamental goal of RANSAC is to fit a parameterized model to a set of observations with outliers. It does this by iteratively sampling the minimum number of observations needed to generate a hypothetical fitting, and then tests the rest of the data to see if they agree (i.e., *inliers*) or disagree (i.e., *outliers*). After iterating a number of times, the hypothesis that has the best agreement and its inliers are chosen. RANSAC is probabilistic in that it can only be guaranteed to obtain a good solution with a certain probability that improves with increasing the number of samples taken. The basic algorithm can be thought of in 5 steps:

1. Select a minimum number of data points.

2. Fit the model to these data points (in our case this is determining a transformation $T$ that explains a set of matched keypoints).
Figure 6.8: Number of RANSAC iterations $k$ needed for different values of $w$, $p$, and $n$. Note the improvement in number of iterations for lower $n$.

3. Test the rest of the data points against the fitted model and determine which are inliers, and which are outliers.

4. Refit the model with the additional inliers chosen in the previous step.

5. Compute the residual error of the refitted model.

In IntenseVO, the $T$ hypothesis is determined with the minimum number of landmarks (normally three in BA, but only two when including surface normal observations as shown in Section 5.1). To compute the transformation, a version of iterative closest point (ICP) [50] is implemented using the unit-length quaternion solution in [90] where we treat each point and surface normal observation as a point observation. As Fischler and Bolles [24] note, needing less minimum landmarks for a hypothesis
6.4. INTENSITY CALIBRATION

has implications on the number of iterations needed to run RANSAC according to:

\[ k = \frac{\ln(1 - p)}{\ln(1 - w^n)}, \quad (6.11) \]

where \( k \) is the number of iterations needed to guarantee success with probability \( p \), if we assume each observation has a probability \( w \) of being an outlier and a minimum of \( n \) observations are needed to compute a hypothesis. It is clear from Figure 6.8 that the number of iterations \( k \) needed drops with \( n \).

RANSAC also allows us to remove specular outliers when we compute the residual error. If the intensity error is large and the other errors are small, the landmark is labelled an inlier but its intensity error and reflectivity are not considered in the optimization problem.

6.4 Intensity Calibration

During image acquisition, ToF camera pixels (like all cameras) collect light over a specific amount of time. This amount of time is referred to as exposure time, denoted \( t_e \) here. The amount of time the pixels are exposed to light has a large effect on the camera image that is produced. Images that are exposed for a shorter time will be darker than images that are exposed for a longer time. Choosing an appropriate exposure time when taking an image is important. If it is set too low, the brightest pixel could be much lower than the maximum intensity value, limiting the intensity resolution. If it is set too high, pixels may become saturated. Examples of what image histograms might look like in these two scenarios are shown in Figure 6.9 and Figure 6.10. To ensure that no information is lost while at the same time maximizing the resolution of the measurement, exposure control algorithms can be implemented.
6.4. INTENSITY CALIBRATION

Figure 6.9: A histogram of an underexposed image. Much of the available storage space is not utilized, resulting in lower resolution intensity data.

that adjust the exposure for new images based on the preceding ones. However, these changes in exposure obviously effect the intensity measurement, independent of all other parameters. This must be accounted for in the presented model before it can be used on real world data. For this reason, a SwissRanger SR4000 camera was selected for real world tests because it allowed the direct control and logging of exposure values, whereas many other (higher resolution) ToF cameras do not allow this level of control.

The model described by (4.10) (see p. 55) used to prove observability was based on the physical principles that determine the power of the returning signal. The equation for this comprises the three main factors: (a) the sensor, (b) the target, and (c) atmospheric parameters. The intensity measurement is linearly proportional to $p_r$. 

126
6.4. INTENSITY CALIBRATION

Figure 6.10: A histogram of an overexposed image of the same scene as the underexposed image. While the intensity resolution is improved, many of the pixels are now saturated.

by $\eta_{\text{LiDAR}} = t_e \eta_{\text{res}}$ where $\eta_{\text{res}}$ is the conversion factor between the power ratio and the number reported by the camera for intensity (i.e., the camera response). Up to this point this value has been treated as constant and was grouped with other constant parameters $\eta_{\text{sys}}$, $\eta_{\text{atm}}$, $a_r$, and $p_e$ to form $\eta$. Because the sensor reports exposure time,
6.4. INTENSITY CALIBRATION

we can separate that from the unknown factors before we group them together:

\[
i = \eta_{res} t_e \rho_r, \quad (6.12)
\]

\[
= \eta_{res} t_e \frac{p_r a_r^2 q}{4 r^2} \eta_{sys} \eta_{atm} \cos \alpha, \quad (6.13)
\]

\[
= \frac{\eta_{res} \eta_{sys} \eta_{atm}}{p_r} \eta \frac{t_e \rho}{r^2} \cos \alpha. \quad (6.14)
\]

To move from simulation to real-world experiments using a ToF camera, we must consider the non-linear nature of \( \eta_{res} \) that has been ignored up until this point. While the number of photons in a static scene that are collected by any one pixel will be linearly proportional to the irradiance, camera pixels convert this to electrical signals in a nonlinear way. This phenomenon is typically referred to as the camera’s response [91, 92]. It is characterised by each pixel’s response curve. Therefore \( \eta_{res} \), and by extension \( \eta \), is a nonlinear function of \( t_e, \rho, \alpha, r \) (all directly affect how many photons are collected by a pixel) and the pixel location on the sensor. This means equation (6.14) should be rewritten as:

\[
i = \eta_{uv}(i_0), \; i_0 = t_e \frac{\rho \cos \alpha}{r^2} \quad (6.15)
\]

where the \( uv \) subscript indicate the pixel location and \( i_0 \) is a value that is proportional to the irradiance at that pixel. The response curve for each pixel, \( \eta_{uv} \), can be modelled as a two term power equation:

\[
\eta_{uv}(i_0) = a_{uv} i_0^b + c_{uv}, \quad (6.16)
\]
6.4. INTENSITY CALIBRATION

where the three parameters $a$, $b$, and $c$ are estimated through calibration and form three look-up tables that are the same dimension as the ToF camera image. This makes computation very quick because the look-up index corresponds directly to the pixel location that is already computed in the full measurement model. All together, we are left with the model of intensity:

$$i = a_{uv}i_0^{b_{uv}+1} + c_{uv}i_0.$$  \hspace{1cm} (6.17)

6.4.1 Calibration Results

To determine the response curve for each pixel, a calibration was performed using a 127mm x 127mm, Spectralon\textsuperscript{®} Lambertian target (Model Number SRT-99-050). At the ToF camera’s operating wavelength of 850 nm, it has a known reflectivity between 97.5-99.5\%. The relatively small target was translated in the camera’s $x$-$axis$ and $y$-$axis$ directions placed at a known distance and orientation with respect to the camera (shown in Figure 6.11) in a $4 \times 8$ grid pattern so that the relatively small target was imaged by all pixels. Alternatively, the distance and orientation of the target could be estimated using the camera’s range measurements of the flat target. Using this information, a set of $i_0$ values was determined at each exposure time, for each pixel using (6.15).

The resulting calibration image can be seen in Figure 6.12 for several exposure times. Intensity values were recorded for all possible exposure times until a pixel was saturated. After correcting for illumination, the measured intensity values and the calculated $i_0$ are used to perform a nonlinear least squares optimization using the Levenberg-Marquardt algorithm to estimate the parameters in (6.17).
6.4. INTENSITY CALIBRATION

Figure 6.11: Calibration experimental setup. A Spectralon® target was mounted such that it could be easily translated in the ToF camera’s x-axis and y-axis directions.

The response curve range for all pixels is shown in Figure 6.13. The range occurs mostly due to a phenomenon called vignetting which is separate from the camera response, and occurs due to several types of systematic error [93]. For our purposes, it is sufficient that the look-up tables encode all of these and can be used to correctly model intensity measurements.
Figure 6.12: Examples of filtered image segments assembled into one image at different exposure times.
6.4. INTENSITY CALIBRATION

Figure 6.13: Response curves across all pixels
Chapter 7

Experiments

By combining the IntenseBA algorithm developed in Chapter 4 and the processing of ToF imagery developed in Chapter 5 it is possible to conduct VO experiments using the SwissRanger 4000 ToF camera that assess the performance of Intense Navigation compared to traditional VO. Several experiments were conducted in the Planetary and Robotics Laboratory at the European Space Research and Technology Centre (ESTEC) in Noordwijk, Netherlands. This chapter describes the experimental setup and the observed results.

7.1 Experimental Setup

The experiments were conducted in the Planetary and Robotics Laboratory (PRL) within the Automation and Robotics Laboratory (ARL) of ESTEC. The PRL has a 8 m \times 8 m test area called the Planetary Utilisation Testbed that attempts to replicate surface appearance on Mars based on imagery from NASA’s Spirit rover. The terrain is approximately Lambertian reflecting, and contains volcanic rocks of various sizes that act as obstacles. The entire terrain is imaged by VICON sensors [86] to provide an absolute reference system and measurements of robots driving in the terrain.
7.1. EXPERIMENTAL SETUP

The SwissRanger SR4000 (specifications are in Table 6.1) was mounted on a six-wheeled triple bogie rover prototype called ExoTer, meant to be a half-scale representation of the ExoMars rover ESA will send to Mars in 2020. The rover was driven in multiple clockwise circles around the terrain, during which it also conducted point turns (minimal translation while turning) as well as curved and straight paths. While the rover sensor suite also includes an IMU, stereo camera, and wheel and joint encoders, these sensors are not used by the IntenseVO algorithm. However, the stereo
7.2. INTENSE VISUAL ODOMETRY

camera images provided some context to the relatively low resolution, black and white imagery that the ToF camera produces. The VICON system tracked the position of highly reflective targets mounted at known locations on the ExoTer rover to sub-millimetre accuracy which is used to produce a ground truth estimate of the sensor pose.

The noise values used by IntenseVO to weight the various observations can be considered tunable parameters. While the SwissRanger 4000 has a reported standard deviation of 0.01 m, the actual value is sensor and environment dependent, with factors such as object reflectivity and distance affecting the accuracy of each observation. While there has been some effort by others to quantify this [94, 95], a value of 0.02 m was found to give the best results for the traditional configuration of VO, therefore this value was used for all configurations. In a similar way, intensity standard deviation depends largely on range, incidence angle and reflectivity of the surface, as well as the exposure time, with a value of 50 providing the best results. Pixel standard deviation is determined from the scale of the keypoint associated with the landmark, and surface normal standard deviations are determined in an online fashion based on the range and pixel observations of points neighbouring a keypoint according to the discussion in Appendix A. A summary of the noise values used in the experiments that follow is shown in Table 7.1.

7.2 Intense Visual Odometry

IntenseVO is performed by combining the IntenseBA algorithm with the processing steps described in Chapter 6. Together these form all of the processing blocks of the VO pipeline that convert the ToF camera data into estimates of the pose of the camera
Table 7.1
Experiment noise magnitudes

<table>
<thead>
<tr>
<th>Noise type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range ($\sigma_r$)</td>
<td>0.02 m</td>
</tr>
<tr>
<td>Pixel ($\sigma_u, \sigma_v$)</td>
<td>computed from keypoint scale</td>
</tr>
<tr>
<td>Intensity ($\sigma_i$)</td>
<td>50</td>
</tr>
<tr>
<td>Surface Normal ($\sigma_\psi_1, \sigma_\psi_2$)</td>
<td>computed online (see Appendix A)</td>
</tr>
</tbody>
</table>

and estimates of the landmarks tracked across two camera images. By compounding the transformations estimated between successive frames, the pose estimate at any time step along the trajectory can be determined. However, each transformation estimate between two frames will contain some error and as transformation estimates are compounded the error of the pose estimate will increase (i.e., dead reckoning) along the trajectory.

Because the amount of error in a transformation estimate is dependent on the observation noise of the ToF camera, not the magnitude of the translation and rotation, less noise is introduced into trajectory estimates if larger distances are driven between frames (the number of camera frames skipped between estimated poses is denoted as $\Delta k$). However, as shown in Chapter 6, the ability to track landmarks across frames degrades with larger rotation and translation changes. In addition, if $\Delta k$ is too large, motions between these frames are not estimated which may be undesirable. Therefore, a user-defined default $\Delta k$, denoted $\Delta k_0$, is set for the distance between frames (it was found that 15 frames provided a good trade-off between number of landmarks detected and the distance travelled between frames for the experiments shown here). If the number of landmarks matched between frames is too low, the algorithm decreases the frame distance and tries again. In addition, if the transformation computed is
7.2. INTENSE VISUAL ODOMETRY

below a set threshold in magnitude, the algorithm waits an additional $\Delta k_0$ frames to avoid adding error to the estimate when the camera is not moving.

At each time step, the transformation $\hat{T}_{k+\Delta k,k}$ and associated uncertainty $\Sigma_{k+\Delta k,k}$ are estimated using IntenseBA. This transformation is used to estimate the pose at time step $k + \Delta k$ by compounding the pose at $k$ with the estimated transformation $\hat{T}_{k+\Delta k,k}$

$$\hat{T}_{k+\Delta k} = \hat{T}_{k+\Delta k,k} \hat{T}_k.$$  \hfill (7.1)

The covariance matrix $\Sigma_{k+\Delta k}$ is computed by using the adjoint of the transformation $\hat{T}_k$, described in [32] to transform the uncertainty of $\hat{T}_{k+\Delta k,k}$ to the global frame $F_g$ and summing with the covariance of $\hat{T}_k$:

$$\Sigma_{k+\Delta k} = \Sigma_k + \mathcal{T}_k \Sigma_{k+\Delta k,k} \mathcal{T}_k^T,$$  \hfill (7.2)

where an adjoint matrix is computed according to

$$\mathcal{T} = \text{Ad} (\mathbf{T}) = \text{Ad} \left( \begin{bmatrix} C & r \\ 0^T & 1 \end{bmatrix} \right) = \begin{bmatrix} C & r^\wedge \\ 0 & C \end{bmatrix}. \hfill (7.3)$$

Three configurations of BA are used; a traditional configuration using only the range and pixel observations, a surface normal configuration that adds surface normal measurements, and an intensity configuration that adds surface normal and intensity measurements. Because RANSAC randomly samples points, the exact set of inliers it outputs can change slightly with each run. To ensure a fair comparison the algorithm is run once per time-step, and the inliers it finds are used for each configuration of BA. Pseudo-code for the IntenseVO algorithm is provided in Algorithm 2.

Require: \( T_0, \Delta k_0, D, I, \sigma_r, \sigma_i \)

output : Optimal estimate \( \hat{T}^*, \Sigma \)

1. \( k = 0 \)
2. \( \Delta k = \Delta k_0 \)
3. while \( k \leq K \) do
4. Load ToF range and intensity images: \( D_k, I_k, D_{k+\Delta k}, I_{k+\Delta k} \)
5. Correct images for illumination
6. Correct images for lens distortion
7. Detect, describe and match keypoints between \( I_k \) and \( I_{k+\Delta k} \)
8. if less than 3 keypoints matched then
9. \( \Delta k = \Delta k - 1 \)
10. go to next while loop iteration
end

11. Compute matched inlier observations \( \{ y_k, R_k, y_{k+\Delta k}, R_{k+\Delta k} \} \) and initial \( \hat{T}_{k+\Delta k,k} \) with RANSAC
12. if less than 3 matched inlier observations then
13. \( \Delta k = \Delta k - 1 \)
14. go to next while loop iteration
end

16. \( \{ \hat{T}^*_{k+\Delta k,k}, \Sigma_{k+\Delta k,k} \} = \text{IntenseBA}(\hat{T}_{k+\Delta k,k}, \{ y_k, R_k, y_{k+\Delta k}, R_{k+\Delta k} \}) \)
17. if \( \hat{T}^*_{k+\Delta k,k} \) is small then
18. \( \Delta k = \Delta k + \Delta k_0 \)
19. go to next while loop iteration
else
22. \( \hat{T}^*_{k+\Delta k} = \hat{T}^*_{k+\Delta k,k} \hat{T}^*_k \)
23. \( \Sigma_{k+\Delta k} = \Sigma_k + \mathcal{T}_k \Sigma_{k+\Delta k,k} \mathcal{T}_k^T \)
24. \( k = k + \Delta k \)
25. \( \Delta k = \Delta k_0 \)
end
7.3. RESULTS AND DISCUSSION

7.3 Results and Discussion

The ExoTer rover was commanded to drive four different trajectories around the Planetary Utilisation Testbed while the VICON sensors recorded the sensor pose. The results of VO performed with three configurations of BA are discussed in this section: (i) traditional BA which uses range and pixel observations; (ii) surface normal BA which uses range, pixel, and surface normal observations; (iii) Intense BA which uses range, pixel, surface normal, and intensity observations. The resulting trajectory estimates are shown in Figure 7.2-Figure 7.4.

Because the VICON system is not synchronized with the ToF camera, it was necessary to interpolate between the two observed poses before and after each image according to

$$T_{vk} = \left( T_{vk}^+ T_{vk}^- \right)^{\frac{t_k - t_{k^-}}{t_k - t_{k^+}}}$$

(7.4)

where $T_{vk}$ is the interpolated pose at $t_k$ computed from the VICON measured pose just before $t_k$ (at $t_{k^-}$), denoted $T_{vk}^-$ and the one measured just after $t_k$ (at $t_{k^+}$), denoted $T_{vk}^+$. The error between the estimated pose at $k$, and the interpolated VICON pose at $k$ is then computed as:

$$\xi_{ek} = \ln \left( T_{vk} \left( \hat{T}_{vk}^* \right)^{-1} \right)^\vee.$$

(7.5)

Note that the error exists in the unconstrained topological space $se(3)$ discussed in Section 4.5.4. The accuracy of each version of VO is measured by the norm of the error in position and orientation states as a percentage of the total distance travelled, which are plotted in Figure 7.6-Figure 7.8.

Here we see very similar results to the simulation, with a slight improvement
7.3. RESULTS AND DISCUSSION

[Diagram showing a perspective view of a trajectory with labels for Vicon, Traditional, Surface Normals, and Intensity.]

Figure 7.2: Perspective view of first trajectory.
7.3. RESULTS AND DISCUSSION

Figure 7.3: Perspective view of second trajectory.
7.3. RESULTS AND DISCUSSION

Figure 7.4: Perspective view of third trajectory.
Figure 7.5: Perspective view of fourth trajectory.
7.3. RESULTS AND DISCUSSION

Figure 7.6: Error in first trajectory estimates as a percentage of total distance travelled.

144
7.3. RESULTS AND DISCUSSION

Figure 7.7: Error in second trajectory estimates as a percentage of total distance travelled
Figure 7.8: Error in third trajectory estimates as a percentage of total distance travelled.
7.3. RESULTS AND DISCUSSION

Figure 7.9: Error in fourth trajectory estimates as a percentage of total distance travelled.
using surface normal observations and a larger improvement coming from the intensity observations. When this is not the case (notably in the orientation error of the second trajectory at the 66 second mark and the third trajectory at the 62 second mark) it appears likely due to the dependence of the intensity observation model on surface normal estimates as we see worse performance in the surface normal configuration as well. When the surface normal estimate is inaccurate the error is magnified by the inclusion of intensity observations. This makes sense, as we saw similar behaviour in the simulation results discussed in Section 5.2.
Chapter 8

Conclusion

This thesis seeks to include intensity observations into the BA problem, improving sensor localization as well as adding new information to the landmarks that are mapped. To begin, a high level overview of the VO pipeline is introduced along with prior work in this area (Chapter 2). The VO pipeline serves as an application to test the contributions of this thesis. The primary contribution of this thesis is to the optimization problem at the end of the pipeline, and so the parameterization of the sensor and landmarks is described (Chapter 3), and then used to develop the IntenseBA algorithm (Chapter 4). The observability of the problem is then verified and a simulation is built to examine how sensitive the problem is to noise, showing the potential for good improvement when intensity and surface normal noise is relatively low. Chapter 6 examines the image processing needed to proceed with each part of the VO pipeline discussed in Chapter 2 when using ToF Camera intensity images (the chosen sensor for testing), and discusses how algorithms like RANSAC benefit from the addition of surface normal observations and can be used to remove specular landmarks that are considered outliers. Finally, Chapter 7 combines all of the preceding work to form the VO pipeline and test the algorithm on real data taken in the
European Space Agency’s Planetary Utilization Testbed, a part of the Automation and Robotics Laboratory. Modest improvement is seen when including the intensity observations, and aligns with what simulation shows for noise magnitudes near the ones observed in these experiments.

8.1 Contributions

IntenseBA is a specialized implementation of Bundle Adjustment, which forms the key component to any VO or SLAM algorithm. The contributions of this thesis are centered around the development of the parameterizations used to include these observations into the problem, and the verification that it is worthwhile to do so (i.e., the problem is observable or solvable). Through simulation it has been verified that IntenseBA is possible and offers considerable improvement to the estimates of sensor pose and landmark states. The next step, applying IntenseBA to a practical application, required extensive image processing of SwissRanger 4000 ToF camera intensity images used in a VO implementation which showed promising results. The primary contributions are listed below.

- The parameterization of intensity observations in terms of sensor and landmark states and the development of IntenseBA which incorporates intensity and surface normal observations into a BA problem.

- Observability analysis and simulation that prove the problem is solvable and what noise magnitudes are appropriate to show improvement in estimation accuracy.
8.2. FUTURE WORK

- Description of image processing techniques needed to form consistent observation predictions of intensity and surface normals, including a calibration procedure that uses camera exposure time to determine the camera response curve at each pixel, and ultimately a prediction of intensity at each pixel given sensor and landmark parameters.

- Performance analysis of several different keypoint detection/description pairs on ToF camera images.

- Description of a two-landmark RANSAC algorithm that reduces the number of iterations needed to run RANSAC with ToF cameras.

- Demonstration of the full VO pipeline operating on ToF intensity and range images and a comparison of the accuracy across three different configurations of BA: traditional BA, surface normal BA, and IntenseBA, all operating on the same set of tracked landmarks.

8.2 Future Work

Through the process of developing the work in this thesis several areas of future work have been identified, listed below.

- The experiments presented in Chapter 7 were limited to an indoor environment with a ground truth provided by a VICON tracking system. However, an outdoor version of the SwissRanger 4000, the SwissRanger 4500, was used to create an outdoor dataset (originally published by the author in [96]). In this experiment a full-scale rover drove multiple kilometers with Global Navigation Satellite System (GNSS) ground-truth observations and additional mounted
8.2. FUTURE WORK

sensors such as an IMU, stereo camera, and 3D scanning LiDAR. Throughout the area traversed, artificial boulders were also placed with known surface shape and positions. The same calibration procedures in Chapter 6 were performed for this sensor and it is planned to apply IntenseBA to this dataset in the near future.

• While the implementation of VO presented here demonstrates the improvements IntenseBA brings to localization, it does not fully demonstrate the improvement to maps possible (shown in simulation) with additional parameters added to each of the landmarks being tracked. A SLAM implementation using IntenseBA would also provide a map of the environment that could be compared to traditional and surface normal configurations.

• As described in Chapter 2, there has been significant progress recently in dense and semi-dense methods of visual localization. While for the purposes of this work, sparse methods still make the most sense (due to limited computational power on space based hardware), it would be interesting to see how the work here could be extended to these methods.

• The results of the observability analysis in Chapter 5 suggest that a version of IntenseBA should be possible without surface normal observations as long as landmarks are observed over three or more frames. Because the VO demonstrated here operates on two frames at a time this wasn’t addressed in this thesis, but is another configuration of a SLAM implementation that would be worth analysis.

• While calibration was performed on the intensity data to form an observation
model of intensity in Chapter 6, other authors [94] have shown range observations from ToF cameras have a dependence on reflectivity. It may be possible to integrate the estimates of reflectivity introduced by IntenseBA to help improve the range observation model.

- During the time the work in this thesis was conducted, MESA’s SwissRanger sensors provided relatively inexpensive means of testing ToF cameras that had controllable exposure times. This was critical to properly calibrate the sensor and construct an intensity observation model. However, the SR4000 used in this thesis has very low pixel resolution (i.e., 144 × 176) that has been improved upon by several manufacturers. It should be possible in the near future to test IntenseBA on higher resolution sensors that also have controllable exposure times.
BIBLIOGRAPHY

Bibliography


BIBLIOGRAPHY


BIBLIOGRAPHY


Appendix A

Computing Surface Normal Measurements from Three-Dimensional Data

Information about surface orientation is lost in the sampling process, however the information still exists in the relationship between a sampled point and points in its neighbourhood [37]. The literature provides two predominant methods for estimating the surface normal of a point and its uncertainty; one based on optimization methods, and one based on averaging methods [37]. Optimization methods seek to minimize a cost function, for instance the distance of points to a local plane [36, 97] such as shown in Figure A.1a. Averaging methods compute the average surface normal of the

Figure A.1: Different ways of estimating surface normal vectors from a set of range measurements
specified point and pairs of its neighbors [98] such as shown in Figure A.1b. While optimization methods generally provide better results, averaging methods are less computationally complex. In the case of this thesis, the method similar to the one described in [36] is used, where the surface normal is estimated from the eigenvectors and eigenvalues of a covariance matrix computed from the set of nearby points, \( \mathbf{p} \), according to

\[
\mathbf{N}(\mathbf{p}) = \frac{1}{M} \sum_{m=1}^{M} (\mathbf{p}_m - \bar{\mathbf{p}})(\mathbf{p}_m - \bar{\mathbf{p}})^T, \quad (A.1)
\]

\[
\mathbf{N} \cdot \mathbf{v}_j = \lambda_j \cdot \mathbf{v}_j, \quad j \in \{0, 1, 2\} \quad (A.2)
\]

where \( M \) is the number of points considered, \( \bar{\mathbf{p}} \) is the mean of the point coordinates, \( \lambda_j \) is the \( j \)th eigenvalue and \( \mathbf{v}_j \) is the \( j \)th eigenvector. The surface normal is the eigenvector that corresponds to the smallest eigenvalue (the direction in which the points exhibit the least amount of variance). Nearby points can be determined in an unordered point cloud using nearest neighbour search algorithms [99], but if the data is ordered (e.g., adjacent pixels in an image) this processing step can be avoided.

However, computing surface normals this way does not include the uncertainty in the point measurements themselves. Each of these points has an associated uncertainty in \( \mathbb{R}^3 \) that was transformed from the measurement space (e.g., range and pixel coordinates). It is not immediately obvious how to include the individual point uncertainty explicitly when computing the surface normal and the surface normal uncertainty. In the literature this detail is usually ignored, and it is common to use the covariance matrix \( \mathbf{N} \) to represent the surface normal uncertainty, however this assumes surface normals exist in \( \mathbb{R}^3 \) and does not consider the measurement noise of
each point.

One possibility is to use the nonlinear inverse observation model, shown in (5.10), and restated here:

\[
\begin{bmatrix}
x_n \\
y_n \\
1
\end{bmatrix}
= K^{-1}
\begin{bmatrix}
u \\
v \\
1
\end{bmatrix},
\quad p_z = \sqrt{r^2 / x_n^2 + y_n^2 + 1},
\quad p = s_T^{-1}(y_T) =
\begin{bmatrix}
x_n p_z \\
y_n p_z \\
p_z \\
1
\end{bmatrix}, \quad (A.3)
\]

to transform a ToF camera pixel and range observation density \(\mathcal{N}(y_{T_0}, R_{T_0})\) and its 8 neighbouring observation densities \(\mathcal{N}(y_{T_m}, R_{T_m}), \quad m = 1 \ldots 8\) to a surface normal observation \(\mathcal{N}(y_n, R_n)\) using two successive unscented transforms [100]. The steps to do so are as follows:

1. Start by transforming (separately) each of the nine observation densities to \(\mathbb{R}^3\) using a set of \(2L + 1\) sigma points (where \(L\) is the degrees of freedom in the observation, in this case \(L = 3\)) sampled from the corresponding observation density. For a density \(\mathcal{N}(y_T, R_T)\) we have

\[
LL^T = R_T, \quad \text{( Cholesky decomposition)} \quad (A.4)
\]

\[
\gamma_{T_0} = y_T, \quad (A.5)
\]

\[
\gamma_{T_i} = y_T + (\sqrt{L + \kappa})L, \quad i = 1 \ldots L \quad (A.6)
\]

\[
\gamma_{T_{i+L}} = y_T - (\sqrt{L + \kappa})L, \quad i = 1 \ldots L \quad (A.7)
\]

where \((\cdot)_i\) is the \(i^{th}\) column.
2. The sigma points are individually passed through the nonlinearity $s_T^{-1}(\cdot)$:

$$
\gamma_{p_i} = s_T^{-1}(\gamma_{T_i}) \in \mathbb{R}^3, \quad i = 0 \ldots 2L
$$

(A.8)

3. The mean of the transformed sigma points is computed as

$$
y_p = \sum_{i=0}^{2L} \alpha_i \gamma_{p_i}, \quad \alpha_i = \begin{cases} 
\frac{\kappa}{L+\kappa}, & i = 0 \\
\frac{1}{2(L+\kappa)}, & \text{otherwise}
\end{cases}
$$

(A.9)

where $\kappa = 2$.

4. The covariance of the transformed sigma points is computed as

$$
R_p = \sum_{i=0}^{2L} \alpha_i (\gamma_{p_i} - y_p)(\gamma_{p_i} - y_p)^T.
$$

(A.10)

This is done for each of the observation densities and the resulting means and covariances are stacked into a single density $\mathcal{N}(y_P, R_P)$ where

$$
y_P = [y_{p_0}, \ldots, y_{p_M}]^T, \quad R_P = \text{blkdiag}(R_{p_0}, \ldots, R_{p_M}).
$$

(A.11)

This density is then passed through the function described in (A.1) using a second unscented transform, and the eigenvector with the smallest eigenvalue from its output is determined. Denoting this nonlinear and undifferentiable process as the function $f(\cdot)$, the steps are as follows

1. Start by transforming the density of stacked points to $S^2$ through a set of $2L + 1$ sigma points (where $L$ is the degrees of freedom in the observation, in this case...
$L = 3 \times 9$), sampled from the transformed observation $\mathcal{N}(\mathbf{y}_P, \mathbf{R}_P)$ according to

$$
\mathbf{L}\mathbf{L}^T = \mathbf{R}_P, \quad \text{(Cholesky decomposition)} \quad (A.12)
$$

$$
\mathbf{\gamma}_P = \mathbf{y}_P, \quad (A.13)
$$

$$
\mathbf{\gamma}_{P_i} = \mathbf{y}_P + (\sqrt{L + \kappa})_i \mathbf{L}, \quad i = 1 \ldots L \quad (A.14)
$$

$$
\mathbf{\gamma}_{P_i+L} = \mathbf{y}_P - (\sqrt{L + \kappa})_i \mathbf{L}, \quad i = 1 \ldots L \quad (A.15)
$$

where $(\cdot)_i$ is the $i^{th}$ column.

2. The sigma points are individually passed through the nonlinearity $f(\cdot)$:

$$
\mathbf{\gamma}_{n_i} = f(\mathbf{\gamma}_{P_i}) \in S^2, \quad i = 0 \ldots 2L \quad (A.16)
$$

3. The mean of the transformed sigma points is computed as

$$
\mathbf{y}_n = \sum_{i=0}^{2L} \alpha_i \mathbf{\gamma}_{n_i}, \quad \alpha_i = \begin{cases} 
\frac{\kappa}{L+\kappa}, & i = 0 \\
\frac{1}{2(L+\kappa)}, & \text{otherwise}
\end{cases} \quad (A.17)
$$

where $\kappa = 2$.

4. The covariance of the transformed sigma points is computed as

$$
\mathbf{R}_n = \sum_{i=0}^{2L} \alpha_i (\mathbf{\gamma}_{n_i} \bigotimes \mathbf{y}_n)(\mathbf{\gamma}_{n_i} \bigotimes \mathbf{y}_n)^T. \quad (A.18)
$$

The result is a transformed observation density $\mathcal{N}(\mathbf{y}_n, \mathbf{R}_n)$. This density represents the surface normal and its uncertainty, incorporating the uncertainty in the range
and pixel observations as well as the variance in the set of neighbouring points.