

EXTENSIONS TO THE VISUAL ODOMETRY PIPELINE FOR THE
EXPLORATION OF PLANETARY SURFACES

by

Paul Timothy Furgale

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Aerospace Science and Engineering
University of Toronto

Copyright © 2011 by Paul Timothy Furgale

Abstract

Extensions to the Visual Odometry Pipeline for the Exploration of Planetary Surfaces

Paul Timothy Furgale

Doctor of Philosophy

Graduate Department of Aerospace Science and Engineering

University of Toronto

2011

Mars represents one of the most important targets for space exploration in the next 10 to 30 years, particularly because of evidence of liquid water in the planet's past. Current environmental conditions dictate that any existing water reserves will be in the form of ice; finding and sampling these ice deposits would further the study of the planet's climate history, further the search for evidence of life, and facilitate in-situ resource utilization during future manned exploration missions. This thesis presents a suite of algorithms to help enable a robotic ice-prospecting mission to Mars. Starting from visual odometry—the estimation of a rover's motion using a stereo camera as the primary sensor—we develop the following extensions: (i) a coupled surface/subsurface modelling system that provides novel data products to scientists working remotely, (ii) an autonomous retrotraverse system that allows a rover to return to previously visited places along a route for sampling, or to return a sample to an ascent vehicle, and (iii) the extension of the appearance-based visual odometry pipeline to an actively illuminated light detection and ranging sensor that provides data similar to a stereo camera but is not reliant on consistent ambient lighting, thereby enabling appearance-based vision techniques to be used in environments that are not conducive to passive cameras, such as underground mines or permanently shadowed craters on the moon. All algorithms are evaluated on real data collected using our field robot at the University of Toronto Institute for Aerospace Studies, or at a planetary analogue site on Devon Island, in the Canadian High Arctic.

Dedication

To all my beautiful ladies¹.

¹To my mother for years of unquestioning love and support; my sisters, Sarah and Samantha, for being pretty much the best, coolest sisters ever; my niece, Agnes, may you never learn that your love of orange juice was the reason you needed dental surgery at such a young age; my friend, Maria, for warmth and strength when they were needed; my BFF, Joanna, for the years of poverty and discomfort we shared and all the things we learned together; my teammate, Claire, for her fearlessness in the face of perilous danger; and my love, Kristina, for our life together and the great adventure ahead.

Acknowledgements

Field robotics is collaborative work, and this thesis would not have been possible without the help and support of many people and organizations.

The mission concept and surface/subsurface modelling work would not have been possible without all of the following: geomorphological and general scientific guidance from Tim Haltigin, Kevin Williams, Michael Daly, and Gordon Osinski; hardware support and technical advice regarding ground-penetrating radar from Peter Annan and Dave Redmond at Sensors&Software; technical assistance and 3D models from the folks at MDA Space Missions including Piotr Jasiobedzki, Stephen Se, Ho-Kong Ng and Nadeem Ghafoor; logistical support for our field testing from members of the Mars Institute, all staff at the Haughton-Mars Project Research Station, and community members from Grise Fiord, Pond Inlet, and Resolute Bay, Nunavut who acted as guides and who were, I am sure, prepared to defend our robot from marauding polar bears.

I have been lucky to have excellent mathematical tutors at the Institute for Aerospace Studies including James Richard Forbes who contributed to the work in Chapter 3; Chi Hay Tong who knows how to take the ball and run with it; John Enright who introduced me to all things celestial; and Tim Barfoot, who taught me everything else.

I would also like to acknowledge the help of many of the members of the Autonomous Space Robotics Lab, specifically Chi Hay Tong for his work on the GPU SURF implementation, his general scepticism, and his donation of many whiteboard hours; Pat Carle for good times with lasers; Braden Stenning for his field testing support, field scotch, collaboration on robot middleware, help with software architecture, and tolerance for the limitations of stereo vision; Colin McManus for his collaboration on the work in Chapter 6 and letting me get some sleep during the epic dataset collection; Andrew Lambert and Hang Dong for collaborative work and rides to Downsview Station; Rehman Merali for help assembling PC-1, the pushcart rover; and Keith Leung for being an understanding desk neighbour.

Contents

1	Introduction	1
2	Stereo Visual Odometry	5
2.1	Image Distortion Correction and Rectification	7
2.2	Keypoint Detection	8
2.3	Stereo Matching	9
2.4	Keypoint Tracking	10
2.5	Outlier Rejection	12
2.6	Nonlinear Numerical Solution	13
2.7	Summary	14
3	State Parameterization for Visual Odometry	15
3.1	A Brief Sketch of Batch Nonlinear Least-Squares	16
3.2	Linearizing Expressions Involving Rotation Matrices	19
3.3	Linearizing Expressions Involving Transformation Matrices	24
3.4	Linearizing Expressions Involving Homogeneous Points	31
3.5	Examples	36
3.5.1	Stereo Camera Transformation Matrix Example	36
3.5.2	Rotation Matrix Priors	40
3.6	Conclusion	43
4	Coupled Surface/Subsurface Modelling	45
4.1	Related Work	47
4.2	Integrated Surface/Subsurface Modelling	48
4.2.1	Two-Dimensional Topographic Correction and Surface Profile . .	49
4.2.2	Three-Dimensional Surface/Subsurface Modelling	49

4.3	Field Testing	53
4.4	Results	55
4.4.1	Visual Odometry	55
4.4.2	Coupled Surface/Subsurface Models	57
4.5	Conclusion and Future Work	59
5	Autonomous Retrotraverse	61
5.1	Related Works	62
5.2	System Overview	66
5.2.1	A Generic Localization Module	68
5.2.2	Route Teaching	69
5.2.3	Route Repeating	77
5.2.4	Parameter Choices	80
5.2.5	Hardware	80
5.3	Field Testing	82
5.3.1	Route Following	83
5.3.2	Route with Large Three-Dimensional Motion and Extreme Lighting Changes	84
5.4	Evaluation	87
5.4.1	Convergence Properties	87
5.4.2	Lighting Dependence	89
5.4.3	Localization Performance during Path Following	89
5.4.4	Keypoint and Feature Usage	91
5.4.5	Teach Pass Failures	93
5.4.6	Repeat Pass Failures	94
5.5	Discussion	97
5.6	Conclusion	99
6	Towards Appearance-Based Methods for Lidar Sensors	101
6.1	Preliminaries	104
6.2	Lighting Sensitivity Analysis	105
6.2.1	Experiment Description	106
6.2.2	Results	107
6.3	Visual Odometry	108

6.3.1	Measurement and Error Terms from Lidar Data	109
6.3.2	Experimental Data	110
6.3.3	Results	110
6.4	Discussion	112
6.5	Conclusion	113
7	Summary and Future Work	115
7.1	Summary of Contributions and Publications	115
7.2	Future Work	118
7.2.1	Autonomous Retrotraverse	118
7.2.2	Appearance-Based Lidar	119
A	Catalogue of Route-Following Experiments	121
B	List of Acronyms	127
	Bibliography	128

Notation

- a : Symbols in this font are real scalars.
 \mathbf{a} : Symbols in this font are real column vectors.
 \mathbf{A} : Symbols in this font are real matrices.
 \mathbf{a} : Symbols in this font are quantities represented in homogeneous coordinates.
 $\sim \mathcal{N}(\mathbf{a}, \mathbf{B})$: Normally distributed with mean \mathbf{a} and covariance \mathbf{B} .
 $E[\cdot]$: The expectation operator.
 $\underline{\mathcal{F}}_a$: A reference frame in three dimensions.
 $(\cdot)^\times$: The cross-product operator that produces a skew-symmetric matrix from a 3×1 column.
 $\mathbf{1}$: The identity matrix.
 $\mathbf{0}$: The zero matrix.
 $\mathbf{p}_a^{c,b}$: A vector from point b to point c (denoted by the superscript) and expressed in $\underline{\mathcal{F}}_a$ (denoted by the subscript).
 $\mathbf{p}_a^{c,b}$: The vector $\mathbf{p}_a^{c,b}$ expressed in homogeneous coordinates.
 $\mathbf{C}_{a,b}$: The 3×3 rotation matrix that transforms vectors from $\underline{\mathcal{F}}_b$ to $\underline{\mathcal{F}}_a$:

$$\mathbf{p}_a^{c,b} = \mathbf{C}_{a,b} \mathbf{p}_b^{c,b}$$

 $\mathbf{T}_{a,b}$: The 4×4 transformation matrix that transforms homogeneous points from $\underline{\mathcal{F}}_b$ to $\underline{\mathcal{F}}_a$:

$$\mathbf{p}_a^{c,a} = \mathbf{T}_{a,b} \mathbf{p}_b^{c,b}$$
.

Chapter 1

Introduction

Mars represents one of the most important targets for the international space exploration community in the near- to mid-term (i.e., 10-30 years). The planet is of particular scientific importance and interest because of the widespread evidence for the presence of water in its geological past (Carr, 1996; Masson et al., 2001). Environmental conditions on Mars today are such that any water reserves will be in the form of ice, either in the polar caps or as ground ice at lower latitudes (Carr, 1996). The detection and analysis of ground ice on Mars is an important pursuit for two reasons. First, study of the content and characteristics of water on Mars is important basic science with relevance in fields such as climate history and astrobiology. Second, sites found with significant deposits of ground ice may be useful for future human exploration missions due to the possibility of in-situ resource utilization.

Polygonal terrain—a network of interconnected trough-like depressions in the ground—is a landform commonly found throughout the polar regions of both Earth (Lachenbruch, 1962; Mackay and Burn, 2002; Fortier and Allard, 2004) and Mars (Mangold, 2005; Levy et al., 2009). In terrestrial environments, these surface features are formed by thermal forcing of frozen terrain under seasonal freezing and thawing, and are often indicative of subsurface wedges of water ice (Lachenbruch, 1962); on Mars, it is believed that such thermal forcing may also be responsible for the creation of polygonal surface features (Levy et al., 2009; Mellon et al., 2008). The Phoenix Lander mission (Smith et al., 2008) has confirmed the presence of a frozen substrate in the Martian polygonal terrain shown in Figure 1.1, but the nature of any underlying massive ice deposits has not yet been determined. It is therefore very important to develop techniques to find and sample subsurface ground ice on Mars.

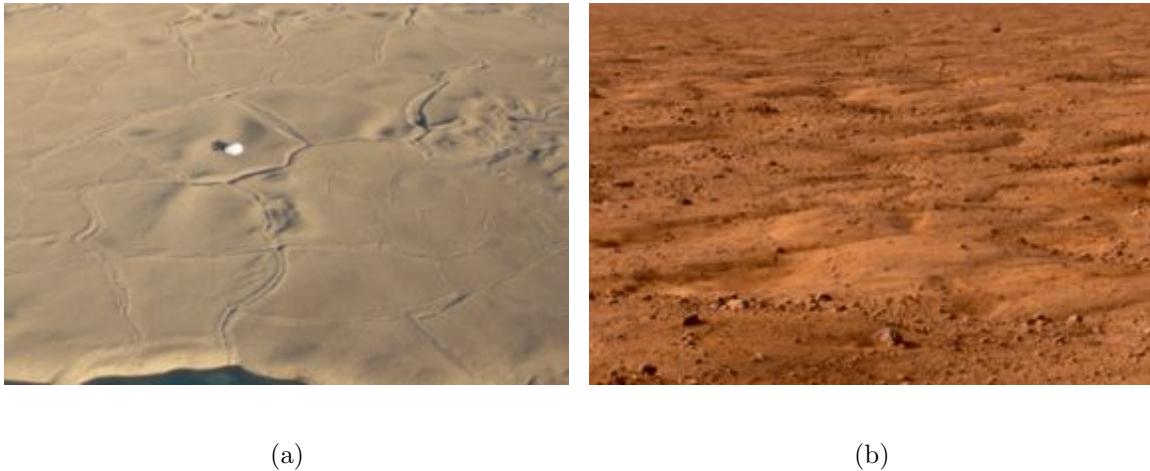


Figure 1.1: Polygonal terrain is a common periglacial landform often indicative of subsurface ice deposits, seen here (a) at our test site on Devon Island, in the Canadian High Arctic (courtesy Mars Institute/Haughton-Mars Project), and (b) from the Phoenix Lander on Mars (courtesy NASA/JPL-Caltech/University of Arizona/Texas A&M University).

This thesis will show how Visual Odometry ([VO](#))—a method of estimating the motion of a vehicle using a stereo camera as the primary sensor—can be utilized and extended to enable specific elements of robotic ice prospecting on Mars. The work will be presented within the context of a mission concept to carry out such ice prospecting using a rover-mounted sensor suite consisting of a light detection and ranging ([lidar](#)) sensor, stereo camera, and Ground-Penetrating Radar ([GPR](#))¹. Hence, we will begin by introducing our mission concept and follow by outlining the specific contributions of this thesis.

Figure 1.2 depicts the top-level steps in our concept. The processes below the dashed line would take place on Earth, while those above would take place on Mars. The labels on the arrows indicate the data products that would be sent back and forth via Earth-Mars communications. According to this mission architecture, we first select a landing site based on orbital imagery (which can reveal the presence of polygonal terrain). We then land and build a large-scale 3D model of the surrounding terrain using a [lidar](#). We use this [lidar](#) scan to (i) select candidate polygon troughs for closer examination using a stereo camera and [GPR](#) and (ii) plan a rover path to deliver these instruments to these

¹The mission concept provides a high-level motivation for the work presented in this thesis, but it is important to note that the algorithms presented here are not specific to polygonal terrain and we do not attempt to advance the current scientific understanding of polygonal terrain; our main objective is to develop and test extensions to the [VO](#) pipeline and each algorithm presented has the potential for wider use in robotic applications.

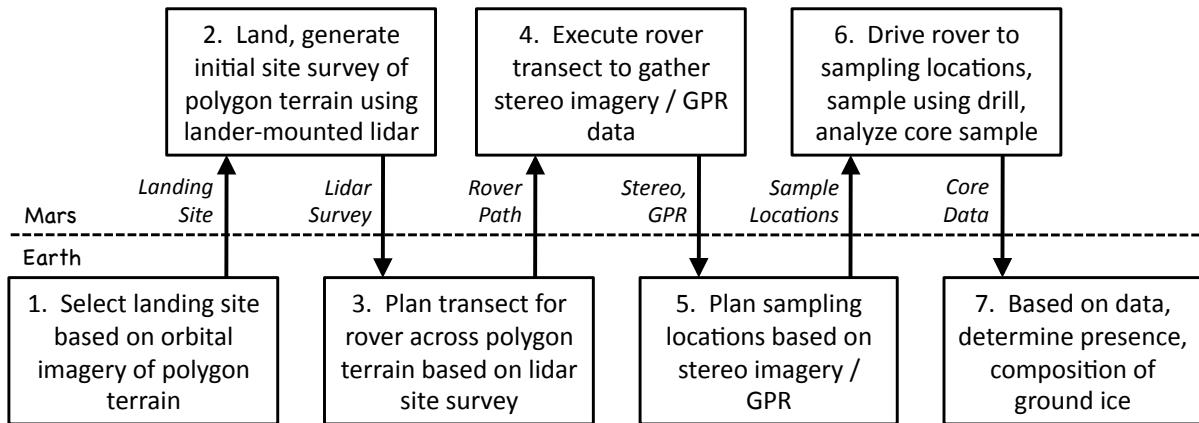


Figure 1.2: Operational steps and data products transferred to/from Earth.

troughs. The rover then drives this planned path collecting stereo imagery and GPR data. The process of collecting a lidar scan, collecting GPR and stereo data along a path, and returning the data to Earth may be iterated any number of times. The data collected along the path are used by the science team to select troughs for subsurface sampling. The rover then returns to these sites, samples ground ice, analyzes the samples' composition, and returns the data to Earth. The specific contributions of this thesis use the VO pipeline to address operational considerations in steps 4, 5, and 6 of this mission concept.

In step 3, the rover must autonomously follow a specific path to collect stereo imagery and GPR data. Accurate path following requires the rover to maintain an accurate position estimate over the length of the path. During the recent Mars Exploration Rover (MER) mission, the rovers utilized wheel odometry measurements to estimate position changes on nominal terrain. However, the MERs have frequently encountered steep slopes and sandy terrain, resulting in large amounts of wheel slip, thereby rendering the odometry measurements unreliable (Maimone et al., 2007). For these high-slip situations, they employ VO over short distances. With some improvements (Johnson et al., 2008), the upcoming Mars Science Laboratory rover will also make use of VO. Chapter 2 describes our VO algorithm, which is used as an enabling technology throughout the rest of this thesis, and Chapter 3 details some of the mathematics used in our VO algorithm.

In step 4, raw data products produced from the stereo imagery and GPR traces are delivered to a science team that will choose a set of sampling locations. The ability to measure surface properties and visualize how they are related to the subsurface GPR data is of critical importance during a geophysical survey. Properties such as eleva-

tion differences and grain size (e.g., sand versus boulders) are particularly important attributes that are always recorded and studied in detail during any field campaign on Earth, typically by a geologist taking measurements using tape measures and the Global Positioning System ([GPS](#)). In addition to being time-consuming and relatively crude in terms of accuracy, these manual methods will not be possible on future rover missions due to lack of availability of both geologists and [GPS](#) on Mars. In Chapter 4, we use the raw sensor data together with the motion estimate from [VO](#) to automate the GPR data collection process and produce novel data products that allow scientists to visualize the relationship between surface features and subsurface structure.

In step 5, the rover must return to a set of specific sampling locations along the outbound traverse. Because the goal in this step is to return to the exact physical locations specified by the science team (the *retrotraverse problem*), it is clear that simply using a dead-reckoning method like [VO](#) is not sufficient to retrace the rover’s path. Dead-reckoning methods, such as wheel odometry or [VO](#), will not solve this problem as localization error will continue to accumulate during retrotraverse. Continuous absolute localization (such as [GPS](#)) would offer a good solution to the retrotraverse problem by ensuring that localization errors are bounded at every point. However, such a system is not currently available on Mars. The teach-and-repeat navigation system described in Chapter 5 uses [VO](#) as an enabling technology within a mapping and localization system that allows the rover to retrace its route without requiring an external absolute localization system or accruing further localization error as it retraces its path.

In Chapter 5 the major system failures in teach-and-repeat navigation are attributed to changing scene appearance caused by changes in ambient lighting. All camera-based localization and mapping systems are reliant on ambient lighting in this way and hence susceptible to such failures. In Chapter 6, we explore the possibility of using an actively-illuminated lidar sensor within our appearance-based [VO](#) framework. Lighting-invariant [VO](#) using an actively illuminated sensor has the possibility to enable navigation into permanently shadowed craters on the Moon as well as teach-and-repeat navigation robust to lighting changes. Finally Chapter 7 summarizes the contributions of this thesis and discusses future work and challenges ahead.

Chapter 2

Stereo Visual Odometry

The core contributions presented in this thesis are based on stereo VO—the estimation of a rover’s position and orientation using a stereo camera as the primary sensor. This chapter will review related work and describe each of the major processing blocks in the sparse stereo VO pipeline.

The use of a stereo pair of cameras (as opposed to a single monocular camera) greatly simplifies the motion estimation problem because the known transformation between the cameras allows the metric structure of the scene to be estimated using a single stereo image pair¹. The scale of the motion cannot be recovered solely from monocular sequences (Hartley and Zisserman, 2004; Davison and Murray, 2002; Nistér, 2004; Strasdat et al., 2010b) so metric odometry using a monocular camera is only possible using some other measurement of scale, such as an inertial measurement unit (Strelow and Singh, 2004; Jones and Soatto, 2011) or wheel odometry.

Early work estimating motion from image sequences may be divided into two categories: (i) optical flow algorithms (dense algorithms) and (ii) interest point tracking algorithms (sparse algorithms) (Aggarwal and Nandhakumar, 1988). While there is some recent work recovering motion (or structure and motion) using optical flow (Corke et al., 2004; Benoit and Ferrie, 2007), the techniques are not widely used in unstructured environments due to the difficulty of jointly estimating the dense scene structure and camera motion without finding explicit correspondences or making assumptions about the nature of the scene. The Efficient Second-order Minimization (ESM) algorithm used for tracking planar patches (Mei et al., 2008), estimating motion (Comport et al., 2010), or

¹The term *structure* in this section refers to the underlying three-dimensional geometry of the scene. This usage is common in computer vision (e.g., Hartley and Zisserman (2004)).

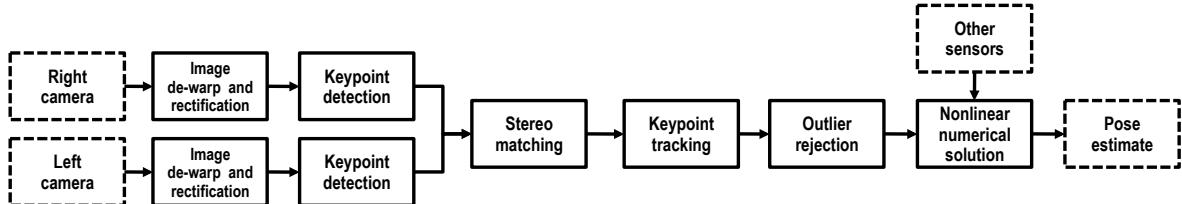


Figure 2.1: The basic sparse stereo VO pipeline includes image distortion correction and rectification, keypoint detection, stereo matching, keypoint tracking, outlier rejection, and a numerical solution for pose.

Sparse methods extract a sparse set of interest points, or *keypoints*, from each incoming image and track these points through the image sequence. There are numerous computational savings derived from the application of sparse methods mainly thanks to years of research from the photogrammetry and computer vision communities. The basic sketch of VO from sparse keypoint tracks was detailed by Moravec in his PhD thesis (Moravec, 1980)². His work included all of the major processing blocks still used for VO today: image capture, image distortion correction, keypoint detection, stereo matching³, keypoint tracking, outlier rejection, and a nonlinear numerical solution for pose. Moravec's work was further developed by Matthies (Matthies and Shafer, 1987; Matthies, 1989) to become the VO implementation deployed on NASA's MERs (Maimone et al., 2006, 2007). The basic outline of the sparse stereo VO pipeline is depicted in Figure 2.1. The remainder of this chapter is a brief overview of the key steps in this pipeline.

²Moravec's thesis is worth a read. As well as a lot of interesting work it is full of amusing anecdotes and great illustrations. Moravec was a pioneer.

³Although in his case, he was matching between nine images taken as a single camera slid along a track.

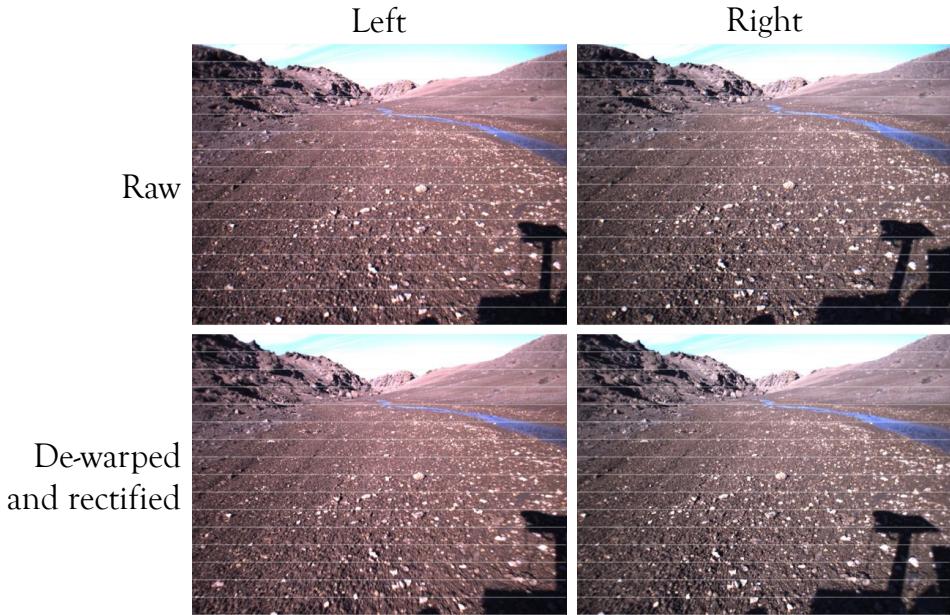


Figure 2.2: Images captured from a stereo camera are corrected for lens distortion and rectified so that they appear to have been captured by a pair of linear cameras with parallel optical axes and aligned image planes. In this idealized configuration, objects observed in one image will appear on the same row in the other image. To illustrate this, transparent horizontal lines have been overlaid on the raw and rectified images shown above.

2.1 Image Distortion Correction and Rectification

Image distortion correction and rectification is processed as one step in the pipeline but it is really performing two separate tasks: (i) *distortion correction*, which attempts to remove any lens distortion so that an individual image appears to have come from an idealized pinhole camera (also known as a *linear camera*) (Hartley and Zisserman, 2004, p. 189), and (ii) *rectification*, the alignment of the two images of a stereo pair so that a point in space projects onto the same row of each image (Hartley and Zisserman, 2004, p. 302). The transformed images after distortion correction and rectification appear to have come from an idealized stereo camera consisting of two linear cameras with parallel optical axes and aligned image planes as shown in Figure 2.2. The mathematical model of such an idealized camera is specified in Section 3.5.1. The two operations are usually combined into a single undistortion map for each camera and pixels in the output image are filled in by looking up the mapping pixel in the input image.

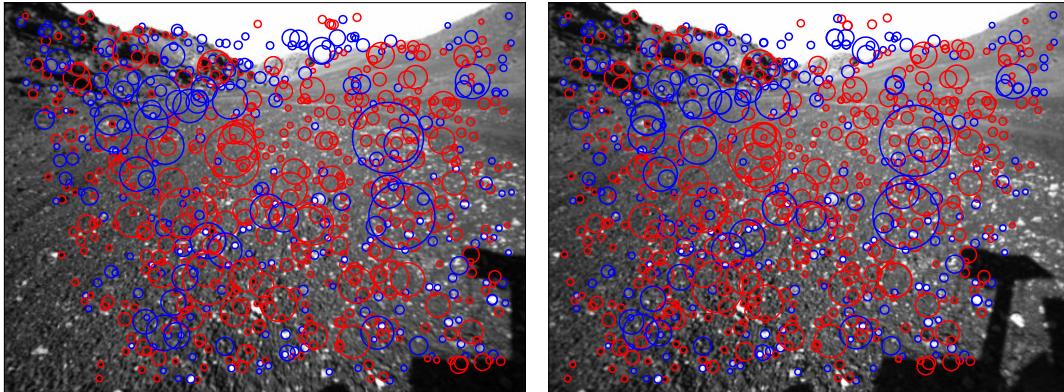


Figure 2.3: This figure shows the strongest SURF keypoints extracted from the left and right images of a stereo pair. SURF is a multiscale keypoint detector and the size of the circle corresponds to the size of the keypoint in the image. SURF also distinguishes between light blobs on dark backgrounds (blue) and dark blobs on light backgrounds (red).

2.2 Keypoint Detection

After distortion correction and rectification, the rectified images are passed to an interest point detection algorithm that finds regions of interest in the image. Interest point detection is an active research area and there are many algorithms from which to choose. The most common choice for VO is the FAST detector (Rosten and Drummond, 2006), which is used by many implementations (Eade and Drummond, 2007; Klein and Murray, 2007; Howard, 2008; Mei et al., 2010; Konolige et al., 2010) because of its high speed and low computational cost. We have chosen our own implementation of the Speeded-Up Robust Features (SURF) algorithm (Bay et al., 2008), which runs on a Graphics Processing Unit (GPU)⁴. The SURF algorithm finds circular regions of interest of different sizes in a monochrome image, as shown in Figure 2.3. The “Speeded-Up” prefix in the algorithm title is a reference to the Scale Invariant Feature Transform (SIFT) algorithm (Lowe, 2004); many of the operations in the SURF algorithm are fast approximations to those used in the SIFT algorithm.

At this step, we also use the SURF algorithm to produce a 64-dimensional description vector, $\mathbf{d} \in \mathbb{R}^{64}$, that encodes local appearance information in a window around each keypoint. Description vectors are normalized so that $\mathbf{d}^T \mathbf{d} = 1$, and the similarity of two keypoints with descriptors \mathbf{d}_a and \mathbf{d}_b is calculated to be $s_{ab} = \mathbf{d}_a^T \mathbf{d}_b$. Pairs of keypoints with a higher similarity score are more likely to be views of the same landmark, and we

⁴Our implementation—Speeded Up Speeded Up Robust Features—has been released as an open-source package and is available at <http://asrl.utias.utoronto.ca/code/gpusurf>

use the similarity scores for data association throughout the rest of the pipeline⁵.

Other VO implementations also use description vectors for data association (Konolige et al., 2010), but the simplest and most common approach is to use the brightness values in a small patch around the keypoint center to describe the keypoint (Nistér et al., 2006; Howard, 2008; Mei et al., 2010). To compare keypoints from different images, two patches may be compared using the sum of absolute differences between the brightness values (Howard, 2008), normalized correlation (Nistér et al., 2006), or the ESM algorithm to find the best warp that aligns two patches (Mei et al., 2010).

In most published VO implementations, a keypoint measurement is modelled as the sum of the true measurement, $\bar{\mathbf{y}}$, and zero mean, independent Gaussian noise, $\delta\mathbf{y}$, with covariance \mathbf{R} :

$$\mathbf{y} = \bar{\mathbf{y}} + \delta\mathbf{y}, \quad \delta\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \quad (2.1)$$

Very few publications are explicit about the values they choose for \mathbf{R} (Maimone et al. (2007) is a notable exception), but with a single-scale keypoint detector⁶ (such as FAST) and no other sensors used, it may be reasonable to assume $\mathbf{R} = \mathbf{1}$ (the noise is Gaussian, isotropic, and equal for every keypoint). As we use SURF, a multiscale detection scheme, it is not appropriate to choose the same uncertainty for every keypoint. However, we use something equally simple, assigning $\mathbf{R} = \sigma^2 \mathbf{1}$, where $\sigma = 2^v$ pixels and v is the *octave* in which the keypoint was detected. Keypoints found in octave 0 were detected in the full-sized image. For each subsequent octave, the image is downsampled by half⁷. Anecdotally, we found that using this simple assignment of uncertainty improved our VO estimates over $\mathbf{R} = \mathbf{1}$, but further study is necessary to clearly show what the correct uncertainty assignment should be for a particular detector.

2.3 Stereo Matching

Next, interest points are matched between images in a stereo pair. Because the images are rectified, the search for a keypoint’s stereo match may proceed across a single row in

⁵We assume that keypoint measurements are generated by point landmarks. Given a keypoint and descriptor in one image, the data association step decides which keypoint in a second image corresponds to a view of the same landmark (or if there is no such keypoint in the second view). This happens during stereo matching and keypoint tracking.

⁶In this context, *single scale* means that each interest point corresponds to a region of approximately the same size in the image.

⁷SURF accomplishes this without actually downsampling the images, but the result is the same.

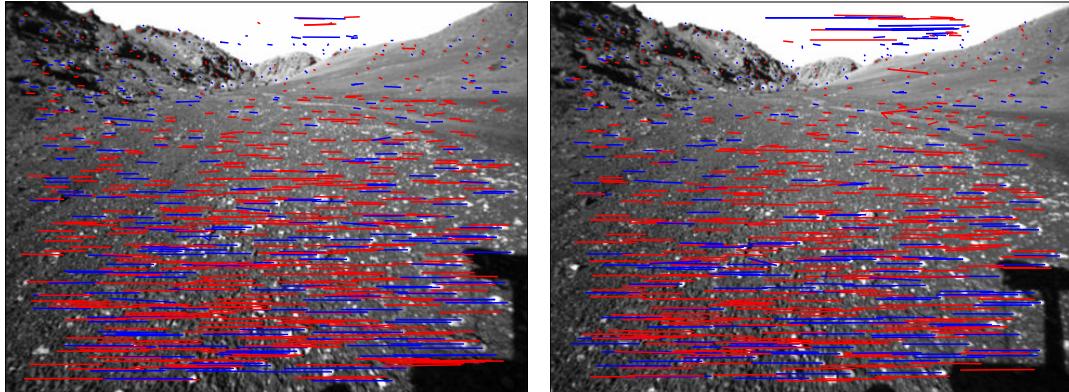


Figure 2.4: This figure shows the stereo keypoints extracted from two temporally consecutive stereo images. Horizontal lines connect the left and right image locations associated as a single stereo keypoint. Longer lines correspond to keypoint observations of landmarks closer to the camera (they have a larger disparity value). A single stereo pair is not sufficient to reject inconsistent stereo measurements and so these sets of measurements often contain outliers, seen most prominently here as large-disparity measurements in the sky.

the other image. As such, some implementations forgo keypoint detection in the second image and use dense stereo processing to find stereo matches (Howard, 2008; Konolige and Agrawal, 2008). Like Nistér et al. (2006) and Mei et al. (2010), we detect keypoints in both images and then search for correspondences between keypoints along rows.

To interact with our GPU implementation of the SURF algorithm, we also implemented our stereo matching algorithm on the GPU. Because SURF is a multiscale keypoint detector that produces subpixel measurements, it is very unlikely that corresponding keypoints will have precisely the same floating-point row value. Therefore, stereo matches are restricted to be between keypoints within the same octave and the allowable vertical distance (in pixels) between two keypoints is proportional to the keypoint uncertainty. Any keypoints that successfully find a stereo correspondence are packaged into stereo keypoints⁸. Results of our stereo matching algorithm are shown in Figure 2.4.

2.4 Keypoint Tracking

After building sets of stereo measurements from each stereo image (as shown in Figure 2.4), temporally consecutive stereo images are searched for keypoint correspondences. We accomplish this by viewing the older image (or some collection of older images in

⁸The exact form of stereo keypoints we use is described in detail in Section 3.5.1 and so we defer our description until then.

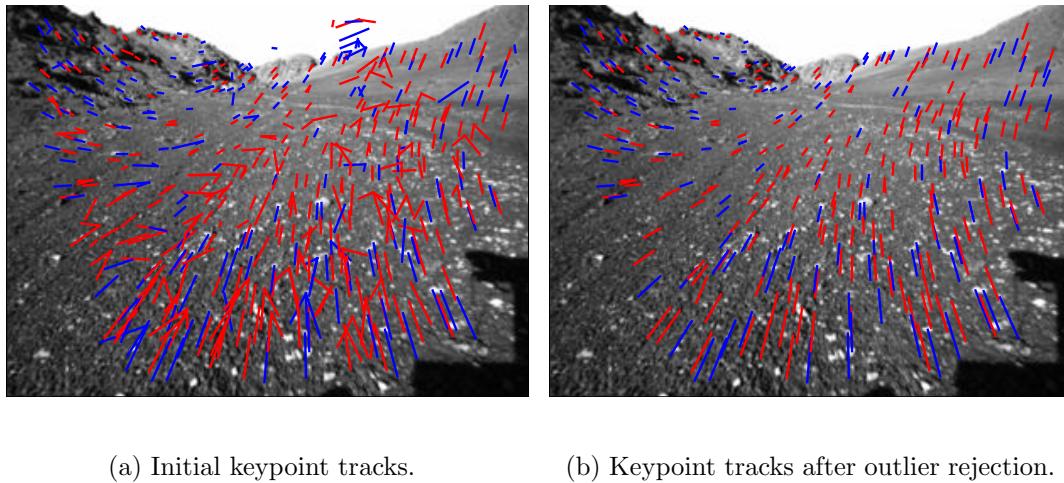


Figure 2.5: A set of landmarks (derived from previous VO timesteps) is projected into the image. For each landmark, the keypoint with the highest descriptor similarity score in a small window around the projection is chosen as a potential correspondence. This set of keypoint tracks is subject to outlier rejection using RANSAC. After outlier rejection, all remaining tracks support a single motion hypothesis.

Chapter 5) as a map consisting of point landmarks, $\mathbf{p}_m^{i,m}$, $i = 1 \dots M$, expressed in some map frame, \mathcal{F}_m , with associated descriptors, \mathbf{d}_i . To make keypoint data associations at time k , a coarse motion estimate is used to predict the observation locations of the landmarks in the current image. The coarse motion estimate can be derived from other onboard sensors (Cheng et al., 2006), a constant velocity assumption (Davison et al., 2007), whole-image processing techniques (Mei et al., 2010), or by the naive assumption that the rover does not move between images. The tracking algorithm then searches in a window around the predicted landmark location for the keypoint in the current image with the strongest descriptor similarity. Figure 2.5(a) shows the result of tracking between the two stereo keypoint sets shown in Figure 2.4.

Keypoint tracks coming out of this stage of the pipeline are based only on appearance (encoded in the descriptors) and proximity (matching within a window); there is no guarantee that this set of tracks describes a consistent motion estimate. One of the underlying assumptions in VO is that the scene is rigid, so any violation of this assumption (e.g., a scene with dynamic objects) will result in feature tracks that do not describe the rover’s motion. However, even with a rigid scene, such as the one shown in Figure 2.5, the initial set of keypoint tracks is polluted by outliers. Chli and Davison (2009) try to combine the tracking, outlier rejection, and motion estimation into a single algorithm

called Active Matching. The results are promising but it is not able to run at frame rate, or with the number of keypoint tracks usually used in stereo VO implementations. Hence, most VO implementations use some form of dedicated outlier rejection after feature tracking.

2.5 Outlier Rejection

As Figure 2.5(a) clearly depicts, keypoint tracks invariably contain mismatches. It is important to find and remove these mismatches before numerical optimization. Many approaches to outlier detection have been developed in the computer vision community, mostly to address the problem of estimating the fundamental matrix relating a pair of images. In essence, this is a statistical model-fitting task; an algorithm must find a model to explain the data, then return the model and all data points that agree with it.

This problem may be solved using RANSAC (Fischler and Bolles, 1981), an algorithm that repeatedly (i) generates a model from a randomly selected minimal set of data, and (ii) scores the model by counting the number of data points with error below a fixed threshold. Many improvements to the original RANSAC algorithm have been proposed. For example, MLESAC (Torr and Zisserman, 2000), scores models using a squared-error cost function, and MAPSAC (Torr, 2002) includes a step to choose the best model class. Motion estimation using a calibrated stereo rig is a significantly simpler task than fundamental matrix estimation because there is no ambiguity as to which model class to choose. The intrinsic camera parameters are obtained through calibration, and the known baseline between the cameras resolves any scale ambiguities. As such, many VO implementations adopt the original RANSAC algorithm (Maimone et al., 2007; Konolige et al., 2007; Mei et al., 2010).

We use a fast alternative to RANSAC developed by Nistér (2005) called *preemptive RANSAC*. Preemptive RANSAC garners speedups over the traditional RANSAC algorithm by generating all hypotheses at once, and then incrementally narrowing the set of plausible hypotheses by iterating the following steps: (i) score a small batch of data points through every remaining hypothesis, (ii) sort the hypotheses by score, and (iii) discard a handful of the least likely hypotheses. This nested loop inversion greatly reduces the amount of computation required, as most hypotheses are discarded *before* all data points have been processed. After outlier rejection, the remaining feature tracks are consistent with a single motion estimate hypothesis (Figure 2.5).

2.6 Nonlinear Numerical Solution

After the outlier rejection step, the remaining feature tracks should be nearly outlier-free. The goal of the nonlinear numerical solution is to find the state variables—a set of variables encoding (i) the change in position and orientation of the camera between images and (ii) the positions of the three-dimensional landmarks in the scene—that best explain the landmark measurements (keypoint tracks) that remain after outlier rejection. The stereo observation model for point landmarks (discussed in detail later, in Section 3.5.1) is nonlinear and hence the most common method of pose solution is nonlinear least-squares estimation. This section provides an overview of related work. Specifics of the solution methods used in this thesis are presented in Chapter 3.

In his pioneering work, Hans Moravec’s motion estimation program retained a world model of three-dimensional landmarks. To solve for the rover’s pose, corresponding keypoints were triangulated and the two point clouds were aligned using a scalar-weighted, nonlinear least-squares approach (Moravec, 1980). Matthies and Shafer (1987) put this approach on firm probabilistic footing through more sophisticated handling of the uncertainties involved—using linearized covariance propagation from uncertainty in image space to uncertainty in triangulated points, and using matrix-weighted nonlinear least-squares to align point clouds resulting from each pair of consecutive images.

There have been other notable pure stereo VO publications including Nistér et al. (2006) and Konolige et al. (2007), both of which introduce Sparse Bundle Adjustment (SBA) (Brown, 1958) as the pose solution method. SBA is an iterative Gauss-Newton minimization algorithm that exploits the specific sparsity pattern in the update step equations to greatly reduce the computational cost of each iteration⁹. SBA enables fast and accurate solutions for motion *and* structure—the landmark positions producing the keypoint measurements. In a way, through the introduction of SBA, VO became Simultaneous Localization and Mapping (SLAM). This connection was articulated clearly by Sibley et al. (2008), who also showed how the matrix sparsity patterns change when a motion model and prior information—staples of the SLAM community (Bailey and Durrant-Whyte, 2006; Durrant-Whyte and Bailey, 2006)—are introduced. Although there is still SLAM research into large-scale localization and mapping using Kalman-, Information-, Particle-Filter forms (Thrun et al., 2001; Barfoot, 2005; Cadena and Neira,

⁹For an excellent introduction to SBA, please see Hartley and Zisserman (2004, p. 597). For a further reading and references, see Triggs et al. (2000).

2010), many researchers have adopted the SBA approach (Strasdat et al., 2010a; Konolige et al., 2010; Sibley et al., 2010), especially for SLAM using cameras as the primary sensor.

Table 2.1: A summary of the input image size and solution method used for VO in different chapters of this thesis.

Chapter	Image Size	Solution Method
Chapter 4	1280×960	A solution for motion similar to that described in Mai-mone et al. (2007) but using the linearization strategies described in Chapter 3. A complete derivation of the estimator is given in Furgale et al. (2010).
Chapter 5	640×480	A solution for position with respect to a static map, formulated differently for each mode required by the autonomous retrotraverse algorithm (Section 5.2.1).
Chapter 6	512×384	A solution for motion and structure based on the sliding window SBA (Konolige et al., 2007), but using the state parameterizations developed in Chapter 3.

We have used a number of different solution methods over the years, all based on batch nonlinear least squares. Chapter 3 outlines our basic solution method and some of the strategies we have used to parameterize landmarks and robot poses. Our goal for the work in this thesis was never to push the metric accuracy of VO to the limit (as it was in other collaborations (Lambert et al., 2011)), so the solution methods have generally been simple. Table 2.1 lists the image sizes and solution methods used in this thesis.

2.7 Summary

In this chapter we have provided a brief overview of the sparse stereo VO pipeline. While there are no direct contributions to the underlying VO pipeline emanating from this thesis work, the algorithm described here is used as the key enabling technology for contributions in the application of VO (i) to support rover based science (Chapter 4), (ii) to enable long-range autonomous navigation in unstructured, three-dimensional terrain (Chapter 5), and (iii) to perform appearance-based motion estimation using a lidar sensor, thereby making VO robust to lighting changes (Chapter 6).

Chapter 3

State Parameterization for Visual Odometry

This chapter will discuss the choice of parameterization for state variables common to estimation problems in robotics. In the VO algorithm described in Chapter 2, our goal is to estimate the motion of a vehicle in three-dimensional space using a set of sparse stereo keypoints tracked through an image sequence. To perform this estimation, we must choose a set of parameters to represent the position and orientation of the vehicle, as well as the positions of the landmarks in the scene. The most common solution methods used for VO are based on batch nonlinear least squares—an iterative numerical optimization technique that requires successive linearization of the individual error terms in the cost function.

The purpose of this chapter is to derive linearization strategies for several state-parameter classes of interest in robotics and, for each derivation, to provide some notation and algebraic tools for manipulating expressions involving these linearized quantities. We provide derivations for the following state-parameter classes:

1. *rotation matrices*: 3×3 matrices that represent elements of the group $SO(3)$,
2. *transformation matrices*: 4×4 matrices that represent elements of the group $SE(3)$ and can be used to represent coordinate frame transformations compactly, and
3. *homogeneous points*: unit-length 4×1 columns that represent Euclidean points encoded in homogeneous coordinates.

While many of the identities derived below may be found scattered throughout other

textbooks and papers, the contribution of this chapter is to collect the results together in one place, and show how they may be derived using a first-principles Taylor-series expansion approach. This avoids having to resort to Lie algebras, matrix exponentials, tensors, and other tools that can be very useful, but are unnecessary if our goal is simply to linearize error terms for nonlinear optimization.

We begin with a brief sketch of the Gauss-Newton solution to the nonlinear least-squares problem, and then derive linearization strategies for rotation matrices, transformation matrices, and homogeneous points. To demonstrate the utility of our approach, we provide two examples that are used in later chapters: (i) an example of linearizing a stereo camera error term, and (ii) an example of forming and linearizing a prior information term on a 3×3 rotation matrix.

3.1 A Brief Sketch of Batch Nonlinear Least-Squares

Many state estimation tasks in robotics reduce to the problem of finding the state parameter vector, \mathbf{x} , that minimizes a scalar squared-error function, $J(\cdot)$, of the form

$$J(\mathbf{x}) := \frac{1}{2} \sum_{n=1}^N \mathbf{e}_n(\mathbf{x})^T \mathbf{W}_n \mathbf{e}_n(\mathbf{x}), \quad (3.1)$$

where $\mathbf{e}_n(\cdot)$ is one of N individual error terms weighted by the matrix \mathbf{W}_n . For stereo VO, the error terms are based on an observation model of the form,

$$\mathbf{z}_n = \mathbf{g}_n(\mathbf{x}) + \mathbf{v}_n, \quad (3.2)$$

where \mathbf{z}_n is an individual observation, $\mathbf{g}_n(\mathbf{x})$ is the (possibly nonlinear) observation model—our model of how some subset of state parameters produced the observation—and \mathbf{v}_n is a random variable representing observation noise. When we assume that \mathbf{v}_n is independent, zero-mean, and Gaussian,

$$\mathbf{v}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_n), \quad E[\mathbf{v}_m \mathbf{v}_n] = \mathbf{0}, \quad (3.3)$$

setting

$$\underbrace{\mathbf{e}_n(\mathbf{x})}_{\text{error}} := \underbrace{\mathbf{z}_n}_{\text{observation}} - \underbrace{\mathbf{g}_n(\mathbf{x})}_{\substack{\text{predicted} \\ \text{observation}}} \quad \text{and} \quad \mathbf{W}_n := E[\mathbf{e}_n \mathbf{e}_n^T]^{-1} = \mathbf{R}_n^{-1} \quad (3.4)$$

makes

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} (J(\mathbf{x})) \quad (3.5)$$

equivalent to the maximum-likelihood estimate of the state given the measurements (Jazwinski, 1970, p. 156). Defining

$$\mathbf{e}(\mathbf{x}) := \begin{bmatrix} \mathbf{e}_1(\mathbf{x}) \\ \vdots \\ \mathbf{e}_N(\mathbf{x}) \end{bmatrix}, \quad \mathbf{R} := \operatorname{diag}\{\mathbf{R}_1, \dots, \mathbf{R}_N\}, \quad (3.6)$$

we may express (3.1) in matrix form,

$$J(\mathbf{x}) = \frac{1}{2} \mathbf{e}(\mathbf{x})^T \mathbf{R}^{-1} \mathbf{e}(\mathbf{x}). \quad (3.7)$$

If $\mathbf{e}(\cdot)$ is a linear function of \mathbf{x} , $J(\cdot)$ is exactly quadratic in \mathbf{x} and we may find its minimum by setting $\frac{\partial J(\mathbf{x})}{\partial \mathbf{x}}^T$ to zero and solving the resulting system of equations. When $\mathbf{e}(\cdot)$ is a nonlinear function, the minimum of $J(\cdot)$ must be found iteratively, using a nonlinear optimization technique (Nocedal and Wright, 2006). In robotics, gradient-based optimization techniques, such as Gauss-Newton or Conjugate Gradient, are commonly used. In this thesis, we have used Gauss-Newton and so a basic sketch of the algorithm is provided here.

Starting with an initial guess for the state, $\bar{\mathbf{x}}$ —arrived at using a lower-fidelity linear method or through a solution using only a portion of the data available—we make the approximation that

$$\mathbf{x} = \bar{\mathbf{x}} + \delta\mathbf{x}, \quad (3.8)$$

for some small update step, $\delta\mathbf{x}$. We then substitute (3.8) into (3.7) and use a first-order Taylor-series approximation to linearize $\mathbf{e}(\cdot)$ about $\bar{\mathbf{x}}$. The result approximates $J(\cdot)$ as quadratic in $\delta\mathbf{x}$,

$$J(\bar{\mathbf{x}} + \delta\mathbf{x}) = \frac{1}{2} \mathbf{e}(\bar{\mathbf{x}} + \delta\mathbf{x})^T \mathbf{R}^{-1} \mathbf{e}(\bar{\mathbf{x}} + \delta\mathbf{x}) \quad (3.9a)$$

$$\approx \frac{1}{2} \left(\mathbf{e}(\bar{\mathbf{x}}) + \left. \frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\bar{\mathbf{x}}} \delta\mathbf{x} \right)^T \mathbf{R}^{-1} \left(\mathbf{e}(\bar{\mathbf{x}}) + \left. \frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\bar{\mathbf{x}}} \delta\mathbf{x} \right), \quad (3.9b)$$

which we write as

$$J(\delta\mathbf{x}) = \frac{1}{2} \mathbf{e}(\delta\mathbf{x})^T \mathbf{R}^{-1} \mathbf{e}(\delta\mathbf{x}), \quad (3.10)$$

where

$$\bar{\mathbf{e}} := \mathbf{e}(\bar{\mathbf{x}}), \quad \mathbf{E} := \left. \frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\bar{\mathbf{x}}}, \quad \mathbf{e}(\delta\mathbf{x}) := \bar{\mathbf{e}} + \mathbf{E}\delta\mathbf{x}. \quad (3.11)$$

Now we may find the minimum of (3.10),

$$\delta\mathbf{x}^* = \underset{\delta\mathbf{x}}{\operatorname{argmin}} (J(\delta\mathbf{x})), \quad (3.12)$$

by expanding $\frac{\partial J(\delta\mathbf{x})}{\partial \delta\mathbf{x}}^T$,

$$\frac{\partial J(\delta\mathbf{x})}{\partial \delta\mathbf{x}}^T = \left(\frac{\partial J(\delta\mathbf{x})}{\partial \mathbf{e}(\delta\mathbf{x})} \frac{\partial \mathbf{e}(\delta\mathbf{x})}{\partial \delta\mathbf{x}} \right)^T \quad (3.13a)$$

$$= \frac{\partial \mathbf{e}(\delta\mathbf{x})}{\partial \delta\mathbf{x}}^T \frac{\partial J(\delta\mathbf{x})}{\partial \mathbf{e}(\delta\mathbf{x})}^T \quad (3.13b)$$

$$= \mathbf{E}^T \mathbf{R}^{-1} \mathbf{e}(\delta\mathbf{x}) \quad (3.13c)$$

$$= \mathbf{E}^T \mathbf{R}^{-1} (\bar{\mathbf{e}} + \mathbf{E}\delta\mathbf{x}), \quad (3.13d)$$

setting it to zero, and solving the resulting linear system of equations for $\delta\mathbf{x}^*$,

$$\mathbf{E}^T \mathbf{R}^{-1} (\bar{\mathbf{e}} + \mathbf{E}\delta\mathbf{x}^*) = 0 \quad (3.14a)$$

$$\mathbf{E}^T \mathbf{R}^{-1} \mathbf{E}\delta\mathbf{x}^* = -\mathbf{E}^T \mathbf{R}^{-1} \bar{\mathbf{e}} \quad (3.14b)$$

$$\delta\mathbf{x}^* = -(\mathbf{E}^T \mathbf{R}^{-1} \mathbf{E})^{-1} \mathbf{E}^T \mathbf{R}^{-1} \bar{\mathbf{e}}. \quad (3.14c)$$

Generally you would not compute the inverse in (3.14c), but rather solve the linear system in (3.14b)¹. The optimal update is then applied to our current guess,

$$\bar{\mathbf{x}} \leftarrow \bar{\mathbf{x}} + \delta\mathbf{x}^*, \quad (3.15)$$

to result in a new (hopefully better) estimate of the state parameters. This process is iterated until $J(\mathbf{x})$ converges to a minimum². The resulting state estimate, \mathbf{x}^* , has covariance (Bell and Cathey, 1993)

$$\mathbf{P} := (\mathbf{E}^T \mathbf{R}^{-1} \mathbf{E})^{-1}, \quad (3.16)$$

which we may think of as

$$\mathbf{x} = \mathbf{x}^* + \delta\mathbf{x}, \quad \delta\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}). \quad (3.17)$$

One assumption we made in the derivation above was that there were no constraints on the values that \mathbf{x} could assume. This presents challenges for many practical estimation

¹Often the specific sparsity of the system of equations is exploited to speed up this step as in Hartley and Zisserman (2004, Appendix 6). For general sparse solution techniques, please see Davis (2006).

²This is the barest sketch of batch nonlinear optimization. For more details on this and other related algorithms, please see Nocedal and Wright (2006) and the excellent Appendix 6 of Hartley and Zisserman (2004).

problems in robotics where a particular choice of parameters to represent a state variable may have singularities or constraints. For example, the set of rotations constitutes a *non-commutative group*, called $SO(3)$. Regardless of the choice of representation (e.g., rotation matrix, unit-length quaternion, Euler angles), a rotation has exactly three degrees of freedom. All rotational representations involving exactly three parameters have singularities (Stuelpnagel, 1964) and all representations having more than three parameters have constraints. The question of how best to parameterize and handle rotations in state estimation is by no means new. There are many rotational parameterizations available, each with its unique advantages and disadvantages (Shuster, 1993). In spacecraft attitude and robotics estimation, the 4×1 unit-length quaternion (a.k.a., Euler-Rodrigues symmetric parameters), the standard 3×3 rotation matrix, and Euler angles are all common (Crassidis et al., 2007).

In this chapter, we derive linearization strategies for a number of common state-parameter classes: (i) 3×3 rotation matrices that represent elements of the group $SO(3)$, (ii) 4×4 transformation matrices that represent elements of the group $SE(3)$ and can be used to represent coordinate frame transformations compactly, and (iii) unit-length 4×1 columns that represent Euclidean points encoded in homogeneous coordinates. The linearization strategies presented have a number of nice properties: they are *minimal*, in the sense that the update parameterization has the same number of degrees of freedom as the underlying state variable, they are *constraint sensitive* in that the equation used to update the state variable preserves constraints on the state, and they are *unconstrained* in that, as long as the update parameters are small, there are no restrictions on the values they may take. Because of these properties, the approaches can be used in unconstrained optimization.

3.2 Linearizing Expressions Involving Rotation Matrices

In this section we derive a method for linearizing expressions involving rotation matrices³. Our approach is a simple first-principles Taylor approximation. To begin, we require the establishment of two identities. Euler's theorem allows us to write a rotation matrix, \mathbf{C} ,

³The contents of this section were derived in collaboration with James R. Forbes and Timothy D. Barfoot and originally appeared in Barfoot et al. (2011a).

in terms of a rotation about a unit-length axis, \mathbf{a} , through an angle, φ (Hughes, 1986),

$$\mathbf{C}(\mathbf{a}, \varphi) = \cos \varphi \mathbf{1} + (1 - \cos \varphi) \mathbf{a} \mathbf{a}^T - \sin \varphi \mathbf{a}^\times, \quad (3.18)$$

where $\mathbf{1}$ is the identity matrix. We may now take the partial derivative of $\mathbf{C}(\mathbf{a}, \varphi)$ with respect to the angle, φ :

$$\frac{\partial \mathbf{C}(\mathbf{a}, \varphi)}{\partial \varphi} = -\sin \varphi \mathbf{1} + \sin \varphi \mathbf{a} \mathbf{a}^T - \cos \varphi \mathbf{a}^\times \quad (3.19a)$$

$$= \underbrace{\sin \varphi (-\mathbf{1} + \mathbf{a} \mathbf{a}^T)}_{\mathbf{a}^\times \mathbf{a}^\times} - \cos \varphi \mathbf{a}^\times \quad (3.19b)$$

$$= -\cos \varphi \mathbf{a}^\times - (1 - \cos \varphi) \underbrace{\mathbf{a}^\times \mathbf{a} \mathbf{a}^T}_{\mathbf{0}} + \sin \varphi \mathbf{a}^\times \mathbf{a}^\times \quad (3.19c)$$

$$= -\mathbf{a}^\times \underbrace{(\cos \varphi \mathbf{1} + (1 - \cos \varphi) \mathbf{a} \mathbf{a}^T - \sin \varphi \mathbf{a}^\times)}_{\mathbf{C}(\mathbf{a}, \varphi)} \quad (3.19d)$$

Thus, our first key identity is

$$\frac{\partial \mathbf{C}(\mathbf{a}, \varphi)}{\partial \varphi} \equiv -\mathbf{a}^\times \mathbf{C}(\mathbf{a}, \varphi), \quad (3.20)$$

where

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}^\times := \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \quad (3.21)$$

defines the usual 3×3 skew-symmetric matrix, which may be used to implement the cross product for 3×1 columns (Hughes, 1986). An immediate application of this is that for any principal-axis rotation, $\mathbf{C}_\psi(\theta)$, about principal axis ψ and through angle θ , we have

$$\frac{\partial \mathbf{C}_\psi(\theta)}{\partial \theta} = -\mathbf{1}_\psi^\times \mathbf{C}_\psi(\theta), \quad (3.22)$$

where $\mathbf{1}_\psi$ is column ψ of the 3×3 identity matrix. Let us now consider an α - β - γ Euler sequence (with $\alpha \neq \beta$ and $\beta \neq \gamma$),

$$\mathbf{C}(\boldsymbol{\theta}) := \mathbf{C}_\gamma(\theta_3) \mathbf{C}_\beta(\theta_2) \mathbf{C}_\alpha(\theta_1), \quad (3.23)$$

where $\boldsymbol{\theta} := [\theta_1 \ \theta_2 \ \theta_3]^T$. Furthermore, select an arbitrary constant 3×1 column, \mathbf{v} . Applying (3.22), we have

$$\frac{\partial (\mathbf{C}(\boldsymbol{\theta}) \mathbf{v})}{\partial \theta_3} = -\mathbf{1}_\gamma^\times \mathbf{C}_\gamma(\theta_3) \mathbf{C}_\beta(\theta_2) \mathbf{C}_\alpha(\theta_1) \mathbf{v} = (\mathbf{C}(\boldsymbol{\theta}) \mathbf{v})^\times \mathbf{1}_\gamma, \quad (3.24a)$$

$$\frac{\partial (\mathbf{C}(\boldsymbol{\theta}) \mathbf{v})}{\partial \theta_2} = -\mathbf{C}_\gamma(\theta_3) \mathbf{1}_\beta^\times \mathbf{C}_\beta(\theta_2) \mathbf{C}_\alpha(\theta_1) \mathbf{v} = (\mathbf{C}(\boldsymbol{\theta}) \mathbf{v})^\times \mathbf{C}_\gamma(\theta_3) \mathbf{1}_\beta, \quad (3.24b)$$

$$\frac{\partial (\mathbf{C}(\boldsymbol{\theta}) \mathbf{v})}{\partial \theta_1} = -\mathbf{C}_\gamma(\theta_3) \mathbf{C}_\beta(\theta_2) \mathbf{1}_\alpha^\times \mathbf{C}_\alpha(\theta_1) \mathbf{v} = (\mathbf{C}(\boldsymbol{\theta}) \mathbf{v})^\times \mathbf{C}_\gamma(\theta_3) \mathbf{C}_\beta(\theta_2) \mathbf{1}_\alpha, \quad (3.24c)$$

where we have made use of the two general identities,

$$\mathbf{r}^\times \mathbf{s} \equiv -\mathbf{s}^\times \mathbf{r}, \quad (3.25a)$$

$$(\mathbf{Cs})^\times \equiv \mathbf{Cs}^\times \mathbf{C}^T, \quad (3.25b)$$

for any 3×1 columns \mathbf{r} , \mathbf{s} and any 3×3 rotation matrix, \mathbf{C} (Hughes, 1986). Combining the results in (3.24) we have

$$\begin{aligned} \frac{\partial (\mathbf{C}(\boldsymbol{\theta})\mathbf{v})}{\partial \boldsymbol{\theta}} &= \left[\frac{\partial(\mathbf{C}(\boldsymbol{\theta})\mathbf{v})}{\partial \theta_1} \quad \frac{\partial(\mathbf{C}(\boldsymbol{\theta})\mathbf{v})}{\partial \theta_2} \quad \frac{\partial(\mathbf{C}(\boldsymbol{\theta})\mathbf{v})}{\partial \theta_3} \right] \\ &= (\mathbf{C}(\boldsymbol{\theta})\mathbf{v})^\times \underbrace{\left[\mathbf{C}_\gamma(\theta_3)\mathbf{C}_\beta(\theta_2)\mathbf{1}_\alpha \quad \mathbf{C}_\gamma(\theta_3)\mathbf{1}_\beta \quad \mathbf{1}_\gamma \right]}_{\mathbf{S}(\boldsymbol{\theta})}, \end{aligned} \quad (3.26)$$

and thus our second key identity is

$$\frac{\partial (\mathbf{C}(\boldsymbol{\theta})\mathbf{v})}{\partial \boldsymbol{\theta}} \equiv (\mathbf{C}(\boldsymbol{\theta})\mathbf{v})^\times \mathbf{S}(\boldsymbol{\theta}), \quad (3.27)$$

which we note is true regardless of the choice of Euler sequence. The matrix, $\mathbf{S}(\boldsymbol{\theta})$, is the usual matrix relating angular velocity to Euler-angle rates (Hughes, 1986).

Having established identities (3.20) and (3.27), we now return to first principles and consider carefully how to linearize a rotation. If we have a function, $\mathbf{f}(\mathbf{x})$, of some variable, \mathbf{x} , then perturbing \mathbf{x} slightly from its nominal value, $\bar{\mathbf{x}}$, by an amount $\delta\mathbf{x}$ will result in a change in the function. We can express this in terms of a Taylor-series expansion of \mathbf{f} about $\bar{\mathbf{x}}$:

$$\mathbf{f}(\bar{\mathbf{x}} + \delta\mathbf{x}) = \mathbf{f}(\bar{\mathbf{x}}) + \left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\bar{\mathbf{x}}} \delta\mathbf{x} + (\text{higher order terms}) \quad (3.28)$$

This presupposes that $\delta\mathbf{x}$ is not constrained in any way. The trouble with carrying out the same process with rotations is that most of the representations involve constraints and thus are not easily perturbed (without enforcing the constraint). The notable exceptions are the three-parameter representations, the most common of which are the Euler angle sequences. These contain exactly three parameters and thus each can be varied independently. For this reason, we choose to use Euler angles in our perturbation of functions involving rotations.

Consider perturbing $\mathbf{C}(\boldsymbol{\theta})\mathbf{v}$ with respect to Euler angles $\boldsymbol{\theta}$, where \mathbf{v} is an arbitrary constant 3×1 column. Letting $\bar{\boldsymbol{\theta}} := [\bar{\theta}_1 \quad \bar{\theta}_2 \quad \bar{\theta}_3]^T$ and $\delta\boldsymbol{\theta} := [\delta\theta_1 \quad \delta\theta_2 \quad \delta\theta_3]^T$, then

applying a first-order Taylor-series approximation we have

$$\mathbf{C}(\bar{\boldsymbol{\theta}} + \delta\boldsymbol{\theta})\mathbf{v} \approx \mathbf{C}(\bar{\boldsymbol{\theta}})\mathbf{v} + \frac{\partial(\mathbf{C}(\boldsymbol{\theta})\mathbf{v})}{\partial\boldsymbol{\theta}} \Big|_{\bar{\boldsymbol{\theta}}} \delta\boldsymbol{\theta} \quad (3.29a)$$

$$= \mathbf{C}(\bar{\boldsymbol{\theta}})\mathbf{v} + ((\mathbf{C}(\boldsymbol{\theta})\mathbf{v})^\times \mathbf{S}(\boldsymbol{\theta}))|_{\bar{\boldsymbol{\theta}}} \delta\boldsymbol{\theta} \quad (3.29b)$$

$$= \mathbf{C}(\bar{\boldsymbol{\theta}})\mathbf{v} + (\mathbf{C}(\bar{\boldsymbol{\theta}})\mathbf{v})^\times \mathbf{S}(\bar{\boldsymbol{\theta}}) \delta\boldsymbol{\theta} \quad (3.29c)$$

$$= \mathbf{C}(\bar{\boldsymbol{\theta}})\mathbf{v} - (\mathbf{S}(\bar{\boldsymbol{\theta}}) \delta\boldsymbol{\theta})^\times (\mathbf{C}(\bar{\boldsymbol{\theta}})\mathbf{v}) \quad (3.29d)$$

$$= (\mathbf{1} - (\mathbf{S}(\bar{\boldsymbol{\theta}}) \delta\boldsymbol{\theta})^\times) \mathbf{C}(\bar{\boldsymbol{\theta}})\mathbf{v}, \quad (3.29e)$$

where we have used (3.27) to get to the second line. Observing that \mathbf{v} is arbitrary, we can drop it from both sides and write

$$\mathbf{C}(\bar{\boldsymbol{\theta}} + \delta\boldsymbol{\theta}) \approx \underbrace{(\mathbf{1} - (\mathbf{S}(\bar{\boldsymbol{\theta}}) \delta\boldsymbol{\theta})^\times)}_{\text{infinitesimal rot.}} \mathbf{C}(\bar{\boldsymbol{\theta}}), \quad (3.30)$$

which we see is the product of an infinitesimal rotation matrix (Hughes, 1986) and the unperturbed rotation matrix, $\mathbf{C}(\bar{\boldsymbol{\theta}})$. It is worth noting that we did not assume the perturbation is of this multiplicative form, but rather showed that it is a consequence of the linearization procedure. Notationally, it is simpler to write

$$\mathbf{C}(\bar{\boldsymbol{\theta}} + \delta\boldsymbol{\theta}) \approx (\mathbf{1} - \delta\boldsymbol{\phi}^\times) \mathbf{C}(\bar{\boldsymbol{\theta}}), \quad (3.31)$$

with

$$\delta\boldsymbol{\phi} := \mathbf{S}(\bar{\boldsymbol{\theta}}) \delta\boldsymbol{\theta}. \quad (3.32)$$

Equation (3.31) is revealing as it tells us how to perturb a rotation matrix when it appears inside any function. This may be done either in terms of perturbations to the Euler angles, $\delta\boldsymbol{\theta}$, or directly through the *rotation vector*, $\delta\boldsymbol{\phi}$.

Rotation matrices fundamentally have three degrees of freedom but are represented by nine parameters. There are therefore six constraints, which may be written as a single matrix orthogonality constraint: $\mathbf{C}\mathbf{C}^T = \mathbf{1}$. Suppose this constraint holds for $\mathbf{C}(\bar{\boldsymbol{\theta}})$. Then for the perturbed rotation matrix according to (3.31) we have

$$\begin{aligned} \mathbf{C}(\bar{\boldsymbol{\theta}} + \delta\boldsymbol{\theta})\mathbf{C}(\bar{\boldsymbol{\theta}} + \delta\boldsymbol{\theta})^T &= ((\mathbf{1} - \delta\boldsymbol{\phi}^\times) \mathbf{C}(\bar{\boldsymbol{\theta}})) ((\mathbf{1} - \delta\boldsymbol{\phi}^\times) \mathbf{C}(\bar{\boldsymbol{\theta}}))^T \\ &= \mathbf{1} - \delta\boldsymbol{\phi}^\times \delta\boldsymbol{\phi}^\times, \end{aligned} \quad (3.33)$$

which we see is correct to first order in $\delta\boldsymbol{\phi}$. For this reason, this approach to linearization may be thought of as constraint-sensitive.

Working in the other direction, suppose we have a perturbation in the form of a rotation vector, $\delta\phi$, and we wish to apply this to a prior value of the rotation, $\mathbf{C}(\bar{\theta})$. In terms of Euler angles, we would like to carry out the update

$$\theta = \bar{\theta} + \mathbf{S}(\bar{\theta})^{-1} \delta\phi. \quad (3.34)$$

However, we would prefer not to use the Euler angles, because $\mathbf{S}(\bar{\theta})^{-1}$ does not exist precisely at the associated singularities. Instead, we would like to simply store and update the rotation as a rotation matrix. The updated rotation matrix, corresponding to the updated Euler angle sequence above, is given by

$$\mathbf{C}(\theta) = \mathbf{C}\left(\bar{\theta} + \mathbf{S}(\bar{\theta})^{-1} \delta\phi\right) \quad (3.35a)$$

$$\approx \left(\mathbf{1} - \underbrace{(\mathbf{S}(\bar{\theta})\mathbf{S}(\bar{\theta})^{-1} \delta\phi)^{\times}}_1 \right) \mathbf{C}(\bar{\theta}) \quad (3.35b)$$

$$\approx (\mathbf{1} - \delta\phi^{\times}) \mathbf{C}(\bar{\theta}), \quad (3.35c)$$

where we have used (3.31), our linearized rotation matrix expression. We then make the observation that setting $\bar{\theta} = \mathbf{0}$ in this last expression reveals

$$\mathbf{C}(\delta\phi) = \mathbf{C}\left(\mathbf{0} + \underbrace{\mathbf{S}(\mathbf{0})^{-1}}_1 \delta\phi\right) \quad (3.36a)$$

$$\approx (\mathbf{1} - \delta\phi^{\times}) \underbrace{\mathbf{C}(\mathbf{0})}_1 \quad (3.36b)$$

$$\approx (\mathbf{1} - \delta\phi^{\times}). \quad (3.36c)$$

Using $\delta\phi$ as an Euler angle sequence to construct a rotation matrix, $\mathbf{C}(\delta\phi)$, is somewhat unsettling (since $\delta\phi$ are not Euler angles), but in the neighbourhood of $\bar{\theta} = \mathbf{0}$, $\delta\phi \approx \delta\theta$, so this is reasonable. In fact, *any* Euler sequence could be used to compute $\mathbf{C}(\delta\phi)$, as they all result in the same linearized expression. Substituting (3.36c) into (3.35c), we arrive at an expression for our rotation matrix update,

$$\mathbf{C}(\theta) = \mathbf{C}(\delta\phi) \mathbf{C}(\bar{\theta}), \quad (3.37)$$

where we have dropped the approximation symbol due to the fact that the rotation matrix constraint, $\mathbf{C}(\theta)\mathbf{C}(\theta)^T = \mathbf{1}$, is satisfied. This update approach allows us to store and update the rotation as a rotation matrix, thereby avoiding singularities and the need to restore the constraint afterwards (i.e., constraint restoration is built in).

The relationship between the rotation vector, $\delta\phi$, and the Euler angle perturbation, $\delta\theta$, expressed in (3.32) is algebraically equivalent to the well-known relationship between rotational velocity, ω , and Euler angle rates (Hughes, 1986):

$$\omega = \mathbf{S}(\theta)\dot{\theta} \quad (3.38)$$

In essence, may think of (3.32) as (3.38) multiplied through by an infinitesimal time increment, dt , such that $\delta\phi = \omega dt$ and $\delta\theta = \dot{\theta}dt$. This insight allows us to perturb a rotation matrix when it appears in any function by using (3.31) in terms of a rotation vector, $\delta\phi$, or by using the well-known $\mathbf{S}(\theta)$ matrix formulae to relate small changes in *any* minimal rotation parameterization, $\delta\theta$, to small changes in $\delta\phi$. An extensive list of formulae for $\mathbf{S}(\theta)$ covering many popular rotation parameterizations is available in Hughes (1986), Table 2.3 on pages 30 and 31.

The results from this section are fundamental to our approach to estimating rotation matrices that show up commonly in estimation problems in robotics. We use these results below (i) to derive similar expressions for 4×4 transformation matrices in Section 3.3, and (ii) in the example of applying a probabilistic prior term to a rotation matrix in Section 3.5.2.

3.3 Linearizing Expressions Involving Transformation Matrices

In this section we derive a similar linearized perturbation expression for 4×4 transformation matrices and develop notation and identities useful for manipulating expressions containing these quantities. Some of the identities in this section may be found in Murray et al. (1994) where they are derived using Lie algebras and applied to problems concerning the dynamics of robotic manipulators. However, they provide no handling of points at infinity and so further notation and concepts are borrowed from Hartley and Zisserman (2004), and Faugeras and Luong (2001). Here we present the material in the context of state estimation and in notation consistent with Section 3.2. We begin by introducing homogeneous coordinates and transformation matrices as a compact method of representing coordinate-frame transformations.

The projective space, \mathbb{P}^3 , is the set of equivalence classes of vectors in $\mathbb{R}^4 - \{\mathbf{0}\}$, under the equivalence relationship $\mathbf{v} \equiv s\mathbf{v}$, for a nonzero scalar, s , and a 4×1 column, \mathbf{v}

(Hartley and Zisserman, 2004; Faugeras and Luong, 2001). Informally, \mathbb{P}^3 can be thought of as the set of lines through the origin of \mathbb{R}^4 ; each element of \mathbb{P}^3 is an infinite subset of \mathbb{R}^4 , and given any point in $\mathbb{R}^4 - \{\mathbf{0}\}$, it maps to a specific element in \mathbb{P}^3 .

Given the coordinates of a point in three-dimensional Euclidean space, $\mathbf{v} =: \begin{bmatrix} x & y & z \end{bmatrix}^T \in \mathbb{R}^3$, any vector $\mathbf{v} =: \begin{bmatrix} v_1 & v_2 & v_3 & v_4 \end{bmatrix}^T \in \mathbb{P}^3$ which satisfies $\mathbf{v} = s [\mathbf{v}^T \ 1]^T$, for some real, nonzero scalar, s , is considered the *homogeneous representation*⁴ of \mathbf{v} . Hence, we define a pair of functions for moving between homogeneous (bold italic symbol) and nonhomogeneous (bold symbol) coordinates,

$$\mathbf{v} = \mathbf{h}(\mathbf{v}) := \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad \mathbf{v} = \mathbf{h}(\mathbf{v}) := \frac{1}{v_4} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}. \quad (3.39)$$

Both \mathbf{v} and \mathbf{v} encode the coordinates of the same point expressed in some coordinate frame. When $v_4 = 0$, the conversion back to \mathbb{R}^3 is not possible, as the corresponding point in \mathbb{R}^3 is infinitely far away from the coordinate frame origin. However, there is no singularity when keeping these points in homogeneous coordinates. The Jacobian matrices for $\mathbf{h}(\cdot)$ and $\mathbf{h}(\cdot)$ (needed later when linearizing error terms for batch estimation) are

$$\frac{\partial \mathbf{h}(\mathbf{v})}{\partial \mathbf{v}} = \begin{bmatrix} \mathbf{1} \\ \mathbf{0}^T \end{bmatrix}, \quad (3.40a)$$

$$\frac{\partial \mathbf{h}(\mathbf{v})}{\partial \mathbf{v}} = \frac{1}{v_4} \begin{bmatrix} \mathbf{1} & -\mathbf{h}(\mathbf{v}) \end{bmatrix}. \quad (3.40b)$$

⁴ According to Kline (1972), what we now know as homogeneous coordinates were first proposed by Augustus Ferdinand Möbius in his work entitled *Der barycentrische Calcul*, published in 1827. Möbius parameterized a point on a plane, $\mathbf{p} = [x \ y]^T$, by considering a fixed triangle within the plane and determining the masses, m_1 , m_2 and m_3 , that must be placed at the triangle vertices to make \mathbf{p} the triangle's center of gravity. Using this system, the coordinates $\mathbf{p} := [m_1 \ m_2 \ m_3]^T$ are not unique as scaling the three masses equally does not change the point location. When the equation of a curve is written in this coordinate system, it becomes homogeneous in m_1 , m_2 and m_3 —every term in the equation has the same degree. Take for example the equation of a circle centered at $[a \ b]^T$ with radius r :

$$(x - a)^2 + (y - b)^2 = r^2$$

Written in homogeneous coordinates with $x = m_1/m_3$ and $y = m_2/m_3$, the equation becomes

$$(m_1 - m_3a)^2 + (m_2 - m_3b)^2 = m_3^2r^2,$$

where every term is now quadratic in the homogeneous coordinates. Similarly, the equation of a parabola, $y = x^2$, becomes $m_2m_3 = m_1^2$.

In homogeneous coordinates, coordinate-frame transformations may be applied to points using a 4×4 transformation matrix, \mathbf{T} ,

$$\mathbf{T}_{1,0} = \begin{bmatrix} \mathbf{C}_{1,0} & \boldsymbol{\rho}_1^{0,1} \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad \mathbf{p}_1^{j,1} = \mathbf{T}_{1,0}\mathbf{p}_0^{j,0}, \quad \mathbf{T}_{1,0}^{-1} = \begin{bmatrix} \mathbf{C}_{1,0}^T & -\mathbf{C}_{1,0}^T\boldsymbol{\rho}_1^{0,1} \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad \mathbf{p}_1^{j,1} = \mathbf{T}_{0,1}^{-1}\mathbf{p}_0^{j,0}, \quad (3.41)$$

where $\mathbf{p}_0^{j,0}$ encodes the coordinates of a vector from the origin of \mathcal{F}_0 to a point j (represented by the superscript $j, 0$), and expressed in \mathcal{F}_0 (represented by the subscript 0), $\mathbf{C}_{1,0}$ is the rotation matrix that takes vectors from \mathcal{F}_0 to \mathcal{F}_1 , and $\mathbf{T}_{1,0}$ is the transformation matrix that takes points from \mathcal{F}_0 to \mathcal{F}_1 . The full complement of subscripts and superscripts is provided in (3.41) for reference. For ease of notation, in the remainder of section we will drop all subscripts and superscripts. Following an approach similar to the one used in Section 3.2, we define a column of parameters,

$$\mathbf{x} := \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\theta} \end{bmatrix}, \quad (3.42)$$

and write the transformation matrix, \mathbf{T} , as

$$\mathbf{T}(\mathbf{x}) = \begin{bmatrix} \mathbf{C}(\boldsymbol{\theta}) & \boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{bmatrix}. \quad (3.43)$$

Using the machinery developed in Section 3.2, we perturb \mathbf{x} about the nominal value,

$$\bar{\mathbf{x}} := \begin{bmatrix} \bar{\boldsymbol{\rho}} \\ \bar{\boldsymbol{\theta}} \end{bmatrix}, \quad (3.44)$$

by a perturbation

$$\delta\mathbf{x} := \begin{bmatrix} \delta\boldsymbol{\rho} \\ \delta\boldsymbol{\theta} \end{bmatrix}, \quad (3.45)$$

to get

$$\mathbf{T}(\bar{\mathbf{x}} + \delta\mathbf{x}) \approx \begin{bmatrix} (\mathbf{1} - (\mathbf{S}(\bar{\boldsymbol{\theta}})\delta\boldsymbol{\theta})^\times)\mathbf{C}(\bar{\boldsymbol{\theta}}) & \bar{\boldsymbol{\rho}} + \delta\boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad (3.46)$$

which we rearrange into the form of a multiplicative update,

$$\mathbf{T}(\bar{\mathbf{x}} + \delta\mathbf{x}) \approx \underbrace{\begin{bmatrix} \mathbf{1} - (\mathbf{S}(\bar{\boldsymbol{\theta}})\delta\boldsymbol{\theta})^\times & \delta\boldsymbol{\rho} + (\mathbf{S}(\bar{\boldsymbol{\theta}})\delta\boldsymbol{\theta})^\times\bar{\boldsymbol{\rho}} \\ \mathbf{0}^T & 1 \end{bmatrix}}_{\mathbf{T}(\bar{\mathbf{x}})} \underbrace{\begin{bmatrix} \mathbf{C}(\bar{\boldsymbol{\theta}}) & \bar{\boldsymbol{\rho}} \\ \mathbf{0}^T & 1 \end{bmatrix}}_{\mathbf{T}(\bar{\mathbf{x}})}. \quad (3.47)$$

This expression may be simplified by substituting in $\delta\phi = \mathbf{S}(\bar{\theta})\delta\theta$ and defining

$$\delta\boldsymbol{\rho} := \delta\boldsymbol{\rho} + \delta\phi^\times\bar{\boldsymbol{\rho}}, \quad (3.48)$$

to get

$$\mathbf{T}(\bar{\mathbf{x}} + \delta\mathbf{x}) \approx \begin{bmatrix} (\mathbf{1} - \delta\phi^\times) & \delta\boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{T}(\bar{\mathbf{x}}), \quad (3.49a)$$

$$= \left(\mathbf{1} - \begin{bmatrix} \delta\phi^\times & -\delta\boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{bmatrix} \right) \mathbf{T}(\bar{\mathbf{x}}). \quad (3.49b)$$

We further simplify the notation by defining the terms

$$\bar{\mathbf{T}} := \mathbf{T}(\bar{\mathbf{x}}), \quad \delta\mathbf{t} := \begin{bmatrix} \delta\boldsymbol{\rho} \\ \delta\phi \end{bmatrix}, \quad (3.50)$$

and the operator $(\cdot)^\boxplus$,

$$\begin{bmatrix} \mathbf{r} \\ \mathbf{s} \end{bmatrix}^\boxplus := \begin{bmatrix} \mathbf{s}^\times & -\mathbf{r} \\ \mathbf{0}^T & 0 \end{bmatrix}, \quad (3.51)$$

for 3×1 columns \mathbf{r} and \mathbf{s} . This allows us to write

$$\mathbf{T}(\bar{\mathbf{x}} + \delta\mathbf{x}) \approx (\mathbf{1} - \delta\mathbf{t}^\boxplus) \bar{\mathbf{T}}, \quad (3.52)$$

which may be compared to our rotation matrix result, (3.31). A similar derivation gives

$$\mathbf{T}(\bar{\mathbf{x}} + \delta\mathbf{x})^{-1} \approx \bar{\mathbf{T}}^{-1} (\mathbf{1} + \delta\mathbf{t}^\boxplus). \quad (3.53)$$

Working in the other direction, suppose we have a perturbation in the form of $\delta\mathbf{t}$, and we wish to apply this to a prior value of the transformation, $\mathbf{T}(\bar{\mathbf{x}})$. Rearranging (3.48) to get

$$\delta\boldsymbol{\rho} = \delta\boldsymbol{\rho} + \bar{\boldsymbol{\rho}}^\times\delta\phi, \quad (3.54)$$

we may write the update to \mathbf{x} as

$$\mathbf{x} = \bar{\mathbf{x}} + \begin{bmatrix} \mathbf{1} & \bar{\boldsymbol{\rho}}^\times \\ \mathbf{0} & \mathbf{S}(\bar{\theta})^{-1} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{\rho} \\ \delta\phi \end{bmatrix}. \quad (3.55)$$

However, we have again run into the situation that we would prefer not to use the Euler angles, because $\mathbf{S}(\bar{\theta})^{-1}$ does not exist precisely at the associated singularities. Instead, we would like to simply store and update the transformation as a transformation matrix.

Substituting our rotation matrix result, (3.37), into (3.49a), results in an equation for the transformation matrix update step,

$$\mathbf{T}(\mathbf{x}) = \begin{bmatrix} \mathbf{C}(\delta\boldsymbol{\phi}) & \delta\boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{T}(\bar{\mathbf{x}}), \quad (3.56a)$$

$$= \mathbf{T}(\delta\mathbf{t})\mathbf{T}(\bar{\mathbf{x}}), \quad (3.56b)$$

where we have dropped the approximation symbol due to the fact that the rotation matrix constraint, $\mathbf{C}(\boldsymbol{\theta})\mathbf{C}(\boldsymbol{\theta})^T = \mathbf{1}$, is satisfied and the resulting expression on the right-hand side is a valid transformation matrix. This update approach allows us to store and update the transformation as a transformation matrix, thereby avoiding singularities and the need to restore the constraint afterwards.

When dealing with infinitesimal transformation matrices, the $(\cdot)^\boxplus$ operator takes on a role similar to that played by the skew-symmetric operator, $(\cdot)^\times$, when dealing with infinitesimal rotation matrices. Now we examine how the perturbation, $\delta\mathbf{t}$, affects the transformation of a point, \mathbf{p} , represented in homogeneous coordinates by \mathbf{p} ,

$$\mathbf{p} := \begin{bmatrix} \mathbf{u} \\ s \end{bmatrix}, \quad \mathbf{T}\mathbf{p} = \begin{bmatrix} \mathbf{Cu} + s\boldsymbol{\rho} \\ s \end{bmatrix}, \quad (3.57)$$

where s is a nonzero scalar and $\mathbf{u} = s\mathbf{p}$, so that $\mathbf{p} = \mathbf{h}(\mathbf{p})$. Applying the perturbation, (3.52), gives us

$$\mathbf{T}\mathbf{p} \approx (\mathbf{1} - \delta\mathbf{t}^\boxplus) \bar{\mathbf{T}}\mathbf{p} \quad (3.58a)$$

$$= \bar{\mathbf{T}}\mathbf{p} - \delta\mathbf{t}^\boxplus \bar{\mathbf{T}}\mathbf{p}, \quad (3.58b)$$

with

$$-\delta\mathbf{t}^\boxplus \bar{\mathbf{T}}\mathbf{p} = \begin{bmatrix} s\delta\boldsymbol{\rho} - \delta\boldsymbol{\phi}^\times(\bar{\mathbf{C}}\mathbf{u} + s\bar{\boldsymbol{\rho}}) \\ 0 \end{bmatrix} \quad (3.59a)$$

$$= \begin{bmatrix} s\delta\boldsymbol{\rho} + (\bar{\mathbf{C}}\mathbf{u} + s\bar{\boldsymbol{\rho}})^\times\delta\boldsymbol{\phi} \\ 0 \end{bmatrix} \quad (3.59b)$$

$$= \underbrace{\begin{bmatrix} s\mathbf{1} & (\bar{\mathbf{C}}\mathbf{u} + s\bar{\boldsymbol{\rho}})^\times \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix}}_{=: (\bar{\mathbf{T}}\mathbf{p})^\boxplus} \underbrace{\begin{bmatrix} \delta\boldsymbol{\rho} \\ \delta\boldsymbol{\phi} \end{bmatrix}}_{\delta\mathbf{t}}, \quad (3.59c)$$

where we have defined the operator, $(\cdot)^\boxplus$, to be

$$\begin{bmatrix} \mathbf{s} \\ t \end{bmatrix}^\boxplus = \begin{bmatrix} t\mathbf{1} & \mathbf{s}^\times \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix}, \quad (3.60)$$

for any 3×1 column \mathbf{s} and scalar t . This demonstrates a useful identity,

$$-\mathbf{c}^\boxplus \mathbf{v} \equiv \mathbf{v}^\boxminus \mathbf{c}, \quad (3.61)$$

for any 4×1 column \mathbf{v} and 6×1 column \mathbf{c} . Using this identity we may write a first-order approximation of how a perturbation, $\delta\mathbf{t}$, produces small changes in the transformed point:

$$\mathbf{T}\mathbf{p} \approx (\mathbf{1} - \delta\mathbf{t}^\boxplus) \bar{\mathbf{T}}\mathbf{p} \quad (3.62a)$$

$$= \bar{\mathbf{T}}\mathbf{p} + (\bar{\mathbf{T}}\mathbf{p})^\boxminus \delta\mathbf{t} \quad (3.62b)$$

Similar results hold for perturbations involving \mathbf{T}^{-1} :

$$\bar{\mathbf{T}}^{-1}\mathbf{p} \approx \bar{\mathbf{T}}^{-1}(\mathbf{1} + \delta\mathbf{t}^\boxplus)\mathbf{p} \quad (3.63a)$$

$$= \bar{\mathbf{T}}^{-1}\mathbf{p} - \bar{\mathbf{T}}^{-1}\mathbf{p}^\boxminus \delta\mathbf{t} \quad (3.63b)$$

Finally, we derive some other useful identities for manipulating expressions involving transformation matrices. First we see that we can push a transformation matrix onto the other side of a perturbation:

$$\mathbf{T}\delta\mathbf{t}^\boxplus = \begin{bmatrix} \mathbf{C} & \boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \delta\phi^\times & -\delta\boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{bmatrix} \quad (3.64a)$$

$$= \begin{bmatrix} \mathbf{C}\delta\phi^\times & -\mathbf{C}\delta\boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{bmatrix} \quad (3.64b)$$

$$= \begin{bmatrix} \mathbf{C}\delta\phi^\times & -\mathbf{C}\delta\boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{bmatrix} \underbrace{\mathbf{T}^{-1}\mathbf{T}}_1 \quad (3.64c)$$

$$= \begin{bmatrix} \mathbf{C}\delta\phi^\times & -\mathbf{C}\delta\boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{C}^T & -\mathbf{C}^T\boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{T} \quad (3.64d)$$

$$= \begin{bmatrix} \mathbf{C}\delta\phi^\times\mathbf{C}^T & -\mathbf{C}\delta\phi^\times\mathbf{C}^T\boldsymbol{\rho} - \mathbf{C}\delta\boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{bmatrix} \mathbf{T} \quad (3.64e)$$

$$= \begin{bmatrix} (\mathbf{C}\delta\phi)^\times & -(\mathbf{C}\delta\phi)^\times\boldsymbol{\rho} - \mathbf{C}\delta\boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{bmatrix} \mathbf{T} \quad (3.64f)$$

$$= \begin{bmatrix} (\mathbf{C}\delta\phi)^\times & \boldsymbol{\rho}^\times\mathbf{C}\delta\phi - \mathbf{C}\delta\boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{bmatrix} \mathbf{T} \quad (3.64g)$$

$$= \left(\underbrace{\begin{bmatrix} \mathbf{C} & -\boldsymbol{\rho}^\times\mathbf{C} \\ \mathbf{0} & \mathbf{C} \end{bmatrix}}_{=: \mathbf{T}^\boxtimes} \begin{bmatrix} \delta\boldsymbol{\rho} \\ \delta\phi \end{bmatrix} \right)^\boxplus \mathbf{T} \quad (3.64h)$$

$$= (\mathbf{T}^{\boxtimes} \delta\mathbf{t})^{\boxplus} \mathbf{T} \quad (3.64i)$$

Stating this result, we have the identity

$$\mathbf{T}\mathbf{c}^{\boxplus} \equiv (\mathbf{T}^{\boxtimes} \mathbf{c})^{\boxplus} \mathbf{T}, \quad (3.65)$$

which holds for any transformation matrix \mathbf{T} and 6×1 column \mathbf{c} . This identity may alternately be written as

$$\mathbf{T}\mathbf{c}^{\boxplus} \mathbf{T}^{-1} \equiv (\mathbf{T}^{\boxtimes} \mathbf{c})^{\boxplus}, \quad (3.66)$$

for comparison with the identity, $\mathbf{C}\mathbf{s}^{\times} \mathbf{C}^T \equiv (\mathbf{Cs})^{\times}$, which is valid for any rotation matrix \mathbf{C} and 3×1 column \mathbf{s} .

The operator, $(\cdot)^{\boxtimes}$, defined as

$$\begin{bmatrix} \mathbf{C} & \boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{bmatrix}^{\boxtimes} := \begin{bmatrix} \mathbf{C} & -\boldsymbol{\rho}^{\times} \mathbf{C} \\ \mathbf{0} & \mathbf{C} \end{bmatrix}. \quad (3.67)$$

produces an invertible matrix with the property

$$\mathbf{T}^{-\boxtimes} := (\mathbf{T}^{\boxtimes})^{-1} = (\mathbf{T}^{-1})^{\boxtimes}. \quad (3.68)$$

This allows us to write (3.65) as

$$\delta\mathbf{t}^{\boxplus} \mathbf{T} \equiv \mathbf{T} (\mathbf{T}^{-\boxtimes} \delta\mathbf{t})^{\boxplus}. \quad (3.69)$$

In a bit of manipulation similar to (3.64), we can derive another useful identity:

$$\mathbf{T}\mathbf{p}^{\boxplus} = \begin{bmatrix} \mathbf{C} & \boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ s \end{bmatrix}^{\boxplus} \quad (3.70a)$$

$$= \begin{bmatrix} \mathbf{C} & \boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} s\mathbf{1} & \mathbf{u}^{\times} \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix} \quad (3.70b)$$

$$= \begin{bmatrix} s\mathbf{C} & \mathbf{Cu}^{\times} \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix} \quad (3.70c)$$

$$= \begin{bmatrix} s\mathbf{C} & (\mathbf{Cu})^{\times} \mathbf{C} \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix} \quad (3.70d)$$

$$= \begin{bmatrix} s\mathbf{1} & (\mathbf{Cu})^{\times} + s\boldsymbol{\rho}^{\times} \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} \mathbf{C} & -\boldsymbol{\rho}^{\times} \mathbf{C} \\ \mathbf{0} & \mathbf{C} \end{bmatrix} \quad (3.70e)$$

$$= \begin{bmatrix} s\mathbf{1} & (\mathbf{Cu} + s\boldsymbol{\rho})^{\times} \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} \mathbf{C} & -\boldsymbol{\rho}^{\times} \mathbf{C} \\ \mathbf{0} & \mathbf{C} \end{bmatrix} \quad (3.70f)$$

$$= \begin{bmatrix} \mathbf{C}\mathbf{u} + s\boldsymbol{\rho} \\ s \end{bmatrix}^\boxminus \begin{bmatrix} \mathbf{C} & -\boldsymbol{\rho}^\times \mathbf{C} \\ \mathbf{0} & \mathbf{C} \end{bmatrix} \quad (3.70g)$$

$$= (\mathbf{T}\mathbf{p})^\boxminus \mathbf{T}^\boxtimes \quad (3.70h)$$

This identity,

$$\mathbf{T}\mathbf{p}^\boxminus \equiv (\mathbf{T}\mathbf{p})^\boxminus \mathbf{T}^\boxtimes, \quad (3.71)$$

may also be written as

$$\mathbf{T}\mathbf{p}^\boxminus \mathbf{T}^{-\boxtimes} \equiv (\mathbf{T}\mathbf{p})^\boxminus, \quad (3.72)$$

which is again similar to $\mathbf{Cs}^\times \mathbf{C}^T \equiv (\mathbf{Cs})^\times$ and $\mathbf{Tc}^\boxplus \mathbf{T}^{-1} \equiv (\mathbf{Tc})^\boxplus$.

The results derived in this section are used to derive the estimators used in Sections 5 and 6. We provide a worked example of how to linearize a stereo camera error term in Section 3.5.1.

3.4 Linearizing Expressions Involving Homogeneous Points

One of the main benefits of using transformation matrices to represent coordinate-frame transformations is that it allows us to use homogeneous coordinates to represent points. When estimating a distant point location in Euclidean coordinates, a Gauss-Newton estimator will often attempt to push the Euclidean point out towards infinity. This causes numerical issues in the linear system of equations, (3.14b), and can cause the estimator to diverge. Homogeneous coordinates have the benefit of representing both near and distant landmarks with no singularities or scaling issues (Triggs et al., 2000). Equation, (3.62b), gives some great intuition about this. If we define the components of \mathbf{p} to be $\mathbf{p} =: [\mathbf{u}^T \ s]^T$, we may restate (3.62b) as

$$\mathbf{T}\mathbf{p} \approx \bar{\mathbf{T}}\mathbf{p} + (\bar{\mathbf{T}}\mathbf{p})^\boxminus \delta\mathbf{t} \quad (3.73a)$$

$$= \begin{bmatrix} \bar{\mathbf{C}}\mathbf{u} + s\boldsymbol{\rho} \\ s \end{bmatrix} + \begin{bmatrix} s\mathbf{1} & (\bar{\mathbf{C}}\mathbf{u} + s\boldsymbol{\rho})^\times \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{\varrho} \\ \delta\boldsymbol{\phi} \end{bmatrix}. \quad (3.73b)$$

As the Euclidean point, \mathbf{p} , represented in homogeneous coordinates, \mathbf{p} , moves away from the coordinate frame origin, s approaches zero. In the limit, we have

$$\lim_{s \rightarrow 0} (\bar{\mathbf{T}}\mathbf{p} + (\bar{\mathbf{T}}\mathbf{p})^\square \delta\mathbf{t}) \approx \lim_{s \rightarrow 0} \left(\begin{bmatrix} \bar{\mathbf{C}}\mathbf{u} + s\boldsymbol{\rho} \\ s \end{bmatrix} + \begin{bmatrix} s\mathbf{1} & (\bar{\mathbf{C}}\mathbf{u} + s\boldsymbol{\rho})^\times \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{\varrho} \\ \delta\phi \end{bmatrix} \right) \quad (3.74a)$$

$$= \begin{bmatrix} \bar{\mathbf{C}}\mathbf{u} \\ 0 \end{bmatrix} + \begin{bmatrix} (\bar{\mathbf{C}}\mathbf{u})^\times \\ \mathbf{0}^T \end{bmatrix} \delta\phi. \quad (3.74b)$$

This is a mathematical statement of what we suspect by intuition—distant landmarks only provide information about a camera’s orientation, $\delta\phi$, not its position; the homogeneous representation automatically encapsulates the different information that can be discerned from near and distant points.

Therefore, when estimating landmark locations, we would like to use a parameterization for landmarks that allows $s \rightarrow 0$. This is also advocated by Triggs et al. (2000). However, a landmark stored in homogeneous coordinates has four parameters representing three fundamental degrees of freedom. Unlike the rotation matrix case, the homogeneous representation is not subject to a constraint. Rather, it has an extra degree of freedom. For example, when estimating motion using the error term derived in Section 3.5.1, multiplying the homogeneous coordinates by a nonzero scalar will result in no change in the objective function. Unconstrained degrees of freedom like this are disastrous for the Gauss-Newton algorithm as they result in an infinite number of possible solutions to the update-step equation, (3.14b), which can cause an implementation of the algorithm to fail.

Hartley and Zisserman (2004) provide a possible solution based on a minimal parameterization of the unit sphere in \mathbb{R}^4 . They let $\boldsymbol{\vartheta}$ be a 3×1 column of point parameters. Using $\varphi := \|\boldsymbol{\vartheta}\|$ and $\mathbf{a} := \boldsymbol{\vartheta}/\varphi$, they define a map from the parameters, $\boldsymbol{\vartheta}$, onto a homogeneous point⁵

$$\mathbf{p}(\boldsymbol{\vartheta}) := \begin{bmatrix} \sin \frac{\varphi}{2} \mathbf{a} \\ \cos \frac{\varphi}{2} \end{bmatrix}. \quad (3.75)$$

⁵This representation has no singularity at $\boldsymbol{\vartheta} = \mathbf{0}$ as $\lim_{\varphi \rightarrow 0} \frac{1}{\varphi} \sin(\frac{\varphi}{2}) = \frac{1}{2}$. Grassia (1998) provides a method of computing this term that is accurate to machine precision. Let ϵ be the smallest increment represented by your floating point type, then

$$\frac{1}{\varphi} \sin(\frac{\varphi}{2}) = \begin{cases} \frac{1}{2} + \frac{\varphi^2}{48} & \text{if } \varphi \leq \sqrt[4]{\epsilon}, \\ \frac{1}{\varphi} \sin(\frac{\varphi}{2}) & \text{otherwise.} \end{cases}$$

Essentially they are trading the degree of freedom inherent in the homogeneous representation for the constraint $\mathbf{p}(\boldsymbol{\vartheta})^T \mathbf{p}(\boldsymbol{\vartheta}) = 1$. This constraint has a hidden benefit in that it keeps the entries of \mathbf{p} finite as \mathbf{p} approaches infinity (Triggs et al., 2000). This parameterization may still represent all points, but it has a singularity in that every $\boldsymbol{\vartheta}$ with $\|\boldsymbol{\vartheta}\| = 2\pi$ maps to the same point, $\begin{bmatrix} 0 & 0 & 0 & -1 \end{bmatrix}^T$. This is similar to parameterizing a rotation using Euler angles; it is minimal but has a singularity.

Our goal here is similar to the previous sections: we would like to derive a linearization method for homogeneous points that allows us to store points in homogeneous coordinates as unit-length 4×1 columns, but with some minimal (3×1) linearized perturbation that, after solution, may be turned into an update step that preserves the unit-length constraint. This goal leads us to a generalization of the linearization strategy used in the sections above.

Given a state variable, \mathbf{x} , that has constraints that must be satisfied, we may follow this general strategy for linearization:

1. Choose an update equation, which we will write $\mathbf{x} \leftarrow \mathbf{x} \oplus \delta\mathbf{x}$, where \oplus represents some (possibly nonlinear) update equation with the following properties:
 - it is *minimal*: the update parameter column, $\delta\mathbf{x}$, is a $D \times 1$ column where D corresponds to the number of underlying degrees of freedom in the state, \mathbf{x} ,
 - it is *constraint sensitive*: after the update is applied, the new value of \mathbf{x} still satisfies any constraints, and
 - it is *unconstrained*: there are no restrictions on the values $\delta\mathbf{x}$ can take and when it is small, it is far away from any singularities in the update equation.
2. Linearize the update equation about the operating point $\delta\mathbf{x} = \mathbf{0}$. This will usually result in greatly simplified, closed-form Jacobian matrices.
3. Wherever \mathbf{x} appears in an error term, substitute in the linearized update equation. Continue to linearize the expression using Taylor-series expansions of nonlinear functions.
4. Use the linearized error terms in the Gauss-Newton algorithm, solving for the optimal update step, $\delta\mathbf{x}^*$. Because the update equation does not impose constraints on $\delta\mathbf{x}$, there is no need to enforce constraints in the optimization algorithm.

5. Update the state, \mathbf{x} , using $\delta\mathbf{x}^*$ in the full nonlinear update equation chosen in Step 1.

Because the update equation was chosen to be constraint-sensitive, the new value of \mathbf{x} should still satisfy any constraints.

The difficult part of this general strategy for linearization is Step 1, finding a suitable update equation that satisfies our three conditions. In the case of homogeneous coordinates, quaternion algebra gives us precisely the tools we need in this regard⁶.

In what is to follow, a *quaternion* will be a 4×1 column that may be written as

$$\mathbf{q} := \begin{bmatrix} \boldsymbol{\epsilon} \\ \eta \end{bmatrix}, \quad (3.76)$$

where $\boldsymbol{\epsilon}$ is a 3×1 and η is a scalar. The quaternion *left-hand compound* operator, $+$, and the *right-hand compound* operator, \oplus , will be defined as

$$\mathbf{q}^+ := \begin{bmatrix} \eta \mathbf{1} - \boldsymbol{\epsilon}^\times & \boldsymbol{\epsilon} \\ -\boldsymbol{\epsilon}^T & \eta \end{bmatrix} \quad \text{and} \quad \mathbf{q}^\oplus := \begin{bmatrix} \eta \mathbf{1} + \boldsymbol{\epsilon}^\times & \boldsymbol{\epsilon} \\ -\boldsymbol{\epsilon}^T & \eta \end{bmatrix}. \quad (3.77)$$

Under these definitions, the *multiplication* of quaternions, \mathbf{q} and \mathbf{r} , which is typically written as $\mathbf{q} \otimes \mathbf{r}$ (Shuster, 1993), may be written equivalently as either

$$\mathbf{q}^+ \mathbf{r} \quad \text{or} \quad \mathbf{r}^\oplus \mathbf{q}, \quad (3.78)$$

which are both products of a 4×4 matrix with a 4×1 column. The *conjugate* operator for quaternions, -1 , will be defined by

$$\mathbf{q}^{-1} := \begin{bmatrix} -\boldsymbol{\epsilon} \\ \eta \end{bmatrix}. \quad (3.79)$$

The set of quaternions forms a *non-commutative group* under both the $+$ and \oplus operations (Shuster, 1993). The *identity element* of this group, $\boldsymbol{\iota} := [0 \ 0 \ 0 \ 1]^T$, is such that

$$\boldsymbol{\iota}^+ = \boldsymbol{\iota}^\oplus = \mathbf{1}, \quad (3.80)$$

where $\mathbf{1}$ is the 4×4 identity matrix. None of the preceding definitions require the quaternions to be of unit length. However, given two unit-length quaternions, \mathbf{q} and \mathbf{r} ,

$$\mathbf{q}^T \mathbf{q} = 1, \quad \mathbf{r}^T \mathbf{r} = 1, \quad (3.81)$$

⁶We use the notational conventions of Barfoot et al. (2011b). Please see that reference for a review of quaternion algebra, relevant identities, and historical perspective.

both the $+$ and \oplus operators preserve the unit length:

$$(\mathbf{q}^+ \mathbf{r})^T (\mathbf{q}^+ \mathbf{r}) = 1, \quad (\mathbf{q}^\oplus \mathbf{r})^T (\mathbf{q}^\oplus \mathbf{r}) = 1 \quad (3.82)$$

Consequently, we may use (3.77) and (3.75) to define a minimal, constraint-sensitive, update equation for homogeneous points,

$$\bar{\mathbf{p}} \leftarrow \mathbf{p}(\boldsymbol{\vartheta})^+ \bar{\mathbf{p}}, \quad (3.83)$$

where $\bar{\mathbf{p}}$ is the 4×1 , unit-length homogeneous point being updated, and $\boldsymbol{\vartheta}$ is a 3×1 column of update parameters. Next we linearize the update equation around the operating point $\bar{\boldsymbol{\vartheta}} = \mathbf{0}$,

$$\mathbf{p}(\boldsymbol{\vartheta})^+ \bar{\mathbf{p}} = \mathbf{p}(\bar{\boldsymbol{\vartheta}} + \delta\boldsymbol{\vartheta})^+ \bar{\mathbf{p}} \quad (3.84a)$$

$$= \mathbf{p}(\mathbf{0} + \delta\boldsymbol{\vartheta})^+ \bar{\mathbf{p}} \quad (3.84b)$$

$$\approx \left(\mathbf{p}(\mathbf{0}) + \frac{\partial \mathbf{p}(\boldsymbol{\vartheta})}{\partial \boldsymbol{\vartheta}} \Big|_{\boldsymbol{\vartheta}=\mathbf{0}} \delta\boldsymbol{\vartheta} \right)^+ \bar{\mathbf{p}}. \quad (3.84c)$$

Here we make the substitutions,

$$\mathbf{p}(\mathbf{0}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \boldsymbol{\iota}, \quad \mathbf{V} := \frac{\partial \mathbf{p}(\boldsymbol{\vartheta})}{\partial \boldsymbol{\vartheta}} \Big|_{\boldsymbol{\vartheta}=\mathbf{0}} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 \end{bmatrix}, \quad (3.85)$$

to get

$$\mathbf{p}(\boldsymbol{\vartheta})^+ \bar{\mathbf{p}} \approx (\boldsymbol{\iota} + \mathbf{V} \delta\boldsymbol{\vartheta})^+ \bar{\mathbf{p}} \quad (3.86a)$$

$$= \bar{\mathbf{p}}^\oplus (\boldsymbol{\iota} + \mathbf{V} \delta\boldsymbol{\vartheta}) \quad (3.86b)$$

$$= \bar{\mathbf{p}} + \bar{\mathbf{p}}^\oplus \mathbf{V} \delta\boldsymbol{\vartheta}, \quad (3.86c)$$

three equivalent forms of the linearized update equation. Now, to linearize an error term involving a homogeneous point, \mathbf{p} , we may substitute in any of the linearized forms on the right-hand side of (3.86). When Gauss-Newton returns the optimal update step, $\delta\boldsymbol{\vartheta}^*$, we may update the current value of the point, $\bar{\mathbf{p}}$ using (3.83). The updated value will still be of unit length. This is the parameterization used for points in Chapter 6, and so we present a full derivation of the linearized error term for a stereo camera model observing homogeneous points in the examples section below.

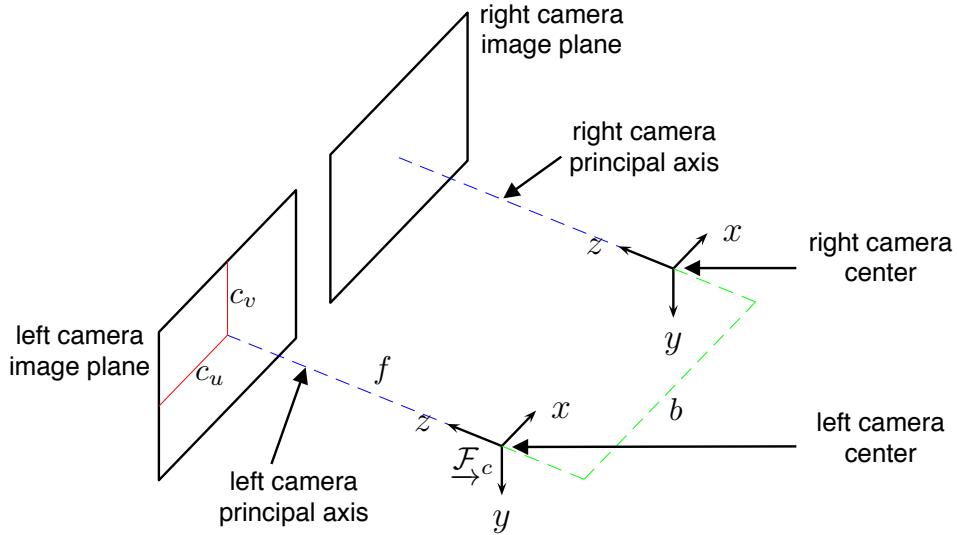


Figure 3.1: The stereo camera model for an idealized pair of cameras with equal focal lengths, parallel optical axes, and aligned image planes.

3.5 Examples

In this section we develop two worked examples using the linearization strategies outlined in this chapter. Section 3.5.1 uses the linearized transformation matrix expressions derived in Section 3.3 to develop a linearized error term for a stereo camera model, and Section 3.5.2 shows how to handle prior information terms on rotation matrices.

3.5.1 Stereo Camera Transformation Matrix Example

This section develops an example using our linearized transformation matrix expression to construct a linearized error term for a stereo camera for use in the Gauss-Newton algorithm. Before using stereo images for estimation, it is common to correct the images for lens distortion and rectify them so that they appear to have come from an idealized pair of cameras with equal focal lengths, parallel optical axes, and aligned image planes. This idealized camera model is shown in Figure 3.1. More details of this preprocessing step and sample images are given in Chapter 2. In this section we assume this preprocessing step has been used, and derive the observation model and linearized error term for an idealized stereo camera observing a point landmark.

Without loss of generality we will assume that the origin of the camera frame, $\underline{\mathcal{F}}_c$, is placed at the left camera's center. The z -axis points out of the lens, and the x - and

y -axes are aligned with horizontal and vertical pixels, respectively. The idealized stereo camera model has the following parameters:

- c_u, c_v : The horizontal and vertical coordinates of the camera's principal point in pixels (from the top left of the image)
- f_u, f_v : The horizontal and vertical focal length in pixels (these values correspond to the physical focal length, f , (mm) divided by the number of pixels per mm in either the horizontal (f_u) or vertical (f_v) direction (Hartley and Zisserman, 2004, p. 156))
- b : The camera *baseline*: the distance between the two centers of projection in meters

Let $\mathbf{p} = [x \ y \ z]^T$ be the position of a landmark expressed in \mathcal{F}_c . The nonlinear stereo observation model, $\mathbf{g}(\cdot)$, projects \mathbf{p} into the rectified images of the stereo camera:

$$\mathbf{y} = \mathbf{g}(\mathbf{p}) := \frac{1}{z} \begin{bmatrix} f_u & 0 & c_u & 0 \\ 0 & f_v & c_v & 0 \\ 0 & 0 & 0 & bf_u \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \quad (3.87)$$

Under this model, the projection, \mathbf{y} , is expressed in *disparity coordinates* (Demirdjian and Darrell, 2002). A landmark projection \mathbf{y} has components,

$$\mathbf{y} =: \begin{bmatrix} u \\ v \\ d \end{bmatrix}, \quad (3.88)$$

where u and v are respectively the horizontal and vertical pixel coordinates in the left image, and d is the *disparity*—the difference between the left and right horizontal pixel locations. These equations become linear when expressed in homogeneous coordinates⁷,

$$\mathbf{y} = \underbrace{\begin{bmatrix} f_u & 0 & c_u & 0 \\ 0 & f_v & c_v & 0 \\ 0 & 0 & 0 & bf_u \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{=: \mathbf{M}} \mathbf{p}, \quad (3.89)$$

⁷Recall from Section 3.3 that we are using bold italic symbols to represent columns in homogeneous coordinates and plain bold symbols for quantities not in homogeneous coordinates. In this case, \mathbf{p} and \mathbf{p} represent the same quantities, as do \mathbf{y} and \mathbf{y} , with (3.39) giving us $\mathbf{p} = \mathbf{h}(\mathbf{p})$ and $\mathbf{y} = \mathbf{h}(\mathbf{y})$.

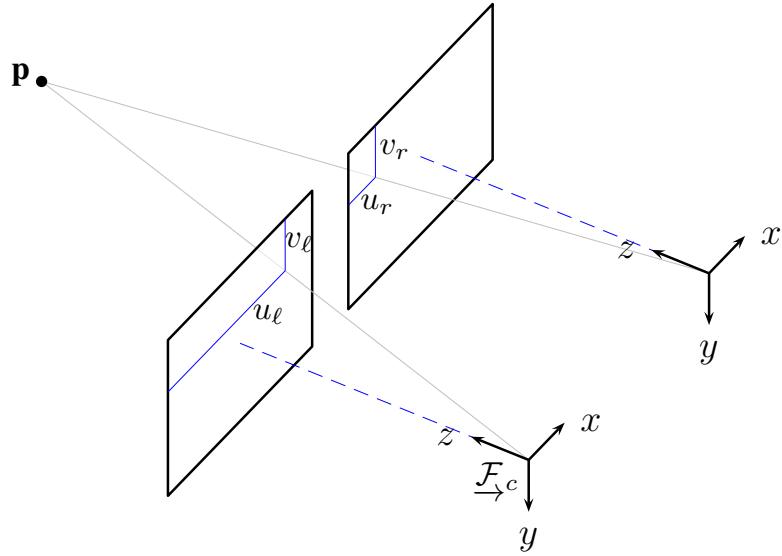


Figure 3.2: An example of a stereo camera measurement of a point landmark. In the idealized stereo camera model, $u_\ell \geq u_r$ and $v_\ell = v_r$.

and the nonlinear form is recovered using (3.39) to get

$$\mathbf{y} = \mathbf{h}(\mathbf{M}\mathbf{p}), \quad (3.90)$$

where \mathbf{M} is the invertible *stereo projection matrix*. Because \mathbf{M} is invertible, we also have

$$\mathbf{p} = \mathbf{M}^{-1}\mathbf{y}, \quad (3.91)$$

which may be used to triangulate points seen in a stereo image. If stereo measurements are made separately in the left and right images, we may define a linear transformation, \mathbf{U} , from disparity coordinates, to the predicted left and right pixel measurements,

$$\begin{bmatrix} u_\ell \\ v_\ell \\ u_r \\ v_r \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}}_{=: \mathbf{U}} \begin{bmatrix} u \\ v \\ d \end{bmatrix}, \quad (3.92)$$

where u_ℓ and v_ℓ are respectively the horizontal and vertical pixel coordinates of the measurement in the left image and u_r and v_r are the right-image coordinates, as shown

in Figure 3.2. To get back to disparity coordinates, we use the linear transformation

$$\begin{bmatrix} u \\ v \\ d \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 1 & 0 & -1 & 0 \end{bmatrix}}_{=: \mathbf{D}} \begin{bmatrix} u_\ell \\ v_\ell \\ u_r \\ v_r \end{bmatrix}. \quad (3.93)$$

Using this camera model, a typical error term for a stereo camera observing landmark j at time k would be

$$\mathbf{e}_{k,j} := \mathbf{y}_{k,j} - \mathbf{h}(\mathbf{M}\mathbf{T}_{c_k,m}\mathbf{p}_m^{j,m}), \quad (3.94)$$

where \mathcal{F}_{c_k} is the camera frame at time k , \mathcal{F}_m is some map frame, $\mathbf{p}_m^{j,m}$ is the homogeneous representation of landmark j expressed in the map frame, $\mathbf{y}_{k,j}$ is the 3×1 observation expressed in disparity coordinates, and $\mathbf{T}_{c_k,m}$ is the transformation matrix that represents the pose of the camera at time k (i.e., it transforms points from the map frame, to the camera frame at time k). The system described in Chapter 5 only produces an estimate of $\mathbf{T}_{c_k,m}$, but the follow-on work in Chapter 6 also estimates the landmark locations. For completeness we will derive the full linearized error term with respect to both the pose variables and the landmark variables. For clarity, we will drop all subscripts.

Let $\delta\mathbf{t}$ represent a 6×1 perturbation of \mathbf{T} about its nominal value, $\bar{\mathbf{T}}$, and let $\delta\boldsymbol{\vartheta}$ represent a 3×1 perturbation to \mathbf{p} about the nominal value, $\bar{\mathbf{p}}$. Applying both perturbations gives us

$$\mathbf{e} = \mathbf{y} - \mathbf{h}(\mathbf{M}\mathbf{T}\mathbf{p}), \quad (3.95a)$$

$$\approx \mathbf{y} - \mathbf{h}(\mathbf{M}(\mathbf{1} - \delta\mathbf{t}^\oplus)\bar{\mathbf{T}}(\boldsymbol{\iota} + \mathbf{V}\delta\boldsymbol{\vartheta})^+\bar{\mathbf{p}}), \quad (3.95b)$$

where we have used (3.52) to perturb \mathbf{T} , and (3.86) to perturb \mathbf{p} . Multiplying through, we drop the products of small terms (consistent with a first-order approximation):

$$\mathbf{e} \approx \mathbf{y} - \mathbf{h}(\mathbf{M}\bar{\mathbf{T}}\bar{\mathbf{p}} - \mathbf{M}\delta\mathbf{t}^\oplus\bar{\mathbf{T}}\bar{\mathbf{p}} + \mathbf{M}\bar{\mathbf{T}}(\mathbf{V}\delta\boldsymbol{\vartheta})^+\bar{\mathbf{p}}) \quad (3.96)$$

At this point, we may apply some of the identities from this chapter to get

$$\mathbf{e} \approx \mathbf{y} - \mathbf{h}\left(\mathbf{M}\bar{\mathbf{T}}\bar{\mathbf{p}} + \mathbf{M}(\bar{\mathbf{T}}\bar{\mathbf{p}})^\ominus\delta\mathbf{t} + \mathbf{M}\bar{\mathbf{T}}\bar{\mathbf{p}}^\oplus\mathbf{V}\delta\boldsymbol{\vartheta}\right) \quad (3.97)$$

where we have applied (3.61) and (3.78). Finally, letting

$$\mathbf{H} := \left. \frac{\partial \mathbf{h}(\mathbf{v})}{\partial \mathbf{v}} \right|_{\mathbf{v}=\mathbf{M}\bar{\mathbf{T}}\bar{\mathbf{p}}} \quad (3.98)$$

from (3.40b), the linearized error term becomes

$$\mathbf{e} \approx \mathbf{y} - \mathbf{h}(\mathbf{M}\bar{\mathbf{T}}\bar{\mathbf{p}}) - \mathbf{H}\mathbf{M}(\bar{\mathbf{T}}\bar{\mathbf{p}})^{\top} \delta\mathbf{t} - \mathbf{H}\mathbf{M}\bar{\mathbf{T}}\bar{\mathbf{p}}^{\top}\mathbf{V}\delta\vartheta. \quad (3.99)$$

Defining

$$\bar{\mathbf{e}} := \mathbf{y} - \mathbf{h}(\mathbf{M}\bar{\mathbf{T}}\bar{\mathbf{p}}), \quad \mathbf{E} := \begin{bmatrix} -\mathbf{H}\mathbf{M}(\bar{\mathbf{T}}\bar{\mathbf{p}})^{\top} & -\mathbf{H}\mathbf{M}\bar{\mathbf{T}}\bar{\mathbf{p}}^{\top}\mathbf{V} \end{bmatrix}, \quad \delta\mathbf{x} := \begin{bmatrix} \delta\mathbf{t} \\ \delta\vartheta \end{bmatrix}, \quad (3.100)$$

we may rewrite (3.99) as

$$\mathbf{e}(\delta\mathbf{t}, \delta\vartheta) = \bar{\mathbf{e}} + \mathbf{E}\delta\mathbf{x}, \quad (3.101)$$

which is of the form required by Gauss-Newton in (3.11). Returning to our initial problem, (3.94), one iteration of the Gauss-Newton algorithm would proceed as follows:

1. Begin with initial pose estimates, $\bar{\mathbf{T}}_{c_k,m}$, $k = 1 \dots K$, and an estimate of the position of each point, $\bar{\mathbf{p}}_m^{j,m}$, $j = 1 \dots M$.
2. For each stereo keypoint measurement, compute $\bar{\mathbf{e}}_{k,j}$ and $\mathbf{E}_{k,j}$ from $\mathbf{y}_{k,j}$, $\bar{\mathbf{T}}_{c_k,m}$, and $\bar{\mathbf{p}}_m^{j,m}$. If an individual pose is held fixed in the optimization, discard the term associated with $\delta\mathbf{t}_{c_k,m}$.
3. Build and solve equation (3.14b) for $\delta\mathbf{t}_{c_k,m}^*$ and $\delta\vartheta_j^*$, the optimal updates to the initial estimates.
4. Update each $\bar{\mathbf{T}}_{c_k,m}$ using $\delta\mathbf{t}_{c_k,m}^*$ according to the procedure in (3.56).
5. Update each $\bar{\mathbf{p}}_m^{j,m}$ using $\delta\vartheta_j^*$ according to the procedure in (3.83).
6. Check for convergence. If not converged, return to step 2.

The linearized error term derived in this section is used to solve for rover poses in Chapter 5, and to solve for rover poses and landmark locations in Chapter 6. The update steps are singularity free and the landmark parameterization handles both near and distant landmarks.

3.5.2 Rotation Matrix Priors

In this section we derive a method for applying a probabilistic prior to a rotation matrix using the results from Section 3.2. If the uncertainty is small, we may approximate a Gaussian prior on the value of a rotation matrix, \mathbf{C} , in the form,

$$\delta\varphi \sim \mathcal{N}(\mathbf{0}, \hat{\mathbf{P}}), \quad \mathbf{C} = (\mathbf{1} - \delta\varphi^{\times}) \hat{\mathbf{C}}, \quad (3.102)$$

where $\hat{\mathbf{C}}$ is the mean rotation matrix, and $\delta\varphi$ is a rotation vector representing uncertainty about the mean. The questions we address in this example are (i) given a guess for \mathbf{C} , how can we compute an error term—a 3×1 column that represents the “distance” that \mathbf{C} is from the mean rotation, $\hat{\mathbf{C}}$ —and (ii) how does this error change with respect to small changes in \mathbf{C} under the linearization strategy derived in Section 3.2? We answer these questions in order, first deriving the error term, and then linearizing the error term.

The Error Term

Given a guess for \mathbf{C} , we would like to form an error term based on our prior belief from (3.102). To understand our error term derivation for rotation matrices, it is useful to understand the method of applying a prior to a typical state variable (i.e., not a rotation). So, given a state variable \mathbf{x} , a Gaussian prior belief may be written as

$$\delta\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \hat{\mathbf{P}}), \quad \mathbf{x} = \hat{\mathbf{x}} + \delta\mathbf{x}, \quad (3.103)$$

where $\hat{\mathbf{x}}$ is the mean and $\delta\mathbf{x}$ represents our uncertainty about the mean. Given a guess for \mathbf{x} , we may form a prior error term by defining

$$\mathbf{e}(\mathbf{x}) := \delta\mathbf{x} = \mathbf{x} - \hat{\mathbf{x}}, \quad (3.104)$$

which gives us

$$E[\mathbf{e}(\mathbf{x})] = E[\delta\mathbf{x}] = \mathbf{0}, \quad E[\mathbf{e}(\mathbf{x})\mathbf{e}(\mathbf{x})^T] = E[\delta\mathbf{x}\delta\mathbf{x}^T] = \hat{\mathbf{P}}. \quad (3.105)$$

This error term becomes part of our maximum-likelihood cost function as

$$J_{\text{prior}}(\mathbf{x}) := \frac{1}{2}\mathbf{e}(\mathbf{x})^T\hat{\mathbf{P}}^{-1}\mathbf{e}(\mathbf{x}). \quad (3.106)$$

Following a similar strategy for rotation state variables, we define $\mathbf{e}(\mathbf{C}) := \delta\varphi$. Substituting this into (3.102) we get

$$(\mathbf{1} - \mathbf{e}(\mathbf{C})^\times)\hat{\mathbf{C}} \approx \mathbf{C} \quad (3.107a)$$

$$(\mathbf{1} - \mathbf{e}(\mathbf{C})^\times) = \mathbf{C}\hat{\mathbf{C}}^T. \quad (3.107b)$$

Assuming that $\mathbf{C}\hat{\mathbf{C}}^T$ is small⁸, it may be approximated by a rotation vector ψ as,

$$\mathbf{C}\hat{\mathbf{C}}^T \approx (\mathbf{1} - \psi^\times). \quad (3.108)$$

⁸Indeed, all of the linear approximations that we are using to handle rotation matrices require the quantities to be small.

Substituting (3.108) into (3.107b) gives us

$$(\mathbf{1} - \mathbf{e}(\mathbf{C})^\times) \approx (\mathbf{1} - \boldsymbol{\psi}^\times), \quad (3.109)$$

which implies

$$\mathbf{e}(\mathbf{C}) = \boldsymbol{\psi}. \quad (3.110)$$

Unlike the standard case derived in (3.103)–(3.106), the computation of the error, $\boldsymbol{\psi}$, is a nonlinear operation that takes a rotation matrix (that we hope is near identity) and computes a 3×1 column. The choice of how to perform this conversion determines the form of the error, and subsequently the form of the linearized error term derived below.

We choose to use the inverse of our rotation-matrix update equation. In this update equation, (3.37), we use a rotation vector to build a rotation matrix by using the entries of the vector as Euler angles. As we showed in Section 3.2, this approximation holds (to first order) as long as the rotation vector is small. The inverse of this process, which we will write as $\boldsymbol{\psi} \leftarrow \mathbf{C}\hat{\mathbf{C}}^T$, involves extracting a 3×1 column of Euler angles, $\boldsymbol{\psi}$, from the rotation matrix $\mathbf{C}\hat{\mathbf{C}}^T$. This is a nonlinear operation that must be linearized in order to use this error term in Gauss-Newton.

The Linearized Error Term

To produce the error term, (3.110), we have used the inverse of our rotation-matrix update equation, which extracts Euler angles, $\boldsymbol{\psi}$, from the rotation matrix, $\mathbf{C}\hat{\mathbf{C}}^T$. For a Gauss-Newton iteration, we start with a guess, $\bar{\mathbf{C}}$, and make the approximation, (3.31), that

$$\mathbf{C} \approx (\mathbf{1} - \delta\phi^\times) \bar{\mathbf{C}}, \quad (3.111)$$

for some small update step, $\delta\phi$. Substituting this into our error function, (3.110), we would like to derive an expression, linear in $\delta\phi$, that describes how small changes in $\delta\phi$ become small changes in $\mathbf{e}(\cdot)$. In the notation of Section 3.1, this would be:

$$\mathbf{e}((\mathbf{1} - \delta\phi^\times) \bar{\mathbf{C}}) \approx \mathbf{e}(\bar{\mathbf{C}}) + \mathbf{E}\delta\phi \quad (3.112)$$

In fact, we know the answer to this question from (3.34), which describes how a perturbation in the form of a rotation vector results in small changes in the Euler angles representing a rotation (to first order). From (3.34), we have

$$\mathbf{e}(\delta\phi) = \bar{\boldsymbol{\psi}} + \mathbf{S}(\bar{\boldsymbol{\psi}})^{-1} \delta\phi, \quad (3.113)$$

where $\bar{\psi} \leftarrow \bar{\mathbf{C}}\hat{\mathbf{C}}^T$. Hence, the linearized error term in our objective function becomes

$$J_{\text{prior}}(\delta\phi) := \frac{1}{2}\mathbf{e}(\delta\phi)^T \hat{\mathbf{P}}^{-1} \mathbf{e}(\delta\phi), \quad (3.114)$$

which is of the form required by Gauss-Newton in (3.11). A prior error term of this form is used to smooth out the localization estimates in Section 5.2.3.

3.6 Conclusion

This chapter has presented a first-principles approach to linearizing expressions involving rotations represented by 3×3 rotation matrices, coordinate-frame transformations represented by 4×4 transformation matrices, and Euclidean points represented as unit-length 4×1 columns. The linearization approach was demonstrated through two examples: (i) linearizing a stereo-camera-model error term, and (ii) forming and linearizing a prior information term on a rotation matrix. Without listing contributions stemming from the collaborative work in Barfoot et al. (2011b), we believe the contributions of this chapter are:

1. A first-principles derivation of the multiplicative constraint-sensitive perturbations of a 4×4 transformation matrix and a unit-length 4×1 homogeneous point given by (3.52) and (3.86) respectively. These may be used to linearize any expression involving a transformation matrix, or homogeneous point.
2. Expressions for updating transformation matrices and unit-length homogeneous points with a constraint-sensitive perturbation are provided in (3.56), and (3.83) respectively. These updates avoid the need to restore constraints afterwards.
3. Development of a number of identities for manipulating expressions involving linearized transformation matrices. These identities are given in (3.61), (3.65), (3.66), (3.69), (3.71), and (3.72).
4. Demonstration of linearizing a stereo-camera-model error term involving a transformation matrix and homogeneous point landmark. The resulting linearized error term, given by (3.99) is used for the VO estimates in subsequent chapters.
5. Demonstration of how to build and linearize an error term representing a Gaussian prior on a rotation matrix. The linearized error term is given in (3.114).

In all cases, the parameterizations we propose are *minimal*, in the sense that the update parameterization has the same number of degrees of freedom as the underlying state variable, they are *constraint sensitive* in that the equation used to update the state variable preserves constraints on the state, and they are *unconstrained* in that, as long as the update parameters are small, there are no restrictions on the values they may take. This allows us to use unconstrained optimization methods without fear of hitting singularities in our underlying representations.

Chapter 4

Coupled Surface/Subsurface Modelling

The use of GPR together with a stereo camera on planetary exploration rover has been proposed several times (Barfoot et al., 2003; Vago et al., 2006; Fong et al., 2008). Used together, surface and subsurface imaging will aid in the search for subsurface ice deposits and evidence of life. On such a mission, cameras would be used for site selection and survey, a GPR would be used to characterize the subsurface structure, and taken together, the surface and subsurface data would be used to select locations for drilling.

Despite this interest, there are still several open issues regarding the use of GPR on a rover platform:

1. Rovers must be able to deliver information about the surface (topography, substrate particle size distribution, and/or the presence of any existing outcrops) that enables the operator to give local geological context to the subsurface data. Surface and subsurface information must be coupled so that the science team can visualize the spatial relationship between surface features identified in images and subsurface features seen in GPR data.
2. For a more complete interpretation of GPR data, the radargram (a two-dimensional subsurface profile) should be corrected for topography (e.g., Busby and Merritt (1999); Lunt and Bridge (2004); Cassidy (2009); Annan (2009)). As planetary exploration rovers have no access to a GPS equivalent, topographic profiles must be generated using other on-board sensors.
3. A flight-ready GPR antenna must satisfy size, mass and power consumption con-

straints and the integration must minimize interference from the rover's metal chassis.

The contribution of this chapter is to address items 1 and 2 by using stereo imagery to enhance GPR data. Specifically, we propose an architecture that enables the automated construction of coupled surface/subsurface models using the specific sensors planned for deployment on near-term robotic exploration missions. Stereo cameras have been deployed on the Mars Exploration Rovers (Maki et al., 2003) and are planned for both the Mars Science Laboratory (Malin et al., 2005) and the ExoMars Mission (Vago et al., 2006). The modelling pipeline uses the VO algorithm described in Chapter 2 to fully automate the GPR data collection procedure and produce two novel data products: (i) a two-dimensional, topography-corrected GPR radargram with the surface topography plotted above, and (ii) a three-dimensional, photo-realistic surface model coupled with a ribbon of GPR data. Each result is derived from only the on-board sensors of the rover, as would be required in a planetary exploration setting.

On Earth, producing a site survey using GPR on rough terrain involves several steps:

1. The GPR antenna is dragged along a transect to collect many vertical GPR traces.
2. The position of the GPR at each trace is determined using traditional surveying techniques or the Differential Global Positioning System (DGPS) integrated directly with the GPR.
3. GPR processing software is used to correct the horizontal spacing and vertical offset of the GPR traces along the transect using the position information from the previous step.
4. The corrected traces are concatenated into a raster image called a *radargram*.

Our approach automates this process by using VO in the place of GPS, thereby enabling the technology to be used in a planetary exploration context.

These techniques have been tested using data gathered at two sites near the Haughton-Mars Project Research Station (HMP-RS) on Devon Island, Nunavut, Canada (Lee et al., 2007). The sites exhibit polygonally-patterned ground, a periglacial landform often indicative of subsurface ice deposits (as discussed in Chapter 1). Stereo images were captured during GPR transects and our integrated surface/subsurface modelling techniques were applied to the resulting data.

The rest of this chapter is organized as follows. Related work regarding the development of GPR for planetary exploration is discussed in Section 4.1. Our integrated surface/subsurface modelling technique is described in Section 4.2. We describe our method of two-dimensional radargram correction in Section 4.2.1 and our three-dimensional surface/subsurface modelling technique in Section 4.2.2. Sections 4.3 and 4.4 outline our field tests on Devon Island and the associated results. Our conclusions are provided in Section 4.5.

4.1 Related Work

Perhaps the most common geophysical tool used in terrestrial geological investigations, GPR relies on differences in subsurface materials' dielectric permittivity to image the layering of different materials beneath the sensor. The GPR transmitter emits a high-energy electromagnetic pulse into the ground at frequencies generally in the range of 10-1000 MHz (Degenhardt and Giardino, 2003). When the signal encounters an interface between layers of differing permittivity, part of the energy is reflected back towards the surface while the remainder is refracted onwards. The reflection/refraction process continues until the signal has attenuated completely or the user-defined time window—the amount of time that the GPR receiver is programmed to search for a return signal—has elapsed (Moorman et al., 2003). Based on the two-way travel time of each reflected pulse, a trace is produced illustrating a series of reflector intensities located beneath the unit. Within the trace, the amplitude of the reflection is proportional to the relative difference in permittivity between adjacent materials (Arcone et al., 1995). When the GPR survey is conducted along a surface transect, individual traces can be combined to produce a radargram, a two-dimensional profile showing continuous subsurface reflective layers.

The imaging of the subsurface of Mars has to date been performed by orbital sounding radars—the Mars Advanced Radar for Subsurface and Ionosphere Sounding (**MARSIS**) on Mars Express (Picardi et al., 2005), and the Shallow Subsurface Radar (**SHARAD**) on Mars Reconnaissance Orbiter (Seu et al., 2007). These instruments have revealed subsurface information to kilometers and hundreds of meters depths, respectively, at resolutions that reveal large-scale features such as buried impact craters, 250 kilometers across (Picardi et al., 2005). Although the scans returned by **SHARAD** have an order of magnitude better depth resolution than those collected by **MARSIS** (Seu et al., 2007), it is still too coarse to pinpoint sampling locations for surface-rover drilling operations. As

such, GPR deployed on a surface rover is a logical complement to the currently-available orbital data.

The application of GPR to frozen terrain was pioneered by Annan and Davis (1976) (cf. Ross et al. (2005)) and is becoming increasingly widespread. Given its established utility in some of Earth’s most extreme environments such as Antarctica (Arcone et al., 2002) and the Canadian Arctic (dePascale et al., 2008), rover-based GPR has thus been proposed for development on a variety of planetary missions (Vago et al., 2006). While previous studies have focused primarily on hardware development and testing (Grant et al., 2003; Kim et al., 2006; Leuschen et al., 2002), understanding the physics of dielectric signal loss in Mars-type substrates (Pettinelli et al., 2007), and possible applications to Mars analogue environments (Arcone et al., 2002; Degenhardt and Giardino, 2003; Williams et al., 2005), little effort has been directed towards the integration of subsurface data with surface-based distance estimates (Barfoot et al., 2003). Because the interpretation of GPR data relies heavily on knowledge of the survey’s spatial orientation and relation to surface features, it is important that any autonomous GPR study have such a capability. Previous advances automating GPR data acquisition have used DGPS (Rial et al., 2005; Fong et al., 2008), a laser theodolite (Lehmann and Green, 1999), or proprioception (Freese et al., 2007)¹ to track the pose of the GPR during data collection. To the best of our knowledge, this work is the first to use VO-derived motion estimates to align GPR data.

4.2 Integrated Surface/Subsurface Modelling

This section will describe our integrated surface/subsurface modelling system. Data flow through the main processing blocks of our system can be seen in Figure 4.1. An attentive reader will notice that the top part of Figure 4.1 is simply the VO pipeline described in Chapter 2. Our modelling system uses the motion estimate derived from VO, the raw stereo images, the GPR data, and some calibration information to produce two novel data products that may be used for scientific study: a two-dimensional topography-corrected radargram, and a three-dimensional coupled surface/subsurface model.

¹Freese et al. (2007) used joint angle encoders to track the pose of a GPR attached to the end of a robotic manipulator.

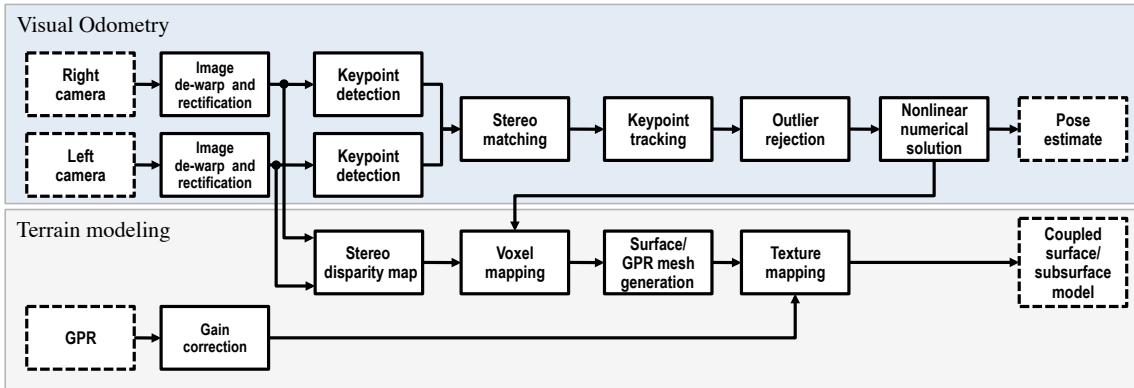


Figure 4.1: An overview of the major processing blocks of our system. A [VO](#) pipeline (top) processes stereo image pairs to solve for the pose of the camera at every image. The pose estimate is used by the terrain modelling pipeline (bottom) to create a three-dimensional coupled surface/subsurface model.

4.2.1 Two-Dimensional Topographic Correction and Surface Profile

When the [GPR](#) transect of interest lies along sloping or undulating terrain, it is typical to perform a topographic survey using standard survey instruments such as [DGPS](#) (Lunt and Bridge, 2004) or an engineer’s level (Busby and Merritt, 1999). The motion estimate from [VO](#) can be used for the same purpose. The height of the antenna at each trace is interpolated from the motion estimate and this is used to apply a topographic correction to the transect.

Additionally, given only the sparse feature points used to produce the motion estimate, we can plot the points that fall near to the path of the antenna and fit a spline that describes the surface. The surface spline improves on the topographic correction as it is able to capture narrow features over which the antenna slides. This extra data allows the user to account for the random clutter field that surface roughness produces in the [GPR](#) data (Daniels, 2004). A sample surface profile is shown in Figure 4.2.

4.2.2 Three-Dimensional Surface/Subsurface Modelling

Surface Mesh Generation

Our three-dimensional, photo-realistic terrain models are generated by passing our motion estimate into the Instant Scene Modeler ([ISM](#)) developed at MDA Space Missions (Barfoot et al., 2006; Se and Jasiobedzki, 2008). At time k we use the sum-of-absolute-

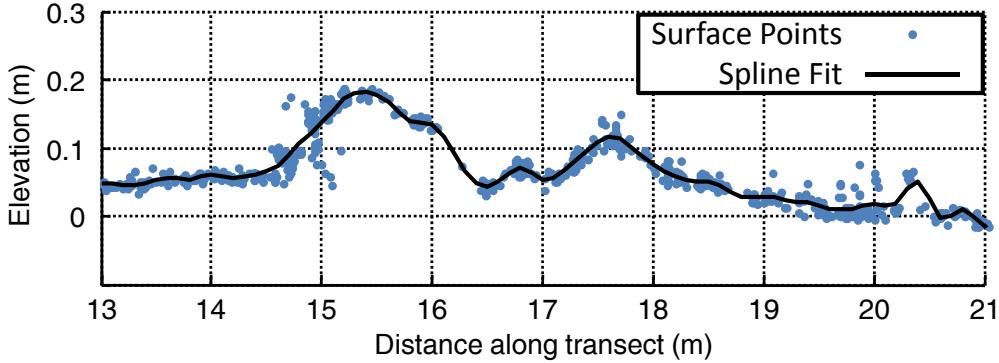


Figure 4.2: Fitting a spline to the surface points (reconstructed from the stereo images) along the GPR antenna’s path produces a high-resolution surface profile. This captures narrow terrain features that may not be present in the motion estimate (due to the antenna’s size having a smoothing effect) and provides further information about surface roughness along the transect.

differences, correlation-based, dense-stereo algorithm to generate a set of M_k scene points expressed in the camera frame, $\underline{\mathcal{F}}_{c_k}$:

$$\{\mathbf{p}_{c_k}^{i,c_k} \mid i = 1 \dots M_k\} \quad (4.1)$$

The motion estimate from VO provides the transformation, \mathbf{T}_{m,c_k} , which takes points from the camera frame at time k to some fixed mapping frame, $\underline{\mathcal{F}}_m$:

$$\mathbf{p}_m^{i,m} = \mathbf{T}_{m,c_k} \mathbf{p}_{c_k}^{i,c_k} \quad (4.2)$$

Aligning dense point clouds from multiple views fills in holes in the data and provides us with the rich volumetric information we need to create a surface mesh. Dense-stereo methods are often used for terrain assessment (Lacroix et al., 2001; Biesiadecki and Mai-mone, 2006) and so the point clouds may be available onboard at no extra computational cost. However, using every point obtained from stereo processing in the terrain model is not efficient. There are many redundant measurements and, due to incorrect matches, occlusions, or lack of texture, the data may contain noise and missing regions. A triangular mesh representation reduces the amount of data when multiple point sets are combined and thus also reduces the amount of bandwidth needed to send the resulting models offboard (e.g., to Earth). Creating a surface mesh fills up small holes and eliminates outliers, resulting in smoother and more realistic reconstructions.

To generate triangular meshes from the merged point set, we employ a voxel-based method (Roth and Wibowo, 1997), which accumulates three-dimensional points with

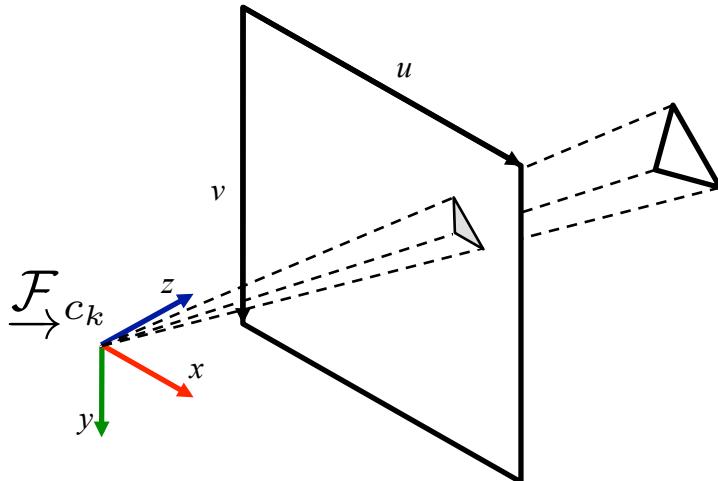


Figure 4.3: The photo-realism of the surface model is achieved by mapping the original colour input images on to the surface mesh. Texture coordinates (u, v) for each triangle in the terrain model are found using (3.90) to project the triangle's vertices into the colour images captured by the rover.

their associated normals. The points are accumulated into voxels at each frame. Outliers are filtered out using their local orientation and by selecting the threshold of range measurements required per voxel for a valid mesh vertex. It takes a few seconds to construct the triangular mesh, which is dependent on the data size and the voxel resolution.

The photo-realistic appearance of the reconstructed scene is created by texture mapping the triangular mesh using the same stereo images used for motion estimation. Texture-mapped surfaces are more visually appealing and easier to interpret as they provide additional surface details. The vertices of each triangle in the mesh are projected into the images using (3.90) as shown in Figure 4.3. As each triangle may be observed in multiple images, one image must be selected to provide the texture for each triangle. The algorithm selects the smallest set of images that will cover all of the triangles. This decreases the amount of bandwidth required to transmit the models, reduces the appearance of texture discontinuities between adjacent triangles, and makes the scene easier to load and view.

Scenes produced by ISM are exported as Virtual Reality Modelling Language (VRML) files and may be viewed in any VRML viewer. The meshes are exported in metric scale based on the motion estimate and the points derived from stereo processing. As such, the resulting three-dimensional model provides a rich record of the site and may be used for measurement of visible surface features (Osinski et al., 2010).

GPR Mesh Generation

After estimating the rover’s motion and building a terrain model, we couple the subsurface data to the surface model. The data from the traces along a rover traverse are associated with the rover’s motion estimate through common time-stamps. With this information, we can build a 3D model of the GPR data.

To display the subsurface data with the surface model, we build a mesh for the GPR data. In a static configuration, we define the left camera frame (discussed in Section 3.5.1) as $\underline{\mathcal{F}}_c$ and the frame attached to the GPR antenna as $\underline{\mathcal{F}}_a$. Calibration of the rover determines $\mathbf{T}_{c,a}$, the static transformation between $\underline{\mathcal{F}}_a$ and $\underline{\mathcal{F}}_c$. The depth of each trace, d , is calculated from the measured signal return time and the estimated velocity of the radar pulse through the subsurface (Hinkel et al., 2001). The electromagnetic wave velocity typical for ice-rich substrates in permafrost is approximately 120 m/ μ s (Fortier and Allard, 2004). The mesh created in Section 4.2.2 is represented in frame $\underline{\mathcal{F}}_m$ and so we create the GPR mesh in the same frame. For each time k in the motion estimate, we create two vertices,

$$\mathbf{v}_{k,0} = \mathbf{T}_{m,c_k} \mathbf{T}_{c,a} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{v}_{k,1} = \mathbf{T}_{m,c_k} \mathbf{T}_{c,a} \begin{bmatrix} 0 \\ 0 \\ -d \\ 1 \end{bmatrix}, \quad (4.3)$$

representing the antenna position ($\mathbf{v}_{k,0}$) and the position at the bottom of the trace ($\mathbf{v}_{k,1}$). For each time $2 \leq k \leq M$, the polygon, $\{\mathbf{v}_{k-1,0}, \mathbf{v}_{k-1,1}, \mathbf{v}_{k,1}, \mathbf{v}_{k,0}\}$, is added to the mesh. The geometry of this polygon construction is shown in Figure 4.4. Using the VO-derived motion estimate to build both the surface model and the subsurface mesh results in a coupled model by construction; all points in both meshes are expressed in the same coordinate frame, $\underline{\mathcal{F}}_m$.

Texture coordinates are assigned to the vertices based on the time-stamp association between the rover poses and the traces in the GPR radargram. Texture mapping the GPR data onto this mesh goes one step beyond standard topographic correction. Because the full six-degree-of-freedom pose of the rover is used to generate the mesh, the texture-mapped traces are corrected for tilt, topography and wheel slip. Future work could involve using the samples from collocated scans to perform automated dense GPR reconstruction as in Lehmann and Green (1999).

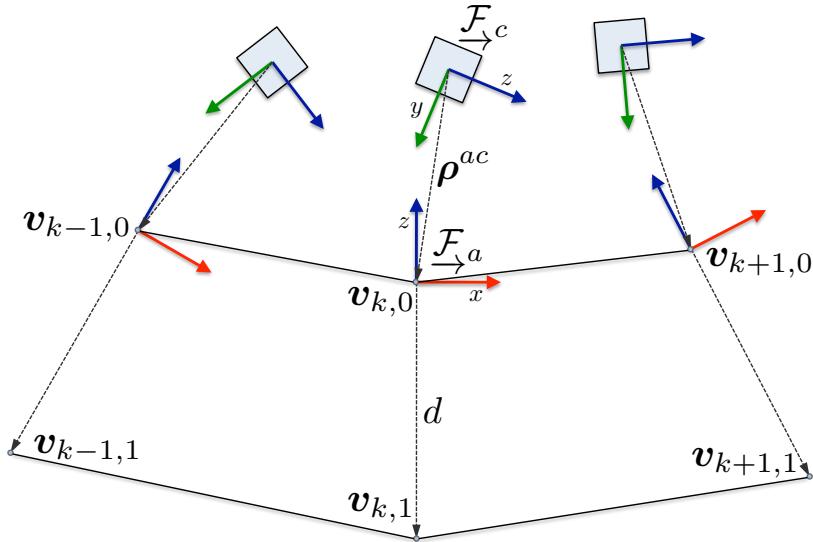
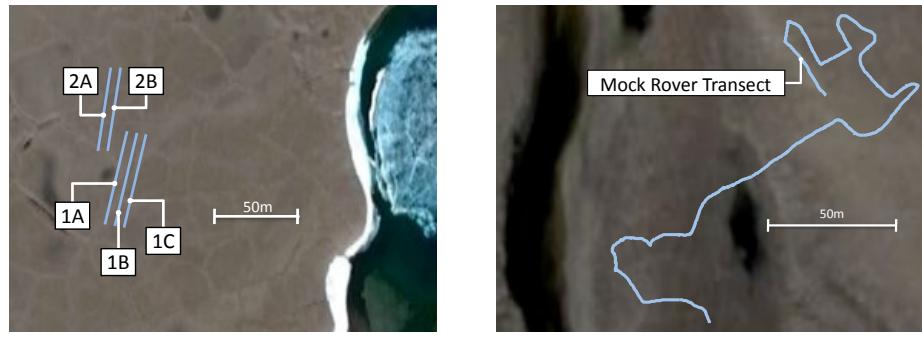


Figure 4.4: The subsurface model is built by reconstructing the trajectory of the GPR antenna over the length of the traverse. From the six-degree-of-freedom motion estimate, the calibration transformation, $\mathbf{T}_{c,a}$, and the scan depth, d , we construct a three-dimensional polygonal mesh, which is texture-mapped with a radargram of the input GPR data. The resulting model goes beyond a typical topographic correction as the traces are corrected for wheel slip, tilt, and topography.

4.3 Field Testing

The experiments described in this chapter were conducted on Devon Island in the Canadian High Arctic, as part of the Haughton-Mars Project (Lee et al., 2007). The HMP-RS is situated just outside the northwest area of the Haughton impact crater, which is located at 75°22' N latitude and 89°41' W longitude. Haughton presents unique qualities for planetary analogue studies because it offers an unusually wide variety of geological features and microbiological attributes of strong planetary analogue value or potential (Cabrol et al., 2010). Haughton is also in a polar desert environment, which presents real challenges to field exploration that are analogous in fundamental ways to those expected in planetary exploration. This site has been used for rover testing in the past (Wettergreen et al., 2002, 2005; Fong et al., 2007, 2008).

Our experiments were conducted approximately 10 kilometers northeast of HMP-RS near Lake Orbiter. This site was selected based on ongoing research into the polygonal terrain it hosts. Image sequences from the stereo camera and GPR data were logged at two sites:



(a) The Lake Orbiter Transects. (b) The Mock Rover Transect.

Figure 4.5: Locations and transects on Devon Island, Nunavut, Canada used for field testing our integrated surface/subsurface modelling technique.

1. The Lake Orbiter Transects: Five straight-line transects were taken at the Lake Orbiter site (Figure 4.5(a)). Each transect is approximately 60 meters long.
2. The Mock Rover Transect: One approximately 357 meter transect at a site that had not been previously studied (Figure 4.5(b)).

The two sites selected for investigation are comprised primarily of poorly sorted angular clasts² ranging from centimeters to tens of centimeters in size. The polygonal shapes measure a few meters to tens of meters between subsequent troughs, with individual troughs averaging approximately 1–2 meters in width and tens of centimeters in depth.

In our experiments, a rover was simulated using the push-cart shown in Figure 4.6. The push-cart was equipped with a stereo camera, a GPR, an on-board computer, and two independent GPS systems (one Real-Time Kinematic) used for ground-truth positioning. Although this was not an actuated rover, our focus in this work is on problems of estimation, and thus it was entirely sufficient as a means to gather data. The GPR (and cart) we used was a Sensors&Software Noggin 250 MHz system (Barfoot et al., 2003). Efforts were made to minimize the effect of the rover body on the GPR data quality (e.g., using plastic parts where possible). The stereo camera was a Point Gray Research Bumblebee XB3 with a 24 cm baseline and 70° field of view, mounted approximately 1 meter above the surface pointing downward by approximately 20°. Each image of the stereo pair was captured at 1280 × 960 pixel resolution.

²Pointy, broken-up bits of rock.

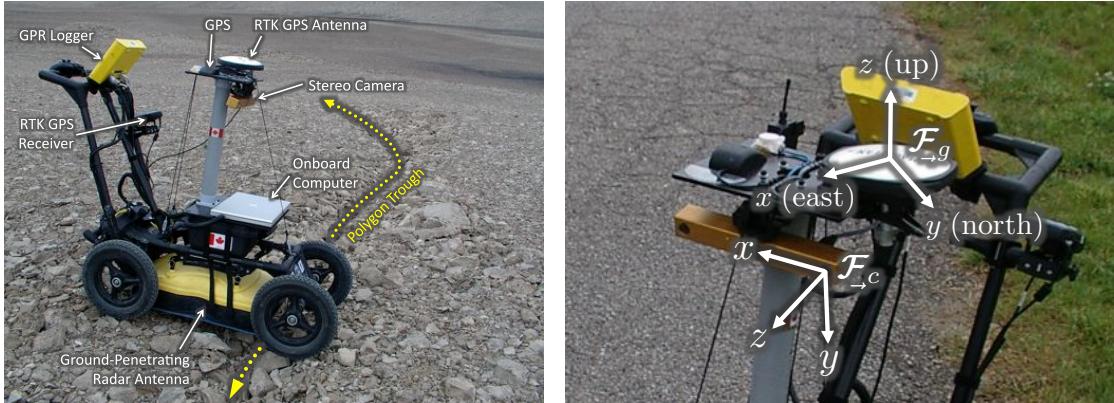


Figure 4.6: The data collection platform used in this study was PC-1, a pushcart rover outfitted with the rover engineering sensors needed to develop and evaluate our coupled surface/subsurface modelling pipeline. This figure shows (left) PC-1 positioned on a polygon trough at our test site at Lake Orbiter on Devon Island, Nunavut and (right) the coordinate frames needed in this chapter.

4.4 Results

Here we present the results of our coupled surface/subsurface modelling pipeline. We first evaluate the VO algorithm against DGPS, and then present the data products produced by our pipeline.

4.4.1 Visual Odometry

The VO algorithm described in Chapter 2 was used to process all data collected at the Lake Orbiter site. We used a real-time kinematic GPS unit as groundtruth for our motion estimate. The GPS frame is depicted in Figure 4.6. The x -axis is aligned with lines of longitude (east positive), the y -axis is aligned with the lines of latitude (north positive) and the z -axis points up. The inclinometers were used to determine the initial pitch and roll of the rover. However, as the data were collected near the north magnetic pole, we were unable to use a compass to determine the initial heading of the vehicle. We determine the initial heading through a least-squares fit of the estimated track to the GPS for a small number of poses at the start of the traverse. These poses are then discarded and are not used when evaluating the linear position error. This is similar to the method used by Nistér et al. (2004).

The results are shown in Table 4.1, which lists the distance travelled and errors for all datasets collected. On the short Lake Orbiter transects (50 to 60 meters), position

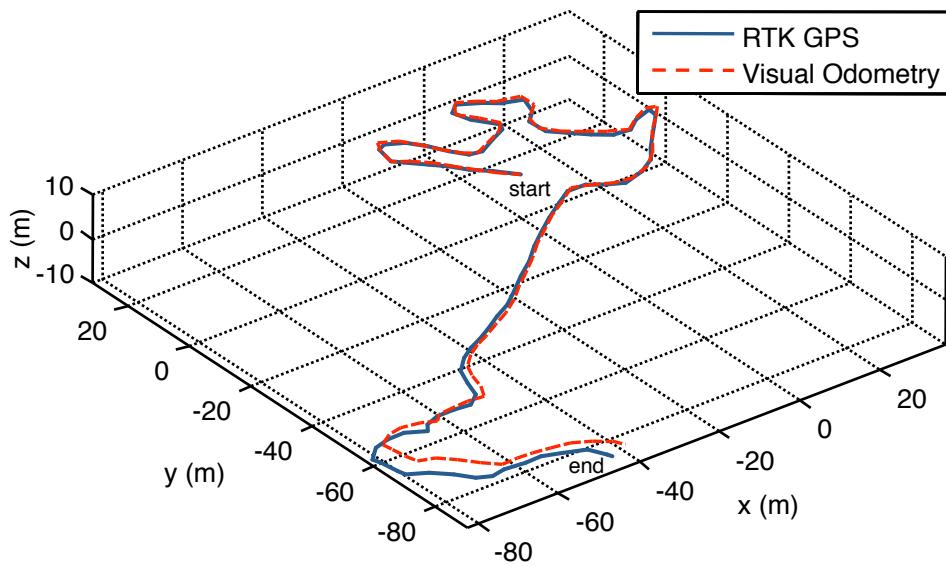


Figure 4.7: Track plots of the GPS and VO estimate on the 357 meter Mock Rover Transect. Linear position error at the end of this traverse was 5.8 meters, or 1.6% of distance travelled.

Table 4.1: VO motion estimate results from the mock rover transect and several short polygon transects. The transect names correspond to the labels in Figure 4.5. The additional *N* or *S* indicates that the transect was collected with the rover travelling North or South.

Transect	Length (m)	Linear Error (m)	% Error	Images
Mock Rover	357.30	5.83	1.63%	4818
poly-1A-S	54.23	2.87	5.29%	333
poly-1B-N	59.63	0.68	1.14%	316
poly-1B-S	60.06	1.24	2.06%	317
poly-1C-N	60.67	1.98	3.26%	327
poly-2A-N	51.49	1.04	2.01%	270
poly-2A-S	50.16	0.25	0.51%	263
poly-2B-N	49.47	1.16	2.34%	260
poly-2B-S	49.05	0.96	1.96%	258

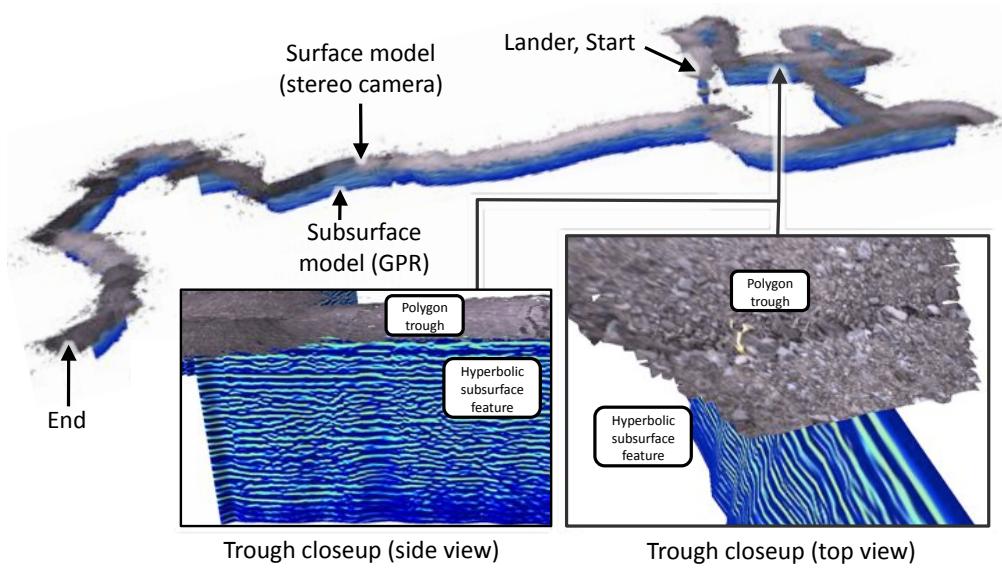


Figure 4.8: Views of the key data product in this chapter: a coupled surface/subsurface model built from stereo imagery and a GPR. The texture-mapped triangle mesh of the surface is displayed above the ribbon of GPR data. The model may be rendered from any viewpoint using off-the-shelf computer components and OpenGL. The inset figures show possible views of one of the polygon troughs along the transect. The coupled model provides situational awareness, allowing operators to explore the interaction of surface topography and subsurface structure even though they are working remotely.

errors were no greater than 5.3% of distance travelled. The results of the estimation on the Mock Rover Transect are plotted in Figure 4.7. The estimate accumulated 1.63% position error over this 357.3 meter traverse.

4.4.2 Coupled Surface/Subsurface Models

The complete coupled surface/subsurface model of the Mock Rover Transect is shown in Figure 4.8. The texture-mapped triangle mesh of the surface is displayed above the ribbon of GPR data. The model may be inspected using a Virtual Reality Modelling Language viewer and rendered from any viewpoint using off-the-shelf computer components and OpenGL.

Figure 4.9 shows the entire corrected GPR radargram produced from data collected at the Mock Rover site. As is evident, the application of topographic corrections provides a more realistic representation of subsurface structure with respect to variations in surface elevation than would an uncorrected model.

GPR is particularly useful for identifying subsurface ice because of the vast differences

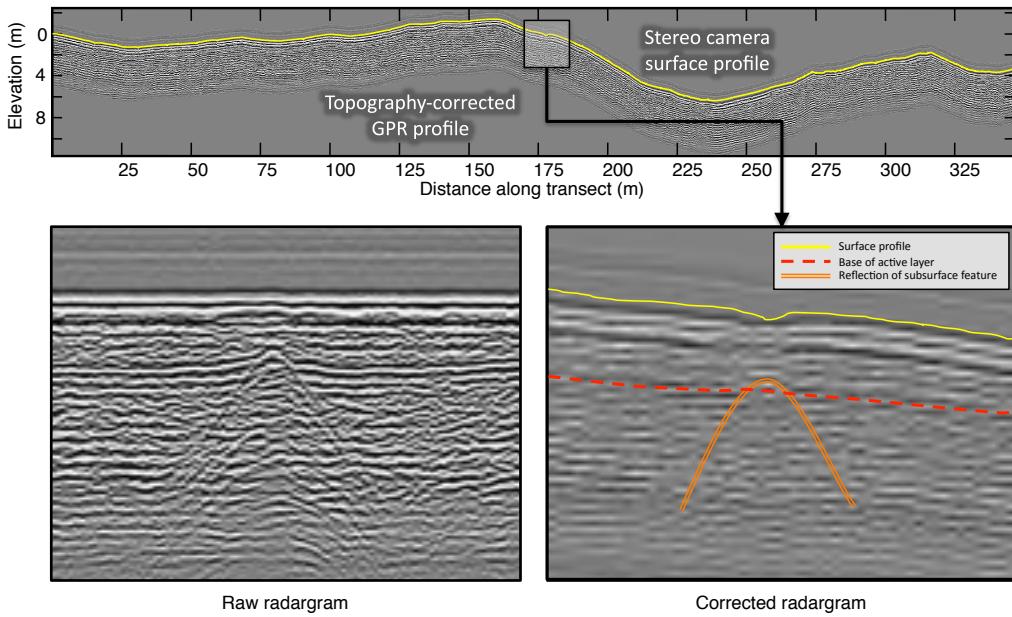


Figure 4.9: The complete corrected GPR radargram of the Mock Rover site (above), produced using the process described in Section 4.2.1. The surface profile derived from stereo processing is plotted in solid/yellow. The bottom plots compare the difference between the corrected and uncorrected radargram for a portion of the data near a large polygon trough. The corrected radargram shows that the base of the seasonally-thawed *active layer* (dashed/red) is nearly flat and that a dip in the surface topography corresponds to a subsurface feature, shown here by a hyperbolic reflection pattern (double-line/orange). As is evident, the topographic correction and surface profile provide further context to aid in the interpretation of the subsurface structure.

in dielectric properties displayed by ice, sediment, water, and air (Milsom, 2003). In ice-rich sediments, it is quite easy to detect the base of the seasonally thawed *active layer* (Moorman et al., 2003; dePascale et al., 2008), because of the immediate increase in permittivity that occurs due to freezing at the top of the permafrost (Scott et al., 1990). In Figure 4.9, a noticeable continuous reflector virtually mirroring the surface topography is located approximately 1 meter beneath the surface along the transect length. This depth corresponds well with the permafrost table located by augering at the highlighted section on Figure 4.9, and thus the reflection is interpreted as the base of the active layer.

In addition to detecting the frozen/unfrozen unconformity within the ground, a boundary between a discrete ground ice body and its enclosing sediments should produce a strong distinguishable reflector because of a sharp decrease in permittivity at the sediment/ice interface (Moorman and Michel, 2000). However, although ice wedges are discrete ice bodies, properly imaging their structure using GPR is not straightforward.

As noted by Hinkel et al. (2001, p. 187), ice wedges “produce exceedingly complex,

high amplitude hyperbolic reflections” due to the conical shape of the emitted GPR pulse. As a result, while ice wedges themselves are roughly triangular in shape—wider at the top and progressively narrowing with depth—their appearance on a radargram more resembles an inverted hyperbola (e.g. Hinkel et al. (2001); Fortier and Allard (2004)).

Figure 4.9 illustrates an example of a hyperbolic subsurface reflection detected within the radargram. At this and other locations along the transect, the hyperbolic reflectors are found immediately beneath the troughs as indicated by small V-shaped depressions in the stereo camera surface profile. Because polygon troughs are the most obvious surface expression of ice wedge locations (Mackay, 1999), the successful coupling of our surface/subsurface model is further supported.

4.5 Conclusion and Future Work

We have presented a coupled surface/subsurface modelling method that uses only a rover-mounted stereo camera and GPR. Specifically, we believe the contributions of this chapter are:

1. Demonstration and field testing of a method to completely automate the GPR data collection process by using stereo VO to derive an estimate of the motion of the GPR antenna over the course of a transect. This method uses only onboard sensors slated to fly on future rover missions, making it suitable for a planetary exploration setting.
2. Demonstration and field testing of a method to use the raw data from a stereo camera and GPR along with the VO motion estimate to produce two novel data products:
 - (a) a topography-corrected radargram plotted with a two-dimensional profile of the surface along the transect (Section 4.2.1), and
 - (b) a photo-realistic three-dimensional surface/subsurface model (Section 4.2.2).

These techniques allow operators to work remotely, surveying the data as if they were on site. The data products allow subsurface structure to be examined in the context of the surface structure, a key scientific technique used by field geologists to identify sites worthy of further study.

Before a GPR is deployed on a flight platform, the significant issue of GPR/rover integration must be addressed (Barfoot et al., 2003). The proximity of the rover to the GPR will produce signal corruption due to the reflections from the metal chassis and electromagnetic signals generated by the actuators. Attempts to shield the antenna from these effects will increase the mass of the payload and may also produce interference at the receiving antenna (Annan, 2009). In short, this is a major issue that must be addressed before rover-based GPR data collection is deployed in space.

Chapter 5

Autonomous Retrotraverse

This chapter describes a complete system for long-range, autonomous operation of a mobile robot in outdoor, unstructured environments. This is achieved using *only* a stereo camera for sensing and a teach-and-repeat operational strategy. During a learning phase—the *teach pass*—the rover is piloted over the desired route (either manually or using some external autonomous system), while the mapping system builds a manifold map composed of overlapping local submaps. The topologically-connected submaps are then used for localization during the autonomous traversal phase—the *repeat pass*. Because our goal was to travel back and forth over long distances where no loops in the path are present, we chose not to formulate the problem as **SLAM**. Rather, the hybrid topological/metric formulation alleviates the requirement for an accurate global reconstruction, while avoiding the unbounded error growth of purely relative motion estimation. Similar systems have been proposed for rovers navigating indoors (Goedeme et al., 2005), in mines (Marshall et al., 2008), or outdoors in planar environments (Royer et al., 2007; Šegvić et al., 2009), but this is the first system shown to work over multi-kilometer autonomous traverses in highly three-dimensional, outdoor, unstructured environments, without the use of **GPS**.

Our visual teach-and-repeat system is appropriate to many scenarios requiring rover autonomy where **GPS** is not available. We envision such a system being used to support Mars-sample-return (Figure 5.1), or automated convoying of equipment between lunar landing sites and habitats. Hence we have tested our system at a Mars/Moon analogue site on Devon Island in the Canadian High Arctic. For completeness, we have also performed tests in an urban setting, over grass, over challenging three-dimensional terrain, and through extreme lighting changes (indoor to outdoor). This chapter reports results



Figure 5.1: We envision our teach-and-repeat navigation framework being used to support Mars-sample-return mission operations (left image credit: NASA/JPL). After sample acquisition, the rover would retrace its route, returning to the lander in a single command cycle. The image on the right shows our rover driving back along its outbound track in a planetary analogue setting on Devon Island, Canada.

for over 32 kilometers of evaluation of our algorithm with 99.6% of the distance travelled autonomously, all without the use of GPS.

5.1 Related Works

In an early paper on vision-based map building, Brooks (1985) outlined some basic principles for robotic mapping:

- The world is inherently three-dimensional. Localization and mapping should reflect this.
- Uncertainty in sensing will lead to maps that are globally inconsistent. However, to enable robot autonomy, maps only need to be locally consistent.

To deal with this, he proposed a map composed of an abstract graph of free-space primitives. Similar in concept, Howard et al. (2006) designed and implemented a multi-agent system that represented the robot's map as a manifold embedded in a higher-dimensional space. Manifold mapping changes the way a map represents the world. A map becomes topological in the sense that it defines a sequence of connected spaces, but the spaces in the map may have a many-to-one correspondence with the world. This topology is represented by dividing the map into a graph of submaps (Bosse et al., 2004; Howard

et al., 2006; Marshall et al., 2008; Eade and Drummond, 2008), or using a continuous relative representation (Sibley et al., 2009; Mei et al., 2009). Incremental errors that would cause inconsistencies in a purely metric map disappear within the manifold representation. As a result, loop-closing decisions may be delayed (Howard et al., 2006) and loops may be closed in constant time, regardless of the size of the map (Sibley et al., 2009). Manifold mapping removes the constraint that maps be globally consistent, but in order to be useful for localization, the neighbourhood around the robot must still appear locally Euclidean.

To see where this local-Euclidean constraint expresses itself in the **SLAM** problem we examine the structure of the basic **SLAM** equations. The **SLAM** problem is formulated probabilistically as the task of estimating the joint posterior density of the map, \mathbf{m} , and vehicle state at time k , \mathbf{x}_k , given all previous measurements, $\mathbf{z}_{0:k}$, control inputs, $\mathbf{u}_{0:k}$, and prior knowledge, \mathbf{x}_0 (Durrant-Whyte and Bailey, 2006):

$$p(\mathbf{x}_k, \mathbf{m} | \mathbf{z}_{0:k}, \mathbf{u}_{0:k}, \mathbf{x}_0) \quad (5.1)$$

Most solutions to this problem involve computing $p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m})$, the likelihood of the measurement vector, \mathbf{z}_k , given the current state and map estimates. The likelihood is then expressed using an observation model, $\mathbf{g}(\cdot)$, such that

$$\mathbf{z}_k = \mathbf{g}(\mathbf{x}_k, \mathbf{m}) + \mathbf{v}_k, \quad (5.2)$$

where \mathbf{v}_k is observation noise. The properties of (5.2) determine the form of the constraint. Most navigation sensors discern something about the geometry in the robot's local neighbourhood and, for a map to be useful, the neighbourhood must appear Euclidean to the sensor suite; any deviation must be small enough to hide in \mathbf{v}_k . This is the motivation behind the adaptive optimization scheme in Sibley et al. (2009), and the choice of submap size in Marshall et al. (2008). If this constraint is satisfied, the map is still useful for localization, even if the global reconstruction is inaccurate.

Visual teach-and-repeat navigation systems have been built on this very concept, combining topologically-connected keyframes with a controller that attempts to drive the robot to the same viewpoints along the path. Our review of teach-and-repeat systems will focus on camera-based systems. Marshall et al. (2008) and Newman et al. (2002) both used planar laser-ranging devices to build teach-and-repeat systems in indoor corridors. The systems are well suited to these environments (an underground mine in the former, and an office building in the latter), but in outdoor, unstructured environments, there is

no guarantee that any walls will be within the range of a laser sensor. Cameras, on the other hand, are not dependent on scene topography. They require only ambient light and scene texture to return useful images. Wide field-of-view and omnidirectional cameras capture the large-scale geometry and appearance of a scene, which is generally unique to a particular viewpoint and somewhat robust to small-scale changes in the scene. In this way cameras are well suited to recognize places previously visited.

Early work in visual teach-and-repeat navigation recognized the key benefit of such a system: an accurate global reconstruction is not needed for a robot to retrace its path (Baumgartner and Skaar, 1994; Brooks, 1985). Systems implementing teach and repeat span the continuum of map-based approaches, as categorized by Bonin-Font et al. (2008). Different systems frame the problem as purely metric (Baumgartner and Skaar, 1994; Royer et al., 2007), combined topological/metric (Šegvić et al., 2009; Goedemé et al., 2007), or purely topological (i.e., only track the position along the path) (Zhang and Kleeman, 2009; Matsumoto et al., 1996). Our system can be described as topological/metric. Localization is performed in three-dimensional space, path tracking is performed in a local planar projected space, and route management is topological.

Appearance-based systems compare large portions of the input image with prototype images captured during the teach pass. These algorithms derive from the work of Matsumoto et al. (1996). They developed a system for autonomous navigation within corridors. During the route-learning phase, an ordered sequence of images was captured using a monocular camera. During route repeating, progress along the sequence is tracked. A template from the center of each new image is correlated with the two nearest images along the route. The steering angle is determined from the correlation peak offset, and the current image index is incremented when the next image returns a higher correlation score than the previous. Jones et al. (1997) extend this basic framework and introduce a second camera to more accurately track the position along the route, while Payá et al. (2007) try to make the process more efficient using principle component analysis. The most impressive demonstration of appearance-based path following was developed by Zhang and Kleeman (2009). They report over 18 km of tests using an omnidirectional imaging system. Position along the route and steering angle determination are similar to Matsumoto et al. (1996), but significant image preprocessing is performed to make the system robust to changing lighting conditions. All of the appearance-based techniques rely on the assumption of planar camera motion. They are therefore not suitable for outdoor unstructured environments.

Another group of algorithms uses image features for mapping and navigation, but relies on the planarity of the camera’s motion to reduce the complexity of the problem. Ohno et al. (1996) use a monocular camera and a bearing-only, two-dimensional localization when navigating between prototype images. Tang and Yuta (2001) develop a similar system for a robot with an omnidirectional camera. They use colour information to describe line features and planar localization based on the bearing of the line correspondences. The algorithm described by Bekris et al. (2006) and Argyros et al. (2005) tracks point features between omnidirectional images. Instead of triangulating the features, they only use the bearing of the measurements and develop a control law to drive the robot between viewpoints. Jun et al. (2000) describe an algorithm that uses range measurements from a stereo camera to detect obstacles, which are projected down to a plane and used for localization while repeating the route. Blanc et al. (2005) developed a system that followed indoor visual routes. As the camera was facing the ceiling, the system could solve three-degree-of-freedom homographies using features tracked between exemplar images and images taken from the robot’s current position. Courbon et al. (2008) extended this work to use an omnidirectional camera. Chen and Birchfield (2006) developed a homing system that uses a KLT point tracker on images captured from a forward-facing monocular camera system. Stored points from the training run are matched with points on the repeat run and all matched points contribute to a simple visual servoing scheme. Goedemé et al. (2005) improve the process of extracting line features by making every part of the algorithm invariant to changes in illumination and viewpoint (assuming the camera is restricted to moving in the plane). They also move to use point features detected using SIFT (Lowe, 2004). During the map-building phase, features are triangulated between views and their three-dimensional coordinates are stored in the map. Three-dimensional localization against the map is performed by observing features and estimating the essential matrix of the camera. Owing to the introduction of local metric 3D information derived from point features, this algorithm, and the similar one described by Booij et al. (2007), could be adapted to work with nonplanar camera motion.

Developing a teach-and-repeat system for outdoor, unstructured environments requires the handling of nonplanar camera motions. Using point image features for localization removes the planarity constraint and enables localization in three dimensions as required by our system. There has been some previous work in this area using forward-facing monocular cameras. The work of Royer et al. (2007) represents one approach

to this task. During the mapping phase, point features detected in a monocular image sequence are tracked between images, and data from the entire route is subject to a large, multi-level estimation routine to find the feature positions and the robot poses. The poses become a reference path and the features are used as a map. To repeat the route, features in the current image are associated with features in the map and used to estimate the rover’s position. In contrast to this global reconstruction approach, Šegvić et al. (2009) develop a system that performs many local reconstructions during the mapping phase, using two-view geometry to triangulate feature points seen in a monocular image sequence. During the repeat traverse, the rover’s three-dimensional pose is estimated using the triangulated features. Interestingly, the three-dimensional pose is only used to localize the robot topologically; the steering angle is derived from a simple visual servoing rule similar to that used by Chen and Birchfield (2006). These algorithms are most similar to the one we propose. However, we use a stereo camera as the main sensor.

A stereo camera provides metric structure within each stereo pair of images, simplifying the reconstruction considerably. A very simple teach-and-repeat mode was built into the stereo navigation system described by Konolige et al. (2009). During a mapping phase, the rover’s path is estimated using VO (Konolige et al., 2007). To repeat the route, they estimate the rover’s position at the start of the route by matching the current view with the first image in the learning sequence. Then the route is repeated without relocalizing against the map. While this method worked for the short paths in question (generally less than 200 meters), longer routes would require localization corrections to maintain global consistency.

We show that it is possible to use stereo vision alone to retrace a long route with nonplanar camera motion in an outdoor, unstructured environment. Our work is based on the VO pipeline described in Chapter 2, but we transform the basic pipeline into a complete mapping and localization system and demonstrate that our algorithm can be used to drive multi-kilometer autonomous routes in a single command cycle.

5.2 System Overview

This section will present a detailed description of our visual teach-and-repeat system. The major processing blocks of our system are depicted in Figure 5.2.

The coordinate frames used in our system are depicted in Figure 5.3. The map frame $\underline{\mathcal{F}}_m$ is the frame in which all three-dimensional estimation occurs. We define $\underline{\mathcal{F}}_{c_k}$ to be

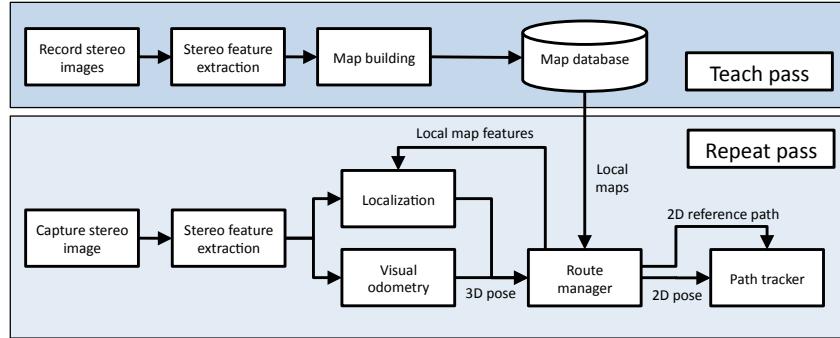


Figure 5.2: An overview of the major processing blocks in our system.

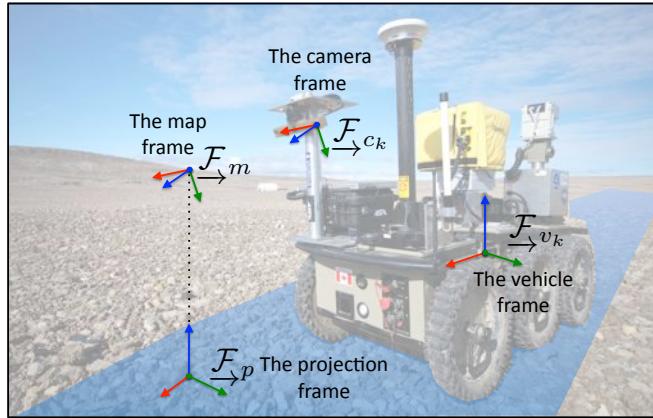


Figure 5.3: Coordinate frames under consideration.

a coordinate frame attached to the left camera of a stereo pair at time k . The attitude of the camera at this time may be described by \mathbf{C}_{m,c_k} , the rotation matrix that transforms vectors from \mathcal{F}_{c_k} to \mathcal{F}_m . Similarly, we define the camera's position as $\rho_m^{c_k,m}$, a vector from the origin of \mathcal{F}_m to the origin of \mathcal{F}_{c_k} (denoted by the superscript), and expressed in \mathcal{F}_m (denoted by the subscript). These may be combined into the transformation matrix, \mathbf{T}_{m,c_k} . Using similar notation, we define a rotation, \mathbf{C}_{v_k,c_k} , and translation, $\rho_{v_k}^{c_k,v_k}$, that may be combined to form the transformation, \mathbf{T}_{v_k,c_k} , between the camera frame, \mathcal{F}_{c_k} , and vehicle frame, \mathcal{F}_{v_k} . This transformation is assumed to be static but it could be time-varying. Finally, a projection frame, \mathcal{F}_p , is defined for each submap. All quantities of interest (the path, landmarks, and the vehicle's current localization) may be projected into the xy -plane of \mathcal{F}_p for a two-dimensional view of the current state, as required by our path-tracking controller.

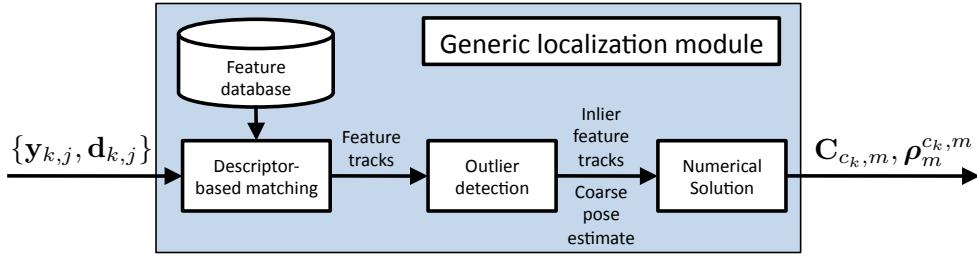


Figure 5.4: An overview of our generic localization module.

5.2.1 A Generic Localization Module

Throughout this work, we use a generic localization module based on the stereo VO algorithm described in Chapter 2. The outline is shown in Figure 5.4. Stereo keypoints are tracked against a feature database, the tracks are subject to outlier detection, and the inlying tracks are used to solve for the current pose of the camera. By substituting different blocks for the feature database and numerical solution, we are able to build all of the different operating modes used for teach-and-repeat navigation: *map building*, *VO*, *submap selection*, and *localization*. We will refer back to this section as we specify the details used in these operating modes. Here we present the specific requirements of each block.

A feature database represents a map against which the robot can localize. To this end, it supplies information about the set of features available for this task:

N : The number of features in the database

$\mathbf{p}_m^{i,m}$: The homogeneous coordinates of feature i with respect to and expressed in \mathcal{F}_m

\mathbf{d}_i : The SURF descriptor associated with feature i

Data association is performed by looking for nearest neighbours in descriptor space. Using this notation and that of Chapter 2, the output of the first block in Figure 5.4 is a list of candidate feature tracks, each associating a feature i in the database to a feature j from the most recent stereo pair.

The candidate tracks are passed to the outlier detection block. We have implemented preemptive RANSAC (Nistér, 2005), as it will on average produce the best set of inliers given a fixed computational budget. Treating the feature database and the incoming stereo keypoints as three-dimensional point clouds (using (3.91) to triangulate each feature), we use the three-point quaternion method of Horn (1987) as our hypothesis

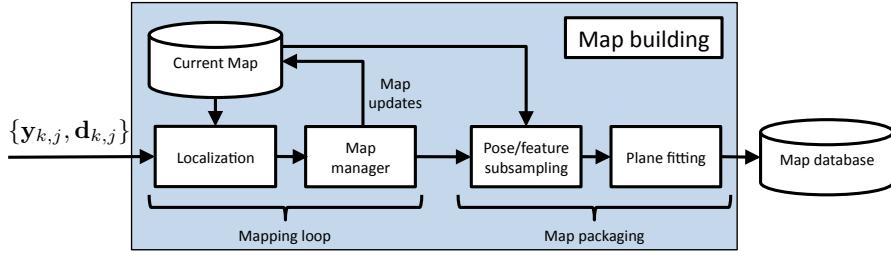


Figure 5.5: An overview of the mapping process.

generator. Preemptive RANSAC generates a set of inlying feature tracks and a coarse estimate of the camera's pose in \mathcal{F}_m .

Finally, the inlying feature tracks are passed to a pose solution method. The pose solution has access to the disparity coordinates of each incoming keypoint, the feature database, the pose estimate supplied by RANSAC, and the camera's pose from the last timestep. Using this data it produces an estimate of the camera's pose in \mathcal{F}_m ,

$$\mathbf{T}_{c_k,m} := \begin{bmatrix} \mathbf{C}_{c_k,m} & \boldsymbol{\rho}_{c_k}^{m,c_k} \\ \mathbf{0}^T & 1 \end{bmatrix}. \quad (5.3)$$

Each solution method is iterative, based on Gauss-Newton minimization, but each operating mode uses a different mathematical formulation based on the requirements of the operating mode. The specific objective function used for each mode is described below.

5.2.2 Route Teaching

The basic process for route teaching involves driving the path once while logging stereo images, and then post-processing the image sequence into a series of overlapping submaps. The post-processing task is shown in Figure 5.5. At the front, a mapping loop incrementally builds the map and estimates the position of the rover within it. Periodically, the map is split, and the raw data is further processed into the format used in the repeat pass.

Teach Pass Localization and Mapping

The mapping loop seems to be solving the SLAM problem. However, the different requirements of this system dictate different design choices. Each submap must be locally consistent and transformations between adjacent submaps must be reasonable. Outside of these constraints, the overall global consistency of the map sequence should not impact

algorithm performance. Because of this, the system does not work toward global consistency. Figure 5.6 shows an example of a five-kilometer map sequence compared to GPS together with the robot's view of the map from either end. Although the reconstruction of the complete path is very inaccurate, locally it is sufficient to enable route following.

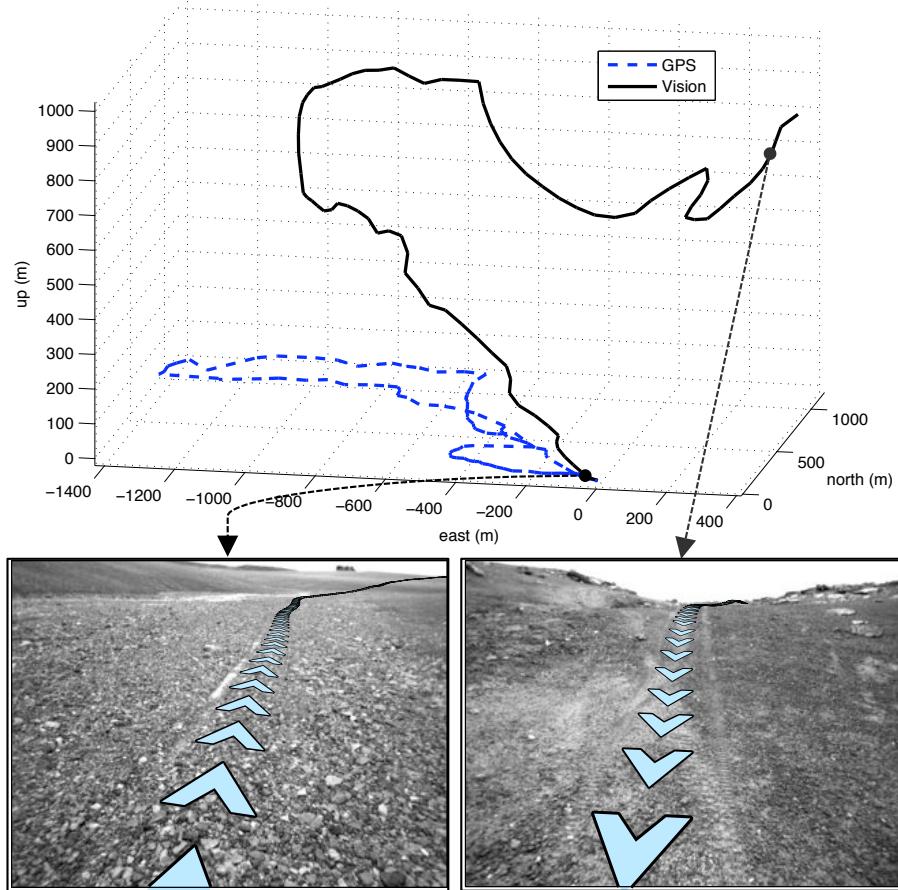


Figure 5.6: The visual reconstruction of a five-kilometer rover traverse plotted against GPS (Top). Although the reconstruction is wildly inaccurate at this scale, locally it is good enough to enable retracing of the route. The bottom images show views from either end of the path, with the reference path plotted as a series of chevrons. To the rover, the map is locally Euclidean.

Submaps are constructed using a specialization of the generic localization module from Section 5.2.1. The system is initialized with the first keypoint list, $\{\mathbf{y}_{0,j}, \mathbf{d}_{0,j}\}$. The map frame \mathcal{F}_m is defined to be the same as \mathcal{F}_{c_0} . All of the keypoints are triangulated using (3.91) and placed in the map. In each subsequent frame, incoming keypoints are tracked against the working database and subjected to outlier detection. Let us use n to index the inlying feature tracks. Each track associates feature i in the map, to keypoint

j . To estimate $\mathbf{T}_{c_k,m}$, we define the error term, \mathbf{e}_n :

$$\mathbf{e}_n := \mathbf{y}_{j,k} - \mathbf{h}(\mathbf{M}\mathbf{T}_{c_k,m}\mathbf{p}_m^{i,m}), \quad (5.4)$$

Letting N_k be the number of feature tracks at time k , we define our objective function, J_k , to be

$$J_k := \frac{1}{2} \sum_{n=1}^{N_k} \mathbf{e}_n^T \mathbf{W}_n \mathbf{e}_n, \quad (5.5)$$

where \mathbf{W}_n is a weighting matrix based on the inverse of the estimated measurement covariance of $\mathbf{y}_{k,j}$. We linearize (5.5) using the method described in Section 3.5.1 and minimize J_k using the Gauss-Newton method. Note that in this case we are only solving for $\mathbf{T}_{c_k,m}$ and not for any of the landmark positions so the term in (3.99) corresponding to the landmark parameters (the $\delta\mathbf{p}$ term) is not used.

When the percentage of features tracked drops below a threshold, τ_f , the pose, $\mathbf{T}_{c_k,m}$, is added to the reference path, and all of the keypoints are added to the map. Using a threshold avoids generating bloated maps while the robot is sitting still, and automatically adjusts the number of features in the map based on the difficulty of the terrain. Using the pose estimated in the previous step, triangulated keypoints are placed into the map in a common frame, $\underline{\mathcal{F}}_m$:

$$\mathbf{p}_m^{i,m} = \mathbf{T}_{c_k,m}^{-1} \mathbf{g}(\mathbf{y}_{j,k}) \quad (5.6a)$$

$$= \mathbf{T}_{c_k,m}^{-1} \mathbf{p}_{c_k}^{i,c_k} \quad (5.6b)$$

The prototype feature in our system is based on the triangulated position and SURF descriptor of the first view only. Incoming keypoints that are not successfully tracked are added to the map as seen. Keypoints that are successfully tracked are already present in the map and so the new observations are discarded. Although there is enough information here to estimate the camera's pose *and* the feature positions—either on the entire map (Royer et al., 2007), or on some sliding window of poses (Konolige et al., 2007; Sibley et al., 2008)—our system has no requirement to build a globally consistent map. Furthermore, our results show that this implementation works for the kind of local, metric localization needed in the repeat pass. While future work may involve some evaluation of the benefits of better reconstruction techniques, local bundle adjustment is not necessary to build a working system.

As poses and features are added to the map, the length of the current reference path is tracked. When the length exceeds a threshold, τ_l , the submap is packaged for the

repeat pass and saved to disk. By changing this parameter, our system scales smoothly between a complete global reconstruction (Royer et al., 2007) and view-sequenced route representations that match against single images along the path (Šegvić et al., 2009). We experimented with different values for τ_l early on in the development of the algorithm and found that higher values (larger submaps) increased the algorithm’s robustness to localization dropouts. However, this robustness came at the cost of increasing computational complexity as larger submaps contained more features. Eventually, we settled on the value $\tau_l = 5$ meters; this was as high as we could set this value and still operate at the framerate necessary to drive the robot at a reasonable speed to conduct long-range experiments.

The map-building step may fail if it is unable to succeed in tracking a minimum number of sparse feature points from the map to the latest image. We have seen this happen if the rover has moved too much between images, with image motion blur experienced in low-light conditions, and in areas with highly repetitive texture. The algorithm tries to deal with single-frame failures by storing the failed image and attempting to track features from the next image. If this fails, the algorithm has no way to associate the current images (and all future images) with the images that have come before. To deal with this, it sets a flag that the end of the current submap is broken, purges the existing map and starts a new map as if it was processing the first image of a sequence. The *broken map* flag is used as a signal to the repeat pass that the algorithm should stop and look for the next section of the map. This will be described further below.

Teach Pass Map Packaging

When a split in the map is triggered, the current set of reference poses and features are packaged for use in the repeat pass. First, the poses are subsampled to satisfy a minimum-spacing constraint, τ_s . This smoothes the path and puts it in a format suitable for our path tracker. All experiments in this chapter use $\tau_s = 0.5$ m. Note that this step only subsamples the poses used as the reference path; all features remain in the map unless they were only observed in one stereo pair. The frame-to-frame tracking process used to build the map is the best possible condition for tracking features; pose changes between images are small and the lighting is consistent. If a feature was unable to be tracked in the teach pass, it is unlikely to be seen in the repeat pass.

The subsampled reference poses give the path of the camera in \mathcal{F}_m , but our path

tracker controls the position of the vehicle, not the camera. We compute the vehicle position using the transformation between the camera and the vehicle frames \mathbf{T}_{c_k, v_k} . The reference path of the vehicle, $\rho_m^{v_k, m}$, and its attitude, \mathbf{C}_{m, v_k} are given by

$$\begin{bmatrix} \mathbf{C}_{m, v_k} & \rho_m^{v_k, m} \\ \mathbf{0}^T & 1 \end{bmatrix} := \mathbf{T}_{c_k, m}^{-1} \mathbf{T}_{c_k, v_k}. \quad (5.7)$$

The projection from three dimensions to two is defined by fitting a plane to the feature points in the current submap. The subsampled features have each passed the consistency test of outlier detection and so they represent a reasonable, sparse reconstruction of the local area. For each feature i , at position $\mathbf{p}_m^{i, m}$ in the submap, we find d_i , the minimum distance between the feature and one of the vehicle reference poses:

$$d_i := \min_k \|\rho_m^{v_k, m} - \mathbf{p}_m^{i, m}\| \quad (5.8)$$

From this distance, we compute a weight, w_i , used in the plane fitting:

$$w_i = \begin{cases} \frac{1}{d_i + \sigma_p} & \text{if } d_i \leq \tau_d \\ 0 & \text{otherwise} \end{cases} \quad (5.9)$$

This weighting term is designed to ensure that the plane fit captures the local ground plane directly along the path that the rover has already traversed. The threshold, τ_d , ensures that features outside of the vehicle corridor are not used for the plane fit and σ_p controls the roll-off of weights as features approach the edge of the vehicle corridor. For all experiments in this chapter, we use $\sigma_d = 0.01$, and $\tau_d = 1.5$. We parameterize the plane by a unit vector, \mathbf{n} , and offset, b , such that any point \mathbf{x} on the plane satisfies

$$\mathbf{n}^T \mathbf{x} + b = 0 . \quad (5.10)$$

From this equation, we define a weighted least-squares problem to solve for \mathbf{n} and b by minimizing J_p ,

$$J_p = \frac{1}{2} \sum_{i=1}^M w_i (\mathbf{n}^T \mathbf{p}_m^{i, m} + b)^2 - \frac{1}{2} \lambda (\mathbf{n}^T \mathbf{n} - 1) , \quad (5.11)$$

where M is the number of features in the submap and λ is a Lagrange multiplier that ensures \mathbf{n} is a unit vector. Solving for the minimum of this equation results in the eigenvalue problem

$$\underbrace{\left(\sum_{i=1}^M w_i (\mathbf{p}_m^{i, m}) (\mathbf{p}_m^{i, m})^T - \frac{1}{W} \left(\sum_{i=1}^M w_i \mathbf{p}_m^{i, m} \right) \left(\sum_{i=1}^M w_i \mathbf{p}_m^{i, m} \right)^T \right)}_{=: \mathbf{A}} \mathbf{n}^* = -\lambda \mathbf{n}^* , \quad (5.12)$$

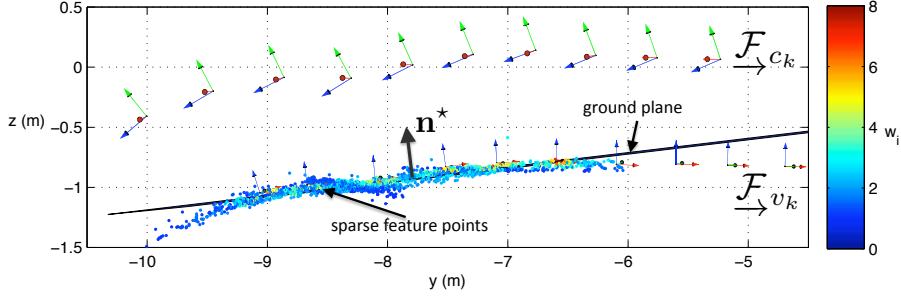


Figure 5.7: Side view of a submap showing the camera frames, \mathcal{F}_{c_k} , the vehicle frames, \mathcal{F}_{v_k} , the sparse feature points, and the ground plane fit to the features.

where

$$W := \sum_{i=1}^M w_i , \quad (5.13)$$

and \mathbf{n}^* , the unit vector that minimizes J_p , is the eigenvector of \mathbf{A} corresponding to its minimum eigenvalue. Figure 5.7 illustrates this process, showing the camera and vehicle poses, the weighted sparse feature points, and the resulting plane fit.

The unit vector \mathbf{n}^* is the normal of the xy -plane of the projection frame, \mathcal{F}_p , expressed in \mathcal{F}_m . We now calculate the rotation, $\mathbf{C}_{m,p}$, that transforms vectors from \mathcal{F}_p to \mathcal{F}_m . Using the shorthand $c_a := \cos(a)$ and $s_b := \sin(b)$, the rotation $\mathbf{C}_{m,p}$ may be parameterized by Euler angles, (α, β, γ) , such that

$$\mathbf{C}_{m,p} = \begin{bmatrix} c_\alpha c_\beta & s_\alpha c_\beta & -s_\beta \\ c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & -s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & c_\beta s_\gamma \\ c_\alpha s_\beta c_\gamma + s_\alpha c_\gamma & -s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma & c_\beta c_\gamma \end{bmatrix} . \quad (5.14)$$

We know that \mathbf{n}^* expressed in \mathcal{F}_p is $\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ which leads to the following constraint:

$$\mathbf{n}^* = \mathbf{C}_{m,p} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -s_\beta \\ c_\beta s_\gamma \\ c_\beta c_\gamma \end{bmatrix} \quad (5.15)$$

Defining the components of $\mathbf{n}^* =: \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix}^T$, we can solve for β and γ :

$$\beta = \arcsin(-n_1) \quad (5.16)$$

$$\gamma = \arctan(c_\beta n_2 / c_\beta n_3) \quad (5.17)$$

The last Euler angle, α , is ambiguous (the plane normal is only a two-degree-of-freedom constraint) so we introduce a final constraint that the x -axis of \mathcal{F}_{v_0} lies in the xz -plane

of $\underline{\mathcal{F}}_p$. Using $\mathbf{C}_{m,p}$ and the vehicle path from (5.7), we can transform the reference path to the projection frame:

$$\boldsymbol{\rho}_m^{v_k,v_0} = \boldsymbol{\rho}_m^{v_k,m} + \mathbf{C}_{c_0,v_0} \boldsymbol{\rho}_{v_0}^{c_0,v_0} \quad (5.18)$$

$$\boldsymbol{\rho}_p^{v_k,v_0} = \mathbf{C}_{m,p}^T \boldsymbol{\rho}_m^{v_k,v_0} \quad (5.19)$$

$$\mathbf{C}_{v_k,p} = \mathbf{C}_{v_k,m} \mathbf{C}_{m,p} \quad (5.20)$$

Finally, we compute a scalar difficulty score for the submap. During the repeat pass, the difficulty level is used to choose the robot's repeat speed to use on a given map. We compute a measure of curvature of the reference path as it captures two common forms of path difficulty: (i) tight turns, and (ii) rough terrain. To this end, we compute the incremental attitude changes of the camera, $\delta\mathbf{C}_k$:

$$\delta\mathbf{C}_k = \mathbf{C}_{c_{k-1},m} \mathbf{C}_{c_k,m}^T \quad (5.21)$$

This attitude change is decomposed into a unit-length axis of rotation, \mathbf{a}_k , and an angle of rotation, ϕ_k . The difficulty, h , is then computed as the root-mean-square attitude change,

$$h = \sqrt{\frac{1}{K} \sum_{k=1}^K \phi_k^2} , \quad (5.22)$$

where K is the number of reference poses.

When building reference trajectories with a fixed length and a fixed spacing of reference poses, K is very consistent across submaps. Furthermore, over these very short distances, the relative pose estimates are very accurate. Because of this, the values of h from (5.22) are comparable between submaps. The dependence of driving speed on terrain difficulty must be tuned for each vehicle/application combination. Table 5.1 lists the speed schedule used for all experiments in this chapter.

At this point, the submap is saved to disk with the following information:

- a vehicle reference path with L poses (indexed by ℓ), $\{\boldsymbol{\rho}_p^{\ell,p}\}$, expressed in $\underline{\mathcal{F}}_p$, calculated from (5.7), (5.18), and (5.19)
- a rotation $\mathbf{C}_{p,m}$ that defines the projection to a local ground plane, calculated from (5.16), (5.17), and (5.14)
- a set of M features (indexed by i), $\{\mathbf{p}_m^{i,m}, \mathbf{d}_i\}$, expressed in $\underline{\mathcal{F}}_m$

Table 5.1: The driving-speed schedule for a deployment of this algorithm would have to be tuned for each vehicle/application combination. This table lists the driving speed used as a function of the difficulty (RMS attitude change).

Difficulty range (deg)	Speed (m/s)
$h < 1.0$	1.00
$1.0 \leq h < 2.0$	0.75
$3.5 \leq h < 8.5$	0.50
$8.5 \leq h$	0.35

- a scalar difficulty score, h , computed from (5.22)
- flags that indicate if the beginning or end of the map is broken

This satisfies the requirements needed to be used as a feature database in the generic localization module described in Section 5.2.1.

Each submap is between 500 kilobytes and 2 megabytes depending on the number of features tracked (which is scene dependent). This size includes extra data that are used solely for algorithm evaluation and not to repeat the route. Averaged over all teach passes, this amounts to 348 megabytes per kilometer. The teach pass processes an image approximately every 0.2 meters, 5000 images per kilometer. An appearance-based approach using the rectified stereo images would occupy 2.9 gigabytes per kilometer and saving all of the keypoints and descriptors would take up 1.3 gigabytes per kilometer (assuming 500 stereo keypoints per frame). By aggregating data, our system offers an order-of-magnitude savings in storage over a pure appearance-based approach and reduces the computational complexity of retrotraverse.

After saving the submap to disk, older poses and features are removed from the database in memory. We build the submaps to overlap by 50% as this ensures data overlap during transitions (Marshall et al., 2008). Poses are removed from the reference path until it is half of the length saved to disk. Any feature not seen by the remaining poses is then removed from the feature database. After this step, the mapping loop continues, processing new keypoint lists, localizing against the feature database, and adding features to the map, until another split is triggered or the image sequence ends. Figure 5.8 shows a short section of a map database, the ground plane of each submap, and the reference path. When the teach pass is complete, a database of maps is available for use in the repeat pass.

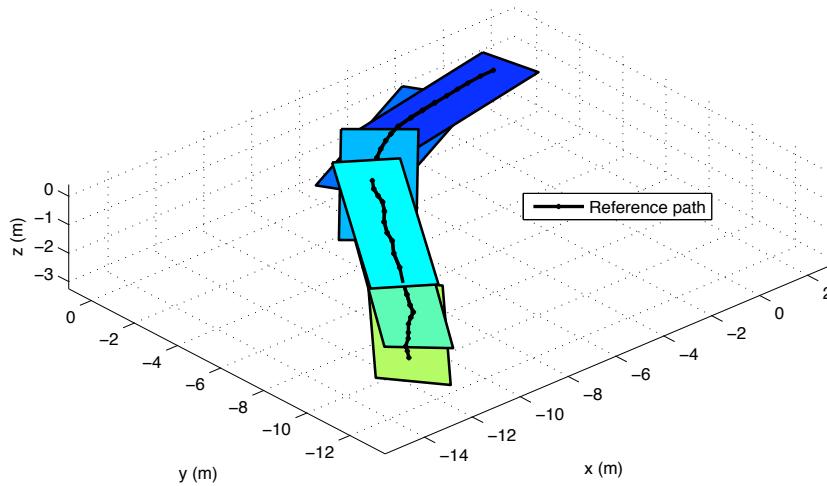


Figure 5.8: A view of six overlapping submaps with the reference path plotted above.

5.2.3 Route Repeating

During the repeat pass, the robot uses the database of submaps to repeat the route. The system we have implemented can start at any place along the path, and repeat the route in either direction, provided the camera is facing the same direction it was facing during the teach pass. Neither direction switching during path following nor local obstacle detection have been implemented, although both should be possible (Marshall et al., 2008). This section will describe the route following algorithm in detail: localization, route management, and failure handling.

Repeat Pass Localization

Three specializations of the generic localization module are used during the repeat pass: *submap selection*, *localization*, and *VO*.

Submap selection is performed at the start of a route or when the robot is lost. One of the submaps built in the teach pass is loaded into memory and used as a feature database. Features are tracked and subjected to outlier detection. If there are enough inlying feature tracks (9 for all experiments in this chapter), the objective function used in the route-teaching phase (5.5) is used to solve for the pose of the camera. If this process is successful, the rover begins to repeat the route, interleaving localization and VO as the route is retraced.

The interleaving of localization and VO during the repeat pass is one of the key strategies that makes this system robust to minor lighting changes, scene changes, and

occlusions. Our first iteration of this project used a formulation similar to Royer et al. (2007) or Šegvić et al. (2009)—all estimation of the robot’s position was based on localization against the map and no form of dead-reckoning was used. This worked well on pavement, and in urban environments, but when we tested our system in grass and rough terrain, the system failed too easily under changing lighting conditions. Realizing that our localization module was based on VO (a purely relative motion estimation method), we implemented a system that would switch back and forth between VO and localization. VO is accurate enough to keep the rover near the path through difficult areas, and periodic localization maintains the global (topological) accuracy that allows long routes to be repeated. This is similar to the method used by Zhang and Kleeman (2009) who use wheel odometry in between their global corrections. We process VO every frame, but given our current hardware, there are not enough computational resources to also perform localization every frame. Hence, we introduce an integer parameter G and only attempt localization when $\text{mod}(k, G) = 0$ (every G^{th} frame). In these experiments we have used $G = 3$. We use the same frame-to-frame VO method as used in Chapter 4.

Localization is similar to submap selection, but our prior knowledge of the rover’s position (from VO) allows us to improve on the position estimate. Using only the process described for submap selection, our system would periodically localize only using distant features. In these cases, the orientation was estimated quite well, but the position of the rover would experience huge jumps. Similar behaviour is described by Diosi et al. (2007). To account for this, a prior information term is added to the error term used to estimate the pose. Let J_{vis} be the error term from (5.5). We add prior information error terms for the position, J_{pos} , and attitude, J_{att} , so that the error term we minimize, J , is

$$J = J_{\text{vis}} + J_{\text{pos}} + J_{\text{att}} . \quad (5.23)$$

Let $\hat{\boldsymbol{\rho}}_m^{c_k, m}$ and $\hat{\mathbf{C}}_{c_k, m}$ be the position and attitude estimated by VO, and let $\boldsymbol{\rho}_m^{c_k, m}$ and $\mathbf{C}_{c_k, m}$ be the position and attitude we are estimating. In this notation,

$$J_{\text{pos}} := \frac{1}{2} (\hat{\boldsymbol{\rho}}_m^{c_k, m} - \boldsymbol{\rho}_m^{c_k, m})^T \mathbf{W}_{\text{pos}} (\hat{\boldsymbol{\rho}}_m^{c_k, m} - \boldsymbol{\rho}_m^{c_k, m}) . \quad (5.24)$$

The form of J_{att} is derived in Section 3.5.2, where $\hat{\mathbf{C}}_{c_k, m}$ is our prior mean:

$$J_{\text{att}} := \frac{1}{2} \mathbf{e} (\delta \boldsymbol{\phi}_{c_k, m})^T \mathbf{W}_{\text{att}} \mathbf{e} (\delta \boldsymbol{\phi}_{c_k, m}) \quad (5.25)$$

The weighting matrices were chosen to be

$$\mathbf{W}_{\text{pos}} := \frac{1}{\sigma_{\text{pos}}^2} \mathbf{1}, \quad \mathbf{W}_{\text{att}} := \frac{1}{\sigma_{\text{att}}^2} \mathbf{1} , \quad (5.26)$$

where $\mathbf{1}$ is the identity matrix. All experiments in this work use $\sigma_{\text{pos}} = 1.0$ and $\sigma_{\text{att}} = 0.1$, which results in a very weak prior. Finally, (5.23) is linearized and solved using the Gauss-Newton method.

The output of the localization block is an estimate of the camera's position, $\rho_m^{c_k,m}$, and attitude, $\mathbf{C}_{c_k,m}$. Equations (5.7), (5.18), and (5.19) are then used to produce $\rho_p^{v_k,p}$, the position of the vehicle in the projection frame. The attitude of the vehicle in the projection frame, \mathbf{C}_{p,v_k} is computed using (5.20), then decomposed into a yaw-pitch-roll Euler-angle sequence. The yaw value of this sequence is the vehicle's heading in the projection frame, θ_k . Defining the components, $\rho_p^{v_k,p} =: \begin{bmatrix} x_k & y_k & z_k \end{bmatrix}^T$, we can express the two-dimensional robot pose, ϱ_k :

$$\varrho_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \quad (5.27)$$

This planar pose of the robot and the projected reference path are passed to a unicycle-model version of the planar path-tracking algorithm adapted from Marshall et al. (2008).

Repeat Pass Route Management

The localization module feeds into a route management system that triggers map hand-offs, schedules the robot's speed based on the path difficulty, and monitors the route-following system for errors. The route manager tracks the closest point on the current reference path. When the vehicle reaches the middle of a reference path, a map handoff is triggered. This involves the following steps:

1. Loading the next submap from disk
2. Updating the feature database used for localization
3. Updating the reference path used by the path tracker
4. Updating the transformation from \mathcal{F}_m to \mathcal{F}_p
5. Setting the robot's speed based on the submap difficulty

Repeat Pass Failure Handling

Route-following failures are detected by monitoring the distance travelled since the last successful localization. When this distance reaches a threshold, τ_g , the rover stops and the

system attempts to recover from the failure. To recover, the system signals the operator that there has been a failure, then searches nearby (topologically) submaps using the submap selection mode to perform wide-baseline matching. If this reinitialization is successful, the rover continues the route. During the search phase, an operator may also reposition the rover on the path (using images from the teach pass images to identify the correct position). Any failures or repositioning of the rover will be noted in our experiments below.

When the algorithm encounters a break in the map as described in Section 5.2.2, the system drives to the end of the current submap, stops, loads the next submap into memory and attempts to localize. If this is successful, the algorithm restarts the rover and continues repeating the route. If this fails, the rover will signal the operator for intervention. The algorithm will then start searching nearby maps (topologically) until submap selection is successful. The operator can then choose to reposition the rover or command it to continue using VO.

5.2.4 Parameter Choices

As in many robotics applications, there are a number of parameters that must be tuned for each deployment. The parameters of our teach-and-repeat system were tuned during algorithm development and then fixed for the experiments reported in this chapter. Throughout the algorithm description above we have tried to elaborate on the intuition behind each of our parameter choices. For clarity, we summarize the main algorithm parameters in Table 5.2 and Table 5.3, along with a description of their functions and some notes about the intuition used to select the parameter value.

5.2.5 Hardware

The experiments described in this chapter were performed using the six-wheeled articulated rover shown in Figure 5.9. Motor control on the rover was performed by a pair of microcontrollers. Vehicle-level motion commands and path tracking were handled by a single embedded PC with a 1.2 GHz Pentium 4 processor and 1 GB of Random Access Memory (RAM). The base was powered by three lithium-ion battery packs, but in order to perform the long-range experiments in this chapter, the on-board power supply was augmented with a Honda 1000 W generator, which supplied power to the base and all of the onboard computers. The computer running the localization and route management

Table 5.2: A list of the major parameters of our teach-and-repeat algorithm, a description of their functions, and the intuition behind the choice of value. Page (1/2).

Parameter	Value	Description
τ_f	45%	(Section 5.2.2) The current set of features is added to the map when the percentage of features tracked from the previous frame drops below τ_f . Using this threshold avoids generating bloated maps when the robot is sitting still and automatically adjusts the number of features per map based on the difficulty of feature tracking on specific terrain.
τ_l	5 meters	(Section 5.2.2) The length of the reference path for each submap. Larger values increase the computational complexity of searching the map for feature correspondences. Smaller values make the system more prone to failure in the face of localization errors. This was the largest value that our computer could process in real time.
τ_s	0.5 meters	(Section 5.2.2) The spacing of poses in the reference path. This controls the fidelity of the path used by the path tracker and the fidelity of the difficulty metric (5.22).
τ_d	1.5 meters	(Section 5.2.2) Controls the width of the vehicle corridor used to build the local ground plane for each submap in (5.9). This was tuned for the width of the vehicle.

was a MacBook Pro with a 2.4 GHz Intel Core 2 Duo processor, 4 GB of RAM, and an NVIDIA GeForce 8600M GT graphics card capable of supporting Compute Unified Device Architecture (CUDA) 1.1. The stereo camera was a Point Grey Research Bumblebee XB3 with a 24 cm baseline and 70° field of view, mounted approximately 1 meter above the surface pointing downward by approximately 20°. Each image of the stereo pair was captured at 640×480 pixel resolution. When possible, we used a pair of Thales DG-16 RTK-GPS units for ground-truth evaluation. These units are rated at 0.4 meter circular error probable (50% of the data should be within a circular area of this radius around the true value). Unfortunately, our radio link was not robust to occlusions; for long routes and near buildings, it was not possible to receive the real-time corrections and only regular GPS was available. The rover was able to track a path while driving

Table 5.3: A list of the major parameters of our teach-and-repeat algorithm, a description of their functions, and the intuition behind the choice of value. Page (2/2).

Parameter	Value	Description
σ_p	0.01	(Section 5.2.2) Controls the weighting roll-off as features approach the edge of the vehicle corridor in the plane fit (5.9). Larger values weight features near the edge of the corridor more. This parameter must be tuned for the expected clutter of the environment.
G	3	(Section 5.2.3) During the repeat pass, localization is processed every G frames (VO is processed every frame). This value would be 1 if possible, causing localization every frame. Higher values reduce the computational complexity of the algorithm.
$\{\sigma_{\text{pos}}, \sigma_{\text{att}}\}$	{1.0, 0.1}	(Section 5.2.3) These terms determine how much to rely on the prior pose estimate (from VO) during localization (5.26). The values chosen result in a weak prior that only significantly changes the solution when localization against the map is uncertain (e.g., when localizing with a small number of distant features.)
τ_g	50 meters	(Section 5.2.3) The distance to travel without successful localization before stopping, signalling the operator, and searching nearby submaps in an attempt to relocalize. This is based on the estimated accuracy of our VO implementation.

fowards or backwards. This allowed us to repeat routes in either direction by keeping the camera facing the same direction as it was during route learning.

5.3 Field Testing

We conducted a number of field trials to test the capabilities of the full teach-and-repeat system and characterize the performance of the localization system. This section will summarize the results of our route-following tests. Our preliminary tests were performed at the University of Toronto Institute for Aerospace Studies (UTIAS). Because of the applicability of our algorithm to planetary exploration, we conducted trials at the HMP-

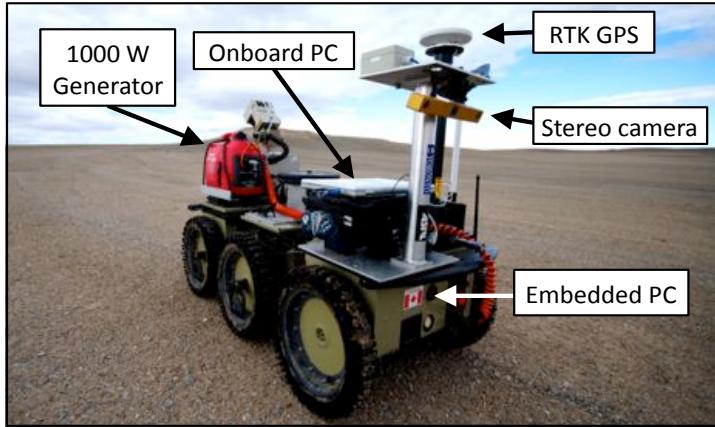


Figure 5.9: The six-wheeled rover platform used in these experiments was outfitted with a stereo camera, an onboard PC to run the teach-and-repeat algorithm, an embedded PC to process path tracking, an RTK-GPS for benchmarking localization performance, and a 1000 W gas generator to provide power during multiple-kilometer traverses.

RS on Devon Island in the Canadian High Arctic (Lee et al., 2007). The HMP-RS is located within a polar desert, which offers an unusually wide variety of geological features of strong planetary-analogue value. Because of this, it has been used for rover testing in the past (Wettergreen et al., 2002, 2005; Fong et al., 2007, 2008). Additionally, the lack of vegetation, low angle of the sun in the sky, and wide range of terrain types make it an ideal site for testing of vision-based algorithms for planetary exploration (Barfoot et al., 2010d).

5.3.1 Route Following

Our teach-and-repeat system has been tested on 27 routes and over 32 kilometers of autonomous driving. Results reported in this chapter are for the algorithm described in Section 5.2. Earlier route-following results during the algorithm’s development are not included. All tests described here were performed using the same code and parameters. Individual teach passes are named according to the convention shown in Figure 5.10. Experiments performed at UTIAS are marked with a *u* and those from Devon Island are marked with a *d*. We used two methods to teach routes. The *h* tag indicates that the rover was piloted by a human and the *a* tag indicates that the rover was driving autonomously. The autonomous teach passes were recorded during trials of a terrain assessment and path-planning algorithm. When the terrain assessment algorithm signalled that its run was complete, the route was taught from logged images, and the rover autonomously returned along its path. Some routes were taught with the camera facing forwards and

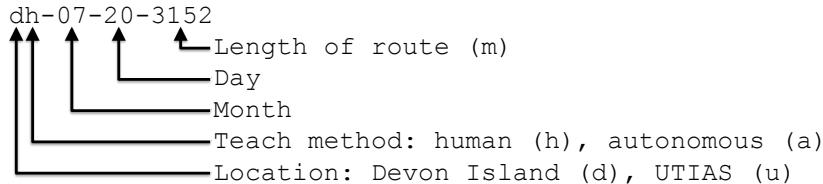


Figure 5.10: The naming convention used for teach passes.

others with the camera facing backwards (as required by other concurrently-running experiments). However, during route repeating, the rover drove forward or backward as necessary to keep the camera pointed in the same direction as it was during route learning. Complete statistics for all teach and repeat passes are given in Appendix A, but the overall results will be summarized here.

Learned routes ranged in length from 47 meters to nearly 5 kilometers. Out of the 27 teach passes, 21 of them successfully built maps without failure. The teach pass failures will be discussed in greater detail in Section 5.4.5. The difficulty of the routes was assessed using an inclinometer to measure the vehicle-frame pitch and roll, and GPS to measure the relative elevation change. During the most extreme routes, the rover experienced up to 118.5 meters of elevation change, as well as pitch and roll deviation from vertical of up to 28° and 22°, respectively.

The 27 teach passes were used to perform 60 repeat passes. Only four of the routes required manual interventions. Four repeat passes were not completed to the end. The repeat pass failures will be discussed in greater detail in Section 5.4.6. The longest autonomous repeat pass was 3.2 kilometers (dh-07-23-4963). There were two autonomous runs of approximately two kilometers (dh-07-20-2120) and ten autonomous runs approximately one kilometer long (uh-05-20-1152, uh-05-21-1170, and dh-07-22-1091). Out of the 32.919 kilometers travelled, only 0.128 kilometers were piloted manually. This is an autonomy rate of 99.6%. In all cases where the rover needed an intervention, it stopped along the path and signalled the operator.

5.3.2 Route with Large Three-Dimensional Motion and Extreme Lighting Changes

To test the operational limits of our algorithm, we built a route at UTIAS where the rover experienced large three-dimensional motion and extreme lighting changes. Figure 5.11 shows an overhead view of the route, the vehicle-frame pitch and roll (as measured by

an inclinometer), and some representative views from the left camera. The rover started inside our indoor test facility on a raised platform. It descended a slope, climbed two hills, ascended a ramp, and then drove through a narrow corridor leading outdoors. There, it traversed an obstacle course, crossed a road, and finished the route by parking in our laboratory. The platform experienced pitch and roll up to 27° and moved from an indoor, low-light environment to outdoors.

We taught this route twice, once during development of the obstacles (uh-07-22-0120), and once after they were complete (uh-07-23-0120). The routes were repeated 7 and 5 times, respectively. Every repeat pass was successful, despite the three-dimensional motion of the camera. Figure 5.12 shows the teach pass corridor (the track of the teach pass laterally extended ± 2 meters for illustration) with the tracks of the seven repeat passes overlaid. Sections where the algorithm experienced localization dropouts are highlighted in blue (shaded). Steep hills in the indoor section caused localization and VO failures due to significant motion blur. At the end of the route, computers and chairs were moved around, changing the appearance of the scene significantly. The experiment highlights the interplay of localization and VO. Where VO fails, localization corrects the error and keeps the rover on the correct path. Where localization fails, VO is accurate enough to carry the rover through to a place it recognizes.

Table 5.4: Difficulty metrics for the teach passes where the rover experienced the most extreme three-dimensional camera motion.

Tag	Elevation change (m)	Min roll (deg)	Max roll (deg)	Min pitch (deg)	Max pitch (deg)
uh-05-20-1152	4.9	-18.4	8.2	-12.0	6.3
uh-05-21-1170	4.9	-17.8	11.9	-14.6	10.3
uh-05-22-0120	3.6	-14.7	11.7	-21.5	27.2
uh-07-23-0120	4.9	-13.3	12.1	-18.5	26.5
dh-07-20-2120	69.2	-22.0	15.9	-28.3	16.9
dh-07-23-4963	118.5	-12.8	13.6	-15.5	12.0
dh-07-30-0347	2.1	-12.2	13.1	-12.1	10.8
dh-07-31-0192	1.1	-9.8	10.9	-17.9	19.1

This experiment was performed before our field trials on Devon Island to prove that the teach-and-repeat system would work on three-dimensional terrain. During our field

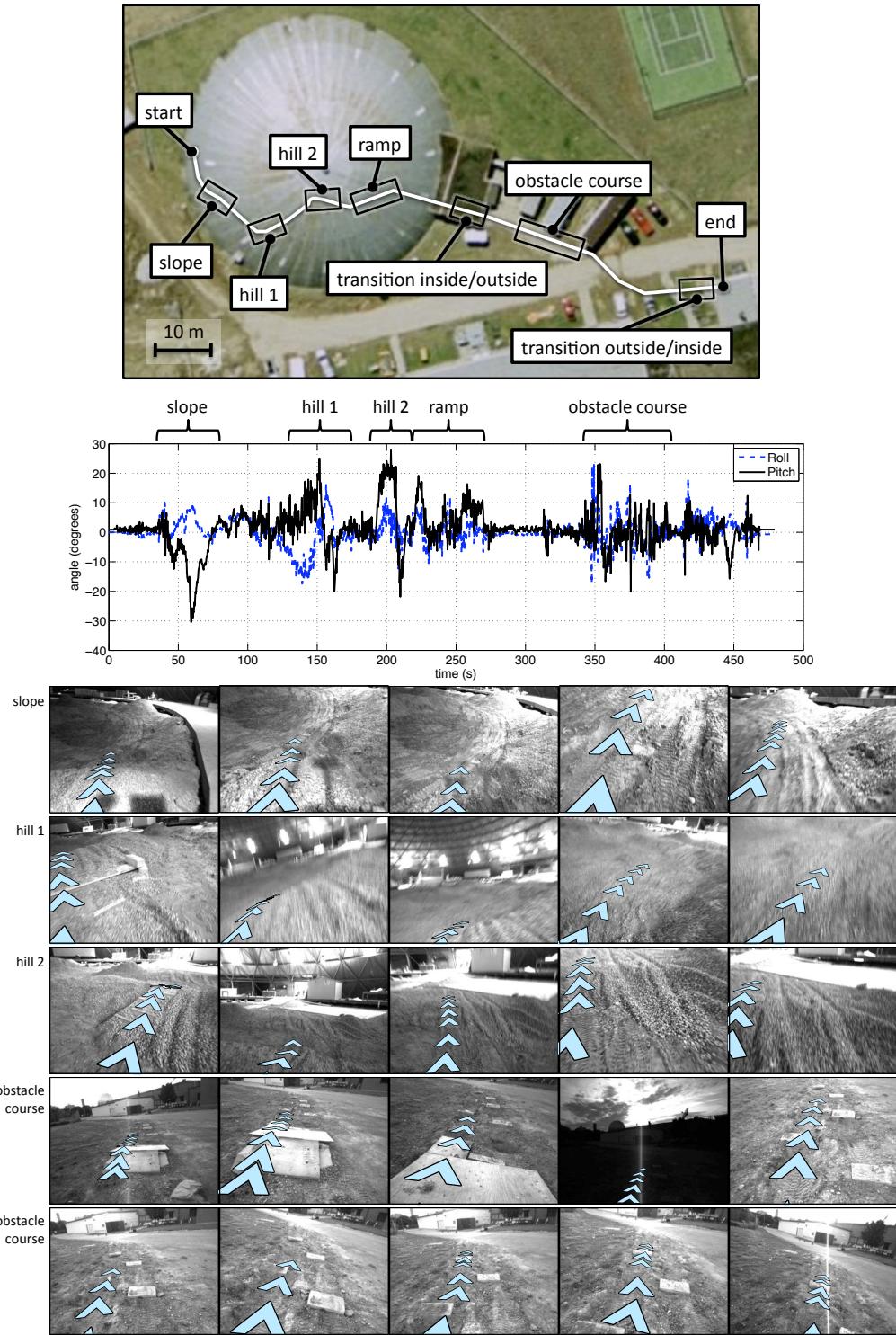


Figure 5.11: Top to bottom: An overhead view of the route built to test nonplanar camera motion and extreme lighting changes (uh-07-22-0120, uh-07-23-0120), the pitch and roll of the rover during the teach pass of route uh-07-22-0120, and short image sequences (left camera) from one repeat run of the route. The path, plotted as chevrons, confirms that localization is indeed performed in three-dimensions.

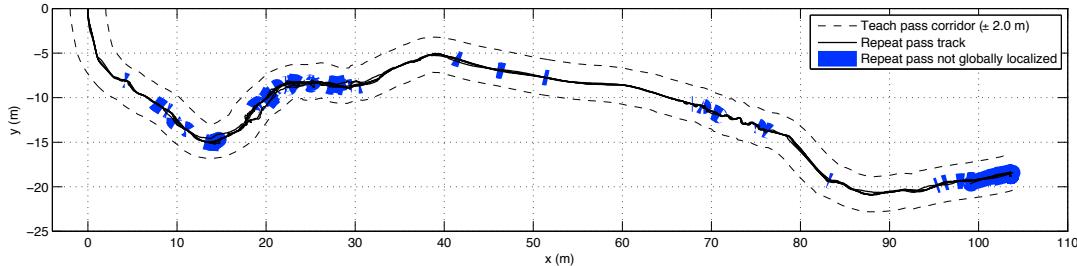


Figure 5.12: The seven repeat passes of route uh-07-22-0120 with the reference path from the teach pass extended laterally ± 2 meters. Localization dropouts, highlighted in blue (shaded), occurred mainly due to scene changes and motion blur.

trials we tested the algorithm over many three-dimensional routes (see the list of the most extreme routes in Table 5.4). In all cases, three-dimensional motion of the camera was not a limiting factor for route following.

5.4 Evaluation

In this section we offer some evaluation of our algorithm to try to describe why it works, and what its strengths and shortcomings are. We examine the convergence properties of the localization algorithm and the properties of the algorithm under changing lighting conditions. We compare the estimated lateral path-tracking error to that measured by GPS, and look at which features are used for localization. Finally, we examine the failure modes experienced by the algorithm.

5.4.1 Convergence Properties

To test the convergence properties of the localization algorithm, we taught a single map on characteristic terrain (from the Devon Island experiments) using a camera on a tripod. After processing the teach pass, the camera was placed in a nominal position in the middle of the map, set to process localization, and perturbed from this nominal position until the localization failed. Perturbations were introduced four ways: as lateral displacements from the path center (0.1 meter increments), and along vehicle-frame yaw, pitch, and roll axes (5° increments). At each increment, 200 localizations were processed.

Figure 5.13 shows the mean inlying feature count for lateral and angular deviations. The curves end when the localization algorithm fails. This experiment shows that the feature count decreases rapidly from the camera's nominal placement. Any curve of this

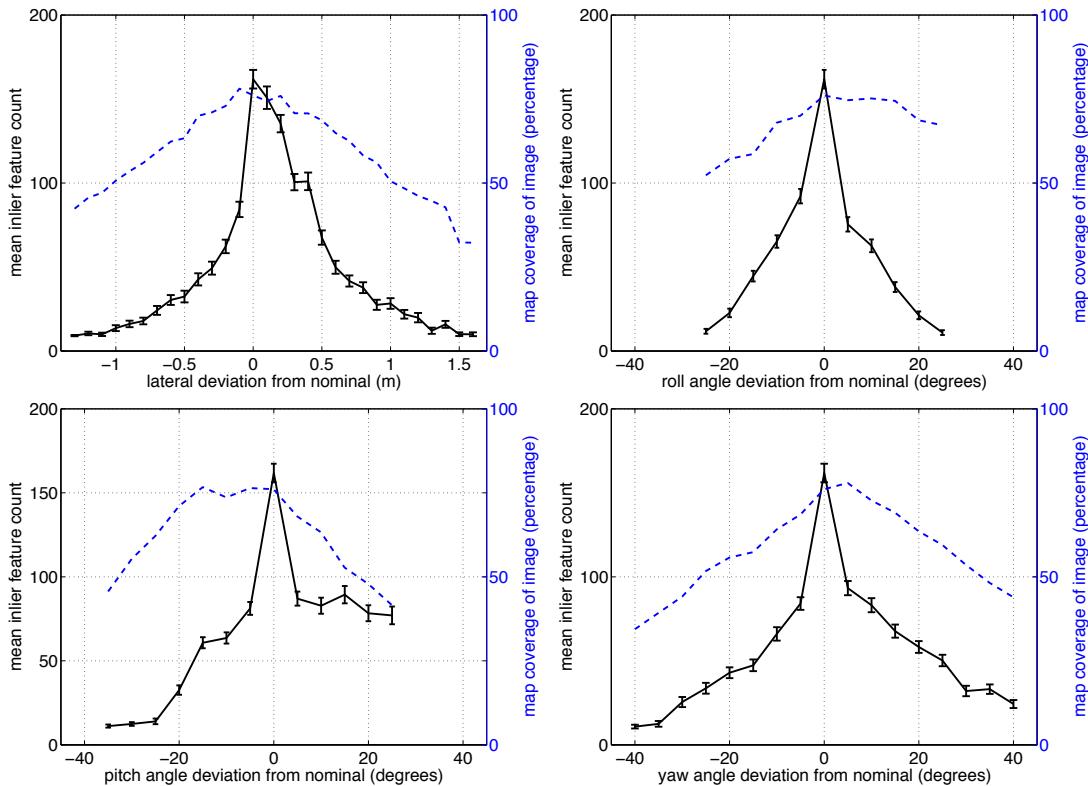


Figure 5.13: Average feature count of the localization algorithm (black, solid line with 1σ bounds) as the camera was displaced laterally from the path, or rotated in place. Each data point is averaged over 200 trials. Angular perturbations were made along the vehicle-frame yaw, pitch, and roll axes. The blue dotted curve shows the percentage of the image covered by the features in the map. While the feature count is correlated to coverage, changes in viewpoint also reduce the ability of the system to associate features.

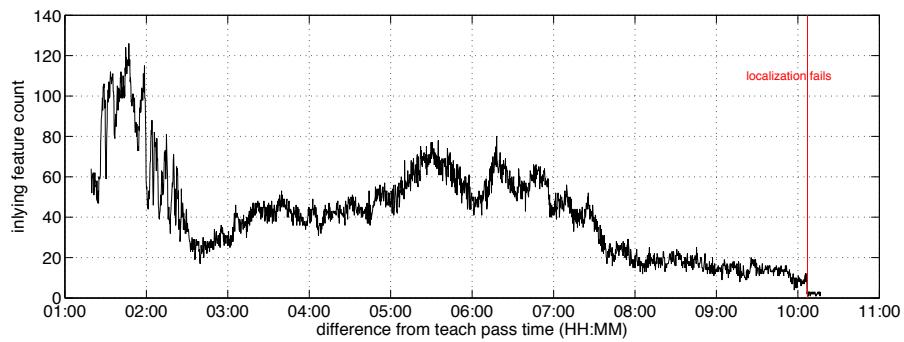


Figure 5.14: Results of testing the localization algorithm performance under changing lighting. Due to time constraints during our field campaign, we were only able to perform a single trial. However, the result here fits very well with the results of our path-following experiments; the SURF feature matching is not robust to extreme lighting changes.

type will be scene-dependent and we believe that the slower drop in feature count with positive lateral displacement may be due to prominent rocks to the right of the path. The experiment shows that localization is possible with up to ± 1 meter lateral displacement from the path, and over $\pm 20^\circ$ angular deviation in all of yaw, pitch, and roll.

5.4.2 Lighting Dependence

We also designed an experiment to show the properties of our algorithm under changing lighting. The SURF feature description algorithm accounts for contrast changes by normalizing the description vector. However, in our experience, descriptor-based matching is very difficult under extreme lighting changes. To illustrate this, we taught a short route and set up a camera to capture an image and perform localization every 30 seconds. The inlying feature count is plotted against time passed in Figure 5.14.

Ten hours after the teach pass, the localization module fails to find enough inlying features. This confirms the lighting dependence that we have seen in our experiments. Strong lighting with a low angle of incidence is particularly problematic in this regard. Similarly, routes taught on overcast days and repeated on sunny days (or the other way around) cause problems. On overcast days, SURF’s blob detector finds points based mainly on surface albedo, whereas during periods of strong lighting, shadowing creates areas of intensified image contrast based on scene structure. Different sets of point features are returned in each case.

5.4.3 Localization Performance during Path Following

This section will characterize the performance of our localization system during path following. Although the reconstruction of the route may not be globally consistent, each small section of the path should have a small reconstruction error. Because of this, we may compare the lateral path-tracking error estimated by the localization algorithm to that measured by GPS. Our GPS unit required line-of-site between the base station and the rover to send the real-time corrections and so we do not have RTK-GPS data for all routes. Figure 5.15 shows the lateral path-tracking error estimated by localization and measured by GPS over a 450-meter-long segment of route da-07-29-0486. This segment of the route had RTK-GPS for both the teach pass and the repeat pass. A blue background highlights the portions of the repeat pass where localization has failed.

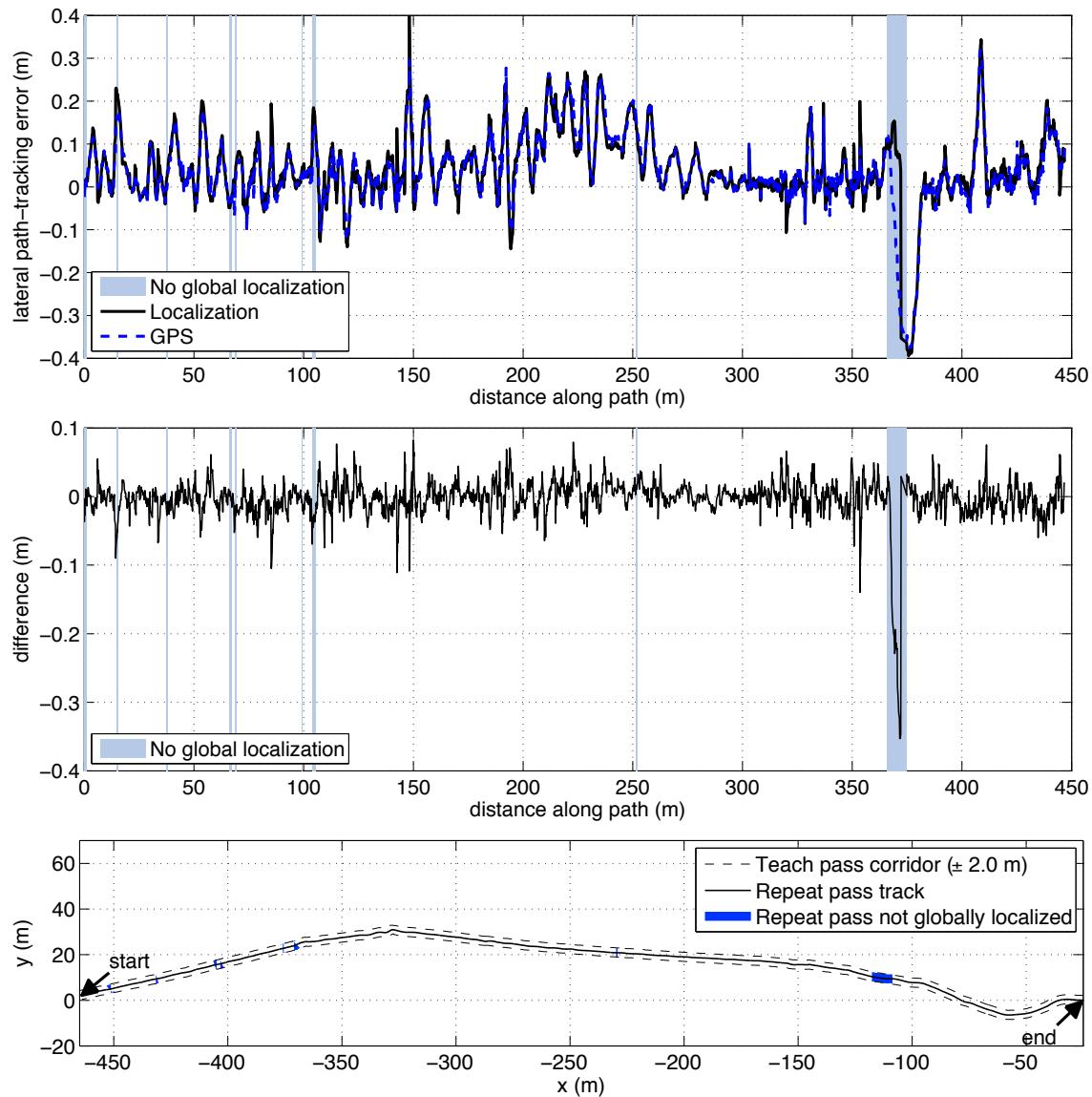


Figure 5.15: Top to bottom: lateral path-tracking error during a repeat pass as estimated by the localization algorithm and measured by RTK-GPS, the difference between these curves, and the track of the rover during this segment. The blue background highlights areas where the localization step failed. When localization is successful, the pose estimate agrees well with GPS.

These figures show two important characteristics of our algorithm. First, when localization is successful, the estimated lateral path-tracking error has good agreement with the same quantity measured by GPS. When localized, none of the differences are larger than 0.2 meters, agreement well within what we can discern with this GPS. Second, it shows that, when the algorithm is unable to globally localize, the estimate may diverge and then reconverge when localization is recovered. This is shown on Figure 5.15 at around 360 meters travelled where the localization drops out for nearly 15 meters. The speed of divergence is a function of the accuracy of our VO algorithm. We have seen the algorithm recover from lateral path-tracking errors of 1.5 meters and localization dropouts of up to 40 meters. In each case, successful localization pulls the estimate back toward global consistency and allows our algorithm to faithfully repeat long routes.

5.4.4 Keypoint and Feature Usage

This section tries to shed some light on which keypoints and features are used by the algorithm to perform localization. To this end, we used data collected during the nine repeat passes of route uh-05-26-0202. This route was taught midday when it was overcast and the first seven repeats were performed on a sunny day, every hour starting at 7:45 am. After the sixth repeat, cloud cover moved in and the additional runs were performed on a different day. The large number of repeats and varying lighting conditions make this route a good candidate for an examination of feature usage.

Figure 5.16(a) shows a histogram of track length (number of observations) for the 132,781 features stored in the map. The figure shows that maps are predominantly populated by features seen in only two images. From there, the track length decreases quite steeply but there are still a small number of features seen many more times. The long tail of this curve has been truncated. The longest track length was 102 frames.

During the repeat pass, we logged which features were used for localization. Figure 5.16(b) shows the relationship of the track length during the teach pass to feature use during the repeat pass. Plotting the mean over all samples shows a strong linear relationship with slope 1. This confirms what intuition would suggest: that unique features seen for a long time during route learning are easily found during route repeating.

To determine which features contribute most to localization, we plotted the feature observations in image space. Figure 5.17(a) shows a clustering of features around the top of the image. When compared to a typical image from this route shown in Figure 5.17(b),

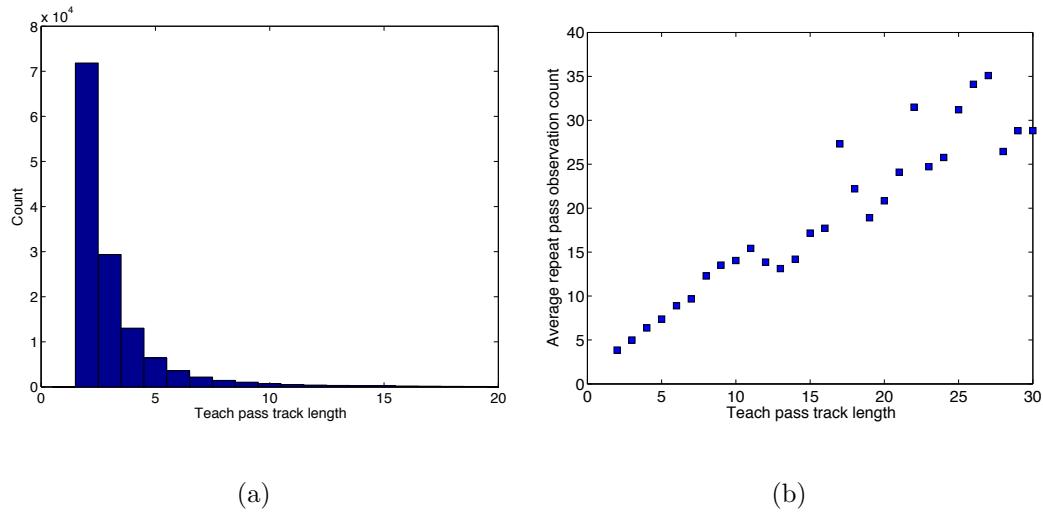


Figure 5.16: Histogram of feature track length for the 132,781 features tracked in the teach pass (a). The mean observation count during route repeating for each track length is shown in (b). Feature track length during route learning correlates strongly with the observation count during route repeating.

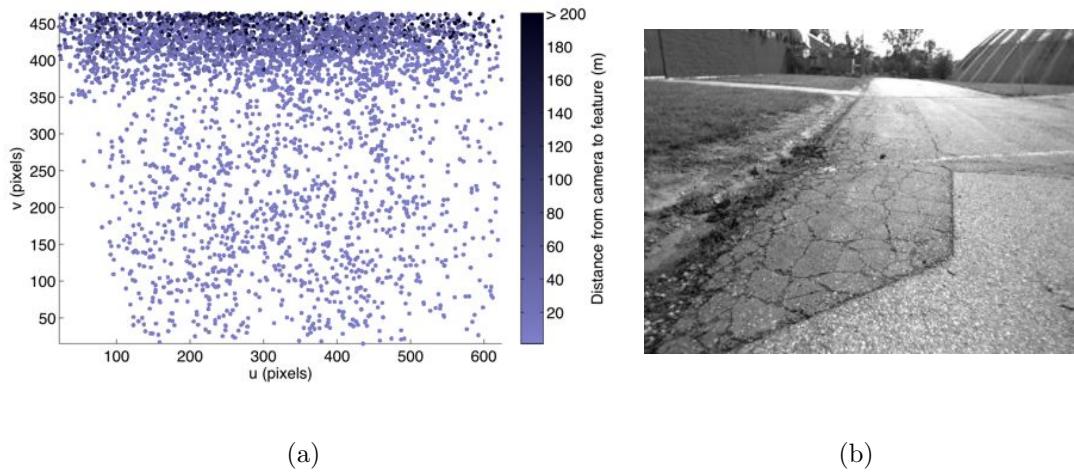


Figure 5.17: A plot of (a) the image locations of 240,534 feature observations from 9 repeat passes and (b) a typical image from this sequence. The features used for localization cluster in a band at the horizon.

Table 5.5: List of teach passes with failures.

Tag	Length (m)	Maps	Features per map	Failures
ua-06-04-0097	96.8	35	1581	3
ua-06-06-0186	185.8	65	1975	1
dh-07-20-2120	2120.0	740	3680	1
da-07-20-0464	464.0	166	1662	1
da-07-21-0453	453.5	161	1960	1
dh-07-23-4963	4962.5	1732	3993	5

it clearly shows that the majority of features used during the repeat pass are distant from the camera—horizon features. Horizon features are good for correcting for rover orientation but, as stereo-based range accuracy decreases with distance from the camera, they are not great for estimating the rover position. In this sense, our algorithm works a lot like VO with globally consistent orientation updates. This also suggests a way forward for future work; it may be possible to reduce the submap size by only using features that have been tracked for multiple frames. This would reduce the computational complexity of finding feature correspondences in the map, and enable localization to be performed more often. However, using only distant features would most likely require more accurate feature position and covariance estimates, suggesting the use of a multi-frame reconstruction method during the teach pass. Alternately, the two-stage estimation algorithm described by Kaess et al. (2009) could be used to decouple the orientation and position estimation problems.

5.4.5 Teach Pass Failures

All of the teach pass failures listed in Table 5.5 were due to large displacement of the camera between images. Sometimes a processing backup would cause our data-logging system to drop images. This was not a problem on many types of terrain, especially where there were strong horizon features or large objects out of the ground plane. However, on flat, repetitive terrain such as that seen in a long section of uh-07-23-4963, even short dropouts caused teaching failures. This is illustrated in Figure 5.18, which shows a pair of consecutive images from this route that caused a failure.

Three out of the five routes with teach-pass failures required no operator intervention.



Figure 5.18: All teach pass failures were the result of large inter-image spacing due to data logging dropouts. This figure shows two consecutive images from uh-07-23-4963 that caused a failure.

The rover simply drove to the end of the broken map, loaded the next map, relocalized, and continued. This exploited the topological knowledge of temporally adjacent maps.

5.4.6 Repeat Pass Failures

Table 5.6 lists all of the repeat pass failures and incomplete routes. Repeat pass failures had two distinct causes. The first was integration with an autonomous terrain assessment and path-planning algorithm, and the second was changing scene appearance.

The route learning algorithm described in this chapter had no problem learning an image sequence with direction switches, but paths that doubled back on themselves were not amenable to our path tracking algorithm. Early on in development, we decided not to implement direction switches. However, the autonomous terrain assessment and path-planning algorithm used to build some of the routes sometimes backed up along its own path to get out of a cul-de-sac. When faced with a knot in the path, the path tracker would command the robot to perform a wide U-turn, ending up 180° to the desired orientation on the path. To deal with this, we developed a preprocessing step that used the motion estimate from the terrain assessment run to automatically identify path knots, and remove the images making up those knots from the sequence. We developed this preprocessing step during some tests at the beginning of June, 2009 (`ua-06-04-*` and `ua-06-06-*`). Failures during this time informed this development process and the knot-removal step worked without fail after that.

Other failures during repeat passes were due to the changing appearance of the scene, mostly because of changing lighting conditions. We encountered several situa-

Table 5.6: List of repeat passes with failures. Column C is the percentage of the total route completed, column A is the percentage of the distance travelled autonomously, and column I is the number of operator interventions.

Teach pass tag	Teach pass start time	Repeat pass start time	C	A	I	Globally localized
uh-05-21-1170	12:16:02	20:26:28	100.0%	92.0%	1	49.6%
		08:03:44	100.0%	98.4%	3	45.0%
ua-06-04-0097	14:48:50	15:10:59	87.4%	95.5%	2	93.4%
ua-06-06-0186	13:11:57	13:55:08	100.0%	98.4%	1	81.5%
dh-07-23-4963	08:50:49	08:49:24	64.9%	100.0%	0	96.3%
		15:07:49	32.8%	100.0%	0	76.2%
dh-07-30-0187	14:35:09	18:37:18	78.4%	100.0%	0	9.9%

tions where a route required manual interventions (uh-05-21-1170) or failed to complete (dh-07-30-0187) at one time of day but was autonomously repeated successfully when the lighting changed. These results agree well with the lighting test in Section 5.4.2. Route uh-05-21-1170, taught at midday in direct sunlight, was repeated six times and only had trouble late in the evening or early in the morning. Route dh-07-30-0187 was in an area made up entirely of fist-sized rocks. The complex shadows created by these rocks were difficult for our algorithm under time changes. Figure 5.19 shows an image from the teach pass of this route along with images from the failed and successful repeat passes.

Flat areas with repetitive texture were particularly difficult under changing lighting conditions. The section of route uh-07-23-4963 already shown in Figure 5.18 was taught when it was partly cloudy with some periods of strong direct sunlight and both repeat passes were attempted when it was overcast. The first repeat pass was attempted forward along the route while the second was attempted backward. Both failed at either end of the same stretch of terrain. Figure 5.20 shows an image from the rover where it stopped on the first repeat pass and a corresponding image from the teach pass. It is clear from



(a) An image from the start of the teach pass

(b) An image from the failed repeat pass started four hours later

(c) An image from the successful repeat pass started the next morning.

Figure 5.19: Images from the start of route dh-07-30-0187 that show the scene changes due to lighting.



(a) An image from the teach pass

(b) The stopping position of the robot on the first repeat pass

Figure 5.20: Images from the teach and repeat passes of route uh-07-23-4963. The rover was unable to localize for 50 meters even though it was clearly on the path for most of the way. Repetitive texture, different lighting, lack of horizon features, and lack of unique three-dimensional objects in the scene were the major causes of localization dropouts.

the image that the rover was no more than 0.5 meters laterally displaced from the path after 50 meters without localization. Although the viewpoint was nearly the same, the scene exhibited repetitive texture, different lighting, a lack of horizon features, and a lack of unique three-dimensional objects; together, this caused the localization system to fail. At this point, we would have repositioned the rover on to the path or piloted it through this section manually but it started to rain. After an hour, we decided the rain would not let up so we secured a tarp over the rover and piloted it home manually. As stated earlier, the second repeat pass came from the other direction. This time, the rover stopped at the teach pass failure shown in Figure 5.18 and was unable to relocalize. We commanded the robot to continue using VO, but it was unable to localize anywhere along the path. We were unable to find time in our test schedule to test the route under weather conditions similar to the teach pass.

5.5 Discussion

Through our extensive field testing and evaluation of this algorithm, we have learned a number of lessons that apply generally to the field of camera-based localization and mapping. First and foremost, we have shown that long-range autonomous navigation in unstructured, three-dimensional terrain is possible using a stereo camera as the only sensor, and using the SURF algorithm to detect and describe visual landmarks. Recent work has shown that it is possible to perform more accurate mapping using SBA (Sibley et al., 2009; Konolige et al., 2007) and optimization over large-scale loops in the trajectory (Newman et al., 2009; Konolige and Agrawal, 2008). However, mapping accuracy did not limit the performance of our algorithm and we feel these advances, while desirable, are not necessary to build a robust long-range navigation system. We see this as an advantage of the hybrid topological/metric formulation over SLAM: the performance of our system does not degrade when the large-scale global reconstruction is inaccurate, whereas this may make a SLAM map unusable (Howard et al., 2006). The rest of this section will outline the major lessons learned throughout this project.

The limitations of the feature detection and description pipeline: The SURF algorithm had a lot of trouble dealing with lighting changes. This was particularly evident on terrain with three-dimensional structure (the rocks in Figure 5.19 or on grass) and less of an issue in urban environments (on concrete and near buildings). Although the SURF

descriptor is normalized to provide some invariance to the effects of lighting, the detector will return different sets of points when shadowing produces areas of high contrast in the image. It is possible that performance could be improved somewhat by preprocessing the images (e.g., the patch normalization in Zhang and Kleeman (2009)) but strong shadows on three-dimensional terrain would continue to cause problems. We have shown this for the SURF algorithm but the results hold for any image-space blob detector. To deal with lighting in the current framework, it would be possible to learn a route several times under different lighting conditions, then dynamically select the “best” map sequence for route repeating (based on matching score or time of day). However, this does not address the main problem that the current feature detection and description paradigm does not deal well with lighting changes in camera imagery.

The utility of dead-reckoning: Interleaving VO and localization was one of the keys to making this algorithm work in practice. VO carries the algorithm through areas with moderate appearance changes and localization keeps the estimate consistent over long distances and corrects for VO failures. While we found VO to be very effective, some form of dead-reckoning *not* based on the camera could be very useful. A planetary exploration rover with power and computational constraints could use wheel odometry between stereo images in low-slip environments. The combination of local submaps and wheel odometry was already used by Marshall et al. (2008) for navigation in underground mines. We also advocate the use of an Inertial Measurement Unit (IMU). As described by Corke et al. (2007), cameras and IMUs are complementary sensors. The use of an IMU in this work might have eliminated teach-pass failures and compensated for VO failures due to motion blur.

The importance of map update: The performance of our system degrades as the environment surrounding a route changes over time. Solutions to this problem—sometimes called persistent mapping (Milford and Wyeth, 2009) or lifelong learning (Konolige and Bowman, 2009)—must be developed before robots can be broadly deployed in service roles. While our system could be patched to remap while the path is being followed, this would not address the underlying structure of the problem, which includes difficult issues such as (i) differentiating static, and dynamic scene elements, (ii) periodic environmental changes (e.g., daily lighting changes or seasonal changes), or (iii) joining disparate maps in the event of a loop closure. We envision the next iteration of our system becoming

much like the one described by Konolige and Bowman (2009), mapping and localizing continuously online while retaining the ability to retrace a known path at any time.

The utility of loop detection: While it was not the focus of this work, the ability to handle loops and networks of paths would increase the number of possible applications of the algorithm. Loop-detection in visual SLAM is an active research area (see the recent review by Williams et al. (2009)) and the incorporation of a fast, accurate loop-detection technique (such as FAB-MAP (Cummins and Newman, 2008)) along with further geometric consistency checking Eade and Drummond (2008) would provide two immediate benefits. First, the signal from a dedicated loop detection algorithm could totally replace the submap selection component of our system. Reliable automatic submap selection would make our algorithm more robust to path-tracking errors or VO failures. Second, this would allow the system to build a graph of connected route segments. The graph representation could be used to plan routes between places on the map (Stenning and Barfoot, 2011). Within the current framework, it should be possible to stack submaps at intersections (one submap per branch). However, navigating on a graph of routes would be more elegantly handled using a continuous relative representation (Sibley et al., 2009; Mei et al., 2009).

5.6 Conclusion

We have presented a complete algorithm for performing long-range rover navigation using a stereo camera as the only sensor. Our system produces a combined topological/metric map consisting of a sequence of small overlapping submaps. As the rover progresses along a path, the nearest submap is swapped into memory. The rover interleaves VO and localization, using VO to carry the algorithm through areas with moderate appearance changes and using localization to ensure the rover ends up in the same physical place at the end of a long path. We have tested our algorithm in an urban setting, over extreme terrain, through indoor-to-outdoor lighting changes, and in a planetary analogue setting in the High Arctic that offered many types of vegetation-free terrain. Of the 32.919 kilometers travelled, 99.6% was traversed autonomously, and in all situations requiring an intervention, the rover stopped and signalled the operator.

Specifically, we believe the contributions of this chapter are:

1. Extension of the VO pipeline to be a complete mapping and localization system

specifically designed to enable long-range autonomous rover navigation using only a stereo camera, as would be required in a planetary exploration context.

2. Demonstration of the system's performance through 32 kilometers of autonomous operation. To the best of our knowledge, this is the first teach-and-repeat navigation system shown to work over multi-kilometer autonomous traverses in highly three-dimensional, outdoor, unstructured environments, without the use of GPS.
3. Evaluation of our system in the following areas:
 - (a) sensitivity to path following errors Section 5.4.1,
 - (b) sensitivity to lighting changes in Section 5.4.2,
 - (c) performance of the localization algorithm during route repeating in Section 5.4.3
 - (d) an evaluation of which image features are most useful for route repeating in Section 5.4.4, and
 - (e) a summary of the major causes of failure of the algorithm in Sections 5.4.5 and 5.4.6.

One shortcoming of the system described in this chapter was its sensitivity to changes in ambient lighting. In the next chapter, we propose one possible solution: the extension of the VO pipeline to actively illuminated lidar sensors that do not require the presence or consistency of ambient light.

Chapter 6

Towards Appearance-Based Methods for Lidar Sensors

Although cameras have emerged as the dominant sensor modality for localization and mapping in three-dimensional, unstructured terrain, all camera-based systems are inherently reliant on the presence of ambient light. This poses a serious problem if we want to use these systems in outdoor environments that lack adequate and consistent light, such as permanently shadowed craters of the Moon. It could also be a problem if we desire to recognize previously visited places under different lighting conditions. As we showed in section 5.4.2, even changes in ambient lighting over the course of a single day can result in recognition failures. There have been some attempts to develop robust odometry systems that are invariant to lighting, such as downward-looking optical odometers (Dille et al., 2009; Wettergreen et al., 2010), and while the results are promising, we would like to develop a system that leverages the years of research into camera-based VO; this would not only provide motion estimation in the dark, but it would enable the use of other appearance-based algorithms such as the applications developed in Chapters 4 and 5.

In this chapter, we explore the possibility of extending the sparse VO pipeline to an actively illuminated sensor, in an effort to reduce or eliminate the dependence on external lighting. Our aim is to take the lessons learned from the success of camera-based systems and apply the same methods to systems that use laser-based sensors, such as the lidar depicted in Figure 6.1. The key insight is that, in addition to range data, lidar sensors also provide intensity information, allowing one to construct an intensity image that looks very similar to a standard greyscale camera image.

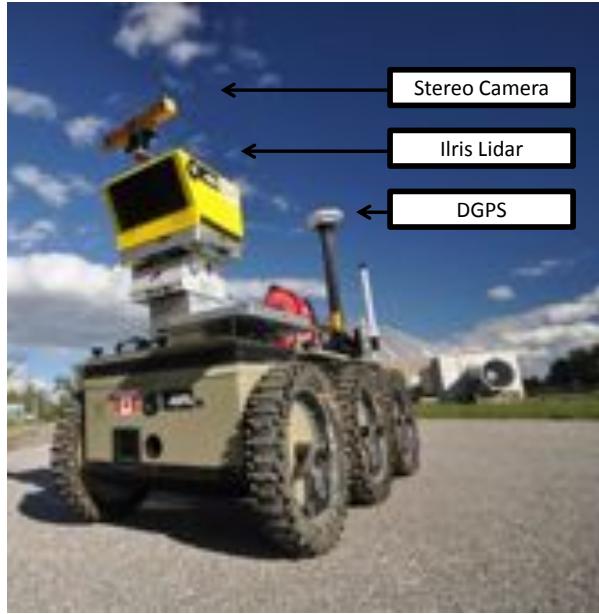


Figure 6.1: Our field robot with a Bumblebee XB3 stereo camera, a Thales DG-16 Differential GPS unit and an Optech Iliris 3D lidar sensor, which is a three-dimensional surveying lidar that has a vertical and horizontal field of view of 40°.

Recognizing that laser intensity images¹ provide a greyscale image of a scene is not a new idea. Kretschmer et al. (2004) point out that in surveying, the intensity images are often used by the surveyor to obtain a photo-realistic impression of the scanned area. In fact, most commercial surveyors use various reflective markers in the scene to act as tie points between different scan positions (Dold and Brenner, 2006). In an effort to automate this scan registration process, Bohm and Becker (2007) developed a marker-free method for point cloud registration that uses point correspondences from the intensity images to estimate the rigid body transformations between point clouds. SIFT features were extracted from the intensity images and RANSAC was used for outlier detection. In order to dampen the areas of low and very high reflectance, histogram equalization was used on all of the raw intensity images (Bohm and Becker, 2007).

In the mobile robotics literature, few have actually incorporated intensity information from a laser sensor for motion estimation. Neira et al. (1999) developed a sensor fusion technique in planar environments using their variant of the Extended Kalman Filter, called the SPfilter, which incorporated both range and intensity data to localize against a known map. Guivant et al. (2000) described a SLAM system that used the intensity

¹Also referred to as *reflectance images*.

data from their lidar to identify reflective markers on landmarks in the environment, which simplified the data association problem.

The most relevant research to date comes from May et al. (2009), who developed 3D mapping and egomotion estimation techniques using a Swiss Ranger Time-of-Flight (ToF) camera, which measures distances based on the phase-shift principle. The Swiss Ranger uses an array of 24 LEDs to simultaneously illuminate a scene, offering the advantage of higher framerates than are possible with scanning lidar. However, ToF cameras often have a limited field of view, short maximum range, and are very sensitive to environmental noise (May et al., 2009).

Using intensity images generated from their ToF camera, May et al. employed two appearance-based methods for motion estimation: a KLT-tracker and frame-to-frame VO using SIFT features. Their results indicated that the SIFT approach yielded more accurate motion estimates than the KLT approach, but less accurate than their iterative-closest-point method, which used a network-based global relaxation algorithm (Borrman et al., 2008).

Although May et al. (2009) demonstrated that frame-to-frame VO might be possible with a ToF camera, the largest environment in which they tested was a 20m long indoor hallway, with no groundtruth. Thus, a number of important questions still remain. In particular, we set out to answer the following questions:

1. How stable are descriptive features in lidar intensity images under changes in ambient light?
2. Can we perform VO using a scanning lidar in a stop-scan-go methodology with comparable results to stereo VO?

In this chapter, we show that it is indeed possible to use the intensity images from a lidar sensor and apply appearance-based techniques that have been traditionally used with camera imagery.

This chapter is organized as follows. In Section 6.1 we describe the lidar sensor used in this study and discuss the image formation and image processing methods required to convert raw lidar intensity data into greyscale intensity images. In Section 6.2, we explore the stability of a representative keypoint detection and description algorithm on both camera images and lidar intensity images collected over a 24 hour period outdoors. In Section 6.3 we validate our approach by implementing VO based on SBA on a sequence

of lidar intensity images. The VO results are compared to stereo VO and DGPS. In Section 6.4 we present a discussion of our findings, with particular emphasis on the limitations and the current obstacles encountered with lidar sensors. The experiments described in this chapter were included as part of McManus et al. (2011)

6.1 Preliminaries

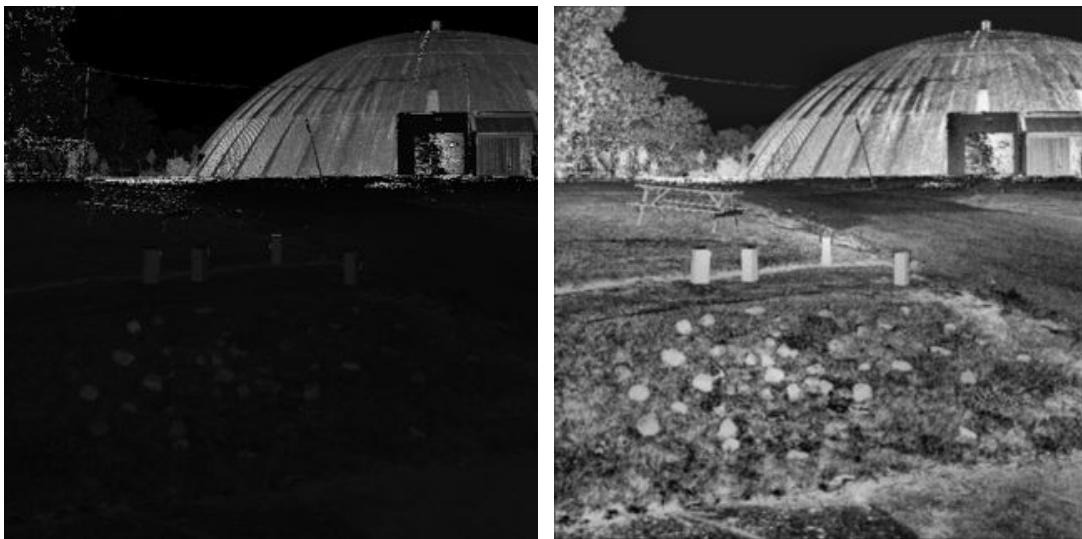
The lidar sensor used in this study was an Optech Ilris 3D, which has a maximum horizontal and vertical field of view of 40 degrees. The Ilris is a ToF lidar that can acquire 2,500 points/second and offers a large dynamic scanning range from 3 meters to over 1 kilometer, with subcentimeter accuracy up to 100 meters (for our experiments, we restricted the maximum range to 500 meters). Since it is a surveying sensor, the Ilris was not designed for high framerate data acquisition and consequently, we had to run our robot in a stop-scan-go fashion for the localization experiment. Having said this, there are currently several options available for high framerate 3D lidar sensors, such as the LVC0702 lidar from Autonosys², and the techniques described in this chapter are applicable to these sensors as well. In addition to the lidar, we also used a Point Grey Research Bumblebee XB3 stereo camera for our VO experiment, in order to provide a comparison with our lidar VO. For groundtruth, we used a Thales DG-16 Differential GPS unit.

To form images from the raw lidar data, we need to develop a camera model. As this lidar provides equally spaced samples in azimuth and elevation, we were able to use the raw data directly in a spherical camera model (see Figure 6.2(a) for an example of a raw lidar intensity image). Once the raw intensity image is formed, image processing is required to equalize the areas of high and low reflectance. Taking a similar approach as Bohm and Becker (2007), we use adaptive histogram equalization and then smooth the image with a Gaussian low-pass filter. The processed image is shown in Figure 6.2(b). The justification for this image processing is described in the next section, since our decision was based on the results from our light sensitivity analysis.

As in other chapters, we used our GPU implementation of the SURF (Bay et al., 2008) algorithm³ to find keypoints in every camera and lidar intensity image. The SURF algorithm returns a strength value for every keypoint. The strength value encodes the

²<http://www.autonosys.com/lidar.htm>

³<http://asrl.utias.utoronto.ca/code/gpusurf>



(a) Raw lidar intensity image. The metal dome in the background is the most prominent object in the image, since its reflectivity is much higher than the surrounding environment.

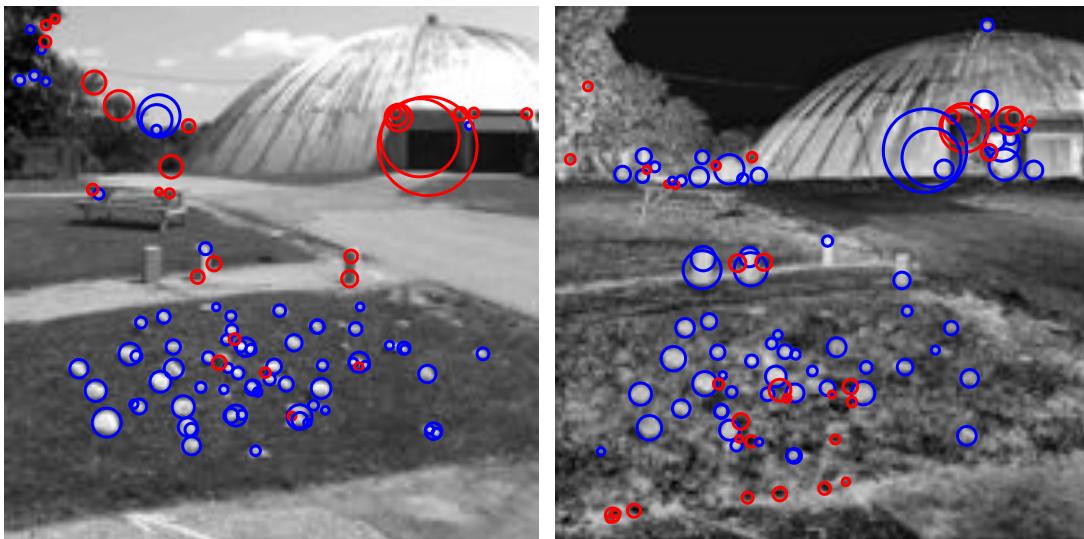
(b) Processed lidar intensity image, after applying adaptive histogram equalization and a Gaussian low-pass filter. The adaptive histogram equalization dampens the areas of high and low reflectance, bringing into focus many rich features in the foreground.

Figure 6.2: Illustrating the image processing required to transform the raw intensity image into a textured greyscale image. All images are 640x640.

intensity of the detector’s response at the keypoint location and scale. Keypoints corresponding to regions with high-contrast boundaries have higher strength values than those with low-contrast boundaries, and the high-contrast keypoints tend to be detected more reliably in the presence of noise in the image. Figure 6.3 shows an example of the strongest 100 SURF keypoints detected in a camera/lidar intensity image.

6.2 Lighting Sensitivity Analysis

In this section, we describe an experiment to analyze the stability of sparse appearance-based methods for lidar intensity images and compare them to camera intensity images, under a variety of lighting conditions. We begin with a description of the experiment and conclude with our results.



(a) Camera intensity image with the 100 strongest keypoints plotted. Obviously, in the case of the camera images, the number of keypoint matches drops off significantly as the time of day changes.

(b) Processed lidar intensity image, with the 100 strongest keypoints plotted. Although the intensity images did change as the ambient light changed, the imagery was relatively stable over a 24h period.

Figure 6.3: Camera and lidar intensity images of the same scene, taken at the same time of day with sample SURF keypoints shown. Blue circles represent light-on-dark blobs and red circles represent dark-on-light blobs.

6.2.1 Experiment Description

This experiment was conducted outdoors over a 24 hour period and consisted of taking camera images and lidar scans at half-hour intervals of an outdoor scene. Since the purpose of this experiment was to analyze the impact that lighting changes had on lidar/camera intensity images, the sensors were mounted on a tripod that remained stationary for the entire experiment. To quantify the stability of the keypoint detection over the 24 hours, we used an adaptive threshold to find the 500 strongest SURF keypoints in every image in the dataset. For both the camera and the lidar, keypoint matching using position, scale, and descriptor similarity was performed between all possible pairs of images. Given $N_{i,j}$ matches between image i and image j , the similarity score, $S_{i,j}$, is

$$S_{i,j} := \frac{N_{i,j}}{N_{\max}}, \quad (6.1)$$

where N_{\max} is the maximum number of possible matches (500 in this case).

Table 6.1: Mean similarity scores for camera/lidar intensity images with different image processing methods.

Image Processing	Camera	Lidar
(i) None	0.092 ± 0.150	0.329 ± 0.101
(ii) Histogram Eq.	0.069 ± 0.135	0.443 ± 0.127
(iii) Gaussian Filter	0.094 ± 0.151	0.375 ± 0.110
(iv) Histogram Eq. + Gaussian Filter	0.073 ± 0.150	0.502 ± 0.161

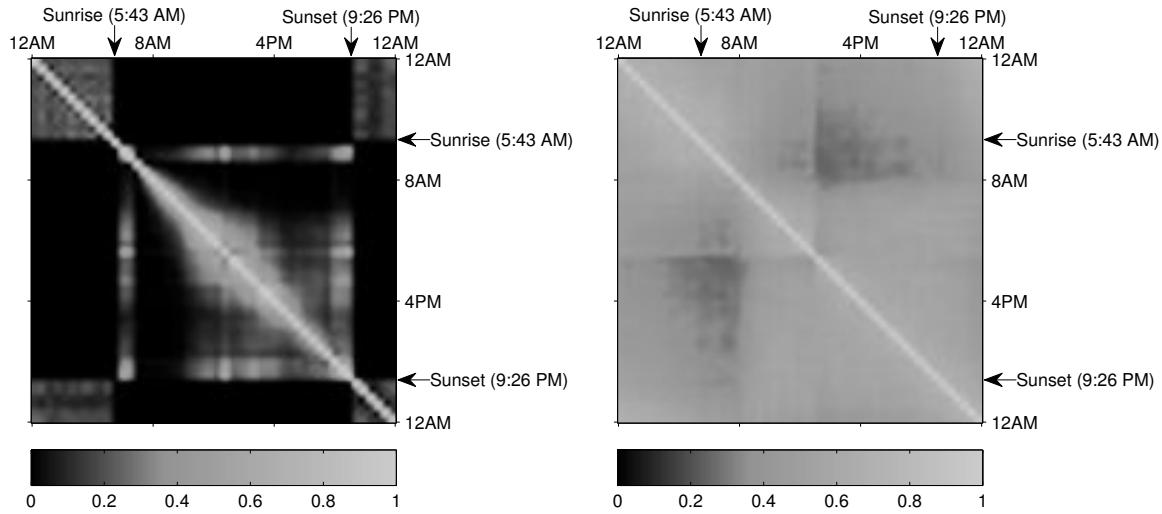
6.2.2 Results

We have created similarity matrices composed of the similarity scores between all pairs of images. Figure 6.4 shows the similarity matrices for the camera and lidar over the entire 24 hour period, where the light values represent a greater number of matches and the dark values represent fewer matches. We compared camera and lidar similarity matrices for four different image processing options: (i) no image processing, (ii) adaptive histogram equalization, (iii) Gaussian low-pass filter, and (iv) adaptive histogram equalization and a Gaussian low-pass filter (only the best two similarity matrices are included as figures). Of the four options, we determined that applying adaptive histogram equalization and a Gaussian low-pass filter was the best for the lidar intensity images, while applying just the Gaussian low-pass filter provided a slight improvement with the camera images. Table 6.1 shows the mean similarity scores and standard deviation for all image processing options.

As expected, for the camera, the number of keypoint matches between daytime and nighttime images drops off significantly (in fact, goes to zero), which produces the dark boxes in the similarity matrix. The structure within the camera's similarity matrix illustrates its sensitivity to ambient light and clearly demonstrates why methods such as visual teach and repeat show a decrease in performance under changing lighting conditions. The similarity score ranged from as low as zero to as high as 0.810 with a mean score of 0.094 ± 0.151 .

In the case of the lidar intensity images, we found that the similarity score does drop off from day to night, indicating a potential sensitivity to ambient light. In fact, the similarity scores exhibited a larger deviation than expected, ranging from as low as 0.250 to as high as 0.738 throughout the entire 24 hour period (the mean score was

0.502 ± 0.161). However, even for the most drastic changes in ambient light (i.e., from light to dark), we were still able to find at least 125 keypoint matches, which is more than sufficient for motion estimation.



(a) Camera intensity image similarity matrix. The similarity score ranged from 0.0–0.810 (i.e., 0–405 keypoint matches) with a mean score of 0.094 ± 0.151 . The similarity matrix exhibits sharp transitions at sunrise and sunset.

(b) Lidar intensity image similarity matrix. The similarity score ranged from 0.250–0.738 (i.e., 125–369 keypoint matches) with a mean score of 0.502 ± 0.161 . The lowest similarity scores occur between scans separated by approximately 12 hours.

Figure 6.4: Similarity matrices over a 24 hour period. The similarity score is $S_{i,j} := N_{i,j}/N_{\max}$, where $N_{i,j}$ is the number of matches between image i and image j and N_{\max} is the maximum number of possible matches, which was 500 in our experiment.

6.3 Visual Odometry

The VO pipeline was the enabling technology for the results presented in Chapters 4 and 5, and hence it represents an excellent algorithm to validate our appearance-based lidar technique. We use an implementation of the sliding window VO algorithm described in Konolige et al. (2007) to produce two motion estimates: one from stereo camera data, and the other from lidar scans. Processing the lidar data only requires differences in two areas: (i) keypoint formation, and (ii) error terms and associated Jacobians. The rest of the code blocks—keypoint tracking, outlier rejection using RANSAC, and SBA—

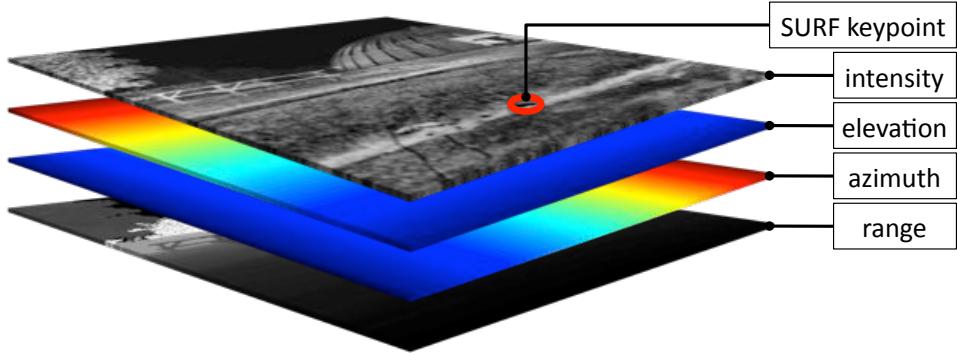


Figure 6.5: The image stack generated from a single lidar scan. SURF keypoints are found in image space at subpixel locations and bilinear interpolation is used to find the azimuth, elevation, and range of the keypoint. Linearized error propagation from image space to azimuth/elevation/range is then used to determine the uncertainty of the measurement.

are identical to the traditional camera-based approach as described in Chapter 2. We therefore restrict ourselves to describing the measurement formation and error terms.

6.3.1 Measurement and Error Terms from Lidar Data

The output of image formation is a stack of images, \mathcal{I} —intensity (\mathcal{I}_ℓ), azimuth (\mathcal{I}_θ), elevation (\mathcal{I}_ϕ), and range (\mathcal{I}_r)—derived from the raw lidar output and shown in Figure 6.5. The stack may be evaluated at any integer row, r , and column, c , as, \mathcal{I}_{rc} , a 4×1 column,

$$\mathcal{I}_{rc} := \mathcal{I}(r, c) = [\ell_{rc} \ \theta_{rc} \ \phi_{rc} \ r_{rc}]^T, \quad (6.2)$$

where ℓ_{rc} , θ_{rc} , ϕ_{rc} , and r_{rc} are the scalar intensity, azimuth, elevation, and range stored at this location in the image stack. Although intensity and range are correlated, we make the simplifying assumption that the elements of each image are independent, identically-distributed samples such that

$$\mathcal{I}_{rc} = \bar{\mathcal{I}}_{rc} + \delta\mathcal{I}_{rc}, \quad \delta\mathcal{I}_{rc} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}), \quad (6.3a)$$

$$\mathbf{R} := \text{diag}\{\sigma_\ell^2, \sigma_\theta^2, \sigma_\phi^2, \sigma_r^2\}, \quad (6.3b)$$

where $\bar{\mathcal{I}}_{rc}$ is the true value, $\delta\mathcal{I}_{rc}$ is zero-mean Gaussian noise, and \mathbf{R} is based on the sensor datasheet⁴.

At each timestep, k , keypoint detection returns a list of image locations, $\mathbf{y}_{k,j} = [u \ v]^T$, with associated covariances, $\mathbf{Y}_{k,j}$, where u , and v are generally not integers.

⁴<http://www.optech.ca/i3dtechoverview-IIris.htm>

We use bilinear interpolation of \mathcal{I} to produce an azimuth/elevation/range measurement, $\mathbf{z}_{k,j}$. The uncertainty, $\mathbf{Q}_{k,j}$, associated with $\mathbf{z}_{k,j}$, is produced by propagation of \mathbf{R} and $\mathbf{Y}_{k,j}$ through the interpolation equations.

The standard error term used in VO systems is *reprojection error*—the difference between the observed keypoint location and the predicted keypoint location given the current state estimate. We use a similar error term based on the spherical camera model used to form the intensity images. As in Section 5.2.2 the output of the outlier rejection step is a set of feature tracks indexed by $n = 1 \dots N$. Using this notation, track n indicates that keypoint j at time k is an observation of landmark i . The error term, \mathbf{e}_n , is

$$\mathbf{e}_n := \mathbf{z}_{k,j} - \mathbf{g}(\mathbf{T}_{c_k,m} \mathbf{p}_m^{i,m}), \quad (6.4)$$

where $\mathbf{p}_m^{i,m}$ is a column of state variables for landmark i expressed in the map frame, \mathcal{F}_m , the transformation matrix, $\mathbf{T}_{c_k,m}$, takes points from \mathcal{F}_m to the camera frame at time k , \mathcal{F}_{c_k} , and $\mathbf{g}(\cdot)$ is our spherical camera model.

6.3.2 Experimental Data

The data for this experiment was collected at the University of Toronto Institute for Aerospace studies. The Ilris lidar, a Point Grey Research Bumblebee XB3 stereo camera, and a Thales DG-16 Differential GPS unit were mounted on our large field robot in the configuration shown in Figure 6.1. The lidar scans, stereo images, and DGPS points were collected every 0.5 meters over a traverse of approximately 200 meters. As previously mentioned, the Ilris is not a high framerate 3D lidar, so we were limited to this stop-and-scan methodology. The dataset took approximately 13 hours to collect, allowing for a wide range of lighting conditions.

6.3.3 Results

Figure 6.6 shows a two-dimensional view of our VO localization results with DGPS for groundtruth. For this dataset, stereo VO outperformed our lidar VO, achieving a total RMS path error of 1.6% of the distance travelled, while the lidar VO achieved a total RMS path error of 2.9% of distance travelled. We believe the stereo motion estimate was more accurate due to longer temporal feature track lengths. The shorter track lengths in the lidar sequence may be due to the sensor’s spatial under-sampling (see the Discussion

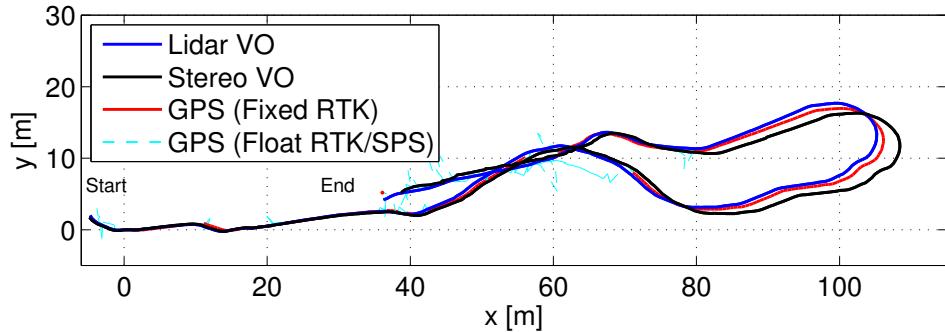


Figure 6.6: Two-dimensional view of localization results with DGPS for groundtruth. Due to external interference, our DGPS switched in and out of its fixed Real Time Kinematic (RTK) mode and into its less accurate float RTK and Standard Positioning Service (SPS) mode, causing noticeable jumps in the data. These have been illustrated with a different colour.

section below), but an investigation of how to track features accurately through intensity images should be a topic of future work.

Figure 6.7 shows the number of inlying keypoint matches for both the lidar and the stereo camera over the entire traverse. For most of the traverse, both lidar and stereo achieved a similar number of matches. However, after sunset, we see a significant decrease in matches for stereo, while the number of matches found with lidar remains unchanged. Thus, on average, we were actually able to detect a greater number of inlying keypoint matches using lidar.

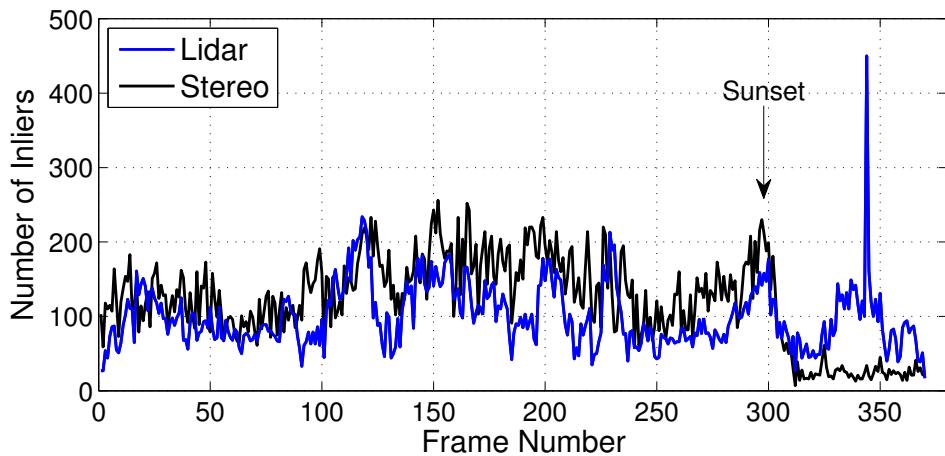


Figure 6.7: Number of inlying keypoints versus frame number.

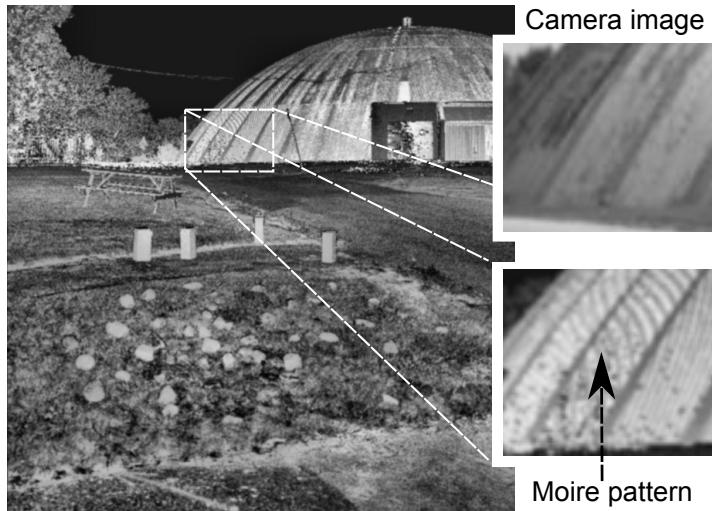


Figure 6.8: Zoomed-in view of the dome in one of the lidar intensity images taken from the light sensitivity experiment. The arching light and dark bands on the surface of the dome create what known as a Moire pattern. A camera image of the same segment of the dome is shown for comparison.

6.4 Discussion

Although we have demonstrated that lidar sensors can be used for appearance-based vision methods, there are a number of unique challenges and limitations that need to be addressed. The most obvious and prominent limitation with a lidar sensor is aliasing, which results from spatial under-sampling of the scene. Figure 6.8 shows an enlarged view of one of the lidar intensity images taken from our 24 hour experiment, where we see spatial aliasing in the form of a Moire pattern (Oster and Nishijima, 1963). As a preliminary treatment of aliasing, we employed a simple Gaussian low-pass filter; however, this was not sufficient to completely eliminate the visual distortion. Clearly more appropriate anti-aliasing algorithms need to be investigated.

Another challenge with scanning lidar is that, in complex environments, small changes in sensor orientation can result in large range deviations since we only measure point samples in the scene; objects that have a high angle of incidence with respect to the laser beam or thin objects such as a sign post, can display large range deviations. Figure 6.9 shows a pixel-wise standard-deviation range image, which was computed by taking all of the range images we gathered from our light sensitivity experiment and computing the standard deviation of the range values. As can be seen, objects such as the power line, the top of the dome, and the chain link fence represent regions were the range can deviate

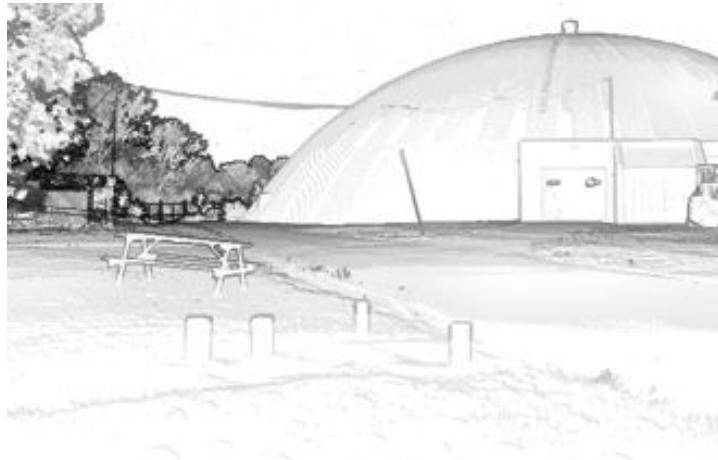


Figure 6.9: Standard deviation range image, where black represents the largest deviation. For visual clarity, we applied adaptive histogram equalization to the raw standard deviation image. Note that objects in the distance, such as the top of the dome, the trees and the fence, exhibit the largest range deviations, which are caused by small perturbations in the orientation of the lidar.

significantly. Since we use bilinear interpolation to compute keypoint measurements, if one of the neighbouring range values used in our interpolation deviates significantly from the others, it results in a biased range reading. For this reason, we rejected any keypoint measurements where the difference between the maximum and minimum range value was beyond a certain threshold, which occurs mostly at structure boundaries.

6.5 Conclusion

This chapter has demonstrated how active laser-based sensors, such as a lidar, can be used for appearance-based motion estimation techniques. Like a camera, a lidar sensor can produce intensity images, allowing the use of sparse keypoint detectors.

Specifically, we believe the contributions of this chapter are:

1. An evaluation of the stability of SURF keypoints in lidar intensity images and traditional camera images of an outdoor scene over a 24 hour period presented in Section 6.2. We demonstrate that lidar sensors do in fact exhibit a sensitivity to changes in ambient light. However, even in the worst case of trying to match keypoints from a daytime scan to a nighttime scan, we were still able to match 25% of the keypoints.

2. An explanation of how to build keypoint measurements from a lidar image stack in Section 6.3.1. The keypoints are suitable for use in the sparse VO pipeline.
3. The extension of the sparse VO pipeline from Chapter 2 to lidar intensity images. To the best of our knowledge, this is the first use of a scanning lidar sensor in the sparse VO pipeline.
4. An evaluation of lidar VO and a comparison with stereo VO over a 200 meter traverse using DGPS for groundtruth in Section 6.3.

This work marks an important step towards enabling appearance-based vision techniques in environments that are not conducive to passive cameras and presents new possibilities for a number of light-dependent autonomous navigation methods, such as the teach-and-repeat navigation algorithm described in Chapter 5.

There is still a great deal of work that needs to be done to fully realize the potential of active laser sensors. In our experiments, we were limited to driving the robot in a stop-scan-go fashion because we did not have access to a high framerate lidar sensor. We anticipate that using the techniques described in this chapter with a sensor that is scanning continuously while the rover is in motion would result in significant image distortion. While such distortion would almost certainly result in degraded motion estimates, it is possible that it would not inhibit the use of such a sensor for teach-and-repeat navigation; as shown in Chapter 5, teach-and-repeat navigation does not require accurate motion estimates.

Chapter 7

Summary and Future Work

This chapter presents a short summary of the contributions emanating from this thesis, together with references to papers published describing this research. We then conclude by suggesting promising directions for future work.

7.1 Summary of Contributions and Publications

The mission concept to find and sample ground ice on Mars and our field campaigns to Devon Island have resulted in a number of overview publications. They include high-level descriptions of the work presented in this thesis along with work done by collaborators toward the same mission concept. Barfoot et al. (2010a) and Barfoot et al. (2010c) describe our 2008 field campaign and our work on the coupled surface/subsurface modelling pipeline within the context of the mission concept. Barfoot et al. (2010b) and Barfoot et al. (2011a) give an overview of our 2009 field tests. These papers discuss results using VO to collect GPR data and then using visual teach and repeat to return to the lander. Barfoot et al. (2010d) evaluates Devon Island as a worthwhile site for rover testing, and provides a high-level summary of all known rover tests at the site.

Results from the first part of Chapter 3 have appeared in Barfoot et al. (2011b), and in the complete estimator derivation in Furgale et al. (2010). In brief, the contributions of Chapter 3 are:

1. A first-principles derivation of the multiplicative constraint-sensitive perturbations of a 4×4 transformation matrix and a unit-length 4×1 homogeneous point given by (3.52) and (3.86) respectively. These may be used to linearize any expression involving a transformation matrix, or homogeneous point.

2. Expressions for updating transformation matrices and unit-length homogeneous points with a constraint-sensitive perturbation are provided in (3.56), and (3.83) respectively. These updates avoid the need to restore constraints afterwards.
3. Development of a number of identities for manipulating expressions involving linearized transformation matrices. These identities are given in (3.61), (3.65), (3.66), (3.69), (3.71), and (3.72).
4. Demonstration of linearizing a stereo-camera-model error term involving a transformation matrix and homogeneous point landmark. The resulting linearized error term, given by (3.99) is used for the VO estimates in subsequent chapters.
5. Demonstration of how to build and linearize an error term representing a Gaussian prior on a rotation matrix. The linearized error term is given in (3.114).

The VO algorithm described in Chapter 2 has been used for collaborations with colleagues in the Autonomous Space Robotics Laboratory in a number of ways. Carle et al. (2010) use VO to constrain relative transformations between lidar scans which are matched to digital elevation maps in order to provide global localization for planetary exploration rovers in the absence of GPS. Lambert et al. (2011) extend the pipeline from Chapter 2 by including a sun sensor and inclinometer directly in the pose solution. Taken together, these added sensors provide global attitude corrections that greatly increase the accuracy of VO over long distances. Portions of the data and code produced while developing this VO algorithm have been released. “Speeded Up Speeded Up Robust Features” is an open source, GPU implementation of the SURF algorithm available at <http://asrl.utias.utoronto.ca/code/gpusurf>. Data collected from our field trials on Devon Island suitable for rover testing have been packaged for ease of use and are available at <http://asrl.utias.utoronto.ca/datasets/devon-island-rover-navigation/>.

The work from Chapter 4 on coupled surface/subsurface modelling appeared in Furgale et al. (2009) and Furgale et al. (2010). The contributions of Chapter 4 are:

1. Demonstration and field testing of a method to completely automate the GPR data collection process by using stereo VO to derive an estimate of the motion of the GPR antenna over the course of a transect. This method uses only onboard sensors slated to fly on future rover missions, making it suitable for a planetary exploration setting.

2. Demonstration and field testing of a method to use the raw data from a stereo camera and GPR along with the VO motion estimate to produce two novel data products:
 - (a) a topography-corrected radargram plotted with a two-dimensional profile of the surface along the transect (Section 4.2.1), and
 - (b) a photo-realistic three-dimensional surface/subsurface model (Section 4.2.2).

The results in Chapter 5 appeared as two conference publications at ICRA 2010 (Furgale and Barfoot, 2010c,b). Furgale and Barfoot (2010c) won the Kuka Service Robotics Best Paper Award at that conference. The complete description of the algorithm and field tests was published in Furgale and Barfoot (2010a). The contributions of Chapter 5 are:

1. Extension of the VO pipeline to be a complete mapping and localization system specifically designed to enable long-range autonomous rover navigation using only a stereo camera, as would be required in a planetary exploration context.
2. Demonstration of the system's performance through 32 kilometers of autonomous operation. To the best of our knowledge, this is the first teach-and-repeat navigation system shown to work over multi-kilometer autonomous traverses in highly three-dimensional, outdoor, unstructured environments, without the use of GPS.
3. Extensive evaluation of system performance.

The work on appearance-based lidar in ongoing and the results in Chapter 6 appeared as part of McManus et al. (2011). The contributions of Chapter 6 are:

1. Evaluation of the stability of SURF keypoints in lidar intensity images and traditional camera images of an outdoor scene over a 24 hour period presented in Section 6.2.
2. Explanation of how to build keypoint measurements from a lidar image stack in Section 6.3.1. The keypoints are suitable for use in the sparse VO pipeline.
3. Extension of the sparse VO pipeline from Chapter 2 to lidar intensity images. To the best of our knowledge, this is the first use of a scanning lidar sensor in the sparse VO pipeline.

4. Evaluation of lidar VO and a comparison with stereo VO over a 200 meter traverse using DGPS for groundtruth in Section 6.3.

7.2 Future Work

Based on the work in this thesis, there are many future directions to pursue, primarily following from the results in Chapters 5 and 6.

7.2.1 Autonomous Retrotraverse

The teach-and-repeat navigation system described in Chapter 5 was only capable of representing routes that were simple linear chains with no branches. The extension of this system to arbitrary graphs of routes raises a number of interesting possibilities for research.

Planning on a network of reusable paths: Acknowledging that globally consistent maps are not needed for robust autonomous rover operations, we may seek to build a complete *exploration system*¹ based on a network of reusable paths. This poses many challenges such as how to define goals, represent obstacles, and plan paths within a world representation that has no single privileged coordinate frame (work already started by Stenning and Barfoot (2011)).

Lifelong localization and mapping: The current state of the art in localization and mapping has shown it is possible to build maps of vast scale. However, maintaining appearance-based maps over time as environments change remains an unsolved problem.

World segmentation: As exploration algorithms enable robots to remain autonomous over long periods, it may be possible for autonomous systems to aggregate the large datasets collected over time with the goal of automatically segmenting the world into discrete spaces (such as rooms), and segmenting the contents of the world into classes of objects, by revisiting places and clustering similarities and differences.

¹An *exploration system* combines the related tasks of localization, mapping, and planning together into a single framework.

Semantic labelling: After clustering and segmenting the world into spaces and object classes, maps may be extended to include a semantic layer that seeks to capture high-level concepts about the world that can then be used by reasoning systems to accomplish complex goals.

7.2.2 Appearance-Based Lidar

The extension of the sparse VO pipeline to scanning lidar described in Chapter 6 is very preliminary and there are still many extensions to be realized.

Lighting-invariant teach and repeat: We show in Chapter 6 that it is possible to perform appearance-based VO using a scanning lidar, so the extension to teach-and-repeat navigation should be straightforward. This should result in a system that is much more robust to changes in ambient lighting—the main cause of failures of the camera-based system from Chapter 5. It should also allow rovers to operate autonomously in the dark, which would enable exploration of challenging locations such as permanently shadowed craters on the Moon.

Advances in joint range/intensity image processing: Because the image stack produced in Chapter 6 is dense—there is a range value for every intensity value—there should be some advances possible in joint range/intensity image processing. This could include joint range/intensity feature detection, dense estimation of scene structure and surface albedo, and the generation of landmark descriptors that encode both appearance and geometric information, and hence are more recognizable under changes in viewpoint.

Motion compensation for continuously scanning lidar in motion: Porting the work from Chapter 6 to a scanning lidar that collects data while the platform is in continuous motion will involve some challenges. One of the assumptions of the image-stack abstraction is that all pixels were captured at the same instant in time. This is not the case when a rover is collecting laser data while in continuous motion. Preliminary study shows that this produces significant image distortion which results in poor metric motion estimates. Designing an estimator that can deal with the continuous motion of the scanner should be possible but it will require careful timestamping of data and may involve some form of interpolation or continuous-time parametric state representation.

Full dark celestial navigation: It has already been shown that fusing visual motion estimation with a sun sensor and inclinometer can greatly increase the accuracy of motion estimation over long distances (Lambert et al., 2011). An actively illuminated sensor that is able to operate in the dark enables the possibility of fusing visual motion estimation with a star tracker and inclinometer. As star trackers measure the bearing of multiple celestial objects at once, the attitude estimates should be even more accurate than those using a sun sensor. However, this work will involve difficulties such as the need for careful timestamp synchronization and the problem of motion blur destroying the visibility of faint stars.

Appendix A

Catalogue of Route-Following Experiments

This appendix includes the complete listing of our route following experiments. Table A.1 lists each teach pass with some basic information such as the length of the route, the number of features per map and the number of failures encountered teaching the route. Table A.2 lists some difficulty metrics for each teach pass: the elevation change reported by GPS, and the roll and pitch variation measured by an inclinometer mounted on the sensor head. The repeat passes are listed in Tables A.3, A.4 and A.5. For each repeat pass we report the start time of both the teach and repeat passes (relevant for lighting effects), the percentage of the route completed, the percentage of run completed autonomously, the number of operator interventions, and the percentage of time the algorithm was globally localized.

Table A.1: List of teach passes.

Tag	Length (m)	Maps	Features per map	Failures
ua-05-17-0726	725.6	253	2937	0
uh-05-20-1152	1151.7	405	3484	0
uh-05-21-1170	1169.7	410	3424	0
uh-05-22-0120	120.0	42	5218	0
uh-07-23-0120	119.7	43	5323	0
uh-05-26-0202	201.8	71	3714	0
ua-06-04-0086	85.6	30	1924	0
ua-06-04-0048	47.7	17	1857	0
ua-06-04-0097	96.8	35	1581	3
ua-06-04-0091	91.1	31	1924	0
ua-06-06-0081	81.5	28	1718	0
ua-06-06-0184	184.4	65	1563	0
ua-06-06-0186	185.8	65	1975	1
ua-06-06-0167	166.9	60	1540	0
dh-07-20-2120	2120.0	740	3680	1
da-07-20-0464	464.0	166	1662	1
da-07-21-0453	453.5	161	1960	1
dh-07-22-1091	1090.9	382	5110	0
dh-07-23-4963	4962.5	1732	3993	5
da-07-23-0557	557.3	200	2126	0
da-07-24-0425	424.6	151	2225	0
da-07-29-0487	486.9	172	2318	0
da-07-29-0486	486.2	176	2415	0
dh-07-30-0347	347.3	124	3147	0
dh-07-30-0187	187.5	66	3322	0
dh-07-30-0153	152.8	54	3145	0
dh-07-31-0192	191.8	68	5431	0

Table A.2: List of teach passes with difficulty metrics.

Tag	Elevation change (m)	Min roll (deg)	Max roll (deg)	Min pitch (deg)	Max pitch (deg)
ua-05-17-0726	7.1	-7.8	12.2	-6.4	5.6
uh-05-20-1152	4.9	-18.4	8.2	-12.0	6.3
uh-05-21-1170	4.9	-17.8	11.9	-14.6	10.3
uh-05-22-0120	3.6	-14.7	11.7	-21.5	27.2
uh-07-23-0120	4.9	-13.3	12.1	-18.5	26.5
uh-05-26-0202	3.7	-5.5	3.5	-8.4	2.4
ua-06-04-0086	1.6	-8.0	2.9	-5.1	1.6
ua-06-04-0048	1.0	-6.7	5.5	-5.1	6.1
ua-06-04-0097	1.3	-6.6	3.3	-7.3	1.9
ua-06-04-0091	1.1	-5.8	3.1	-5.7	4.3
ua-06-06-0081	0.9	-9.4	4.0	-5.9	2.3
ua-06-06-0184	1.3	-7.9	3.7	-5.7	2.3
ua-06-06-0186	1.5	-10.4	4.9	-6.8	7.4
ua-06-06-0167	3.1	-8.8	3.1	-4.0	2.4
dh-07-20-2120	69.2	-22.0	15.9	-28.3	16.9
da-07-20-0464	8.7	-12.0	4.5	-9.7	6.9
da-07-21-0453	9.4	-13.0	4.3	-10.0	11.6
dh-07-22-1091	32.4	-7.4	10.0	-11.0	11.2
dh-07-23-4963	118.5	-12.8	13.6	-15.5	12.0
da-07-23-0557	36.7	-8.7	12.0	-11.7	13.4
da-07-24-0425	7.4	-6.7	5.9	-7.4	18.3
da-07-29-0487	9.1	-9.0	4.1	-13.0	7.5
da-07-29-0486	9.1	-6.6	3.4	-11.0	6.8
dh-07-30-0347	2.1	-12.2	13.1	-12.1	10.8
dh-07-30-0187	1.5	-6.1	8.5	-12.3	7.7
dh-07-30-0153	1.5	-6.9	5.9	-10.6	3.6
dh-07-31-0192	1.1	-9.8	10.9	-17.9	19.1

Table A.3: List of repeat passes (1/3). Column C is the percentage of the total route completed, column A is the percentage of the distance travelled autonomously, and column I is the number of operator interventions.

Teach pass tag	Teach pass start time	Repeat pass start time	C	A	I	Globally localized
ua-05-17-0726	12:27:12	14:27:02	100.0%	100.0%	0	41.1%
		12:50:02	100.0%	100.0%	0	91.8%
uh-05-20-1152	12:04:45	09:25:48	100.0%	100.0%	0	88.8%
uh-05-21-1170	12:16:02	20:26:28	100.0%	92.0%	1	49.6%
		12:54:14	100.0%	100.0%	0	99.2%
		15:28:08	100.0%	100.0%	0	97.4%
		16:14:01	100.0%	100.0%	0	99.1%
		08:03:44	100.0%	98.4%	3	45.0%
		09:04:15	100.0%	100.0%	0	76.3%
uh-05-22-0120	18:13:22	10:23:40	100.0%	100.0%	0	96.8%
		10:56:05	100.0%	100.0%	0	94.1%
		11:18:14	100.0%	100.0%	0	96.0%
		11:41:32	100.0%	100.0%	0	91.7%
		11:59:30	100.0%	100.0%	0	95.8%
		13:04:15	100.0%	100.0%	0	89.1%
		11:17:39	100.0%	100.0%	0	90.8%
uh-07-23-0120	16:53:43	18:39:23	100.0%	100.0%	0	93.6%
		19:00:53	100.0%	100.0%	0	97.3%
		19:12:07	100.0%	100.0%	0	95.8%
		19:22:49	100.0%	100.0%	0	97.8%
		19:33:36	100.0%	100.0%	0	95.5%

Table A.4: List of repeat passes (2/3). Column C is the percentage of the total route completed, column A is the percentage of the distance travelled autonomously, and column I is the number of operator interventions.

Teach pass tag	Teach pass start time	Repeat pass start time	C	A	I	Globally localized
uh-05-26-0202	11:14:39	07:44:53	100.0%	100.0%	0	95.9%
		08:53:08	100.0%	100.0%	0	95.5%
		09:41:09	100.0%	100.0%	0	88.7%
		10:41:07	100.0%	100.0%	0	87.5%
		11:39:57	100.0%	100.0%	0	95.0%
		12:38:06	100.0%	100.0%	0	94.7%
		13:34:20	100.0%	100.0%	0	98.8%
		14:38:29	100.0%	100.0%	0	86.0%
		14:58:39	100.0%	100.0%	0	88.8%
ua-06-04-0086	12:57:06	13:12:38	100.0%	100.0%	0	99.4%
ua-06-04-0048	14:41:09	14:51:15	100.0%	100.0%	0	97.2%
ua-06-04-0097	14:48:50	15:10:59	87.4%	95.5%	2	93.4%
ua-06-04-0091	16:26:46	16:37:13	100.0%	100.0%	0	84.5%
ua-06-06-0081	10:01:59	10:08:55	100.0%	100.0%	0	100.0%
ua-06-06-0184	10:25:07	11:35:48	100.0%	100.0%	0	99.7%
ua-06-06-0186	13:11:57	13:55:08	100.0%	98.4%	1	81.5%
ua-06-06-0167	21:09:38	21:45:16	100.0%	100.0%	0	100.0%
dh-07-20-2120	10:18:18	09:45:12	100.0%	100.0%	0	97.4%
		11:05:46	100.0%	100.0%	0	89.9%
da-07-20-0464	16:09:42	16:46:56	100.0%	100.0%	0	94.4%

Table A.5: List of repeat passes (3/3). Column C is the percentage of the total route completed, column A is the percentage of the distance travelled autonomously, and column I is the number of operator interventions.

Teach pass tag	Teach pass start time	Repeat pass start time	C	A	I	Globally localized
da-07-21-0453	17:26:10	18:07:05	100.0%	100.0%	0	98.4%
dh-07-22-1091	08:55:08	10:35:23	100.0%	100.0%	0	99.5%
		11:14:58	100.0%	100.0%	0	100.0%
		12:00:56	100.0%	100.0%	0	99.2%
		13:40:39	100.0%	100.0%	0	99.9%
		14:13:58	100.0%	100.0%	0	97.5%
dh-07-23-4963	08:50:49	08:49:24	64.9%	100.0%	0	96.3%
		15:07:49	32.8%	100.0%	0	76.2%
da-07-23-0557	17:19:32	18:21:22	100.0%	100.0%	0	99.2%
da-07-24-0425	16:24:23	17:20:45	100.0%	100.0%	0	98.4%
da-07-29-0487	11:41:31	15:28:45	100.0%	100.0%	0	96.7%
da-07-29-0486	16:04:04	15:28:45	100.0%	100.0%	0	96.7%
		17:00:29	100.0%	100.0%	0	96.0%
dh-07-30-0347	12:03:40	12:49:47	100.0%	100.0%	0	89.5%
dh-07-30-0187	14:35:09	18:37:18	78.4%	100.0%	0	9.9%
		10:02:35	100.0%	100.0%	0	75.7%
dh-07-30-0153	16:33:18	16:59:04	100.0%	100.0%	0	93.9%
dh-07-31-0192	11:40:01	12:27:23	100.0%	100.0%	0	100.0%
		12:50:01	100.0%	100.0%	0	92.7%

Appendix B

List of Acronyms

CUDA	NVIDIA's Compute Unified Device Architecture http://www.nvidia.com/object/cuda_home_new.html
DGPS	Differential Global Positioning System
ESM	Efficient Second-order Minimization (Malis and Benhimane, 2005)
GPR	Ground-Penetrating Radar
GPS	Global Positioning System
GPU	Graphics Processing Unit
HMP-RS	Haughton-Mars Project Research Station
IMU	Inertial Measurement Unit
ISM	Instant Scene Modeler
lidar	light detection and ranging
MARSIS	Mars Advanced Radar for Subsurface and Ionosphere Sounding
MER	Mars Exploration Rover
RAM	Random Access Memory
RANSAC	random sample consensus (Fischler and Bolles, 1981)
RTK-GPS	Real-Time Kinematic Global Positioning System

SBA	Sparse Bundle Adjustment
SHARAD	Shallow Subsurface Radar
SIFT	Scale Invariant Feature Transform (Lowe, 2004)
SLAM	Simultaneous Localization and Mapping
SURF	Speeded-Up Robust Features (Bay et al., 2008)
ToF	Time-of-Flight
VO	Visual Odometry
VRML	Virtual Reality Modelling Language

Bibliography

- Aggarwal, J. and Nandhakumar, N. (1988). On the computation of motion from sequences of images: A review. *Proceedings of the IEEE*, 76:917–935. (ref. page 5)
- Annan, A. (2009). Electromagnetic principles of ground penetrating radar. In Jol, H. M., editor, *Ground Penetrating Radar Theory and Applications*, pages 1 – 40. Elsevier, Amsterdam. (ref. pages 45 and 60)
- Annan, A. and Davis, J. (1976). Impulse radar sounding in permafrost. *Radio Science*, 11:383–394. (ref. page 48)
- Arcone, S., Lawson, D., and Delaney, A. (1995). Short-pulse radar wavelet recovery and resolution of dielectric constants within englacial and basal ice of matanuska glacier, alaska, usa. *Journal of Glaciology*, 41:68–86. (ref. page 47)
- Arcone, S., Prentice, M., and Delaney, A. (2002). Stratigraphic profiling with ground-penetrating radar in permafrost: A review of possible analogs for mars. *Journal of Geophysical Research*, 107(E11):5108. (ref. page 48)
- Argyros, A. A., Bekris, K. E., Orphanoudakis, S. C., and Kavraki, L. E. (2005). Robot homing by exploiting panoramic vision. *Auton. Robots*, 19(1):7–25. (ref. page 65)
- Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (slam): part ii. *Robotics & Automation Magazine, IEEE*, 13(3):108–117. (ref. page 13)
- Barfoot, T., D’Eleuterio, G., and Annan, P. (2003). Subsurface surveying by a rover equipped with ground penetrating radar. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2541–2546, Las Vegas, NV. (ref. pages 45, 48, 54, and 60)

- Barfoot, T., Furgale, P., Stenning, B., and Carle, P. (2010a). Field testing of rover gn&c techniques to support a ground-ice prospecting mission to mars. In *Proceedings of the Planetary Rovers Workshop, IEEE International Conference on Robotics and Automation (ICRA)*., Anchorage, Alaska, USA. (ref. page 115)
- Barfoot, T., Furgale, P., Stenning, B., Carle, P., Thomson, L., Osinski, G., and Daly, M. (2010b). Field testing of rover gn&c techniques to support a ground-ice prospecting mission to mars. In *Proceedings of the 10th International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Sapporo, Japan. (ref. page 115)
- Barfoot, T., Furgale, P., Stenning, B., Carle, P., Thomson, L., Osinski, G., Daly, M., and Ghafoor, N. (2011a). Field testing of a rover guidance, navigation, & control architecture to support a ground-ice prospecting mission to mars. *Robotics and Autonomous Systems*, 59(6):472–488. (ref. pages 19 and 115)
- Barfoot, T., Se, S., and Jasobedzki, P. (2006). *Visual Motion Estimation and Terrain Modelling for Planetary Rovers*, chapter 4. Intelligence for Space Robotics. TSI Press, Albuquerque, NM. (ref. page 49)
- Barfoot, T. D. (2005). Online visual motion estimation using fastslam with sift features. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 579–585. (ref. page 13)
- Barfoot, T. D., Forbes, J. R., and Furgale, P. T. (2011b). Pose estimation using linearized rotations and quaternion algebra. *Acta Astronautica*, 68(1-2):101–112. (ref. pages 34, 43, and 115)
- Barfoot, T. D., Furgale, P. T., Osinski, G. R., Ghafoor, N., and Williams, K. (2010c). Field testing of robotic technologies to support ground-ice prospecting in martian polygonal terrain. *Planetary and Space Science*, special issue on “Exploring other worlds by exploring our own: The role of terrestrial analogue studies in planetary exploration”, 58(4):671–681. (ref. page 115)
- Barfoot, T. D., Furgale, P. T., Stenning, B. E., Carle, P. J. F., Enright, J. P., and Lee, P. (2010d). Devon island as a proving ground for planetary rovers. In Angeles, J., Boulet, B., Clark, J., Kovacs, J., and Siddiqi, K., editors, *Brain, Body, and Machine*:

- Proceedings of an International Symposium on the Occasion of the 25th Anniversary of the McGill University Centre for Intelligent Machines, Advances in Intelligent and Soft Computing* 83, pages 269–281, Montreal, Quebec. Springer. (ref. pages 83 and 115)
- Baumgartner, E. T. and Skaar, S. B. (1994). An autonomous vision-based mobile robot. *Automatic Control, IEEE Transactions on*, 39(3):493–502. (ref. page 64)
- Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359. (ref. pages 8, 104, and 128)
- Bekris, K., Argyros, A., and Kavraki, L. (2006). Exploiting panoramic vision for bearing-only robot homing. *Imaging Beyond the Pinhole Camera*, pages 229–251. (ref. page 65)
- Bell, B. and Cathey, F. (1993). The iterated kalman filter update as a gauss-newton method. *Automatic Control, IEEE Transactions on*, 38(2):294 –297. (ref. page 18)
- Benoit, S. and Ferrie, F. (2007). Towards direct recovery of shape and motion parameters from image sequences. *Computer Vision and Image Understanding*, 105(2):145–165. (ref. page 5)
- Biesiadecki, J. and Maimone, M. (2006). The Mars Exploration Rover surface mobility flight software: driving ambition. *Aerospace Conference, 2006 IEEE*, page 15 pp. (ref. page 50)
- Blanc, G., Mezouar, Y., and Martinet, P. (2005). Indoor navigation of a wheeled mobile robot along visual routes. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3354–3359. (ref. page 65)
- Bohm, J. and Becker, S. (2007). Automatic marker-free registration of terrestrial laser scans using reflectance features. In *Proceedings of the 8th Conference on Optical 3D Measurement Techniques*, pages 338–344, Zurich, Switzerland. (ref. pages 102 and 104)
- Bonin-Font, F., Ortiz, A., and Oliver, G. (2008). Visual navigation for mobile robots: A survey. *Journal of Intelligent and Robotic Systems*, 53(3):263–296. (ref. page 64)
- Booij, O., Terwijn, B., Zivkovic, Z., and Kroese, B. (2007). Navigation using an appearance based topological map. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3927–3932. (ref. page 65)

- Borrman, D., Elseberg, J., Lingemann, K., Nuchter, A., and Hertzberg, J. (2008). Globally consistent 3d mapping with scan matching. *Journal of Robotics and Autonomous Systems*, 65(2). (ref. page 103)
- Bosse, M., Newman, P., Leonard, J., and Teller, S. (2004). Simultaneous localization and map building in large-scale cyclic environments using the atlas framework. *The International Journal of Robotics Research*, 23(12):1113–1139. (ref. page 62)
- Brooks, R. (1985). Visual map making for a mobile robot. In Al, editor, *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 824–829. (ref. pages 62 and 64)
- Brown, D. C. (1958). A solution to the general problem of multiple station analytical stereotriangulation. Rca-mtp data reduction technical report no. 43 (or afmtc tr 58-8), Patrick Airforce Base, Florida. (ref. page 13)
- Busby, J. P. and Merritt, J. W. (1999). Quaternary deformation mapping with ground penetrating radar. *Journal of Applied Geophysics*, 41(1):75–91. (ref. pages 45 and 49)
- Cabrol, N. A., Andersen, D. T., Stoker, C. R., Lee, P., McKay, C. P., and Wettergreen, D. S. (2010). Other analogs to mars: high altitude, subsurface, desert, and polar environments. In Doran, P. T., Lyons, W. B., and McKinght, D. M., editors, *Life in Antarctic Deserts and other Cold Dry Environments*, number 5 in Cambridge Astrobiology, pages 258–305. Cambridge University Press, Cambridge, UK. (ref. page 53)
- Cadena, C. and Neira, J. (2010). Slam in $O(\log n)$ with the combined kalman - information filter. *Robotics and Autonomous Systems*, 58(11):1207–1219. (ref. page 13)
- Carle, P., Furgale, P. T., and Barfoot, T. D. (2010). Long-range rover localization by matching lidar scans to orbital elevation maps. *Journal of Field Robotics*, 27(3):344–370. (ref. page 116)
- Carr, M. H. (1996). *Water on Mars*. Oxford University Press, New York. (ref. page 1)
- Cassidy, N. J. (2009). Ground penetrating radar data processing, modelling and analysis. In Jol, H. M., editor, *Ground Penetrating Radar Theory and Applications*, pages 141 – 176. Elsevier, Amsterdam. (ref. page 45)

- Chen, Z. and Birchfield, S. (2006). Qualitative vision-based mobile robot navigation. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2686–2692. (ref. pages 65 and 66)
- Cheng, Y., Maimone, M., and Matthies, L. (2006). Visual odometry on the Mars exploration rovers-a tool to ensure accurate driving and science imaging. *Robotics & Automation Magazine, IEEE*, 13(2):54–62. (ref. page 11)
- Chli, M. and Davison, A. (2009). Active Matching for visual tracking. *Robotics and Autonomous Systems*, 57(12):1173–1187. (ref. page 11)
- Comport, A. I., Malis, E., and Rives, P. (2010). Real-time Quadrifocal Visual Odometry. *The International Journal of Robotics Research*, page 0278364909356601. (ref. page 5)
- Corke, P., Lobo, J., and Dias, J. (2007). An introduction to inertial and visual sensing. *Int. J. Rob. Res.*, 26(6):519–535. (ref. page 98)
- Corke, P., Strelow, D., and Singh, S. (2004). Omnidirectional visual odometry for a planetary rover. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 4, pages 4007–4012 vol.4. (ref. page 5)
- Courbon, J., Mezouar, Y., and Martinet, P. (2008). Indoor navigation of a non-holonomic mobile robot using a visual memory. *Autonomous Robots*, 25(3):253–266. (ref. page 65)
- Crassidis, J. L., Markley, F. L., and Cheng, Y. (2007). A survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28. (ref. page 19)
- Cummins, M. and Newman, P. (2008). FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 27(6):647–665. (ref. page 99)
- Daniels, D., editor (2004). *Ground Penetrating Radar*. Peter Peregrinus Ltd. (ref. page 49)

- Davis, T. A. (2006). *Direct Methods for Sparse Linear Systems*, volume 2 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA. (ref. page 18)
- Davison, A. and Murray, D. (2002). Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 865–880. (ref. page 5)
- Davison, A., Reid, I., Molton, N., and Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1052–1067. (ref. page 11)
- Degenhardt, J. J. and Giardino, J. R. (2003). Subsurface investigation of a rock glacier using ground-penetrating radar: Implications for locating stored water on Mars. *Journal of Geophysical Research (Planets)*, 108:8036. (ref. pages 47 and 48)
- Demirdjian, D. and Darrell, T. (2002). Using multiple-hypothesis disparity maps and image velocity for 3-d motion estimation. *Int. J. Comput. Vision*, 47(1-3):219–228. (ref. page 37)
- dePascale, G., Pollard, W., and Williams, K. (2008). Geophysical mapping of ground ice using a combination of capacitive coupled resistivity and ground-penetrating radar, northwest territories, canada. *Journal of Geophysical Research*, 113:F02S90. (ref. pages 48 and 58)
- Dille, M., Grocholsky, B., and Singh, S. (2009). Outdoor downward-facing optical flow odometry with commodity sensors. In *Proceedings Field & Service Robotics (FSR '09)*, Cambridge, MA, USA. (ref. page 101)
- Diosi, A., Remazeilles, A., Segvic, S., and Chaumette, F. (2007). Outdoor visual path following experiments. *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 4265–4270. (ref. page 78)
- Dold, C. and Brenner, C. (2006). Registration of terrestrial laser scanning data using planar patches and image data. In *Proceedings of the ISPRS Commission V Symposium 'Image Engineering and Vision Metrology'*, volume XXXVI, Dresden, Germany. (ref. page 102)

- Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part i. *Robotics & Automation Magazine, IEEE*, 13(2):99–110. (ref. pages 13 and 63)
- Eade, E. and Drummond, T. (2007). Monocular slam as a graph of coalesced observations. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. (ref. page 8)
- Eade, E. and Drummond, T. (2008). Unified loop closing and recovery for real time monocular SLAM. In *19th British Conference on Machine Vision*. (ref. pages 63 and 99)
- Faugeras, O. and Luong, Q.-T. (2001). *The Geometry of Multiple Images*. The MIT Press, Cambridge, MA, USA. (ref. pages 24 and 25)
- Fischler, M. and Bolles, R. (1981). Random sample and consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395. (ref. pages 12 and 127)
- Fong, T., Allan, M., Bouyssounouse, X., Bualat, M., Deans, M., Edwards, L., Fluckiger, L., Keely, L., Lee, S., Lees, D., To, V., and Utz, H. (2008). Robotics site survey at haughton crater. In *Proceedings of the 9th International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Los Angeles, CA. (ref. pages 45, 48, 53, and 83)
- Fong, T., Deans, M., Lee, P., and Bualat, M. (2007). Simulated lunar robotic survey at terrestrial analog sites. In *Proceedings of the 38th Lunar and Planetary Science Conference*, League City, Texas. (ref. pages 53 and 83)
- Fortier, D. and Allard, M. (2004). Late holocene syngentic ice-wedge polygons development, bylot island, canadian arctic archipelago. *Canadian Journal of Earth Sciences*, 41:997–1012. (ref. pages 1, 52, and 59)
- Freese, M., Matsuzawa, T., Oishi, Y., Debenest, P., Takita, K., Fukushima, E. F., and Hirose, S. (2007). Robotics-assisted demining with gryphon. *Advanced Robotics*, 21(15):1763–1768. (ref. page 48)
- Furgale, P. and Barfoot, T. (2010a). Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics, special issue on “Visual mapping and navigation outdoors”*, 27(5):534–560. (ref. page 117)

- Furgale, P., Barfoot, T. D., and Ghafoor, N. (2009). Rover-based surface and subsurface modeling for planetary exploration. In Howard, A., Kelly, A., and Iagnemma, K., editors, *Proceedings of the International Conference on Field and Service Robotics (FSR)*, Springer Tracts in Advanced Robotics 62, pages 499–508, Cambridge, MA. (ref. page 116)
- Furgale, P. T. and Barfoot, T. D. (2010b). Stereo mapping and localization for long-range path following on rough terrain. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4410–4416, Anchorage, Alaska, USA. (ref. page 117)
- Furgale, P. T. and Barfoot, T. D. (2010c). Visual path following on a manifold in unstructured three-dimensional terrain. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 534–539, Anchorage, Alaska, USA. ICRA 2010 Kuka Service Robotics Best Paper Award. (ref. page 117)
- Furgale, P. T., Barfoot, T. D., Osinski, G. R., Williams, K., and Ghafoor, N. (2010). Field testing of an integrated surface/subsurface modeling technique for planetary exploration. *International Journal of Robotics Research*, special issue on “Field and Service Robotics”, 29(12):1529–1549. (ref. pages 14, 115, and 116)
- Goedemé, T., Nuttin, M., Tuytelaars, T., and Van Gool, L. (2007). Omnidirectional vision based topological navigation. *International Journal of Computer Vision*, 74(3):219–236. (ref. page 64)
- Goedemé, T., Tuytelaars, T., and Gool, L. V. (2005). Omnidirectional sparse visual path following with occlusion-robust feature tracking. In *6th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras, OMNIVIS05, in Conjunction with ICCV 2005*. (ref. page 65)
- Goedeme, T., Tuytelaars, T., Van Gool, L., Vanacker, G., and Nuttin, M. (2005). Feature based omnidirectional sparse visual path following. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005)*, pages 1806–1811. (ref. page 61)
- Grant, J. A., Schutz, A. E., and Campbell, B. A. (2003). Ground-penetrating radar as a tool for probing the shallow subsurface of mars. *Journal of Geophysical Research*, 108(E4). (ref. page 48)

- Grassia, F. S. (1998). Practical parameterization of rotations using the exponential map. *journal of graphics, gpu, and game tools*, 3(3):29–48. (ref. page 32)
- Guivant, J., Nebot, E., and Baiker, S. (2000). Autonomous navigation and map building using laser range sensors in outdoor applications. *Journal of Robotic Systems*, 17(10):565–583. (ref. page 102)
- Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision, Second Edition*. Cambridge University Press, Cambridge, UK. (ref. pages 5, 7, 13, 18, 24, 25, 32, and 37)
- Hinkel, K., Doolittle, J., Bockheim, J., Nelson, F., Paetzold, R., Kimble, J., and Travis, R. (2001). Detection of subsurface permafrost features with ground-penetrating radar, barrow, alaska. *Permafrost and Periglacial Processes*, 12:179–190. (ref. pages 52, 58, and 59)
- Horn, B. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642. (ref. page 68)
- Howard, A. (2008). Real-time stereo visual odometry for autonomous ground vehicles. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3946–3952. Ieee. (ref. pages 8, 9, and 10)
- Howard, A., Sukhatme, G. S., and Mataric, M. J. (2006). Multirobot simultaneous localization and mapping using manifold representations. *Proceedings of the IEEE*, 94(7):1360–1369. (ref. pages 62, 63, and 97)
- Hughes, P. C. (1986). *Spacecraft Attitude Dynamics*. John Wiley & Sons, New York. (ref. pages 20, 21, 22, and 24)
- Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory*. Academic Press, Inc, New York, NY, USA. (ref. page 17)
- Johnson, A. E., Goldberg, S. B., Cheng, Y., and Matthies, L. H. (2008). Robust and efficient stereo feature tracking for visual odometry. In *ICRA*, pages 39–46. (ref. page 3)

- Jones, E. S. and Soatto, S. (2011). Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*. (ref. page 5)
- Jones, S., Andresen, C., and Crowley, J. (1997). Appearance based process for visual navigation. In *Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on*, volume 2, pages 551–557 vol.2. (ref. page 64)
- Jun, K. K., Miura, J., and Shirai, Y. (2000). Autonomous visual navigation of a mobile robot using a human-guided experience. *Robotics and Autonomous Systems*, 40:2–3. (ref. page 65)
- Kaess, M., Ni, K., and Dellaert, F. (2009). Flow separation for fast and robust stereo odometry. In *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Kobe, Japan. (ref. page 93)
- Kim, S. S., Carnes, S., Haldemann, A., Ho Wah Ng, E., Ulmer, C., and Arcone, S. (2006). Miniature ground penetrating radar, crux gpr. *Aerospace Conference, 2006 IEEE*, pages 7 pp.–. (ref. page 48)
- Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan. (ref. page 8)
- Kline, M. (1972). *Mathematical Thought from Ancient to Modern Times*, volume 3. Oxford University Press Inc., New York, NY, USA. (ref. page 25)
- Konolige, K. and Agrawal, M. (2008). Frameslam: From bundle adjustment to real-time visual mapping. *Robotics, IEEE Transactions on*, 24(5):1066–1077. (ref. pages 10 and 97)
- Konolige, K., Agrawal, M., Blas, M. R., Bolles, R. C., Gerkey, B., Solà, J., and Sundaresan, A. (2009). Mapping, navigation, and learning for off-road traversal. *J. Field Robot.*, 26(1):88–113. (ref. page 66)
- Konolige, K., Agrawal, M., and Solà, J. (2007). Large scale visual odometry for rough terrain. In *Proceedings of the International Symposium on Research in Robotics (ISRR)*. (ref. pages 12, 13, 14, 66, 71, 97, and 108)

- Konolige, K. and Bowman, J. (2009). Towards lifelong visual maps. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. (ref. pages 98 and 99)
- Konolige, K., Bowman, J., Chen, J., Mihelich, P., Calonder, M., Lepetit, V., and Fua, P. (2010). View-based maps. *The International Journal of Robotics Research*, 29(8):941–957. (ref. pages 8, 9, and 14)
- Kretschmer, U., Abymar, T., Thies, M., and Frohlich, C. (2004). Traffice construction analysis by use of terrestrial laser scanning. In *Proceedings of the ISPRS working group VIII/2 'Laser-Scanners for Forest and Landscape Assessment'*, volume XXXVI, pages 232–236, Freiburg, Germany. (ref. page 102)
- Lachenbruch, A. (1962). Mechanics of thermal contraction cracks and ice-wedge polygons in permafrost. *Special paper to the Geological Society of America*, 70:69. (ref. page 1)
- Lacroix, S., Mallet, A., and Bonnafous, D. (2001). Autonomous rover navigation on unknown terrains functions and integration. In *Experimental Robotics VII*, volume 271/2001 of *Lecture Notes in Control and Information Sciences*, pages 501–510. Springer Berlin / Heidelberg. (ref. page 50)
- Lambert, A., Furgale, P., Barfoot, T. D., and Enright, J. (2011). Visual odometry aided by a sun sensor and inclinometer. In *Proceedings of the IEEE Aerospace Conference*, page #1279, Big Sky, MT. (ref. pages 14, 116, and 120)
- Lee, P., Braham, S., Boucher, M., Schutt, J., Glass, B., Gross, A., Hine, B., McKay, C., Hoffman, S., Jones, J., Berinstain, A., Comptois, J.-M., Hodgson, E., and Wilkinson, N. (2007). Haughton-mars project: 10 years of science operations and exploration systems development at a moon/mars analog site on devon island, high arctic. In *Proceedings of the 38th Lunar and Planetary Science Conference*, pages 2426–2427, League City, Texas. (ref. pages 46, 53, and 83)
- Lehmann, F. and Green, A. G. (1999). Semiautomated georadar data acquisition in three dimensions. *Geophysics*, 64(3):719–731. (ref. pages 48 and 52)
- Leuschen, C., Kanagaratnam, P., Yoshikawa, K., Arcone, S., and Gogineni, P. (2002). Field experiments of a surface-penetrating radar for mars. *Geoscience and Remote*

- Sensing Symposium, 2002. IGARSS '02. 2002 IEEE International*, 6:3579–3581 vol.6. (ref. page 48)
- Levy, J., Head, J., and Marchant, D. (2009). Thermal contraction crack polygons on mars: Classification, distribution, and climate implications from hirise observations. *Journal of Geophysical Research*, 114(E01007). (ref. page 1)
- Lovegrove, S. and Davison, A. (2010). Real-time spherical mosaicing using whole image alignment. In *Proceedings of the 11th European conference on computer vision conference on Computer vision: Part III*, pages 73–86. Springer-Verlag. (ref. page 6)
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110. (ref. pages 8, 65, and 128)
- Lunt, I. A. and Bridge, J. S. (2004). Evolution and deposits of a gravelly braid bar, sagavanirktoq river, alaska. *Sedimentology*, 51(3):415–432. (ref. pages 45 and 49)
- Mackay, J. (1999). Periglacial features developed on the exposed lake bottoms of seven lakes that drained rapidly after 1950, tuktoyaktuk peninsula area, western arctic coast, canada. *Permafrost and Periglacial Processes*, 10:39–63. (ref. page 59)
- Mackay, J. R. and Burn, C. R. (2002). The first 20 years (1978-1979 to 1998-1999) of ice wedge growth at the illisarvik experimental drained lake site, western arctic coast, canada. *Canadian Journal of Earth Sciences*, 39(1):95–111. (ref. page 1)
- Maimone, M., Cheng, Y., and Matthies, L. (2007). Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3):169–186. Maimone, Mark Cheng, Yang Matthies, Larry. (ref. pages 3, 6, 9, 12, and 14)
- Maimone, M., Johnson, A., Cheng, Y., Willson, R., and Matthies, L. (2006). Autonomous navigation results from the Mars Exploration Rover (MER) mission. In *Experimental Robotics IX*, volume 21 of *Springer Tracts in Advanced Robotics*, pages 3–13. Springer Berlin / Heidelberg. (ref. page 6)
- Maki, J. N., III, J. F. B., Herkenhoff, K. E., Squyres, S. W., Kiely, A., Klimesh, M., Schwochert, M., Litwin, T., Willson, R., Johnson, A., Maimone, M., Baumgartner, E., Collins, A., Wadsworth, M., Elliot, S. T., Dingizian, A., Brown, D., Hagerott, E. C., Scherr, L., Deen, R., Alexander, D., and Lorre, J. (2003). Mars exploration rover

- engineering cameras. *JOURNAL OF GEOPHYSICAL RESEARCH*, 108(E12). (ref. page 46)
- Malin, M. C., Bell, III, J. F., Cameron, J., Dietrich, W. E., Edgett, K. S., Hallet, B., Herkenhoff, K. E., Lemmon, M. T., Parker, T. J., Sullivan, R. J., Sumner, D. Y., Thomas, P. C., Wohl, E. E., Ravine, M. A., Caplinger, M. A., and Maki, J. N. (2005). The Mast Cameras and Mars Descent Imager (MARDI) for the 2009 Mars Science Laboratory. In Mackwell, S. and Stansbery, E., editors, *36th Annual Lunar and Planetary Science Conference*, volume 36 of *Lunar and Planetary Inst. Technical Report*, pages 1214–+. (ref. page 46)
- Malis, E. and Benhimane, S. (2005). A unified approach to visual tracking and servoing. *Robotics and Autonomous Systems*, 52(1):39–52. (ref. page 127)
- Mangold, N. A. (2005). High latitude patterned grounds on mars: Classification, distribution, and climatic control. *Icarus*, 174:336–359. (ref. page 1)
- Marshall, J., Barfoot, T., and Larsson, J. (2008). Autonomous underground tramping for center-articulated vehicles. *J. Field Robot.*, 25(6-7):400–421. (ref. pages 61, 63, 76, 77, 79, and 98)
- Masson, P., Carr, M. H., Costard, F., Greeley, R., Hauber, E., and Jaumann, R. (2001). Geomorphologic evidence for liquid water. *Space Science Reviews*, 96:333–364. (ref. page 1)
- Matsumoto, Y., Inaba, M., and Inoue, H. (1996). Visual navigation using view-sequenced route representation. *1996 IEEE International Conference on Robotics and Automation*, 1:83–88 vol.1. (ref. page 64)
- Matthies, L. and Shafer, S. (1987). Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*, 3(3). (ref. pages 6 and 13)
- Matthies, L. H. (1989). *Dynamic stereo vision*. PhD thesis, Carnegie-Mellon University. (ref. page 6)
- May, S., Fuchs, S., Malis, E., Nuchter, A., and Hertzberg, J. (2009). Three-dimensional mapping with time-of-flight cameras. *Journal of Field Robotics*, 26(11-12):934–965. (ref. page 103)

- McManus, C., Furgale, P. T., and Barfoot, T. D. (2011). Towards appearance-based methods for lidar sensors. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), to appear.*, pages 1930–1935, Shanghai, China. (ref. pages 104 and 117)
- Mei, C., Benhimane, S., Malis, E., and Rives, P. (2008). Efficient homography-based tracking and 3-d reconstruction for single-viewpoint sensors. *IEEE Transactions on Robotics*, 24(6):1352–1364. (ref. page 5)
- Mei, C., Sibley, G., Cummins, M., Newman, P., and Reid, I. (2009). A constant time efficient stereo SLAM system. In *BMVC*. (ref. pages 63 and 99)
- Mei, C., Sibley, G., Cummins, M., Newman, P., and Reid, I. (2010). Real: A system for large-scale mapping in constant-time using stereo. *International Journal of Computer Vision*, pages 1–17. Special issue of BMVC. (ref. pages 6, 8, 9, 10, 11, and 12)
- Mellon, M. T., Arvidson, R. E., Marlow, J. J., Phillips, R. J., and Asphaug, E. (2008). Periglacial landforms at the phoenix landing site and the northern plains of mars. *Journal of Geophysical Research*, 113(E00A23). (ref. page 1)
- Milford, M. and Wyeth, G. (2009). Persistent navigation and mapping using a biologically inspired slam system. *The International Journal of Robotics Research*, In Press. (ref. page 98)
- Milsom, J. (2003). *Field Geophysics, 3rd Edition*. Wiley. (ref. page 58)
- Moorman, B. and Michel, F. (2000). Glacial hydrological system characterization using ground-penetrating radar. *Hydrological Processes*, 14:2645–2667. (ref. page 58)
- Moorman, B. J., Robinson, S. D., and Burgess, M. M. (2003). Imaging periglacial conditions with ground-penetrating radar. *Permafrost and Periglacial Processes*, 14(4):319–329. (ref. pages 47 and 58)
- Moravec, H. (1980). *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. PhD thesis, Stanford University. Available as Stanford AIM-340, CS-80-813 and republished as a Carnegie Mellon University Robotics Institute Technical Report to increase availability. (ref. pages 6 and 13)

- Murray, R., Li, Z., and Sastry, S. (1994). *A mathematical introduction to robotic manipulation*. CRC. (ref. page 24)
- Neira, J., Tardos, J., Horn, J., and Schmidt, G. (1999). Fusing range and intensity images for mobile robot localization. *IEEE Transactions on Robotics and Automation*, 15(1). (ref. page 102)
- Newcombe, R. and Davison, A. (2010). Live dense reconstruction with a single moving camera. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1498–1505. IEEE. (ref. page 6)
- Newman, P., Leonard, J., Tardos, J. D., and Neira, J. (2002). Explore and return: experimental validation of real-time concurrent mapping and localization. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 2, pages 1802–1809 vol.2. (ref. page 63)
- Newman, P., Sibley, G., Smith, M., Cummins, M., Harrison, A., Mei, C., Posner, I., Shade, R., Schroeter, D., Murphy, L., Churchill, W., Cole, D., and Reid, I. (2009). Navigating, recognizing and describing urban spaces with vision and lasers. *Int. J. Rob. Res.*, 28(11-12):1406–1433. (ref. page 97)
- Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *Ieee Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770. (ref. page 5)
- Nistér, D. (2005). Preemptive ransac for live structure and motion estimation. *Machine Vision and Applications*, 16(5):321–329. (ref. pages 12 and 68)
- Nistér, D., Naroditsky, O., and Bergen, J. (2004). Visual odometry. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 01:652–659. (ref. page 55)
- Nistér, D., Naroditsky, O., and Bergen, J. (2006). Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3. (ref. pages 9, 10, and 13)
- Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, 2nd edition. (ref. pages 17 and 18)

- Ohno, T., Ohya, A., and Yuta, S. (1996). Autonomous navigation for mobile robots referring pre-recorded image sequence. In *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, volume 2, pages 672–679 vol.2. (ref. page 65)
- Osinski, G. R., Barfoot, T. D., Ghafoor, N., Izawa, M., Bangerjee, N., Jasiobedzki, P., Tripp, J., Richards, R., Auclair, S., Sapers, H., Thomson, L., and Flemming, R. (2010). Lidar and the mobile scene modeler (mSM) as scientific tools for planetary exploration. *Planetary and Space Science*, special issue on “Exploring other worlds by exploring our own: The role of terrestrial analogue studies in planetary exploration”, 58(4):691–700. (ref. page 51)
- Oster, G. and Nishijima, Y. (1963). Moire patterns. *Scientific American*, 208:54–63. (ref. page 112)
- Payá, L., Reinoso, O., Gil, A., Pedrero, J., and Ballesta, M. (2007). Appearance-based multi-robot following routes using incremental pca. *Knowledge-Based Intelligent Information and Engineering Systems*, pages 1170–1178. (ref. page 64)
- Pettinelli, E., Burghignoli, P., Pisani, A. R., Ticconi, F., Galli, A., Vannaroni, G., and Bella, F. (2007). Electromagnetic propagation of gpr signals in martian subsurface scenarios including material losses and scattering. *Geoscience and Remote Sensing, IEEE Transactions on*, 45(5):1271–1281. (ref. page 48)
- Picardi, G., Plaut, J. J., Biccari, D., Bombaci, O., Calabrese, D., Cartacci, M., Cicchetti, A., Clifford, S. M., Edenhofer, P., Farrell, W. M., Federico, C., Frigeri, A., Gurnett, D. A., Hagfors, T., Heggy, E., Herique, A., Huff, R. L., Ivanov, A. B., Johnson, W. T. K., Jordan, R. L., Kirchner, D. L., Kofman, W., Leuschen, C. J., Nielsen, E., Orosei, R., Pettinelli, E., Phillips, R. J., Plettemeier, D., Safaeinili, A., Seu, R., Stofan, E. R., Vannaroni, G., Watters, T. R., and Zampolini, E. (2005). Radar Soundings of the Subsurface of Mars. *Science*, 310(5756):1925–1928. (ref. page 47)
- Rial, F. I., Pereira, M., Lorenzo, H., and Arias, P. (2005). Acquisition and synchronism of GPR and GPS data: application on road evaluation. In Bruzzone, L., editor, *Image and Signal Processing for Remote Sensing XI*, volume 5982, pages 598219.1–598219.8. SPIE. (ref. page 48)

- Ross, N., Harris, C., Christansen, H., and Brabham, P. (2005). Ground penetrating radar investigations of open-system pingos, adventdalen, svalbard. *Norsk Geografisk Tidsskrift - Norwegian Journal of Geography*, 59:129–138. (ref. page 48)
- Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, volume 1, pages 430–443. (ref. page 8)
- Roth, G. and Wibowoo, E. (1997). An efficient volumetric method for building closed triangular meshes from 3-d image and point data. In *Proceedings of the conference on Graphics interface '97*, pages 173–180, Toronto, Ont., Canada, Canada. Canadian Information Processing Society. (ref. page 50)
- Royer, E., Lhuillier, M., Dhome, M., and Lavest, J.-M. (2007). Monocular vision for mobile robot localization and autonomous navigation. *Int. J. Comput. Vision*, 74(3):237–260. (ref. pages 61, 64, 65, 71, 72, and 78)
- Scott, W., Sellman, P., and Hunter, J. (1990). Geophysics in the study of permafrost, in geotechnical and environmental geophysics. In Ward, S., editor, *Geotechnical and Environmental Geophysics*, number 5 in Investigations in Geophysics. Society of Exploration Geophysicists. (ref. page 58)
- Se, S. and Jasiobedzki, P. (2008). Stereo-vision based 3d modeling and localization for unmanned vehicles. *International Journal of Intelligent Control and Systems*, 13(1):47–58. Special Issue on Field Robotics and Intelligent Systems. (ref. pages 6 and 49)
- Seu, R., Phillips, R. J., Biccari, D., Orosei, R., Masdea, A., Picardi, G., Safaeinili, A., Campbell, B. A., Plaut, J. J., Marinangeli, L., Smrekar, S. E., and Nunes, D. C. (2007). Sharad sounding radar on the mars reconnaissance orbiter. *J. Geophys. Res.*, 112(E5). (ref. page 47)
- Shuster, M. D. (1993). A survey of attitude representations. *Journal of the Astronautical Sciences*, 41(4):439–517. (ref. pages 19 and 34)
- Sibley, G., Matthies, L., and Sukhatme, G. (2008). A sliding window filter for incremental slam. In *Unifying Perspectives in Computational and Robot Vision*, volume 8 of *Lecture Notes in Electrical Engineering*, pages 103–112. Springer US. (ref. pages 13 and 71)

- Sibley, G., Mei, C., Reid, I., and Newman, P. (2009). Adaptive relative bundle adjustment. In *Robotics Science and Systems (RSS)*, Seattle, USA. (ref. pages 63, 97, and 99)
- Sibley, G., Mei, C., Reid, I., and Newman, P. (2010). Vast-scale outdoor navigation using adaptive relative bundle adjustment. *The International Journal of Robotics Research*, 29(8):958–980. (ref. page 14)
- Smith, P. H., Tamppari, L., Arvidson, R. E., Bass, D., Blaney, D., Boynton, W., Carswell, A., Catling, D., Clark, B., Duck, T., DeJong, E., Fisher, D., Goetz, W., Gunnlaugsson, P., Hecht, M., Hipkin, V., Hoffman, J., Hviid, S., Keller, H., Kounaves, S., Lange, C. F., Lemmon, M., Madsen, M., Malin, M., Markiewicz, J., Marshall, C., McKay, C., Mellon, M., Michelangeli, D., Ming, D., Morrise, R., Renno, N., Pike, W. T., Staufer, U., Stoker, C., Taylor, P., Whiteway, J., Young, S., and Zent, A. (2008). Introduction to special section on the phoenix mission: landing site characterization experiments, mission overviews, and expected science. *Journal of Geophysical Research*, 113(E00A18). (ref. page 1)
- Stenning, B. and Barfoot, T. D. (2011). Path planning on a network of paths. In *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT. (ref. pages 99 and 118)
- Strasdat, H., Montiel, J., and Davison, A. (2010a). Real-time monocular SLAM: Why filter? In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2657–2664, Anchorage, Alaska, USA. IEEE. (ref. page 14)
- Strasdat, H., Montiel, J., and Davison, A. (2010b). Scale drift-aware large scale monocular SLAM. In *Proceedings of Robotics: Science and Systems (RSS)*. (ref. page 5)
- Strelow, D. and Singh, S. (2004). Motion estimation from image and inertial measurements. *International Journal of Robotics Research*, 23(12):1157–1195. (ref. page 5)
- Stuelpnagel, J. (1964). On the parameterization of the three-dimensional rotation group. *SIAM Review*, 6(4):422–430. (ref. page 19)
- Tang, L. and Yuta, S. (2001). Vision based navigation for mobile robots in indoor environment by teaching and playing-back scheme. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 3072–3077 vol.3. (ref. page 65)

- Thrun, S., Burgard, W., and Fox, D. (2001). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press. (ref. page 13)
- Torr, P. H. S. (2002). Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *International Journal of Computer Vision*, 50(1):35–61. (ref. page 12)
- Torr, P. H. S. and Zisserman, A. (2000). Mlesac: a new robust estimator with application to estimating image geometry. *Comput. Vis. Image Underst.*, 78(1):138–156. (ref. page 12)
- Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (2000). Bundle adjustment – a modern synthesis. In Triggs, B., Zisserman, A., and Szeliski, R., editors, *Vision Algorithms '99*, volume LNCS 1883, pages 298–372. Springer-Verlag, Berlin Heidelberg. (ref. pages 13, 31, 32, and 33)
- Vago, J., Gardini, B., Kmínek, G., Baglioni, P., Gianfiglio, G., Santovincenzo, A., Bayón, S., and van Winnendaal, M. (2006). ExoMars - searching for life on the Red Planet. *ESA Bulletin*, 126:16–23. (ref. pages 45, 46, and 48)
- Šegvić, S., Remazeilles, A., Diosi, A., and Chaumette, F. (2009). A mapping and localization framework for scalable appearance-based navigation. *Comput. Vis. Image Underst.*, 113(2):172–187. (ref. pages 61, 64, 66, 72, and 78)
- Wettergreen, D., Dias, M., Shamah, B., Teza, J., Tompkins, P., Urmson, C., Wagner, M., and Whittaker, W. (2002). First experiment in sun-synchronous exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3501–3507, Washington, DC. (ref. pages 53 and 83)
- Wettergreen, D., Moreland, S., Skonieczny, K., Jonak, D., Kohanbash, D., and Teza, J. (2010). Design and field experimentation of a prototype lunar prospector. *International Journal of Robotics Research*, 29(1):1550 – 1564. (ref. page 101)
- Wettergreen, D., Tompkins, P., Urmson, C., Wagner, M., and Whittaker, W. (2005). Sun-synchronous robotic exploration: Technical description and field experimentation. *International Journal of Robotics Research*, 24(1):3–30. (ref. pages 53 and 83)

- Williams, B., Cummins, M., Neira, J., Newman, P., Reid, I., and Tardós, J. (2009). A comparison of loop closing techniques in monocular slam. *Robot. Auton. Syst.*, 57(12):1188–1197. (ref. page 99)
- Williams, K., Grant, J., , and Schutz, A. (2005). Ground-penetrating radar in mars analog terrains: Testing the strata instrument. In *Workshop on Radar Investigations of Planetary and Terrestrial Environments*, Houston, TX. (ref. page 48)
- Zhang, A. M. and Kleeman, L. (2009). Robust appearance based visual route following for navigation in large-scale outdoor environments. *Int. J. Rob. Res.*, 28(3):331–356. (ref. pages 64, 78, and 98)