STAR TRACKER AND LIDAR BASED NIGHT-TIME PLANETARY NAVIGATION

by

Ilija Jovanovic, B.Eng. Aerospace Engineering Ryerson University, 2013

A Thesis presented to Ryerson University

in partial fulfillment of the requirements for the degree of

Master of Applied Science in the program of Aerospace Engineering

Toronto, Ontario, Canada, 2015 © Ilija Jovanovic 2015

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

STAR TRACKER AND LIDAR BASED NIGHT-TIME PLANETARY NAVIGATION

Ilija Jovanovic, Master of Applied Science, Aerospace Engineering Ryerson University, Toronto, September 2015

Abstract

Planetary navigation on planets with no global positioning infrastructure is a challenge which has been overcome in the past by using day-light dependent sensors. This thesis develops a platform to test a star tracker and LIDAR based night-time navigation system. Testing of the system confirmed its feasibility despite not meeting target performance. Global position was acquired to within five kilometres using the star tracker and refined to within 23% of true motion for a 13 m path using the LIDAR.

Thesis Supervisor: Dr. John Enright Associate Professor, Ryerson University, Department of Aerospace Engineering

Acknowledgements

I would like to thank my supervisor, John Enright, for his guidance and support throughout this project. My lab peers for their continued supply entertainment and stress relief. Thomas Dzamba for the tremendous help, guidance, and friendship he has given me over the past two years. My Family and friends. Suzanna Vader for providing support and companionship which has been the greatest source of encouragement and motivation.

Contents

	Aut Abs	hors Dec tract	$\begin{array}{c} \text{claration} $	•	· ·	•	•	•	· ·	•	ii iii
	Ack	nowledg	ments								iv
	List	of Table	es								vii
	List	of Figur	res								ix
				-		-	-	-		-	
1	Intr	oductio	on								1
	1.1	Scope a	and Approach								1
	1.2	Motiva	ation and Goals								1
~	-										
2	Bac	kgroun									3
	2.1	Visual	Odometry	·		·	•	·		·	3
		2.1.1	Methods	·		•	•	•		•	4
		2.1.2	Existing Implementations	•		•	•	•		•	5
		2.1.3	Proposed Night-Time Implementation	•			•	•		•	6
	2.2	Absolu	te Position and Attitude Measurements	•				•			6
		2.2.1	Methods								7
		2.2.2	Proposed and Night-Time Implementations								7
	2.3	Mather	matical Framework								8
		2.3.1	Major Measurements								8
		2.3.2	Minor Measurements								10
_	_	_									
3	Rov	ver Dev	velopment								11
3	Rov 3.1	er Dev Constr	velopment								11 11
3	Rov 3.1 3.2	er Dev Constr Frame	velopment ruction		 		•				11 11 12
3	Rov 3.1 3.2 3.3	er Dev Constr Frame Inclino	velopment ruction		 	•		•	 		11 11 12 13
3	Rov 3.1 3.2 3.3 3.4	er Dev Constr Frame Inclino LIDAF	velopment cuction of Reference ometer R		 				 		11 11 12 13 14
3	Rov 3.1 3.2 3.3 3.4 3.5	er Dev Constr Frame Inclino LIDAF Star Tr	velopment ruction of Reference ometer		 			•	 		11 11 12 13 14 15
3	Rov 3.1 3.2 3.3 3.4 3.5 Visu	rer Dev Constr Frame Inclino LIDAF Star Tr	velopment ruction of Reference ometer R racker	•	 	•	• • •	•	· · ·		11 11 12 13 14 15
3	Rov 3.1 3.2 3.3 3.4 3.5 Vist	rer Dev Constr Frame Inclino LIDAF Star Tr ual Odd	velopment ruction of Reference ometer ometer R racker ometry nt Light Sensitivity		· · ·	•	•	•	 		 11 12 13 14 15 17
3	Rov 3.1 3.2 3.3 3.4 3.5 Visu 4.1	Yer Dev Constr Frame Inclino LIDAF Star Tr ual Odd Ambier Vigual	velopment ruction of Reference ometer ameter cometer ameter cometer ameter cometer ameter cometer ameter cometry nt Light Sensitivity Odemetry	· · ·	· · ·	· · · · · ·	• • • •	• • •	· · ·		 11 11 12 13 14 15 17 17 10
3	Rov 3.1 3.2 3.3 3.4 3.5 Viss 4.1 4.2	rer Dev Constr Frame Inclino LIDAF Star Tr ual Odd Ambien Visual	velopment suction of Reference ometer suction R Name R Name Name Name Succession	· · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · ·	· · · · · · ·	• • • •	· · · · · ·	· · ·	11 11 12 13 14 15 17 17 19 20
3	Rov 3.1 3.2 3.3 3.4 3.5 Vis 4.1 4.2	Yer Dev Constr Frame Inclino LIDAF Star Tr ual Odd Ambier Visual 4.2.1	velopment of Reference ometer ometer cometry nt Light Sensitivity Odometry Algorithms Filtering Mattian Thresholding	· · · · · · · · · · · · · · · · · · ·	· · ·	· · · · · · · · ·	· · · · · · · ·	· · · · · · · ·	· · · · · ·	· · ·	 11 11 12 13 14 15 17 17 19 20 21
3	Rov 3.1 3.2 3.3 3.4 3.5 Viso 4.1 4.2	Yer Dev Constr Frame Inclino LIDAF Star Tr ual Odd Ambien Visual 4.2.1 4.2.2	velopment ruction of Reference ometer amount	· · · · · · · · · · · · · · · · · · ·	· · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · ·	· · ·	 11 11 12 13 14 15 17 17 19 20 21
3	Rov 3.1 3.2 3.3 3.4 3.5 Vist 4.1 4.2	Yer Dev Constr Frame Inclino LIDAF Star Tr ual Odd Ambien Visual 4.2.1 4.2.2 4.2.3	velopment of Reference ometer acker racker ometry nt Light Sensitivity Odometry Algorithms Filtering Motion Thresholding acker	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · ·	· · · ·	 11 11 12 13 14 15 17 17 19 20 21 22 22
3	Rov 3.1 3.2 3.3 3.4 3.5 Vis 4.1 4.2	Yer Dev Constr Frame Inclino LIDAF Star Tr ual Odd Ambier Visual 4.2.1 4.2.2 4.2.3 4.2.4	velopment of Reference ometer ometer acker ometry nt Light Sensitivity Odometry Algorithms Filtering Motion Thresholding J-D Point Cloud Minimization	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	 11 11 12 13 14 15 17 17 19 20 21 22 23
3	Rov 3.1 3.2 3.3 3.4 3.5 Viso 4.1 4.2	Yer Dev Constr Frame Inclino LIDAF Star Tr Ual Odd Ambier Visual 4.2.1 4.2.2 4.2.3 4.2.4 4.2.5	velopment ruction of Reference ometer a a acker racker ometry nt Light Sensitivity Odometry Algorithms Filtering Motion Thresholding Jonetry Jonetry Correction Jonetry Correction </td <td>· · · · · · · · · · · · · · · · · · ·</td> <td>· · · · · · · · · · · ·</td> <td></td> <td></td> <td>· · · · · · · · · · · ·</td> <td>· · · · · · · · · · · ·</td> <td>· · · · · · · · · · · · · · · · · · ·</td> <td> 11 11 12 13 14 15 17 17 19 20 21 22 23 26 </td>	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · ·			· · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	 11 11 12 13 14 15 17 17 19 20 21 22 23 26
3	Rov 3.1 3.2 3.3 3.4 3.5 Vist 4.1 4.2	Yer Dev Constr Frame Inclino LIDAF Star Tr ual Odd Ambien Visual 4.2.1 4.2.2 4.2.3 4.2.4 4.2.5 4.2.6	velopment ruction of Reference ometer ameter	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	11 11 12 13 14 15 17 17 19 20 21 22 23 26 32
3	Rov 3.1 3.2 3.3 3.4 3.5 Vis 4.1 4.2	Yer Dev Constr Frame Inclino LIDAF Star Tr ual Odd Ambier Visual 4.2.1 4.2.2 4.2.3 4.2.4 4.2.5 4.2.6 Visual	velopment of Reference ometer ometer acker racker ometry nt Light Sensitivity Odometry Algorithms Filtering Motion Thresholding J-D Point Cloud Minimization J	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · ·				· ·	· · · · · · · · · · · · · · · · · · ·	 11 12 13 14 15 17 17 19 20 21 22 23 26 32 34
3	Rov 3.1 3.2 3.3 3.4 3.5 Viso 4.1 4.2	Yer Dev Constr Frame Inclino LIDAF Star Tr ual Odd Ambier Visual 4.2.1 4.2.2 4.2.3 4.2.4 4.2.5 4.2.6 Visual 4.3.1	velopment of Reference ometer a acker racker ometry nt Light Sensitivity Odometry Algorithms Filtering Motion Thresholding Joneter Correction Joneter Fourier Transform Method Discrete Fourier Transform Method Motion for Discrete Fourier Transform Method		· ·				· · · · · ·	· · · · ·	 11 11 12 13 14 15 17 17 19 20 21 22 23 26 32 34 34
3	Rov 3.1 3.2 3.3 3.4 3.5 Vist 4.1 4.2	Yer Dev Constr Frame Inclino LIDAF Star Tr Hal Odd Ambien Visual 4.2.1 4.2.2 4.2.3 4.2.4 4.2.5 4.2.6 Visual 4.3.1 4.3.2	velopment of Reference ometer ometer acker racker ometry nt Light Sensitivity Odometry Algorithms Filtering Motion Thresholding Joneter Correction Joneter Fourier Transform Method Discrete Fourier Transform Method Odometry Testing Other Transform Method Thresholding for Discrete Fourier Transform Method LIDAR with Inclinometer Validation		· ·	· · · · · · · · · · · · · · · · · · ·			· ·	· · · · · · · · ·	$\begin{array}{c} 11 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 17 \\ 17 \\ 19 \\ 20 \\ 21 \\ 22 \\ 23 \\ 26 \\ 32 \\ 34 \\ 34 \\ 35 \end{array}$

5 Geolocation	n
---------------	---

6	System Integration	43
	6.1 Minor Measurement Sensor Fusion	43
	6.2 Major Measurement Sensor Fusion	44
	6.3 Complete System Test	46
	6.3.1 Global Navigation Test	46
7	Conclusion and Future Work	49

References

39

51

List of Tables

2.1	Major measurement state variables, their descriptions, and the sensors(s) necessary to obtain them.	9
2.2	Minor measurement state variables, their descriptions, and the sensors(s) necessary to obtain them.	9
$3.1 \\ 3.2 \\ 3.3$	Inclinometer specifications	14 16 16
6.1	Integrated system test	48

viii

List of Figures

3.1	The rover which was assembled to serve as a test platform	11
3.2	Schematic of rover power distribution system. Black and red lines represent power lines and yellow lines represent communication	12
3.3	Frames of reference used for navigation and the information needed to convert between them.	13
3.4	Projection of gravity vector onto the inclinometer's yz -plane and the angle θ_X measured by the inclinometer.	14
3.5	An example of the LIDAR's brightness map return, 3.5a, and the correspond- ing point cloud with proportional axis, 3.5b	15
4.1	MESA Imaging SR4000 LIDAR camera and its coordinate frame [MESA Imaging]	17
4.2	Example of an image taken of a region exposed to direct sunlight: (4.2a) equal- ized histogram of brightness map for exposure with sunlight, (4.2b) equalized histogram of brightness map for exposure with clouds casting shade, and (4.2c)	
19	a photograph of the scene	18
4.3	odometry. In real-time operation, only one of the blue blocks is executed.	19
4.4	Effect of different types of LIDAR filtering on the z-component maps with blue indicating the closest part of the scene and yellow the furthest: (4.4a) unfiltered, (4.4b) low-pass filtered, (4.4c) outlier filtered, and (4.4d) photo- graph of the scene. Distortions in images 4.4a, 4.4b, and 4.4c are due to the	
	plotting of the data as the point cloud is properly rectified	21
4.5	Illustration of a truncation example.	26
4.0	the finite difference of the depth map, 4.6b. Nominal shift was positive ten	97
4.7	Depth map of image and model for a single row with zero-padding necessary for a FFT, (4.7a), and the numerical derivative of the depth map with zero-	21
	padding, $(4.7b)$.	28
4.8	Absolute values of transfer functions corresponding to and x direction trans- lation. Image (4.8a) is an example of a single row's LTI transfer function.	
	Image (4.8b) is the averaged LTI transfer function of all rows.	30
4.9	Demonstration of contrast enhancement from a raw image, (4.9a), and an im- age passed through a histogram equalization filter, (4.8b). Since both images	
	own light and appear brighter, such as the box in the lower region of the image	33
4.10	Scatter plot of row crest factor for two subsequent exposures with no motion	
	between them. The nominal shift for this case would be zero	35

4.11	Forward motion estimate from VO when using inclinometer corrections under acceleration. The image count is proportional to time. The curve consists of	
4.12	no rover motion at the beginning followed by constant velocity travel Preliminary results of VO testing for all four motions tracked by the camera.	36
	This data set consists of 1000 exposures and a 40 m driven distance	37
5.1	Simplified schematic of frame transformations	39
$\begin{array}{c} 6.1 \\ 6.2 \end{array}$	Diagram illustrating the spacing of LIDAR and star tracker readings Global position estimate of the rover using the star tracker, inclinometer, and LIDAR together. The performance of each of the three LIDAR methods is	44
	displayed	47
6.3	(Surface position estimates through sensor fusion using no initial position and a five kilometre measurement noise for the star tracker's global position	
	estimates. The start point of the path is at the origin.)	48

CHAPTER 1

Introduction

This project develops a platform to test a surface navigation system which does not rely on navigation infrastructure or daylight. On other planets in our solar system where there is no orbiting navigation infrastructure, such as GPS, systems which do not require such infrastructure are the only way surface vehicles can navigate in real time. Every such system which has been implemented relies on daylight, prohibiting night-time travel.

A group of navigation methods which do not rely on infrastructure are those which integrate local measurements. These systems do not provide absolute information by directly measuring a vehicles position or attitude but instead provide relative information by measuring changes in the state. Inevitably, noise in the measurements leads to drift in the state estimate. The addition of systems capable of directly measuring some or all of the state attenuates drift.

1.1 Scope and Approach

Three sensors were used to create a navigation system for this thesis. A star tracker and inclinometer are used to provide a coarse but absolute global position and a light detecting and ranging (LIDAR) camera performs visual odometry, providing precision relative measurements. The goal is to combine these sensors into a systems capable of night-time planetary navigation. To test the system, a rover platform was constructed for this thesis.

1.2 Motivation and Goals

When a rover's primary objective requires travelling, it is undesirable to stop every night due to an inability to navigate. There are several obstacles which need to be overcome for a rover to be able to travel at night, one of them being a system of sensors which can navigate at night.

Curiosity is the first non-solar powered Mars rover, having the power capability to continue moving at night. This demonstrates that one of the biggest design limitations to night-time travelling, power availability, can be overcome.

The second major limitation is that the lubricant used on all rover missions, Castrol Braycote 601 EF, becomes ineffective at Martian night time temperatures [Tillman 2009]. This can be overcome by switching to a solid lubricant which does not lose effectiveness at lower temperatures.

This thesis aims to address the third major limitation: developing a sensor system capable of navigating in the absence of light.

Chapter 2

Background

For this thesis, planetary navigation is broken down into two major constituents: relative and absolute navigation. The former is based on systems which integrate relative changes in position and attitude, referred to as state, to determine the total change in the state. Absolute navigation relies on systems which can directly measure all or some part of the state.

This chapter will provide a brief background of relative and absolute navigation systems relevant to this thesis. After this, the mathematical framework for interpreting sensor measurements into a navigation solution will be presented.

2.1 Visual Odometry

Visual odometry (VO) is a method of local motion determination from successive images of an optical system. Assuming a static environment, the successive images reveal the motion of the environment with respect to the optics. Inverting this result gives the optics's motion relative to the scene, typically referred to as egomotion. However, in order to determine motion of a camera, equivalent spatial scaling of the image is required.

The quantity and type of spatial information available depends on the optics system configuration. A typical configuration for obtaining this information is a stereo camera system which employs two cameras with known cross bore-sight separation. The disparity of objects between the perspectives of the two cameras, coupled with a camera model, provides a 3-D vector to a given object in the camera's frame of reference.

The focus in this thesis is on a LIDAR configuration which only uses a single camera. A MESA Imaging SR4000 LIDAR camera is used for this thesis. It works by actively emitting bursts of modulate light and measuring the returning phase of the modulated signal for each pixel. By knowing the speed of light, it is able to determine the distance to the sampled part of the scene for each pixel; this method is also referred to as time-of-flight (TOF). Passing

this information through a camera model provides position vectors, in the LIDAR's frame of reference, for each pixel.

Comparing VO to two more common types of odometry, wheel odometry and inertial odometry, VO has notable advantages. Wheel odometry uses encoders on the wheels to measure revolutions made and relates that to motion. Inertial odometry uses measurements of an inertial navigation system (INS), typically comprised of a accelerometer and gyroscope, and integrates the higher order measurements to arrive at displacement. The advantage of VO over wheel odometry is that no errors are incurred from the wheels sliding on soft or loose ground. In comparison to inertial odometry, VO can better limit solution drift at low speeds. It does this by relating the current image to images older than the immediately prior one, greatly reducing solution drift at low speeds and absolutely bounding error when stationary; an INS will continuously accumulate errors, even when stationary.

2.1.1 Methods

Visual odometry requires solving for the 3-D motion of a camera, referred to as the egomotion problem, between successive images for the purpose of navigation [Irani *et al.* 1994, Maimone *et al.* 2007]. Estimation of egomotion manifests itself in a wide range of 3-D visual problems including visual odometry, mapping, object registration, and camera stabilization. Although each of these applications have different end goals, they all share the need to determine the camera(s) motion by referencing the environment.

With this in mind, a simple breakdown of any visual odometry problem becomes:

- 1. Register regions and/or features between two images.
- 2. Calculate egomotion between registered data.
- 3. Integrate egomotion to determine total motion.

Step three is common for all visual odometry methods since it separates visual odometry from other egomotion applications.

Feature detection is a method of image registration, the identification of common regions or features between images. It works by repeatable identification of high frequency structures in an image. Features are referred to as *structures* since feature detectors look for patterns in high frequency changes, otherwise they would identify noise. Repeatable identification is the property which enables registration of features between images. Two common algorithms for feature detectors look for regions with high perpendicular gradients typical of object corners [Harris & Stephens 1988]. The gradients are based on the partial derivatives of the image across

the rows and columns. Harris features can also identify edges if they are prominent or there is a lack of corner features. The SURF algorithm uses the properties of the determinant of the Hessian matrix to detect features. In order to improve on speed, the second order Gaussian derivative filter is approximated by a box filter [Bay *et al.* 2008]. This method detects corners and blobs but not edges.

In a stereo camera system, visual features common to both cameras are mapped to feature vectors and used for egomotion calculation. This approach can be extended to a LIDAR system, making use of the algorithms developed for stereo systems. From Scaramuzza's and Fraundorfer's review of VO [Scaramuzza & Fraundorfer 2011], the most common approach for determining the egomotion between two feature sets is optimizing a weighted leastsquares where the weighting is proportional to the inverse length of the feature vectors. One improvement to the least-squares formulation, proposed by Nister [Nister et al. 2004], is to first reject outlier vectors using a random sample consensus (RANSAC) filter [Fischler & Bolles 1981]. Milella and Siegwart [Milella & Siegwart 2006] propose using an iterative closest point (ICP) [Besl & McKay 1992] algorithm to further refine the motion solution obtained from the least-squares optimization. ICP differs from other formulation by not requiring the features to be registered and instead assumes that the feature sets contain enough common features. At each iteration, the ICP algorithm registers feature vectors to their closest neighbours and re-optimizes the least-squares problem to determine egomotion [Besl & McKay 1992]. This process is repeated until alignment of the vector sets stops improving. It is important to note that, although an ICP implementation is useful for aligning feature vectors that are effectively randomly scattered, alignment of consistently spaced vectors would not benefit registration being repeated.

This thesis also spends time looking at a non-feature based registration methods which makes use of the Fourier transform's properties. The theory for this was first developed by De Castro and Morandi [De Castro & Morandi 1987] who demonstrated the ability of a Fourier transform to register two images which have been translated and rotated within a plane. This theory was applied to VO by Goecke et al. [Goecke *et al.* 2007] who used a Fourier-Mellin transform to recover rotation, translation, and scale parameters between two images. This method does not require any iterative solving and is global, meaning it uses all of the pixels in both images to arrive at a solution unlike feature based methods which only use parts of an image. The limitation their method is its inability to deal with occlusion and parallax effects.

2.1.2 Existing Implementations

In recent missions, surface navigation on other planets has been performed using a stereo VO system during the day time. This is due to two reasons: the first being that solar powered rovers, such as *Opportunity*, have no power at night [Biesiadecki *et al.* 2005]; the second reason is that the passive cameras of the stereo VO system require sunlight. As an example, the Mars *Curiosity* rover performs visual odometry and obstacle avoidance by combining

information from three stereo camera systems. At night, wheel encoders and INS systems can perform odometry but cannot substitute the camera's obstacle avoidance, such as that implemented on *Curiosity* [Grotzinger *et al.* 2012]. Being unable to avoid obstacles at night renders wheel odometry and INS equally unusable.

The Mars Exploration Rovers (MER) uses a similar system, but with fewer stereo pairs, and have a target position accuracy of no more than 10% error for a 100 m drive [Maimone *et al.* 2007]. As a method for long distance navigation, the MERs VO system is insufficient. The system can be supplement with wheel encoders and INS systems to increase accuracy and mitigate drift but without absolute measurements, this will also accumulate error.

2.1.3 Proposed Night-Time Implementation

The ability to navigate at night could be beneficial to planetary rover missions which need to travel to a destination. Every moment a rover spends on the surface it risks being disabled by environmental factors or random equipment failures. Therefore, the faster a rover can, the higher the chances of mission success will be. This thesis uses a LIDAR camera to overcome the need for illuminated surroundings of a passive camera system. With active lighting, the LIDAR camera can perform VO and obstacle avoidance at night, enabling night-time navigation, and by extension, night-time travel.

Use of an active lighting system does come with some drawbacks:

- Consumes more power than a passive system without illumination.
- Range is limited by light emitter brightness.
- Performance degrades during the day [Piatti & Rinaudo 2012].

2.2 Absolute Position and Attitude Measurements

Absolute state measuring refers to a sensor's ability to directly measure a vehicles position or attitude. The addition of any such measurement can absolutely limit drift in at least one dimension of the state estimate. This discussion will focus on methods which reference against celestial objects, omitting potential systems which rely on Earth-based infrastructure. To date, no mission to other planets was been equipped with anything other than local measurement-based navigation systems.

2.2.1 Methods

Celestial objects are fixed in an inertial frame, making them useful to reference against. Knowing the location of these object in the body frame of a spacecraft, along with inertial position and orientation of a planet, can provide state information relative to the planet. Two sensors which provide absolute measurements by referencing celestial objects are sun sensors and star trackers.

A Sun sensor measures the vector pointing towards the sun. From predictions of the inertial position of the Sun and an estimate of the sensor's position with respect to the Sun, the Sun vector can be related to the inertial frame. This can then be used to obtain a partial attitude solution. Similarly, a star tracker is a camera which detects and identifies stars in its field of view. By identifying at least three stars, sufficient information is available for a complete attitude solution.

2.2.2 Proposed and Night-Time Implementations

A proposed planetary navigation system adds a sun sensor and inclinometer to a stereo camera system, such as those described above. Knowing the sidereal orientation of the planet and an estimate of the vehicle's position on the surface, the sun sensor and inclinometer can determine the vehicle's absolute heading. With multiple measurements, this system can also provide global position. A sun sensor measures the vector pointing to the sun while the inclinometer corrects the sun sensor's orientation to the local geoid. The addition of this information mitigates drift in the position solution introduced by integration [Enright *et al.* 2009].

Similar to how sun sensors can provide an absolute measurement of one dimension of attitude (heading) during the day, a star tracker, given an inclinometer correction to the local geoid, can provide a position and heading solution at night [Enright *et al.* 2012]. Star trackers provide the following benefits for geolocation:

- Star trackers provide absolute position and heading as opposed to just heading from a single measurement.
- A star tracker has higher accuracy, giving a better heading solution [Wertz et al. 2011].
- The star tracker system does needs fewer measurements to determine position.

The Sinclair Interplanetary ST-16 star tracker is used for this project. Based on it pointing accuracy and Earth radius, it is capable of determining global position to approximately 200 m on Earth with adequate inclinometer accuracy (typical inclinometer accuracy is not a limiting factor [Jewell Instruments LLC]). While this is an accurate solution in the global

context, more refined motion estimates need supplementary measurements. Since an absolute position solution is available when using a star tracker, fusing integrated local measurements absolutely limits their drift, making long distance navigation feasible. For this reason, a LIDAR based VO system is used for high precision positioning.

2.3 Mathematical Framework

Throughout this thesis, variables with an under-bar will denote arrays of scalars while boldface variable will represent physical vectors or rotation matrices. Variables that are both boldface and have an under-bar will denote arrays of physical vectors or matrices. Where appropriate context is given, lower-case indices will index dimensions where the corresponding upper-case letter is the size of the dimension. Lastly, again with appropriate context, uppercase subscripts X, Y, and Z will denote the x, y, or z components of the un-subscripted vector.

The frames of references used throughout this thesis provide the grounding for the mathematical framework. Because the navigation system was developed for testing on Earth, the frames of reference are with respect to Earth as well. For this framework, the topocentric has its origin at the rover and is defined as North-East-Up: the y-axis is pointed north, the x-axis is pointed East, and the z-axis is pointed up. The measurements are broken down into two data tiers: major measurements and minor measurements. Major measurements are concerned with absolute position and orientation on the planet while minor measurements are relative to a prior state. Major measurements are indexed by j and minor measurements are indexed by k such that there a K minor measurements between successive major measurements.

The major measurements use geolocation provided by the star tracker to determine absolute position and treats VO motion provided by the LIDAR as velocity. Table 2.1 shows state information for the major measurements.

For the Minor measurements, table 2.2 shows the state variables.

2.3.1 Major Measurements

For the major measurements, the star tracker and inclinometer directly directly provide absolute position and attitude with the exception of altitude. Altitude is determined by integrating the upwards velocities, as calculated by the LIDAR. Equation 2.1 is a mathematical representation of this.

$$z_p = \mathbf{C}_{j(j-1)} \Delta z_{VO,j-1} + \mathbf{C}_{j(j-2)} \Delta z_{VO,j-2} + \dots + \mathbf{C}_{j1} \Delta z_{VO,1}$$
(2.1)

	Description	Sensor(s)
Variable		
	Longitude on the Earth, measured in radians.	Star tracker, Inclinometer
λ_j		~
ϕ_j	Latitude on the Earth, measured in radians.	Star tracker, Inclinometer
	Altitude on the Earth, measured in meters.	LIDAR, Inclinometer
z_j	This only relatively known.	
~	Body pitch relative to the surface in radians.	Inclinometer
$\theta_{incl,X,j}$	This is the inclinometer model corrected pitch.	
~	Body roll relative to the surface in radians.	Inclinometer
$\theta_{incl,Y,j}$	This is the inclinometer model corrected roll.	
	Heading from north, in radians.	Star tracker, Inclinometer
α_j		
$\Delta x_{VO,j}$	Longitudinal velocity, in radians per second.	LIDAR, Inclinometer
$\Delta y_{VO,j}$	Latitudinal velocity, in radians per second.	LIDAR, Inclinometer
$\Delta z_{VO,j}$	Upwards velocity, in meters per second.	LIDAR, Inclinometer

Table 2.1: Major measurement state variables, their descriptions, and the sensors(s) necessary to obtain them.

Table 2.2: Minor measurement state variables, their descriptions, and the sensors(s) necessary to obtain them.

	Description	Sensor(s)
Variable		
	x-displacement relative to initial body frame,	LIDAR, Inclinometer
$x_{VO,k}$	in meters.	
	y-displacement relative to initial body frame,	LIDAR, Inclinometer
$y_{VO,k}$	in meters.	
	z-displacement relative to initial body frame,	LIDAR, Inclinometer
$z_{VO,k}$	in meters.	
~	Body pitch relative to the surface in radians.	Inclinometer
$\theta_{incl,X,k}$	This is the inclinometer model corrected pitch.	
~	Body roll relative to the surface in radians.	Inclinometer
$\theta_{incl,Y,k}$	This is the inclinometer model corrected roll.	
	Heading from north, in radians.	LIDAR, Star tracker, Incli-
α_k		nometer

where $C_{j(j-1)}$ is the rotation matrix from measurement j-1 to j.

Based on Eq. 2.1, at j = 0, the altitude would always be 0 m. For this reason, the altitude is said to be relative to the starting altitude.

The velocities for the major measurement j are the net displacements of the K minor mea-

surements between j and j - 1. In equation form:

$$\begin{bmatrix} \Delta x_{VO,j} \\ \Delta y_{VO,j} \\ \Delta z_{VO,j} \end{bmatrix} = \frac{1}{\Delta t_{j(j-1)}} \begin{bmatrix} x_{VO,K} \\ y_{VO,K} \\ z_{VO,K} \end{bmatrix}$$
(2.2)

where $\Delta t_{j(j-1)}$ is the time between j and j-1.

2.3.2 Minor Measurements

The minor measurements are taken in-between major major measurements with K minor measurements for each major measurement. Position in the minor measurements is obtained using visual odometry and is relative to the previous major measurement's navigation frame. In equation form, the change in position from the previous major measurement to the current minor one is:

$$\mathbf{q}_{T,k} = \mathbf{C}_{k(k-1)}\mathbf{q}_{T,k-1} + \mathbf{C}_{k(k-1)}\mathbf{C}_{(k-1)(k-2)}\mathbf{q}_{T,k-2} + \dots + \mathbf{C}_{k1}\mathbf{q}_{T,1}$$
(2.3)

where \mathbf{q}_T is the translation vector.

Part of the rotation between successive frame is a relative measurement provided by the LIDAR. As a result, a rotation from the current measurement k to the reference major measurement j is not available. The rotation between minor measurements is then constructed using Eq. 2.4 which combines relative and absolute sensor measurements.

$$\mathbf{C}_{k(k-1)} = \mathbf{R}_Z \left(\alpha_k - \alpha_{k-1} \right) \mathbf{R}_Y \left(\widetilde{\theta}_{incl,Y,k} - \widetilde{\theta}_{incl,Y,k-1} \right) \mathbf{R}_X \left(\widetilde{\theta}_{incl,X,k} - \widetilde{\theta}_{incl,X,k-1} \right)$$
(2.4)

Chapter 3

Rover Development

For this project a rover was assembled to act as a test bed, seen in Fig. 3.1. The rover has a computer, a Tuff Tilt Digital inclinometer, an ST-16 star tracker, and an SR4000 LIDAR camera.



Figure 3.1: The rover which was assembled to serve as a test platform.

3.1 Construction

The rover was constructed using a commercial-off-the-shelf chassis. The motors and steering servo were separately specified based on the chassis mounting interfaces for such components.

The primary method of control for the rover is a hobby radio controller and receiver. The rover also has an on-board computer, router, and instrument box called the "Green Box". The instrument box contains a port server through which the star tracker and inclinometer communicate and powers the two sensors.

The power system is based on a 12 volt battery supply which feeds to all systems and a 12 volt step-up battery for the Green Box. Figure 3.2 gives a schematic of the power distribution system and communication chains.



Figure 3.2: Schematic of rover power distribution system. Black and red lines represent power lines and yellow lines represent communication.

3.2 Frame of Reference

The development of the rover and its navigation system requires a consistent set of reference frames. These come from the rover's sensors and the planetary body. Figure 3.3 shows the frames of reference used and the information needed to convert between linked ones.



Figure 3.3: Frames of reference used for navigation and the information needed to convert between them.

The mounting angles of the LIDAR and inclinometer in Fig. 3.3 are the nominal rotations between the body frame and the sensor frame. The star tracker's mounting angle is calibrated for and is discussed later in the thesis.

3.3 Inclinometer

The Inclinometer used for this thesis is the Jewell Instruments Tuff Tilt Digital inclinometer. It works by measuring the angle of two pendulums which are each only free to rotate about one axis. The pendulums are align perpendicular to one another such that one measures to x-rotation of the rover and the other measures y-rotation.

Mathematically, the angle of the pendulums is the projection of the gravity vector onto two perpendicular planes, the inclinometer's xz and yz planes. The angles between the projections and z-axis are the returns of the inclinometer. An example of this is shown in Fig. 3.4.

The measurements taken by the inclinometer, $\theta_{incl,X}$ and $\theta_{incl,Y}$ cannot directly be used to construct a rotation matrix because there is coupling of the two values. An inclinometer, just as the one given by Enright et al. [Enright *et al.* 2012] is needed to convert the measurements into a rotation matrix. Assuming a nominal inclinometer geometry, the rotation matrix is:

$$\mathbf{C}_{CN} = \mathbf{R}_X \left(\widetilde{\theta}_{incl,X} \right) \mathbf{R}_y \left(\widetilde{\theta}_{incl,y} \right) \mathbf{R}_Z \left(\frac{\pi}{2} \right)$$
(3.1)



Figure 3.4: Projection of gravity vector onto the inclinometer's yz-plane and the angle θ_X measured by the inclinometer.

where $\tilde{\theta}_{incl,X}$ and $\tilde{\theta}_{incl,Y}$ are given by:

$$\theta_{incl,Y} = \operatorname{asin}\left(\sin\left(-\theta_{incl,Y}\right)\cos\left(\theta_{incl,X}\right)\right)$$
$$\widetilde{\theta}_{incl,X} = -\operatorname{asin}\left(\frac{\cos\left(-\theta_{incl,Y}\right)\sin\left(\theta_{incl,X}\right)}{\widetilde{\theta}_{incl,Y}}\right)$$
(3.2)

Some useful specifications of the inclinometer are given in table 3.1.

 Table 3.1: Inclinometer specifications.

Angular Range	$\pm 50 \text{ degrees}$
Resolution	0.0001 degree
Static Repeatability	0.0003 degree (1 arc second)
Data Rate	10 samples/s

3.4 LIDAR

The LIDAR camera used is a MESA Imaging SR4000 LIDAR camera. It returns a rectangularly indexed M by N 3-D point cloud representation of the imaged environment, represented

by $\underline{\mathbf{I}}^k$, and brightness map, \underline{A}^k , where the superscript k indicates the index of the image. A single point's vector from the point cloud is given by \mathbf{I}_{mn}^k , and A_{mn}^k for a single pixel's brightness. In the context of visual odometry, the prior data, k - 1, will be referred to as the model, and the current will be referred to as the image. Figure 3.5 gives an example of a point cloud return with its corresponding brightness map.



Figure 3.5: An example of the LIDAR's brightness map return, 3.5a, and the corresponding point cloud with proportional axis, 3.5b

To help resolve y-rotation from x-translation in the LIDAR's frame, the camera is pointed down (x-rotation) by ten degrees. This introduces more parallax between the upper and lower parts of the image when looking a flat terrain. Rotating the Camera down also helps limit its field of view to the unambiguous range. The LIDAR mounting angle is assumed to be nominal and is not calibrated for.

Some useful specifications of the LIDAR are given in table 3.2.

3.5 Star Tracker

The star tracker used is an Sinclair interplanetary ST-16 star tracker. It identifies observed star patterns to those in a catalogue and uses that information to return its orientation

Accuracy	1 cm
Wavelength	850 nm
Unambiguous Range	5 m
Data Rate	30 frames/s

 Table 3.2:
 LIDAR camera specifications.

relative to the Earth-centred inertial (ECI) frame. Some useful specifications of the star tracker are given in table 3.3.

Accuracy (cross bore-sight)	7 arc seconds
Accuracy (bore-sight)	56 arc seconds
Data Rate	0.5 frames/s

 Table 3.3:
 Star tracker specifications.

Chapter 4 Visual Odometry

Visual odometry was done using the MESA Imaging SR4000 LIDAR camera. This camera returns a three dimensional point cloud representation of the photographed environment and an brightness map. Figure 4.1 provides a better view of the camera's coordinate frame in which the point cloud is given. The LIDAR uses time-of-flight (TOF) to determine the length of each pixel's vector with the vector direction being pre-defined by the optics' geometry [MESA Imaging]. Invalid returns are typically caused by the either the scene being too far away to be effectively illuminated or light rays returning to the camera after multiple reflections [MESA Imaging].

Absolute coordinates



Figure 4.1: MESA Imaging SR4000 LIDAR camera and its coordinate frame [MESA Imaging].

For the visual odometry section of this thesis, the LIDAR camera's frame of reference will be used.

4.1 Ambient Light Sensitivity

Time-of-flight LIDARs rely on being able to detect to the phase of the reflected light emitted by the camera. However, the reflections become less prominent with increasing ambient brightness, eventually causing points furthest away in the image to return invalid values. The camera is incapable of resolving parts of the image exposed to direct sunlight, as see in Fig. 4.2. The saturated yellow regions in Fig. 4.2a have a straight border across the middle of the images consistent with the building's shadow seen in Fig. 4.2c, suggesting that the saturated regions correspond to parts of the scene exposed to direct sunlight.

While the shadow borders may offer good features in the brightness map, the distance to the saturated regions in Fig. 4.2a cannot be calculated, making it difficult to measure motion relative to those features. The result is that this LIDAR camera is not usable in direct sunlight.





Figure 4.2: Example of an image taken of a region exposed to direct sunlight: (4.2a) equalized histogram of brightness map for exposure with sunlight, (4.2b) equalized histogram of brightness map for exposure with clouds casting shade, and (4.2c) a photograph of the scene.

4.2 Visual Odometry Algorithms

Figure 4.3 is a graphical representation of the implemented process for getting a solution from successive exposures. The three parallel algorithms, shown in blue, each serve the function of estimating motion between the image (latter exposure) and the model (former exposure) but use different methods to do so. Each exposure consists of x, y, z, and brightness maps. The following sections will discuss the functions and justification of each step in Fig. 4.3.



Figure 4.3: Flow chart depicting the process for acquiring a motion solution through visual odometry. In real-time operation, only one of the blue blocks is executed.

The VO algorithms looked at are a point cloud minimization, and Fourier transform based method, and a feature based method. They focus on determining forward motion of the rover (x-translation), turning of the body due to steering inputs (y-rotation), side slipping of the wheels (z-translation), and vertical motion (y-translation). De-coupling of x-translation and y-rotation is the more challenging task as they have very similar effects on the captured image. Changes in pitch (z-rotation) and roll (x-rotation) are calculated using the incli-

nometer and not the VO system. The reasoning is that the inclinometer provides a solution that is orders of magnitude more accurate. Thus, when fusing VO estimates with those of the inclinometer, the appropriate measurement noise makes the VO contribution negligible.

4.2.1 Filtering

The raw image returned by the LIDAR is too noisy to be reliably used in motion estimation. After each image is taken, the process starts by filtering the image to improve VO reliability. Two candidate methods, a simple low-pass filter and an outlier detection and replacement filter, are considered. The effects of these two filters are shown in Fig. 4.4.

The low-pass filter is a three by three Gaussian filter. It has the advantage of being more than ten times faster than the alternative. However, where the outlier filter completely removes erroneous information from the solution, the low-pass simply attenuates it.

The outlier filter was developed for this thesis and uses three types of information to flag outliers. Once an outlier is flagged, that point is removed and replaced by the average of the eight neighbouring points which were not flagged. The methods and order for the flagging process are:

- 1. Pixels whose brightness is beyond an accepted range are flagged. Erroneous brightnesses have shown to be strongly associated with noise pixels.
- 2. Pixels that have solutions very close to the detector are flagged. Not only will the rover be actively avoiding obstacles, an obstacle would require a particular geometry to come within 5 cm of the camera while not colliding with any part of the rover. Therefore, points closer than this are likely invalid.
- 3. The image is convolved with a point filter. High brightness returns from this filtering process indicate the point is a local outlier and is flagged.

The greatest advantage of the outlier filter is its ability to re-construct missing parts of a scene. In Fig. 4.4a, the closest part of the scene is the dark blue region in the middle, however, from Fig. 4.4d, it is obvious that the centre of the image is actually furthest away. This is due to the centre being too far to be sufficiently illuminated by the LIDAR. The yellow specks in the middle of Fig. 4.4a are the points which had valid returns. Knowing this, the low-pass filter (fig. 4.4b) attenuates the valid specks while the outlier filter uses their information to re-construct part of that scene, as seen in Fig. 4.4c. In practice, reconstruction of parts of the scene that are out of range of the LIDAR is undesirable as the validity of those results is questionable. However, for the purpose of demonstrating the ability of the outlier filter to perform this function, Fig. 4.4c uses such a scenario.

The low-pass was selected for its computational speed. In practice, the low-pass filter accomplishes the goal of filtering LIDAR images but at a lower computational expense.





Figure 4.4: Effect of different types of LIDAR filtering on the z-component maps with blue indicating the closest part of the scene and yellow the furthest: (4.4a) unfiltered, (4.4b) low-pass filtered, (4.4c) outlier filtered, and (4.4d) photograph of the scene. Distortions in images 4.4a, 4.4b, and 4.4c are due to the plotting of the data as the point cloud is properly rectified.

4.2.2 Motion Thresholding

Motion thresholding simply refers to the decision block in Fig. 4.3. Each motion estimator has some tolerance as to how much the scene can change between images and still be able to arrive at a solution. The selected bounds are 1 cm and ten 15 cm. The lower bound is selected to avoid small estimates with a large noise fraction and the upper bound is the empirically determined tolerance of motion for the VO algorithms. Equation 4.1 mathematically shows the criteria for acceptable motion.

$$0.01 \text{ m} < x_{VO,k} < 0.15 \text{ m} \tag{4.1}$$

If the criteria of Eq. 4.1 are not met, then the current image is discarded and the model is kept. This is repeated p time to a maximum of four more times:

$$\underline{\mathbf{I}}^{k-1} = \underline{\mathbf{I}}^{k-1-p}, \quad p = 1, 2, 3, 4, 5 \tag{4.2}$$

The maximum for p was chosen based on the VO algorithm's motion tolerance. Given the rover's speed and the frame rate of the camera, the change between five images is expected to be just under the 15 cm upper bound such that the sixth image will over. As a result, using up to a five image difference to calculate motion allows for potential full use of the upper motion bound.

If the maximum image difference was reached without an acceptable solution, the motion estimate across that period of time is defaulted to no motion and the error that introduces is attributed to poor algorithm robustness.

4.2.3 Inclinometer Correction

As mentioned previously, the inclinometer can provide a more accurate solution for x-rotation and z-rotation in comparison to VO. Images captured by the LIDAR camera during minor measurements are therefore corrected using the inclinometer measurements prior to processing. This section will describe the inclinometer correction process in detail.

The goal of minor measurements is to produce a relative translation in prior major measurements navigation frame. Based on Eq. 2.3, the rotation matrix from reference navigation frame to the LIDAR frame is needed to achieve this goal. However, from Eq. 2.4, this rotation is indeterminate using only the two inclinometer measurements. VO could provide rover heading, α_k , and give a complete rotation but the goal is to use the inclinometer prior to performing VO on the current image. This means that, at the time the inclinometer correction is applied, at most the VO model's heading, α_{k-1} , is known but the heading of the image, α_k , is not.

Since the goal of the inclinometer corrections is to improve LIDAR accuracy and not give motion in the reference navigation, the effect of heading changes is neglected:

$$\mathbf{R}_{Z} \left(\alpha_{k} - \alpha_{k-1} \right) \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
(4.3)

In addition to the assumption of Eq. 4.3, the inclinometer model is not used for the correction and the rotation used is instead:

$$\mathbf{C}_{k(k-1)} = \mathbf{R}_X \left(\theta_{incl,X,k-1} - \theta_{incl,X,k} \right) \mathbf{R}_Y \left(\theta_{incl,Y,k-1} - \theta_{incl,Y,k} \right)$$
(4.4)

This assumption helps simplify the implementation. The error introduced by not using the inclinometer model is negligible in comparison the LIDAR accuracy, the sensor being corrected using this rotation.

These assumption are valid since the goal of the inclinometer correction is to improve the performance of the VO algorithms by improving alignment LIDAR images. The residual alignment errors are small in comparison to the starting error.

The implementation of the correction starts by translating the image maps in the vertical direction to correct for the *y*-rotation in Eq. 4.4; this aligns the rows of the image and model. For this part of the discussion, $\tilde{\mathbf{I}}^k$ will denote the uncorrected point cloud.

$$\tilde{\mathbf{I}}_{mn}^{k} = \tilde{\mathbf{I}}_{(m-o)n}^{k}$$

$$o = \frac{\theta_{incl,Y,k-1} - \theta_{incl,Y,k}}{\mu_{N}}$$
(4.5)

where μ_N is the apparent angular height of a row. The maps are then truncated to only include paired rows.

$$m = o + 1, o + 2, \dots, M \tag{4.6}$$

So far, the rows of the image and model are registered but the geometric rotation between their point clouds is still present. To align the point clouds, a 13-rotation in the camera's frame is applied to the point clouds.

$$\mathbf{I}^{k} = \mathbf{R}_{Z} \left(\theta_{incl,X,k-1} - \theta_{incl,X,k} \right) \mathbf{R}_{X} \left(\theta_{incl,Y,k-1} - \theta_{incl,Y,k} \right) \widetilde{\mathbf{I}}^{\kappa}$$
(4.7)

Lastly, the image maps are rotated about the bore-sight (z-rotation) and cropped. The centre of rotation of the maps is offset to correct for the ten degree mounting offset of the LIDAR relative to the inclinometer.

4.2.4 3-D Point Cloud Minimization

Minimization of two 3-D point clouds is the minimization of the least-squares errors between registered point pairs. Registration of 3-D point clouds is commonly solved using global iterative methods such as iterative closest point (ICP) [Besl & McKay 1992] or feature based methods such as fast point feature histogram (FPFH) [Rusu *et al.* 2009]. In this study, registration of point clouds of two subsequent LIDAR exposures was done by exploiting the consistently space point cloud.

The first property is that all the points are arranged in a grid corresponding to the camera's detector. This property leads to the assumption that if a pixel in the image is registered to one in the model, that pixel's neighbour corresponds to the respective neighbour in the model.

This greatly reduces the computational cost of registration in comparison to registering every pixel individually as is done in ICP.

The second property is that the update rate of the LIDAR is fast enough relative to the travel speed so that we assume negligible parallax is introduced between images. In this case, the 2-D z-map (considering only the z-components of the points) of the image and the model are panned versions of one another with some value biasing caused by y-rotation. The biasing comes from the objects decreasing in z-distance as they are rotated towards the borders of the detector.

Using the properties of consistently spaced points and negligible parallax, registration is done in four steps, as outlined below:

- 1. Convert image and model into 2-D z-maps.
- 2. Calculate Harris corner energy map of each z-map.
- 3. Cross-correlate Harris energy maps [Harris & Stephens 1988].
- 4. Use the offset of the correlation maximum to register the pixels.

The Harris corner energies are used in the correlation to emphasizes features alignment, attenuating the influence of featureless regions of the image. Empirical comparisons between the cross-correlations of the z-maps and Harris energy maps showed that the Harris maps produced a smaller and higher brightness maximum value.

Given a rectangularly indexed point cloud $\underline{\mathbf{I}}^k$ which consist M by N points \mathbf{I}_{mn}^k , the 2-D z-map is given by \underline{I}_Z^k and consists of points $I_{Z,mn}^k$. The transformation of an image to its Harris energy map, \underline{I}_H^k , is described by:

$$\underline{I}_{H}^{k} = \mathcal{H}(\underline{I}_{Z}^{k}) \tag{4.8}$$

where \mathcal{H} denotes the Harris corners transformation [Harris & Stephens 1988].

Using this convention, the maximum of the of the Harris energy map cross correlation of the image, $\underline{\mathbf{I}}^{k}$, with the model, $\underline{\mathbf{I}}^{k-1}$, is given by:

$$\underline{S} = \underline{I}_{H}^{k} \star \underline{I}_{H}^{k-1}$$

$$S_{uv} = \max(\underline{S})$$
(4.9)

Provided that both $\underline{\mathbf{I}}^k$ and $\underline{\mathbf{I}}^{k-1}$ are of size M by N, the point $\underline{\mathbf{I}}^{k-1}(M/2, N/2)$ is registered to $\underline{\mathbf{I}}^k(M-u, N-v)$.

In the same fashion as ICP, once registration of the two point clouds is complete, the problem turns to minimizing error between point pairs, earlier defined as egomotion. In the LIDAR's frame of reference, objective function for this minimization is the least squares problem:

$$f = \frac{1}{E} \sum_{e=1}^{E} \|\mathbf{i}_e^k - \mathbf{R}(\mathbf{q}_R) \left(\mathbf{i}_e^{k-1} - \mathbf{q}_T\right)\|^2$$
(4.10)

where, E is the total number of points, \mathbf{i}_e is the vector of the e^{th} point in \mathbf{I} , \mathbf{q}_R is the rotation quaternion, and \mathbf{q}_T is the translation vector, such that:

$$E = MN$$
 and $\mathbf{q} = \begin{bmatrix} \mathbf{q}_R \\ \mathbf{q}_T \end{bmatrix}$

To solve the least squares problem, a non-linear formulation of the trust-region-reflective optimizer (TRRO) was used [f. Coleman & Li 1996]. This method was quickest and simplest to implement as it is the default optimizer for bounded non-linear least squares solver in MATLAB.

Since image pixels are matched to the model's pixels rather than the interpolating model's data, sub-pixel residuals of motion are not accounted for. Not addressing these error reduces the computational cost by removing the need for multiple iterations of registration and interpolation. A quick argument will demonstrate that these residuals are absolutely bounded for any series of images. So, although there is always residual error, over longer distances, it account for a smaller percentage of the displacement.

To demonstrate the error bounding, assume an initial image, $\underline{\mathbf{I}}^1$, and model, $\underline{\mathbf{I}}^0$, with a relationship given by:

$$\underline{\mathbf{I}}^{1} = \mathbf{R}(\mathbf{q}_{R,calc} + \mathbf{q}_{R,res})\underline{\mathbf{I}}^{0} - (\mathbf{q}_{T,calc} + \mathbf{q}_{T,res})$$
(4.11)

In equations 4.11, \mathbf{q}_{calc} represents the motion calculated by the estimator and \mathbf{q}_{res} is the sub-pixel motion truncated by the estimator.

To explain the truncation, suppose the LIDAR undergoes linear motion, starting at point A and moving through B and on to C. At each point, an exposure is taken such that $\underline{\mathbf{I}}^0$ is the exposure at A, $\underline{\mathbf{I}}^1$ is the exposure at B, and $\underline{\mathbf{I}}^2$ is the exposure at C. Suppose $\mathbf{q}_{calc|A,B}$, the estimated motion from A to B, underestimates by truncating a quarter of a pixel and $\mathbf{q}_{calc|A,C}$ estimates with no error. This scenario is illustrated in Fig. 4.5.

In this case, the distance from A to B is not an integer number of pixels, resulting in $\underline{\mathbf{I}}^1$ being biased towards A by a quarter pixel (note the distinction between the position of the estimate



Figure 4.5: Illustration of a truncation example.

 $\underline{\mathbf{I}}^1$ and the physical position B). Since C is assumed to be a whole number of pixels away from $\underline{\mathbf{I}}^0$, when $\mathbf{q}_{calc|B,C}$ is calculated, it has a value of D + 0.75 pixels, where D is the rounded down integer number of pixels between B and C. Therefore, $\mathbf{q}_{calc|B,C}$ is rounded up, causing $\mathbf{q}_{calc|B,C}$ to over estimate by a quarter pixel. The result is then $\mathbf{q}_{calc|A,B} + \mathbf{q}_{calc|B,C} = \mathbf{q}_{calc|A,C}$. This logic can be extended for any number of intermediate points P - 1:

$$\mathbf{q}_{calc|1,P} = \sum_{p=1}^{P-1} \mathbf{q}_{calc|p,p+1}$$
(4.12)

According to eq. 4.12, the total truncation error of P-1 motions is the same as that for a single motion, which is at most half a pixel.

A summarized version of the algorithm:

- 1. Use Harris corners energy maps of the image and model to register them.
- 2. Using a least-squares formulation determine the motion which minimizes the distance between the registered point clouds.

4.2.5 Discrete Fourier Transform Method

This approach is similar to the Fourier domain deconvolution methods used for image registration such as that of De Castro and Morandi [De Castro & Morandi 1987]. However, instead of using a 2-D Fourier transform to extract in-plane translation and rotation, a novel approach which uses a series of 1-D transforms is used to determine translation and rotation in 3-D. This method does image registration and egomotion together, as opposed to using the Fourier transform to only register the image.

The LIDAR output is a 3-D point cloud where row and column indexing vary along the y and x directions, respectively, as defined by Fig. 4.1. To determine motion in the x-direction,

1-D Fourier transforms of the euclidean norms of the rows are used. The m^{th} row of the image and model are represented by $\underline{\mathbf{I}}_m^k$ and $\underline{\mathbf{I}}_m^{k-1}$, respectively, and their euclidean norms are \underline{I}_m^k and \underline{I}_m^{k-1} , respectively.

In order to deconvolve the image and model in the Fourier domain, the issue of the spatial domain row norms having a DC bias needs to be addressed. The problem arises from a necessity to append zero-padding to each row to prevent circular wrapping when using the fast Fourier transform (FFT) algorithm. The sudden change to all zero values creates a sharp drop, seen in Fig. 4.7a, which resembles a prominent feature shared by both image and model rows. However, knowing that the drop, near point 150 in Fig. 4.7a, is the result of the data ending, the true offset between the red and blue curves becomes apparent. Various methods for removing the drop were tested with two showing the most promise:

- Subtract the end value of the row from the whole row, eliminating the jump in value prior to the zero-padding. While this would be a very effective method for the example in Fig. 4.7a, cases where the model and image have substantially different end values would result in different offsets, potentially altering the majority of the row (in the context of DFT deconvolution, a sign change to part of one signal due to DC and not the other makes deconvolving unreliable).
- Taking the finite difference of the rows and deconvolving those. This works since the derivative tends to have no bias while still preserving unique characteristics of the row. As see in Fig. 4.7b, the shift between the red and blue rows is still perceivable but without the jump in value near index 150. Figure 4.6 also provides a histogram showing the effectiveness of this method.



Figure 4.6: Histogram of per row shift estimations using raw depth maps, 4.6a, and using the finite difference of the depth map, 4.6b. Nominal shift was positive ten pixels. Majority of points in 4.6a were exactly zero.



Figure 4.7: Depth map of image and model for a single row with zero-padding necessary for a FFT, (4.7a), and the numerical derivative of the depth map with zero-padding, (4.7b).

The finite difference method was selected for its reliability.

The discrete Fourier transform of a given row of the image and model is then given by:

$$\underline{X}_{m} = \mathcal{F}\left(d\underline{I}_{m}^{k-1}, 2N-3\right)$$

$$\underline{Y}_{m} = \mathcal{F}\left(d\underline{I}_{m}^{k}, 2N-3\right)$$
(4.13)

where \mathcal{F} denotes the Fourier transform, and $d\underline{I}_m^k$ is the derivative of \underline{I}_m^k . The second argument of \mathcal{F} is the size of the resulting Fourier transform. In order to prevent circular wrapping of the Fourier transform, the size must be the size of the two rows minus one. Since taking the derivative reduced the size of each row to N - 1, the required size of the DFT is hence 2N - 3. The linear time-invariant (LTI) transfer function relating each row of the image to the corresponding row of the model is then:

$$\underline{H}_m = \frac{\underline{Y}_m}{\underline{X}_m} \tag{4.14}$$

The impulse response function of each row is found by taking the inverse Fourier transform of the transfer function.

$$\underline{h}_m = \mathcal{F}^{-1}\left(\underline{H}_m\right) \tag{4.15}$$

The maximum of the impulse response function \underline{h}_m is then:

$$h_{\mathcal{M},m} = \max\left(|\underline{h}_m|\right) \tag{4.16}$$

where the subscript \mathcal{M} is the index of the maximum point in the m^{th} row. Since the \mathcal{M} is not necessarily the same for each row, the correct form would be $\mathcal{M}(m)$, however, \mathcal{M} is used for conciseness.

The maximums of the individual rows contain enough noise that simply averaging their values would not produce a reliable result. Instead, all the \underline{h}_m transfer functions are averaged together into \underline{h}' . This is then passed through a five point Gaussian low-pass filter to obtain \underline{h} . The filter helps remove high frequency noise making the peak more defined. Figure 4.8 gives two examples which emphasizes the difference between the transfer function of a single row, \underline{h}_m , an the averaged one, \underline{h} .

The maximum of the averaged transfer function of all the rows, \underline{h} , is then:

$$h_{\mathcal{N}} = \max(|\underline{h}|) \tag{4.17}$$

The subscript \mathcal{N} is used since, unlike \mathcal{M} , the index of the maximum value of <u>h</u> is unrelated to the row index m. A motion estimate is calculated using the value of the index \mathcal{N} :

$$T_{X,est} = \mathcal{N}w_x \tag{4.18}$$

where w_x is the average spatial sampling frequency of the LIDAR in the x-direction. The calculation of the average spatial sampling is:

$$w_x \approx \sum_{1}^{M} \frac{\left(I_{X,1}^{k-1} - I_{X,N}^{k-1}\right)}{N} \tag{4.19}$$

where the subscript X denotes the x-component. Equation 4.19 is an approximation because the pixel angularly evenly space, not spatially.

The motion estimate, \mathbf{T}_{est} , is used as a first guess in a final step which is uncoupling the translation in the *x*-direction from any rotation in the *y*-direction. For small translations and rotations, the two motion are indistinguishable, however, as the magnitude of the motion increases, the effects of parallax become more prominent. This arises from the 2-D map used to obtain the Fourier transforms being the distance map, $||x_{mn}||$. In this context, a *y*-rotation would introduce no parallax, causing each row to shift by the same number of pixels. Alternatively, translation would cause a greater shift in features closer to the camera and a smaller shift in features further away. The magnitude of these effects can be measured by comparing the relationship between the shifts of each row and their effective distance from the camera.

To get the effective distance to the camera, the row features need to be considered since they have the greatest influence on the calculated shift. In this case, the foreground objects



Figure 4.8: Absolute values of transfer functions corresponding to and x direction translation. Image (4.8a) is an example of a single row's LTI transfer function. Image (4.8b) is the averaged LTI transfer function of all rows.

against the background form the most discernible features and govern the calculated shift. Therefore, the effective distance to the camera, d_{eff} , is taken to be the average distance of the closest 10% of points in a row; this is chosen based on empirical testing.

Analytically, the pixel shift of a row resulting from translation and rotation is:

$$\mathcal{N} = \frac{\theta_{R,Y} + \operatorname{atan}\left(\frac{T_X}{d_{eff}}\right)}{\mu} \tag{4.20}$$

where μ is the apparent angular width of a pixel, $\theta_{R,Y}$ is the *y*-rotation of the camera, and T_X is the *x*-translation. From Eq. 4.20, for small translations, the pixel shift is inversely proportional to the effective distance of the row.

Prior to a RANSAC filter being used to determine inliers, some rows are eliminated as valid data points based on their crest factors. All rows whose $|\underline{h}_m|$ had a crest factor greater than four are passed into a RANSAC filter to determine the inlier from among them. The RANSAC filter used Eq. 4.21 as a test model.

$$\mathcal{N} = \frac{\theta_{R,Y} + \frac{T_X}{d_{eff}}}{\mu} \tag{4.21}$$

Using the inliers and \mathbf{T}_{est} as a first guess, the least-square problem given by Eq. 4.22 is solved using a TRRO [f. Coleman & Li 1996].

$$f = \left| \frac{\theta_{R,Y} + \frac{T_X}{\underline{d}_{eff}}}{\mu} - \underline{\mathcal{N}} \right|$$
(4.22)

where $\underline{\mathcal{N}}$ and \underline{d}_{eff} are arrays containing the respective values for the inliers.

To determine y-translation, the above process is repeated except a RANSAC algorithm is not used to decouple y-translation from x-rotation. Decoupling is not necessary since x-rotation is corrected for by the inclinometer prior to this. The y-translation is therefore calculated by the column-wise equivalent of Eq. 4.18.

Next, the image and model distance maps are aligned using the calculated offset \mathcal{N} . The z-maps of the overlapping portions of the image and model are differenced and the average, this is taken as the z-translation in the body frame.

Since the camera is tilted down by approximately ten degrees, the DFT approach differs from the other two in that, even with the inclinometer correction, its results are still relative to the detector plane. As a result, a last step is to undo the ten degree tilt of the camera to obtain rotation and translation in the body frame.

With the detailed algorithm presented, a summarized version is as follows:

- 1. Calculate the Euclidean norm of the point cloud to generate distance maps \underline{I}^k and \underline{I}^{k-1} .
- 2. Take the numerical derivative of the distance maps's rows.

- 3. Take the Fourier transform of the rows \underline{I}_m^k and \underline{I}_m^{k-1} .
- 4. Divide the Fourier transform of the image rows with those of the model.
- 5. Take the inverse Fourier transform of the quotient to obtain the spatial domain LTI function \underline{h} .
- 6. Use spatial sampling to determine the mean translation \mathbf{T}_{est} .
- 7. Use RANSAC to determine the rows of the image and model which best satisfy the relationship between effective row distance and pixel shift.
- 8. Use a non-linear least squares formulation to determine the optimal y-rotation and x-translation values for the inlier rows using \mathbf{T}_{est} for an initial guess.
- 9. Repeat steps 1 to 6 using columns instead of rows to determine y-translation.
- 10. Align the image and model distance maps and difference them to find the z-translation.
- 11. Convert transforms into the body frame.

4.2.6 Feature Tracking

The most common approach to egomotion estimation and that used by Barfoot et al., is tracking a small number of features and measuring the motion between their vectors [Barfoot *et al.* 2012]. When implemented on a LIDAR, this method uses features of the LIDAR's brightness map rather than features derived from the 3-D point cloud. By using the brightness, features will not necessarily spawn on geometric boundaries where parallax is influential. While such features are still useful, having a sample of texture features (parallax invariant) would help determine outliers among the geometric features.

The algorithm begins by both image and model brightness maps, A^k and A^{k-1} respectively, being passed through MATLAB's built in implementation of a histogram equalization (HE) filter, **histeq**. This is a necessary step for improving global image contrast, increasing the number of identifiable features, and reducing the prominence of bright objects. Figure 4.9 shows the effect of an HE filter on feature visibility.

$$A_{HE}^{k} = \text{histeq}(A^{k}) \tag{4.23}$$

The HE filtered brightness maps are then used to detect speeded up robust feature (SURF) vectors using MATLAB's detectSURFFeatures implementation. Conveniently, MATLAB also has an implementation for describing SURF features, extractFeatures, which is used to describe detected features. More information about the details of SURF feature detection and description is given in [Bay *et al.* 2008].



Figure 4.9: Demonstration of contrast enhancement from a raw image, (4.9a), and an image passed through a histogram equalization filter, (4.8b). Since both images are brightness plots in the infra-red spectrum, certain objects will emit their own light and appear brighter, such as the box in the lower region of the image.

Now that both image and model brightness maps have a corresponding set of described features, they are matched using the MATLAB implementation matchFeatures. This function uses the algorithm developed by Lowe [Lowe 2004] to perform an exhaustive search for unique feature pairs.

The locations of the matched features on the detector are then used to generate 3-D points from the LIDAR's 3-D point cloud using linear interpolation.

With 3-D representation of each feature pair, the points are passed into a RANSAC filter which, unlike the formulation used for the DFT/IDFT RANSAC algorithm, uses the egomotion problem for its model. This formulation is similar in form to Eq. 4.10 and is given as:

$$\underline{\mathbf{F}}^{k} = \mathbf{R}(\mathbf{q}_{R}) \left(\underline{\mathbf{F}}^{k-1} - \mathbf{q}_{T} \right)$$
(4.24)

where $\underline{\mathbf{F}}^k$ and $\underline{\mathbf{F}}^{k-1}$ are the matched features' vectors for the image and model respectively.

Solving for the x-translation and y-rotation first, Eq. 4.24 becomes:

$$q_{T,X} = \underline{F}_X^{k-1} - \underline{F}_X^k \cos\left(q_{R,Y}\right) - \underline{F}_Z^k \sin\left(q_{R,Y}\right)$$
$$0 = \underline{F}_Z^{k-1} - \underline{F}_Z^k \cos\left(q_{R,Y}\right) + \underline{F}_X^k \sin\left(q_{R,Y}\right)$$
(4.25)

Assuming the rotation between frames is less than 0.1 rad, the small angle approximation can be applied to linearize and solve Eq. 4.25. This approximation helps simplify the implementation. The error model for the RANSAC filter is the distance between transformed feature vectors of the model and the feature vectors of the image:

$$\operatorname{Error} = \|\underline{\mathbf{F}}^{k} - \mathbf{R}(\mathbf{q}_{R}) * \left(\underline{\mathbf{F}}^{k-1} - \mathbf{q}_{T}\right)\|$$
(4.26)

After the RANSAC filter establishes the inliers, those feature vector pairs are passed into a TRRO using the least squares formulation given by Eq. 4.26; this provides the xyztranslation and y-rotation solutions.

The summarized feature based algorithm is:

- 1. Pass the image and model brightness maps, A^k and A^{k-1} respective, through a histogram equalization filter.
- 2. Detect the SURF features for A_{HE}^k and A_{HE}^{k-1} .
- 3. Describe detected SURF features.
- 4. Match detected features.
- 5. Pass match feature vector pairs into a RANSAC filter to determine inliers.
- 6. Use inlier pairs in a least squares TRRO to determine motion.

4.3 Visual Odometry Testing

Prior to the rover having been made operational, preliminary results using only the LIDAR camera were obtained. These results motivated algorithm design decisions and provided early insight into the competitiveness of mentioned algorithms. These results will be shown in this section.

4.3.1 Thresholding for Discrete Fourier Transform Method

As was mentioned in section 4.2.5, row LTI transfer functions with crest-factors below four were discarded under the pretext that they harbour too much noise. Figure 4.10 show the shift predicted for each row along with its crest factor for two exposures with no motion between them. As is evident from the figure, rows with crest factors below four had a lot of spread in predicted shift and few points which were close to the nominal value of zero pixel shift.



Figure 4.10: Scatter plot of row crest factor for two subsequent exposures with no motion between them. The nominal shift for this case would be zero.

4.3.2 LIDAR with Inclinometer Validation

In section 4.3.3, VO was validated without the use of an inclinometer. This section will look at VO *x*-translation estimates to determine if there is any adverse effect of using the inclinometer to correct LIDAR images. The primary concern was whether acceleration of the rover experienced by the inclinometer would affect the gravity vector readings enough to degrade VO estimates. The inclinometer measures the direction of the local acceleration, including the affect of the rover's acceleration; this coincides with the gravity vector when the rover is not accelerating. Figure 4.11 looks at just such a case, using successive LIDAR images to plot motion. However, as can be seen in Fig. 4.11, the sudden acceleration at image 45 has no noticeable effect on the solution and the results, the distance travelled is accurate to within the accuracy of the instrument used to measure it.

4.3.3 Performance Testing

Using only the LIDAR camera, telemetry was collected by driving the rover in a straight line at low speed through a grass field with foliage in the background. In total, 1000 exposures



Figure 4.11: Forward motion estimate from VO when using inclinometer corrections under acceleration. The image count is proportional to time. The curve consists of no rover motion at the beginning followed by constant velocity travel.

were take over a distance of 40 m. Figure 4.12 shows the results of this testing.

The point cloud minimization method failed to provide any meaningful estimate of the rover's state and was furthest from truth in all motions. The DFT method and feature tracking methods performed similarly with the the feature based method having a smoother path. Their translation plots track nominal while their rotation plots exhibit increasing error. Although they under-estimated x-translation, the addition of heading from a star track would flag the slow procession in y-rotation as false, adding it instead to the x-translation. The camera's geometry is such that the coupling of x-translation to y-rotation converts one radian to one meter for small motions. Since this data consists of the 1000 small motions added together, adding the star track would ad the procession of the y-rotation to x-translation increasing its final value. This would bring the final x-translation of the DFT to approximately 22 m and that of the feature based method to approximately 34 m.

Adopting performance targets of the MERs, the acceptable margin of error for x-translation would be four meters from the true value of 40 m. The further addition of the inclinometer



Figure 4.12: Preliminary results of VO testing for all four motions tracked by the camera. This data set consists of 1000 exposures and a 40 m driven distance.

correction could make the difference needed to meet that target.

CHAPTER 5

Geolocation

Geolocation was obtained with a Sinclair Interplanetary ST-16 star track and a Jewell Instruments Tuff Tilt Digital inclinometer. Global position is derived by relating Earth-centredinertial (ECI) star vector obtained in the navigation frame to the to the Earth-centred-Rotational (ECR) frame. Figure 5.1 shows the relationship between different frames of reference and the information needed to transform one into the other.



Figure 5.1: Simplified schematic of frame transformations.

Starting with the body frame, Φ_B , the inclinometer frame, Φ_C , is the rotation through the inclinometer mounting angle, \mathbf{C}_{CB} . From this, the navigation frame, Φ_N , can be obtained using the rotation measured by the inclinometer, \mathbf{C}_{NC} .

$$\mathbf{C}_{NB} = \mathbf{C}_{NC} \mathbf{C}_{CB} \tag{5.1}$$

Next, the heading of the rover determines the rotation between the topocentric frame, Φ_T , and Φ_N . Defining the topocentric frame as north east up, where the y-axis is pointing north,

a z-rotation of the heading angles, \mathbf{C}_{TN} , rotates Φ_N into Φ_T . A heading angle of zero would mean Φ_N and Φ_T and aligned.

Going in the opposite direction, Φ_B can be rotated through the star tracker's mounting angle, \mathbf{C}_{SB} to obtain the star tracker frame, Φ_S . The star tracker then measures the quaternion which relate its frame to the ECI frame, Φ_I , through the rotation \mathbf{C}_{IS} .

$$\mathbf{C}_{IB} = \mathbf{C}_{IS} \mathbf{C}_{SB} \tag{5.2}$$

Lastly, the sidereal time, provided by a clock along with an accurate model of the Earth's rotation, provides the rotation between the ECI frame and the ECR frame, C_{RI} , where C_{RI} is the rotation from Φ_I to Φ_R . This step is done using MATLAB's implementation dcmeci2ecef which converts using the IAU-2000/2006 reduction [Petit & Luzum 2010].

Having the ability to go from Φ_R to Φ_N , the z-axis of the navigation frame can be described ECR frame can be used to obtain a global position. The inclinometer cannot provide a complete rotation from Φ_B to Φ_T since it does not have heading information. However, the z-axis of Φ_T is parallel with that of Φ_N and this property is what allows the z-axis of Φ_N to be used instead. The $\hat{\mathbf{k}}$ vector of Φ_T , can be constructed in Φ_C using the inclinometer model [Enright *et al.* 2012].

$$\widehat{\mathbf{k}}_{B} = \mathbf{C}_{BC} \begin{bmatrix} \sin\left(-\theta_{incl,Y}\right)\cos\left(-\theta_{incl,X}\right) \\ \cos\left(-\theta_{incl,Y}\right)\sin\left(-\theta_{incl,X}\right) \\ \cos\left(-\theta_{incl,Y}\right)\cos\left(-\theta_{incl,X}\right) \end{bmatrix}$$
(5.3)

where $\widehat{\mathbf{k}}_B$ is $\widehat{\mathbf{k}}$ of Φ_N but in the body frame, as denoted by the subscript. The angles $\theta_{incl,Y}$ and $\theta_{incl,X}$ are the pendulum angles of the inclinometer. The rotation \mathbf{C}_{NC} , also taken from the inclinometer model, is:

$$\beta = -\operatorname{asin}\left(\frac{\mathbf{k}_{X,C}}{\|\mathbf{k}\|}\right)$$
$$\gamma = \operatorname{asin}\left(\frac{\mathbf{k}_{Y,C}}{\cos\left(\beta\right)\|\mathbf{k}\|}\right)$$
$$\mathbf{C}_{CN} = \mathbf{R}_{X}\left(\gamma\right)\mathbf{R}_{Y}\left(\beta\right)\mathbf{R}_{Z}\left(\pi\right)$$
(5.4)

where $\mathbf{k}_{X,C}$ and $\mathbf{k}_{Y,C}$ are the x and y components of \mathbf{k} in Φ_C , respectively. The rotation $\mathbf{Rz}(\pi)$ comes from Φ_C being π radians away from the y-axis point in the forward direction of the rover when the heading angle is zero.

The result of Eq. 5.3 can then be transformed into the ECR frame using the rotations established above.

$$\widehat{\mathbf{k}}_R = \mathbf{C}_{RI} \mathbf{C}_{IS} \mathbf{C}_{SB} \widehat{\mathbf{k}}_B \tag{5.5}$$

With the topocentric frame's z-axis in Φ_R , the longitude, λ , and geodetic latitude, ϕ' , can

be used to calculate the rotation between Φ_R and Φ_T , represented by \mathbf{C}_{TR} .

$$\lambda = \operatorname{atan2}\left(\widehat{\mathbf{k}}_{Y,R}^{2}, \widehat{\mathbf{k}}_{X,R}^{2}\right)$$

$$\phi' = \operatorname{atan}\left(\frac{\mathbf{k}_{Z,R}}{\sqrt{\widehat{\mathbf{k}}_{Y,R}^{2} + \widehat{\mathbf{k}}_{X,R}^{2}}}\right)$$
(5.6)

where $\hat{\mathbf{k}}_{X,R}$, $\hat{\mathbf{k}}_{Y,R}$, and $\hat{\mathbf{k}}_{Z,R}$ are the x, y, and z components of $\hat{\mathbf{k}}_{R}$, respectively.

$$\mathbf{C}_{TR} = \mathbf{Ry}\left(\frac{\pi}{2} - \phi'\right) \mathbf{Rz}\left(\frac{\pi}{2} + \lambda\right)$$
(5.7)

It is now possible to solve for the heading of the rover by equating the product of rotation matrices going counter-clockwise in Fig. 3.3, starting with the navigation frame, to the rotation matrix \mathbf{C}_{TN} .

$$\mathbf{C}_{TN} = \mathbf{C}_{TR} \mathbf{C}_{RI} \mathbf{C}_{IS} \mathbf{C}_{SB} \mathbf{C}_{BC} \mathbf{C}_{CN}$$
(5.8)

By definition, \mathbf{C}_{TN} is only a z-rotation through the heading, α . Using the y-axis in the navigation frame, $\hat{\mathbf{j}}$, as a datum:

$$\widehat{\mathbf{j}}_T = \mathbf{C}_{TN} \widehat{\mathbf{j}}$$

$$\alpha = \operatorname{atan2}\left(\widehat{\mathbf{j}}_{X,T}, \widehat{\mathbf{j}}_{Y,T}\right)$$
(5.9)

where $\hat{\mathbf{j}}_{X,T}$ and $\hat{\mathbf{j}}_{X,T}$ are the x and y components of $\hat{\mathbf{j}}_T$, respectively.

Lastly, the geocentric latitude is calculated. To calculate geocentric latitude, ϕ the WGS84 convention is used. This thesis will not go into detail on how the WGS84 is implemented but the complete standard and its implementation is presented in [Mularie 1984].

Chapter 6 System Integration

Sensor fusion for the rover is performed at two rates to avoid averaging or down-sampling of the data. The LIDAR camera and inclinometer are capable of a comparably higher operation frequency than the star tracker. Additionally, the star tracker is more accurate when the rover is stationary while the LIDAR is needed when the rover is in motion. For this reason the LIDAR and star tracker were chosen to not operate in parallel.

If the star tracker were used while the rover was in motion, the epoch of the star image would need to be synchronized with the equivalent inclinometer measurement. While this is not complex to implement using linear interpolation, the sensitivity of these sensors to arcsecond rotations could require high order interpolation to account for angular acceleration or even oscillations of the rover's suspension. For this reason the rover is stopped and one second is given for the rover's dampers to kill oscillations before the star tracker and inclinometer take readings in quick succession. Although the epochs are not strictly synchronised, the topocentric orientation is assumed constant during the star tracker exposure for a stationary rover.

The resulting acquisition pattern consists of high frequency inclinometer and LIDAR pairs being taken while the rover is moving followed by the rover being stopped to take a star tracker and inclinometer readings, as illustrated by Fig. 6.1. The higher rate data acquisition forms the minor measurements and the lower rate data acquisition forms the major measurements.

6.1 Minor Measurement Sensor Fusion

Because the VO algorithms will not try and estimate pitch and roll in the body frame, the VO and inclinometer outputs form a determinate solution. At each LIDAR exposure, the inclinometer is used to both determine body pitch and roll as well as correct the LIDAR image using with that information. Afterwards, the VO algorithm determines translation in all directions as well as body yaw, forming a complete solution. The VO solutions are integrated for the duration of the K high rate measurements, providing net motion.



Star tracker and inclinometer reading, rover is stationary

LIDAR and inclinometer reading, rover is moving

Figure 6.1: Diagram illustrating the spacing of LIDAR and star tracker readings.

6.2 Major Measurement Sensor Fusion

Low rate sensor fusion is updated once every star tracker reading. This process takes the net translation provided by the VO and treats it as a single measurement. The inclinometer again serves two purposes in this step, it is used to obtain a global position and heading using the star tracker reading and to estimate body pitch and roll. The heading is then used to correct the VO heading estimate and re-run the algorithm with LIDAR *y*-rotation a fixed parameter. This is done by subtracting the average difference between the net change in heading measured by VO, $\Delta \alpha_{VO}$, from the change in heading between the two most recent star tracker reading $\Delta \alpha$.

$$\underline{q}_{R\alpha} = \underline{q}_{RV} - (\Delta \alpha - \Delta \alpha_{VO})/K \tag{6.1}$$

where $\underline{q}_{R,\alpha}$ is the set of heading corrected LIDAR *y*-rotation estimates and *K* is the total number of LIDAR images between the two most recent star tracker readings. By forcing a modified *y*-rotation through the VO algorithm, the optimizations arrived at a new *x*-translation value.

Because this part of the sensor fusion is over-determined, a Kalman filter is implemented to performed a statistically weighted fusion of sensor measurements.

The inclinometer x and y tilt angle vector will be represented by θ_{incl} and $\mathbf{q}_{T,VO}$ will represent the heading corrected translation estimate given by the VO algorithm. The linear state transition model of the Kalman filter:

where $\Delta \lambda_{VO}$ and $\Delta \phi_{VO}$ are the changes in longitude and latitude as calculated the VO, respectively, z is the z-translation in the topocentric frame, and θ is the x and y rotation in the body frame. The blank regions of the transition matrix are filled with the appropriate zero arrays.

The linear observation model of the filter is then:

$$\begin{bmatrix} \lambda_{j} \\ \phi_{j} \\ \widetilde{\theta}_{incl,X,j} \\ \widetilde{\theta}_{incl,Y,j} \\ \alpha_{j} \\ \Delta x_{VO,j} \\ \Delta y_{VO,j} \\ \Delta z_{VO,j} \end{bmatrix} = \begin{bmatrix} 1 & & & & & & 0 \\ 1 & & & & & & \\ & 0 & 0 & 0 & 1 & & \\ & & -1 & 0 & & & \\ & & & -1 & 0 & & \\ & & & -1 & 0 & & \\ & & & 0 & r_{E} & & \\ 0 & & & & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_{j} \\ \phi_{j} \\ z_{j} \\ \theta_{X,j} \\ \theta_{Y,j} \\ \alpha_{j} \\ \Delta \lambda_{VO,j} \\ \Delta \phi_{VO,j} \\ \phi_{VO,j} \\ QT,Z,j \end{bmatrix}$$
(6.3)

where r_E is the radius of the Earth in meters. Again, all empty regions of the observation matrix are filled with an appropriate number of zeros.

The process noise for the Kalman filter, \underline{Q} is selected to best reflect the expected motion from one data frame to another. The measurement noise, \underline{R} is selected to reflect the expect repeatability of the measurements.

The initial state covariance, \underline{P} is a diagonal matrix with diagonal elements:

$$\begin{bmatrix} \frac{5000}{r_E} & \frac{5000}{r_E} & 0 & \frac{1}{1000} & \frac{1}{1000} & \frac{1}{1000} & 0 & 0 & 0 \end{bmatrix}$$
(6.4)

The equations of the Kalman filter prediction step are:

$$\mathbf{s}_{j|j-1} = \underline{F}_{j} \mathbf{s}_{j-1|j-1}$$

$$\underline{P}_{j|j-1} = \underline{F}_{j} \underline{P}_{j-1|j-1} \underline{F}_{j}^{\mathrm{T}}$$
(6.5)

where, \underline{F} is the state transition matrix from Eq. 6.2 and \mathbf{s} is the major measurement state vector in Eq. 6.2.

The equations for the Kalman filter update step are:

$$\underline{A}_{j} = \underline{\rho}_{j} - \underline{H}_{j} \mathbf{s}_{j|j-1} \\
\underline{C}_{j} = \underline{H}_{j} \underline{P}_{j|j-1} \underline{H}_{j}^{\mathrm{T}} + \underline{R}_{j} \\
\underline{D}_{j} = \underline{P}_{j|j-1} \underline{H}_{j}^{\mathrm{T}} \underline{C}_{j}^{-1} \\
\mathbf{s}_{j|j} = \mathbf{s}_{j|j-1} + \underline{D}_{j} \underline{A}_{j} \\
\underline{P}_{j|j} = (\underline{1} - \underline{D}_{j} \underline{H}_{j}) \underline{P}_{j|j-1}$$
(6.6)

where, <u>*H*</u> is the observation matrix in Eq. 6.3, $\underline{\rho}$ is the measurement vector in Eq. 6.3, and <u>1</u> is an identity matrix.

6.3 Complete System Test

The complete tests were performed using all of the rover's sensors. For these tests, the rover would stop when acquiring star tracker readings and move when acquiring LIDAR readings. Between successive individual star tracker readings, 200 LIDAR images were taken. After the rover was done moving, a one second delay allowed all motion to stop before the next star tracker reading was taken. The inclinometer collected data any time one of the other two sensors did.

6.3.1 Global Navigation Test

For the global navigation test, the rover travelled a distance of 13 m through a field with foliage in the background. The rover's heading was South-East and moved in roughly a straight line. The initial position of the rover was given to the low rate Kalman filter. Although unfortunate, technical difficulties with this part of testing results in less data being collected than would be preferred. Figure 6.2 shows the global position estimate of this test.

In terms of path length accuracy and qualitative heading, the point cloud minimization method performed the worst while the other two methods had more similar results. Both the DFT and feature based methods traced out a path of ten meters (including z-translation), three meters short of truth. The feature based method did provide a straighter path which qualitatively better resembled the true path. Table 6.1 shows the full results of testing.

The underestimation of the true motion is most likely due to the LIDAR camera intermittently failing to communicate on the night of the test along with an un-noticed shift in the rover's suspension. The suspension shift pointed the LIDAR too far down to see the foliage at all terrain angles, leaving the VO algorithms with few features in the field of view.



Figure 6.2: Global position estimate of the rover using the star tracker, inclinometer, and LIDAR together. The performance of each of the three LIDAR methods is displayed.

To obtain the results in Fig. 6.2, the Kalman filter's measurement noise had to be changed from the initial estimate. Originally, the global position estimate of the star track was believed to be accurate to within 200 m. Alone, the star tracker's global position estimates had an accuracy of about five kilometres. However, in order to obtain a straight line for the estimated path, the respective measurement noise was increased to an effective accuracy of 100 km. The star tracker measurements are devalued to maintain to the relative confidence between the star trackers's position estimate and the LIDAR's. Figure 6.3 shows the motion estimated through sensor fusion with no initial position and confidence in the star tracker's

	Distance Travelled (m)	Path Length (m)	Path Heading (deg)
Expected	13 m	13	120
Point Cloud Minimization	30.2	46.7	29.2
DFT/IDFT	6.5	10.3	131
Feature Tracking	8.85	10.9	110

 Table 6.1: Integrated system test.

position estimates being the expected five kilometres of measurement noise.



Figure 6.3: (Surface position estimates through sensor fusion using no initial position and a five kilometre measurement noise for the star tracker's global position estimates. The start point of the path is at the origin.)

It is worth noting that although the direction of travel in Fig. 6.3 is in the opposite direction of what is expected and the start point is approximate five kilometres South-East of the true position (origin of the graph), the estimated position is moving towards the rover's true position.

Chapter 7

Conclusion and Future Work

This thesis confirmed the ability of a star tracker and inclinometer to provide a global position estimate to within five kilometres and heading during night-time. A method for supplementing LIDAR measurements with heading and inclinations was developed and shown to work. Based on the comparisons of three VO algorithms, the feature based method is the most accurate. Throughout testing, VO performance was variable, ranging between 23% and 15% drift, well above the target of 10%. To meet these requirements, more rigorous hardware testing needs to be implemented along with better algorithm tuning. Primary concerns are accuracy of sensor calibrations during temperature changes and more adaptive RANSAC inlier thresholding.

The outcome of this is that the goal of developing a working test platform has been met. While more work is needed to refine the navigation algorithms and systems to meet navigation targets, the ability to use the three sensors on a moving platform and acquire meaningful data was demonstrated.

The next step in this project would be to perform this test over a longer distance where drift from VO would be noticeable. A reduced drift when a star tracker is added would provide strong support for the use of coarse global position. Along with this, it would also be beneficial to test other non-global (sparse), VO algorithms; those similar to the feature based one. It is likely that the inclusion of more data points does not provide the solvers with more meaningful information. Lastly a test where the rover does not move and subsequent star tracker measurements are used to localize the rover's position through filtering would reinforce the effectiveness of global positioning with a star tracker.

References

- [Barfoot et al. 2012] Barfoot, Timothy D., McManus, Colin, Anderson, Sean, & et al., "Into Darkness: Visual Navigation Based on a Lidar-Intensity Image Pipeline," Proceedings of the International Symposium on Robotics Research, 2012.
- [Bay et al. 2008] Bay, Herbert, Ess, Andreas, Tuytelaars, Tinne, & Van Gool, Luc, "SURF:Speeded Up Robust Features," Computer Vision and Image Understanding, Vol. 110, No. 3, 2008, Pages: 346 – 359.
- [Besl & McKay 1992] Besl, Paul J., & McKay, Neil D., "A Method for Registration of 3-D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, 256, 1992, Page: 239.
- [Biesiadecki et al. 2005] Biesiadecki, Jeffrey J., Baumgartner, Eric T., Bonitz, Robert G., & et al., "Mars Exploration Rover Surface Operations: Driving Opportunity at Merdiani Planum," Systems, Man and Cybernetics, 2005 IEEE International Conference on, Vol. 2, October, 2005, Pages: 1823 – 1830.
- [De Castro & Morandi 1987] De Castro, E., & Morandi, C., "Registration of Translated and Rotated Images Using Finite Fourier Transforms," *Pattern Analysis and Machine Intelligence, IEEE transactions on*, Vol. PAMI-9, No. 5, 1987, Pages: 700 – 703.
- [Enright et al. 2009] Enright, John, Furgale, Pauk, & Barfoot, Time, "Sun Sensing for Planetary Rover Navigation," *IEEE Aerospace Conference*, 2009.
- [Enright et al. 2012] Enright, John, Barfoot, Time, & Soto, Marcela, "Star Tracking for Planetary Rovers," Aerospace Conference, 2012 IEEE, Big Sky, MT, 2012, Pages: 1 – 13.
- [f. Coleman & Li 1996] f. Coleman, Thomas, & Li, Yuying, "An Interior Trust Region Approach For Nonlinear Minimization Subject to Bounds," SIAM J. OPTIMIZATION, Vol. 6, No. 2, May, 1996, Pages: 418 – 445.
- [Fischler & Bolles 1981] Fischler, Martin A., & Bolles, Robert C., "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, Vol. 24, No. 6, 1981, Pages: 381 395.
- [Goecke et al. 2007] Goecke, Roland, Asthana, Akshay, Pettersson, Niklas, Petersson, Lars, & et al., "Visual Vehicle Egomotion Estimation using the Fourier-Mellin Transform," *Proceedings of the 2007 IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, 2007, Pages: 450 – 455.
- [Grotzinger et al. 2012] Grotzinger, John P., Crisp, Joy, Vasavada, Ashwin R., & et al., Robert C. Anderson, "Mars Science Laboratory Mission and Science Investigation," Space Science Review, Vol. 170, No. 1-4, September, 2012, Pages: 5 – 56.

- [Harris & Stephens 1988] Harris, Chris, & Stephens, Mike, "A Combined Corner and Edge Detector," In Proc. of Fourth Alvey Vision Conference, 1988, Pages: 147 – 151.
- [Irani et al. 1994] Irani, M., Rousso, B., & Peleg, S., "Recovery of ego-motion using image stabilization.," Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on, Seattle, June, 1994, Pages: 454 – 460. IEEE.
- [Jewell Instruments LLC] Jewell Instruments LLC. "Tuff Tilt Digital: Uniaxial & Biaxial Tiltmeter," . Rapport technique, 850 Perimeter Road, Manchester, NH.
- [Lowe 2004] Lowe, David G., "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, Vol. 60, No. 2, 2004, Pages: 91 – 110.
- [Maimone et al. 2007] Maimone, Mark, Cheng, Yang, & Matthies, Larry, "Two years of visual odometry on the Mars Exploration Rovers," Journal of Field Robotics, Special Issue on Space Robotics, Vol. 24, 2007.
- [MESA Imaging] MESA Imaging. "SR4000/SR4500 User Manual," . Rapport technique, AG, Technoparkstrasse 1, 8005 Zurich.
- [Milella & Siegwart 2006] Milella, Annalisa, & Siegwart, Roland, "Stereo-Based Ego-Motion Estimation Using Pixel Tracking and Iterative Closest Point," *Proceedings of the Fourth IEEE International Conference on Computer Vision Systems*, 2006.
- [Mularie 1984] Mularie, William M. "Department of Defense World Geodetic System 1984: Its Definition and Relationships with Local Geodetic Systems," . Rapport technique 8350.2 Third Edition, National Imagery and Mapping Agency, 3200 South Second Street, St. Louis, MO, 1984.
- [Nister et al. 2004] Nister, D., Naroditsky, O., & Bergen, J., "Visual odometry," Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, Vol. 1, 2004, Pages: I-652 - I-659.
- [Petit & Luzum 2010] Petit, Gérard, & Luzum, Brian. "IERS Convention (2010)," . IERS Technical Note 36, International Earth Rotation and Reference System Services, Frankfurt, 2010.
- [Piatti & Rinaudo 2012] Piatti, Dario, & Rinaudo, Fulvio, "SR-4000 and CamCube3.0 Time of Flight (ToF) Cameras: Tests and Comparison," *Remote Sensing*, Vol. 4, No. 4, 2012, Pages: 1069 – 1089.
- [Rusu et al. 2009] Rusu, Radu Bogdan, Blodow, Nico, & Beetz, Michael, "Fast Point Feature Histogram (FPFH) for 3D Registration," *IEEE International Conference on Robotics* and Automation, Kobe, May, 2009, Pages: 3212 – 3217. IEEE.
- [Scaramuzza & Fraundorfer 2011] Scaramuzza, Davide, & Fraundorfer, Friendrich, "Visual Odometry. Part I: The First 30 Years and Fundamentals," *IEEE Robotics & Automa*tion Magazine, 2011, Pages: 80 – 92.

- [Tillman 2009] Tillman, James E, "Mars temperature overview," Online publication. URL http://www-k12. atmos. washington. edu/k12/, retrieved, 2009, Pages: 07–15.
- [Wertz et al. 2011] Wertz, James R., Everett, David F., & Puschell, Jeffery J., editeurs, Space Mission Engineering: The New SMAD, Microcosm Press, 2011.