Probabilistic Terrain Analysis and Roughness Estimation

Stephanos Loizou

MInf Project (Part 1) Report
Master of Informatics
School of Informatics
University of Edinburgh
2017
Abstract

Terrain classification is becoming an important issue in robotic navigation. Traditional approaches deal with classifying terrain into discrete categories but as autonomy levels increase, different approaches are required to successfully navigate through arbitrary environments. Outdoor environments, such as forests, do not follow any structure and thus traditional terrain classification approaches are not sufficient. This project explores the problem of terrain classification and roughness estimation for the purpose of traversability of Unmanned Ground Vehicles in unstructured environments. Pro-prioeptive and Exteroceptive methods from literature are explored. A simple, novel roughness estimation metric is proposed and implemented. The proposed approach borrows ideas from Probabilistic Terrain Analysis [1MA06], Terrain Roughness Identification [WR14] and Roughness Map for Autonomous Rovers [LEC13]. The approach produces Drivability Index maps that are representative of the terrain presented.
Acknowledgements

I would like to thank my supervisor, Dr. Maurice Fallon, for his support and help throughout this project as well as the rest of the members of the Robot Perception Group at the University of Edinburgh, especially Georgi Tinchev for helping me get acquainted with the robot and in conducting the data collection required for this project.
# Table of Contents

## 1 Introduction
1.1 Project Overview ........................................... 2
1.2 Objectives .................................................. 2
1.3 Report Outline .............................................. 2

## 2 Background and Related Work
2.1 Definition of the problem and Terminology .................. 5
  2.1.1 Definition of Traversability .......................... 5
  2.1.2 Terrain and environment types ....................... 6
  2.1.3 Approaching Terrain Analysis .......................... 6
  2.1.4 Refining the problem .................................. 7
2.2 Technical Background ....................................... 7
  2.2.1 Laser Based Sensors ................................ 7
  2.2.2 Image Sensors ......................................... 8
  2.2.3 Proprioceptive Sensors ............................... 9
  2.2.4 Pointclouds and Heightmaps ......................... 9
  2.2.5 Pose Estimation ..................................... 10
  2.2.6 Coordinate Frames and Geometric Transformations .... 11
2.3 Related Work ............................................... 11
  2.3.1 Proprioceptive Methods ............................... 12
  2.3.2 Exteroceptive Methods ............................... 12
  2.3.3 Terrain Roughness Rating ............................ 15
2.4 Motivation, Approach and Theory ............................ 17

## 3 Design Process .............................................. 21
3.1 Hardware Platform ......................................... 21
  3.1.1 Background ........................................... 21
  3.1.2 ClearPath Husky UGV ................................ 21
3.2 Sensors .................................................... 21
3.3 Software Platform ......................................... 22
3.4 Algorithm Input/Output Specification ....................... 23
3.5 Classification Pipeline and Algorithm Design ............. 24
3.6 Supporting Infrastructure ................................ 24
  3.6.1 Robot Description, Coordinate Frames and Transformations .. 25
  3.6.2 LiDAR pointcloud accumulation ....................... 25
  3.6.3 Stereo Camera pointcloud reconstruction ......... 26
3.6.4 Pointcloud cropping ........................................... 26
3.6.5 Projection to 2D grid ........................................... 27
3.7 Methodology and evolution of the system .......................... 27
3.8 Experiment Design .................................................. 28
3.8.1 Experiments in Simulation ...................................... 28
3.8.2 Testing and calibration on real data .............................. 30
3.8.3 Design Evaluation .................................................. 31
3.9 Terrain Labelling and Process ...................................... 32
3.10 Identified Difficulties in Evaluation ............................... 33
3.11 Summary ............................................................. 33

4 Implementation .......................................................... 35
4.1 Prepossessing ............................................................ 35
4.2 Terrain Classification and Drivability Index ......................... 36
4.3 Updating the Global Map ............................................. 37

5 Results ................................................................. 41
5.1 Evaluation around George Square south entrance ............... 41
5.1.1 Area 1 ............................................................... 42
5.1.2 Area 2 ............................................................... 46
5.1.3 Area 3 ............................................................... 49
5.2 Summary ............................................................... 51

6 Conclusion ............................................................... 53
6.1 Conclusions ............................................................ 53
6.1.1 Summary of work undertaken ................................... 53
6.1.2 Summary of evaluation .......................................... 54
6.1.3 Summary of achievements and lessons learned ................ 54
6.2 Further Development .................................................. 55

A Statistical Tests .......................................................... 63

B Supplementary Results ................................................. 65
B.1 Average Deviation ..................................................... 65
B.2 George Square North .................................................. 66
B.3 Path Segmentation ..................................................... 68
List of Figures

2.1 Demonstration of how a LiDAR sensor works ........................................... 7
2.2 A visualization of epipolar geometry ......................................................... 8
2.3 Visualizing terrain ....................................................................................... 10
2.4 Visualization of the 6 Degrees of Freedom (DOF) ..................................... 10
2.5 Stanley’s PTA system ................................................................................... 14
2.6 Example output of Roughness Mapping ..................................................... 17

3.1 ClearPath Husky as configured ................................................................. 23
3.2 Visualisation of LiDAR FOV ....................................................................... 25
3.3 Labelling function tests in simulation ......................................................... 28
3.4 Simulated Rough Terrain Test ................................................................... 29
3.5 Simulated objects test .................................................................................. 29
3.6 ROC curves for determining the vehicle traversal height parameter ....... 31

5.1 Test Location Reference Picture ................................................................. 42
5.2 Drivability Index maps for Area 1 ............................................................... 43
5.3 Road Recall Rate of Area 1 ......................................................................... 44
5.4 Obstacle Misclassification Rate for Area 1 ................................................ 45
5.5 Local Classifications for Area 1 .................................................................. 46
5.6 Drivability Index maps for Area 2 ............................................................... 47
5.7 Road Recall Rate for Area 2 ......................................................................... 48
5.8 Local Classifications for Area 2 .................................................................. 48
5.9 Drivability Index maps for Area 3 ............................................................... 49
5.10 Road Recall and Obstacle Misclassification Rates for area 3 .................... 50
5.11 Local Classifications for Area 3 .................................................................. 51

B.1 Average Deviation from labelled roughness ............................................. 66
B.2 Stereo Drivability Map for George Square North ..................................... 67
B.3 LiDAR Drivability Map for George Square North ..................................... 67
B.4 Path Segmentation Experiments - Stereo ................................................ 68
B.5 Path Segmentation Experiments - Lidar ..................................................... 68
Chapter 1

Introduction

This project investigates the problem of navigation of medium sized unmanned ground vehicles (UGVs) in unstructured outdoor environments and the different techniques currently available to perceive the terrain and subsequently implement an algorithm to classify and rank terrain in order to be used for selecting a suitable, in terms of drivability, path.

In recent years there has been a lot of interest in autonomous vehicles of various types and sizes ranging from micro-sized aerial vehicles flying in very sparse environments to full-sized trucks and lorries driving in our crowded urban environments. Advancements in hardware such as battery technologies and low-powered processors and falling prices of sensors have allowed the creation of many new types of vehicles and uses for them.

Navigation of these vehicles depends on knowledge of the environment. For some applications and types of vehicles such as Unmanned Aerial Vehicles (UAVs), Global Positioning Systems (GPS) and ranging sensors such as sonar and infrared are sufficient to navigate safely since they can move in a straight line between way-points. However, ground vehicles of all types have the extra problem of terrain traversability.

One way of dealing with this problem is by providing the vehicles with maps of the area / terrain of where it needs to navigate but having pre-programmed maps of every place robots need to go is impossible. Also, UGVs can be used for the purpose of actually creating these maps by traversing previously unmapped or even completely unknown terrain and recording the environment using numerous extrinsic sensors such as cameras and laser sensors.

This project aims to implement and assess a method for quantifying the quality of the terrain around the ground vehicle by assigning a drivability rating.
1.1 Project Overview

In this project, we aim to implement an algorithm that the classifies the terrain around an Unmanned Ground Vehicle into drivable and undrivable areas and subsequently give an estimate of the surface roughness in the immediate path of the robot in order for it to be used as an indication of the speed at which the robot should move. The proposed approach for Drivability Index (DI), uses elements from the work of S. Thrun et al. [TMA06] on the Probabilistic Terrain Analysis System that was used to tackle the 2005 DARPA Robotics Challenge and borrows ideas the Roughness Index proposed by Wilson [WR14] and Roughness Mapping by Loh [LEC13] to extend it in order to make it suitable for the type of UGV and sensors involved in this project. Our platform is a rugged and much slower moving vehicle which does not have to follow visually-distinguishable paths, such as roads and footpaths, and is used as data collection platform for other experiments and projects such as pointcloud alignment and tree classification.

The most prominent differences between the aforementioned methods and ours is that we exchange the capability of the algorithm to produce terrain classification in far distances with the capability to estimate the roughness of what is considered drivable in the immediate vicinity of the robot. The roughness estimates are based on statistical tests which aim to reflect the significance of the data to suggest we can traverse it.

In this report, the ideas behind the implementation are explained and subsequently implemented and tested. For comparison, the Roughness Index metric which was part of the inspiration for this project is also implemented in order to form a baseline to compare our implementation against.

1.2 Objectives

- Understand the issues in terrain analysis.
- Examine methods developed for roughness identification for wheeled vehicles.
- Develop a solution for the available platform.
- Implement terrain classification and roughness rating.
- Evaluate the performance of the method.
- Examine the strengths and limitations of the proposed approach.
- Implement the algorithm in a real-time scenario.

1.3 Report Outline

The following chapters document the course of the project. Chapter 2 deals with refining the problem at hand, explaining relevant concepts and technical knowledge,
examining relevant literature and subsequently proposing a solution suitable for the needs of this project. Chapter 3 summarises the available hardware and software infrastructure, defines the design choices for the specific implementation of the proposed approach and subsequently examines the methodology adopted for the completion of this project, including the evaluation procedure. Chapter 4 describes the implementation of the proposed approach. Chapter 5 focuses on the evaluation and analysis of the results obtained for applying the approach on real data collected in the George Square area. In Chapter 6 is a summary of the work undertaken and results obtained and possible further steps for improving the approach based in the results. In the end, we outline the plan of the second part of this project.
Chapter 2

Background and Related Work

In this chapter, the background knowledge that was gathered in order to understand and realize the project is explained with a discussion on the different aspects of the problem of perception and analysis of terrain, based on the current literature and research. Subsequently we discuss the suitability of the identified methods and assess their theoretical capability to provide a relevant analysis for the terrain that our robot would have to traverse.

2.1 Definition of the problem and Terminology

In this section, we define the basic concepts relevant for this project and then refine the scope of this work.

The limited amount of research in the area of the roughness of the terrain for the purpose of driving means there is still a few terms that have been used to describe this type of analysis. In this text the terms drivability and traversability are used interchangeably while others such as navigability, manoeuvrability and mobility have also been used in literature to describe a vehicles moving capability in relation to some terrain.

2.1.1 Definition of Traversabality

Traversability is defined by George Papadakis [Pap13] as:

The capability of a ground vehicle to reside over a terrain region under an admissible state wherein it is capable of entering given its current state, this capacity being quantified by taking into account a terrain model, the robotic vehicle model, the kinematic constraints of the vehicle and a set of criteria which the optimality of an admissible state can be assessed.

As stated in the definition, the properties of the terrain and the properties of the vehicle should both be taken into consideration in order to assess this ability since it is defined as the interaction between the terrain and the vehicle.
2.1.2 Terrain and environment types

One major consideration in tackling autonomous navigation of ground vehicles is the variability of the environment. Navigating across structured and unstructured environments presents a number of different challenges ranging from detecting large natural stationary obstacles such as trees and large rocks or identifying moving ones such as humans walking around the robot to road and path segmentation and following. In recent years, there has been a lot of research on terrain perception of all types which is a direct result of the interest autonomous vehicles, especially for commercial applications. A stellar example of this is the automotive industry with numerous car manufacturers working to bring Advanced Driver Assistance Systems and subsequently complete autonomy to their vehicles.

The DARPA Robotics Challenge has been a major influence on the way we see autonomous ground navigation today with the early challenges presented to the contestants being the navigation across the Mojave Desert. The challenge ran in two consecutive years, 2004 and 2005, which found no contestants finishing in the first one and all but one finishing the second year.

Since then, a lot of research has been conducted on analysis of urban environments, which includes road detection and segmentation, lane detection, obstacle identification and tracking. The advancements in these areas have allowed technologies with industrial applications to be developed and have since found military and commercial applications, ranging from autonomous military trucks to Advanced Driver Assistance Systems in mainstream cars. The capabilities of these systems are very promising. Currently, many of the top companies in the automotive industry have stated that fully autonomous passenger vehicles will be on public roads before 2022.

While the current focus of the media is on commercially deployable technologies for urban environments, such as the aforementioned driver-less cars and indoor robots such as warehouse and house-cleaning robots, there are many applications which require traversal of unstructured and harsh terrains. Examples of such applications range from space exploration missions to land surveying and forest mapping. Tackling rough unstructured terrains presents a very different set of requirements than urban environments and subsequently requires a different approach.

2.1.3 Approaching Terrain Analysis

There are 3 main ways in approaching Terrain Analysis:

- Binary Classification - In the context of Traversability, that would be classifying terrain into Drivable and Non-Drivable.

- Multi-class Classification - Classification into discrete categories such as Grass, Gravel, Asphalt, Dirt and Obstacles.

- Regression - Assigning a numerical value such Roughness Estimation or confidence level of traversability to a piece of terrain.
2.2. Technical Background

The following information has been identified as relevant to the problem we are trying to solve. The physical details of the sensors as well as the interpretation of the data that is collected from them are important to the implementation of this project as they directly affect the performance and capabilities of the system.

2.2.1 Laser Based Sensors

One of the major advancements that have enabled this acceleration in the field of autonomous vehicles is the improvements and decrease in cost of LiDAR sensors. LiDAR is an acronym for Light Detection and Ranging and consists of a laser range-finding device that transmits light beams, in the form of laser, and measures the time it takes for them to return to the device. [Gig]. These sensors usually employ a single laser range-finding device (1 beam) and a mirror that rotates in some plane and reflects the beam in a circular pattern around the device. [Wik17] Because of physical construction restrictions, LiDAR sensors usually do not offer 360 degree scanning. To enhance their usability in autonomous vehicles, LiDAR sensors are sometimes mounted to spin in a plane orthogonal to the scanning plane of the device. Some devices are also able to measure the relative intensity of the returning beams. To reconstruct the

![Figure 2.1: Simple demonstration of how a LiDAR sensor works. The diagrams shows a laser ray scanning a plane as it bounces on the spinning mirror. Adapted from Wikipedia [Wik17]](image)
data gathered by the sensor at each pulse of the laser the angle at which the mirror in the LiDAR is and the pose of the whole device are needed. From this, we can deduce where in 3D space each beam bounced of. This project makes use of a Hokuyo UTM-30LX-EW the details of which will be specified later.

### 2.2.2 Image Sensors

The term Image Sensors is used to refer to all sensors that use the visible light spectrum to record data about the environment. There are many variations of imaging sensors such as colour or grey-scale, CMOS or CCD (the underlying light-intensity-measuring device) and of the resolution at which the sensor captures data. Monocular Vision is a term used to describe systems which only have a single camera and produce a 2-dimensional image from a single point of view at each instance. Stereo Imaging is a term used for pairs of images taken in the same scene from a known displacement of each other. By utilizing knowledge about the physical distance between the point-of-view of the two images and the underlying imaging model (pinhole model) a disparity image is calculated which is forms the base of the 3D reconstruction using epipolar geometry.

Figure 2.2: A visualization of epipolar geometry illustrating the relationship between the point $p$, the camera centers $c_0$ and $c_1$, the epipoles $e_0$ and $e_1$ and the corresponding image points $x_0$ and $x_1$. 
2.2.3 Proprioceptive Sensors

Proprioceptive Sensing refers to the ability to feel oneself. In the case of robotics this refers to the ability of a robot to sense its own motion, both relative to the world and within itself. In this project, proprioception is used in two ways. The position of the x,y position and yaw of the robot is determined using rotary encoders in the wheels. This can be augmented with Inertial Measurement Unit (IMU) data to estimate the 3-dimensional pose of the robot. A rotary encoder is also used in the LiDAR sensor can tell us which way the planar scanner is pointing.

2.2.4 Pointclouds and Heightmaps

Data collected or produced about the surrounding environment needs to be stored in some meaningful representation. Pointclouds are the most popular way to represent 3D objects and environments. Pointclouds are collections of points in 3D space. They have at least 3 attributes (x,y,z) specifying their position in space but can carry other attributes to model their information such as colour or uncertainty. Pointclouds can be organised into a few different logical structures. Some examples are storing them in a way that resembles a 2d grid, as voxels which are the 3-dimensional equivalent of pixels, octrees which are voxel-based tree structures for storing 3D data, or even left unorganised as a collection of points [Mea82]. This implementation of pointclouds should be independent of the algorithm but can play an important role in its performance because of the differences when querying the data-structure for occupancy, inserting new points and other operations. Generally, unorganised pointclouds are the most space efficient since they only hold a list of points, while organised pointclouds encode information about empty space so they occupy more space. Octrees are a middle ground as they compensate by recursively splitting space into voxels and therefore empty spaces are encoded in less detail than occupied spaces. As a result, organised pointclouds are faster to query for occupancy than unstructured pointclouds but are usually slightly slower to insert new points into. The selection of this representation ultimately depends on the density of our pointclouds and the insertion and queries that will be run on them.

Height-maps are an alternative representation method for 3D environments. Height-maps are simply augmented 2D grids which encode 3D space in an organised, fixed-spaced grid which have the advantage of being able to be stored in a matrix in a very easy to interpret representation. Each cell in the matrix represents some information about the given square. For example, it could hold a triplet such as (height, uncertainty, number of observations). The obvious downside of this representation is that of each x,y location we can only have one value / height. An example is shown in figure 2.3 where the bar height encodes the estimated height and colour encodes uncertainty of height.
2.2.5 Pose Estimation

A very important issue when considering analysis of the terrain/environment for any kind of vehicle is its physical position and/or motion over time. Pose estimation usually refers to the full information required to place the object in a 3D environment, that is its position and orientation. On each of the 3 axes we have both translation and rotation which means 6-Degrees-of-Freedom are required to fully specify a pose in 3D (x,y,z,roll,pitch,yaw)\(^2\).4

In order for the sensor’s data to be meaningful when accumulated over time it has to be mapped around the vehicle’s position as it moves. This is especially true for the LiDAR sensor as it can only record 1 beam at a time. This means that a reliable pose estimation is required if the sensor data is to be successfully integrated over time.
2.3 Related Work

In the past few years there has been a lot interest in Autonomous Vehicles and thus a lot of research around environment and terrain perception. Methods which aim to classify
the drivability of terrain of terrain currently fall under 3 main categories as identified by Panagiotis Papadakis [Pap13]:

- Proprioceptive
- Exteroceptive - Appearance Based
- Exteroceptive - Geometric Based

### 2.3.1 Proprioceptive Methods

Methods based on proprioceptive sensors tend to use the vibrations generated when the vehicle drives over terrain. The most popular formulation of the problem is classification of the terrain into discrete categories (such as asphalt or gravel) based on the vibration signature produced by an Inertia Measurement Unit (IMU) and fed to classifiers such as Support Vector Machines (SVMs) and Neural Networks (NN) to perform the classification. DuPont et al. [DuP+08] proposed an approach that uses the frequency domain of vibrations to classify terrain into 6 distinct categories (Packed Gravel, Loose Gravel, Sparse Grass, Tall Grass, Asphalt and Sand). The proposed system used 10 second intervals of vertical acceleration, the pitch rate, and the roll rate at 200Hz from an on-board IMU. Using data collected from the terrains mentioned, 831-feature vectors were constructed using Fast Fourier Transform (FFT) to extract significant signals which are then used to train a Probabilistic Neural Network. The system classified the terrain into the given classes with average accuracy reaching 90% but was shown to be susceptible to speed variations. Speed-independent approaches based on the same principles have been developed such the system proposed by Ward and Iagnemma [WI09] which models the car’s dynamics (using a quarter-car model) and combines it with vibration data to create feature vectors which are then used to train an SVM giving an average accuracy of 90% across all tested speeds. The most important drawback of proprioceptive methods is the fact that terrain is classified after we drive over it and therefore cannot be used proactively to avoid rough terrain. This means that while some action is possible such as slowing down when travelling over continuous rough terrain, these approaches cannot offer any useful information in advance especially in the presence of short patches of terrain that should be avoided. Also, the complication that arises is that the particular vehicles dynamics (suspension model and characteristics) has to be known and modelled in order to properly adjust evaluate the frequency of vibrations. This makes such approaches more complex to analyse and less portable.

### 2.3.2 Exteroceptive Methods

Appearance and Geometric-based methods are going to be examined as one since a number of approaches use one to enhance the other or even try to extract geometry from appearance based descriptors.
Laible, Khan, Bohlmann and Zell (2012)\cite{Lai+} have proposed a system to classify terrain into discrete categories such as gravel, grass and asphalt using a LiDAR sensors which is considered lighting-invariant. The system down-samples the pointcloud into a 2D grid and for each cell stores 11 features (maximum and standard deviation of height, minimum, maximum, range, median, mean and standard deviation of intensity, distance to sensor, angle of incidence to ground, number of points incident in the cell). Random Forests were trained on labeled patches of terrain and classification accuracy was tested against a camera based method. The results showed a clear advantage of the LiDAR based method under different lighting conditions, with True Positive Rates above 90% across 5 different light-settings and showcased that 3D LiDAR data and the simple features selected provided more information about the terrain than the much more elaborate and proven feature descriptors used by Khan \textit{et al.} \cite{KKZ11} (Local Binary Patterns (LBP), Local Ternary Patterns (LTP) and TSURF). While this approach gives very good results, it does not give any information about the geometric properties of the terrain which could impede the vehicles motion.

One of the most influential approaches in autonomous vehicle development was developed by Thrun, Montemerlo and Aron (2005). The approach was designed to tackle driving at high speeds in a desert environment in the DARPA Grand Challenge of 2005\cite{TMA06}. The success of this approach stemmed from not attempting to reconstruct an accurate 3D model of the world but instead attempts to model the uncertainty of each incident laser ray down-sampled into a compact 2D structure and perform statistical tests on the aggregated measurements and their uncertainties to label terrain into 3 distinct categories.

Let $X_l^i$ denote the value of the $X$ coordinate of a point collected indexed by $l$ and collected at time $k$ and similarly for $Y$ and $Z$.

- **Obstacle** - If there are two points $(X_l^i, Y_l^i, Z_l^i)$ and $(X_l^j, Y_l^j, Z_l^j)$ whose Euclidian X-Y distance to the 2D-point of interest $(X_q,Y_q)$ is less that some distance $\varepsilon$ and $|Z_j - Z_l|$ is greater than some distance $\delta$

- **Drivable** - If not an obstacle and there is at least 1 point within $\varepsilon$ distance from the 2D-point of interest $(X_q,Y_q)$.

- **Uncertain** - If no points are found within $\varepsilon$ distance from the 2D-point of interest $(X_q,Y_q)$.

Note that the points do not need to be captured in the same time index to be considered a valid witness of an obstacle and that the 2D grid mentioned above is created with knowledge of $\varepsilon$ and can therefore be adjusted using a grid-size of $\varepsilon/2$ so that all queries mentioned above can be resolved by only considering the 8 immediate neighbours of a cell. The labeling function is then performed as a statistical test on the aggregated data in the grid at some chosen level of confidence.

In their implementation, the LiDAR sensor is pointed forwards and down, scanning a single line on the terrain surface perpendicular to the cars direction of travel and the described labelling function was unable to provide stable results. To overcome this, the noise in the data acquisition was modeled by a first order Markov model:
\[
\begin{align*}
\left( \frac{x_k}{\Psi_k} \right) &= \left( \frac{x_k}{\Psi_k} \right) + \beta_k + \gamma_k
\end{align*}
\] (2.7)

where \( x \) and \( \Psi \) are the vehicles position and orientation at time \( k \). The asterisks denote the estimated pose of the vehicle, corrupted by noise over time \( \beta \) and momentary noise \( \gamma \). If Gaussian noise is assumed:

\[
\beta_k \sim N(\beta_{k-1}, B)
\] (2.8)

\[
\gamma_k \sim N(0, C)
\] (2.9)

and \( B \) and \( C \) are of the form:

\[
B = diag(\sigma_{xyz}^2, \sigma_{xyz}^2, \sigma_{\phi\theta\psi}^2, \sigma_{\phi\theta\psi}^2, \sigma_{\phi\theta\psi}^2)
\] (2.10)

\[
C = diag(\tau_{xyz}^2, \tau_{xyz}^2, \tau_{\phi\theta\psi}^2, \tau_{\phi\theta\psi}^2, \tau_{\phi\theta\psi}^2)
\] (2.11)

The parameters \( \sigma_{xyz}^2, \sigma_{\phi\theta\psi}^2, \tau_{xyz}^2, \tau_{\phi\theta\psi}^2 \) are tuned using a discriminative learning algorithm in such a way that they maximize the discriminative accuracy of the resulting labelling on pre-labeled data.

To extend the working range of the approach, the resulting labelling of 3D pointcloud is then used to project a quadrangle patch of drivable road in-front of the vehicle onto the image produced by the monocular vision system. This patch is used to learn a drivable-road model using Gaussian Mixture Models (GMM) in RGB space which is then used to classify the rest of the image regions into drivable or non-drivable. \[\text{[Dah+06]}\]

This method has the advantage of being able to label any type of terrain quite far in advance and also adapt to new terrains as the surface type changes. However it still does not deal with the problem of terrain roughness estimation.

Figure 2.5: Stanley’s PTA system. Left: Stanley’s system labels a drivable patch using 3D LiDAR data. Middle: Raw image from camera. Right: Drivable patch projected onto the 2D image. The system learns a GMM of what is drivable and classifies the rest of the image based on the learned model. Red and Blue indicate drivable and non-drivable terrain respectively.
2.3.3 Terrain Roughness Rating

Two approaches have been used as inspiration for this project which aim to rate the terrain traversability. Both approaches are geometric-based and try to provide numerical assessment of the terrain.

A simple Roughness Index (RI) metric was introduced by Wilson & Ramirez-Serrano [WR14] that intends to quantify the traversability of terrain from a pointcloud as follows:

\[ RI = \frac{1}{h} \sqrt{\frac{1}{n} \sum_{i=1}^{n} (e_i - \bar{e})^2} \]

This approach relates the standard deviation of the points within a square patch of terrain to the vehicle’s ground clearance to obtain an index for the roughness of that patch. The resulting index ranges from 0 to \(\infty\) but the proposed threshold for untraversable cells is 1. A major consideration for this approach is that it does not account for the differences between cells. This means that 2 cells which are adjacent, have low standard deviations on their own but very different average heights would be marked as smooth terrain. Furthermore, at small grid sizes (< 10 cm), smooth obstacles, such as a box, can be marked as smooth, traversable terrain, especially when aligned with the grid. On the opposite end, at large grid sizes (> 20cm) a lot of information would be lost and would result in missing terrain details and possibly completely miss obstacles.

A much more elaborate roughness mapping technique is proposed by Jonathan Loh, Elkaim, and Curry [LEC13] for NASA’s K-Rex rover. This will be referred to as Roughness Mapping. In this the 3D pointcloud is stored in a 2D grid representation and then uses both neighbouring-cell and within-cell attributes in the calculations to give a rating to the terrain.

Initially, for each cell the best plane and its normal is calculated. The fit is updated when new points are gathered.

\[ z = Ax + By + C \] (2.12)

Once the planes are fitted, the calculated normals are used to create a smoothness scale by computing the dot product of the normal at the given cell with the normals of neighbouring cells and summing over a specified window as follows:

\[ N = \frac{(-A, -B, 1)}{||(-A, -B, 1)||} \] (2.13)

\[ s_N = |N(x, y) \cdot N(x + i, y + j)| \] (2.14)

\[ s(x, y) = \sum_{x=-K}^{K} \sum_{y=-K}^{K} s_N \] (2.15)
This provides the main framework by which terrain smoothness is computed. In addition to that, in order to provide information about the terrain within each cell, 3 different metrics are used which aim to detect certain conditions.

First, the residuals from the plane fit are used to calculate a residual cost which is then used as a weighting factor for the importance of the given cell.

\[
\text{residualCost} = \max\left(0, \min\left(1, \frac{\log_{10}(\text{residual}^{-1})}{\log_{10}(0.5^{-1})}\right)\right)
\]  \hspace{1cm} (2.16)

Second, cell coverage is calculated for each cell using the determinant of the covariance matrix of the incident points in the x and y direction. The two eigen-values of \(\lambda_1\) and \(\lambda_2\) are extracted using Principal Component Analysis (PCA). This is used to give less weight to cells which do no have high coverage of points which could result in unstable plane fitting.

\[
\text{coverage} = \begin{cases} 0 & \text{if } \lambda_1 \leq \omega \text{ or } \lambda_2 \leq \omega \\ 1 & \text{else} \end{cases}
\]  \hspace{1cm} (2.17)

where \(\omega\) was selected experimentally.

Third, the slope of a cell is calculated as a ratio between the plane parameters and the pitch and roll thresholds of the vehicle.

\[
\text{slope} = \max\left|\frac{|A|}{\Phi_{\text{thrsh}}}\right|, \left|\frac{|B|}{\theta_{\text{thrsh}}}\right|
\]  \hspace{1cm} (2.18)

\[
\text{s}_{\text{slope}} = \begin{cases} 0 & \text{if slope} \geq 1 \\ 1 - \text{slope}^\eta & \text{else} \end{cases}
\]  \hspace{1cm} (2.19)

where A and B are the plane coefficients and \(\Phi_{\text{thrsh}}\) and \(\theta_{\text{thrsh}}\) are the vehicle roll and pitch thresholds respectively. Parameter \(\eta\) must be optimised for each application. In the presented approach, \(\eta\) selected experimentally such that the smoothness cost is 0.5 at 50° slopes.

The dot-product-based smoothness calculation would be unable to detect steps in the terrain, thus one further step is employed. The plane of the selected cell is projected into the centre of each of the neighbouring cells and a height difference \(\Delta\) between the projected centre height and height of the fitted plane at the centre of the neighbouring cell is recorded. Then the cost is calculated as:

\[
\text{s}_{\mu}(x,y) = 1 - \min\left[\frac{\Delta}{\delta}, 1\right]
\]  \hspace{1cm} (2.20)

where \(\delta\) is the vehicles obstacle clearance parameter.

At the final stage, total smoothness cost is calculated using the metrics mentioned above for a given cell in two passes. Initially a smoothness cost \(s_{\text{center}}\) for each cell is calculated based on said metrics of the cell and once the smoothness cost is established
for all the cells, a second pass is used to smooth the cost of each cell by according to some weighting factor $\alpha$ which determines the importance of the smoothness $s_{center}$ of neighbouring cells against the smoothness of the given cell. In particular, more weight is given to cells which have a good plane fit and slope and have sufficient coverage. The specific details of the weighting are not relevant to our implementation and are therefore omitted.

In summary, this approach combines a 5 different metrics to assess the traversability qualities of the terrain. The dot product of normals establishes changes in slopes between cells and their neighbours, while the 3 within cell attributes, $\text{residualCost}$, $\text{coverage}$ and $s_{slope}$ are used to detect cells that could be problematic either by being untraversable or unreliable for use in the roughness assessment. Finally the $s_{\mu}$ indicator is used to detect steps in the terrain that the normals of fitted planes would be unable to distinguish. The final roughness cost is in the range of $[0,1]$.

![Image](image_url)

Figure 2.6: Left: Digital Elevation Map (DEM) of the NASA Ames Marscape. The boxes labelled 1, 2 and 3 mark rough hillside, uneven terrain on on level ground and summit of hill respectively. Middle and Right: Results of the Roughness Mapping method with grid size 0.5m. The range of green to red marks smooth to rough terrain respectively

While this approach performs a much more comprehensive analysis of the terrain, some differences in applying this to a different UGV were identified.

1. LiDAR sensor used produces very dense pointclouds (700000 points/s vs < 50000 points/s for our sensor)

2. The speed of the Lunar Rover is quite low. (0.01 m/s vs 1 m/s for our robot)

The approach relies on the accumulation of points over time in the 2D grid to which planes are fitted. At faster speeds, we expect the accumulation process to become much more noisy and when using less dense pointclouds we expect the plane fitting process to be much more unreliable.

### 2.4 Motivation, Approach and Theory

This project presents an implementation of a simple, novel approach for estimating terrain roughness that borrows ideas from the 2 approaches identified in section 2.3.3 but also from well established work presented in section 2.3.2.
The main idea behind the algorithm presented is similar to that of Wilson & Ramirez-Serrano [WR14] in that it attempts to relate the vehicles obstacle clearance to the distribution of points in an area. It is also similar to Jonathan Loh, Elkaim, and Curry [LEC13] approach summarized in section 2.3.2 in that it sums over the neighbours of a square patch to create a Drivability Index (DI) map. To overcome the problems identified above we will use a probabilistic approach to identify the roughness/unevenness of the terrain and we will perform this identification locally, i.e. as soon as we have a sweep of the ground before updating the global map.

We attempt to achieve a similar outcome to the Roughness Mapping by using statistical tests to express our confidence in the terrain such that the rating given to each cell represents the probability/certainty that it falls within a certain measurable range from neighbouring cells which can be then related either directly to the height of the vehicle or otherwise used in the decision making logic of the vehicles.

We approach the problem by down-sampling the input pointcloud using ideas from Thrun, Montemerlo and Aron(2005) [TMA06] in order to create a 2D grid that would enable us to apply similar labelling as mentioned in 2.3.2. This provides us with a simple binary classification of drivable and undrivable areas. The statistical tests of obstacle labelling are then inverted to test the statistical significance of the data to suggest that the terrain corresponding to a cell falls within $\delta$ vertical distance from its surroundings by considering both the height and standard deviation of neighbouring cells of the 2D grid. For this purpose we will use the average probability of neighboring cells being within said distance as the Drivability Index (DI).

$$DI(x,y) = \Phi(s_{ground}(x,y); 0, 1)$$ (2.21)

$$s_{ground}(x,y) = \sum_{i=-K}^{K} \sum_{j=-K}^{K} \frac{\delta - |Z_{x+i,y+j} - Z_{x,y}|}{\sqrt{s^2_{x+i,y+j} + s^2_{x,y}}} \text{for } i \neq 0, j \neq 0$$ (2.22)

where $\Phi(s, m, \sigma)$ is the Cumulative Distribution Function up to $s$ for the Normal Distribution with mean $m$ and standard deviation $\sigma$, $\delta$ is the clearance parameter of our vehicle, $Z, s$ and $N$ is the mean height, standard deviation and Number of points respectively.

The resulting mapping of the terrain could either be expressed as a probability that the point in terrain falls within $d$-cm of it surrounding or each grid square could be given a rating $d$-cm which represents the height that we are confident squares fall within at a given level of certainty $p\%$.

A Global Drivability Index Map will then be updated such that each the newest rating has 50% weight.

$$s^{\tau+1}_{ground}(x,y) = 0.5s^{\tau+1}_{ground}(x,y) + 0.5s^{\tau}_{ground}(x,y)$$ (2.23)
By considering the height difference between cells in probabilistic manner, we aim to incorporate both the unevenness of the terrain and the slope of the ground expressed as the confidence that the terrain is within the vehicles capabilities. Furthermore, this approach should not present problems with stepped terrain and should be more robust to cell coverage presenting problems for the plane fitting which could arise from the nature of the LiDAR sensor producing lines of points.
Chapter 3
Design Process

This chapter describes the platform used to implement the project and subsequently documents the requirements that arise from both the intended outcomes of the project and the hardware available to implement it. At the end of the chapter, attention is shifted to the evaluation of the project, where experiments are designed to test the projects success both qualitatively and quantitatively.

3.1 Hardware Platform

3.1.1 Background

The wheeled robot and other hardware used in this project belongs to the Robot-Perception-Group at the UoE and is used as a platform to test work in the field of perception by members of the group and but is also used in other areas such as robotic hand kinematics as it is simplifies control of the platform compared to a humanoid platform.

3.1.2 ClearPath Husky UGV

The robot used is a ClearPath Robotics Husky UGV shown in figure 3.1. The relevant characteristics of the robot for this project are also summarised in table 3.1 but important to note are the vehicles size (990 x 670 x 390mm) and ground clearance (130mm) as well as the angles it can climb (45°) and traverse (30°) and its maximum speed (1 m/s).

3.2 Sensors

The sensors that are on the robot and are available to use are:
**Table 3.1: Relevant Specifications of the Clearpath Husky Robot**

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTERNAL DIMENSIONS (L x W x H)</td>
<td>990 x 670 x 390 mm</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>50 kg</td>
</tr>
<tr>
<td>WHEELS</td>
<td>330 mm diameter</td>
</tr>
<tr>
<td>GROUND CLEARANCE</td>
<td>130 mm</td>
</tr>
<tr>
<td>MAX SPEED</td>
<td>1.0 m/s</td>
</tr>
<tr>
<td>MAX CLIMB GRADE</td>
<td>45 (100% Slope)</td>
</tr>
<tr>
<td>MAX TRAVERSAL GRADE</td>
<td>30 (58% Slope)</td>
</tr>
<tr>
<td>DRIVERS AND APIs</td>
<td>ROS, C++, and Python.</td>
</tr>
</tbody>
</table>

1. Carnegie Robotics Multisense SL Stereo and LiDAR Combination - This is the main sensor package used and contains the Multisense S7 and Hokuyo UTM-30LX sensors. The advantage of this sensor is that: 1) The LiDAR cradle rotates orthogonal to the plane of scanning so it provides a 360° scan of the environment, 2) the transformation between the two sensor’s frames is well specified by the manufacturer.

2. Carnegie Robotics Multisense S7 - This provides the stereo images in a single package, which implies a very accurate disparity calculation, it has a Field Of View (FOV) of 80° x 45° and can produce a depth resolution of Depth Resolution +/- 0.3mm at 1m and +/- 30 mm at 10m.

3. Hokuyo UTM-30LX LiDAR Range Finder - This runs at 40Hz(40 lines/s), with an FOV of 270° and a range of 0.1m to 30m. Its angular resolution is 0.25° and its accuracy is +/- 30mm at 1-10m and +/- 50mm at 10-30m.


5. MicroStrain 3DM-GX3-25 IMU - a high-performance Attitude Heading Reference System (AHRS) which combines an accelerometer, gyro, magnetometer and temperature sensors to provide static and dynamic orientation and inertial measurements.

The Multisense SL is mounted on motorised Pan-Tilt-Unit which can be controlled electronically.

### 3.3 Software Platform

The current software platform uses the Robot Operating System (ROS) [Bad+16] and Lightweight Communication Marshaling (LCM) [HOM09] for intra-process communication. ROS is a tool developed for easy interaction with robots and is the operating system used by the robot and LCM was developed for handling message passing and data marshaling where high bandwidth and low latency is required. This project aims to use the already implemented software platform to add terrain classification and rating to the already existing features of the robot. The robot is used by the Robot
Perception Group at the University of Edinburgh and already has software to control it remotely and odometry from both vision and the wheels. In addition to that, the Gazebo [KH] physics simulator is used for testing, and Director [Mar+17], a robotics interface and visualisation software is used to view the outputs of the algorithms.

3.4 Algorithm Input/Output Specification

The project aims to firstly classify the terrain into drivable and non-drivable and for all drivable area estimate the probability that the area’s height falls within a specified vertical distance $\delta$ compared to its neighbours related to the capabilities of the vehicle it is used for. We aim to perform the classification locally rather than trying to align and overlap data from multiple sensors and/or consecutive sensor measurements of the
same terrain since our proposed approach uses the standard deviation of measurements. Doing this would introduce deviations in our data samples for each terrain patch which stem from imperfect transformations between them and imperfect modelling of the accumulating noise in the pose estimation of the robot as well as the momentary noise due to vehicle vibration which also increases with distance from the sensor.

This system should receive a pointcloud as input along with the details of the sensor that captured it, including pose and accuracy. This input pointcloud should be a single scan of the environment which in the case of the LiDAR sensor can be captured every half rotation of the Multisense (details will be covered in section 3.6) while our stereo camera is capable of producing pointclouds at a rate of 15Hz.

The output should be a labelled pointcloud which identifies traversable terrain labelled with the probability of being within the parameters mentioned above.

### 3.5 Classification Pipeline and Algorithm Design

According to the hardware and algorithm specification above the following steps were identified for implementing the pipeline to support the algorithm on the described platform:

1. Capture a pointcloud which contains a single scan of the environment/terrain.
2. Crop the pointcloud to a local frame around the robot
3. Crop the pointcloud to remove non-terrain points
4. Project the pointcloud to a 2D Grid with mean, standard deviation and number of points for each cell
5. Perform classification on the 2D grids
6. Project classification back to 3D pointcloud (both locally and globally).

### 3.6 Supporting Infrastructure

In order to realise the terrain classification and roughness indication procedures, some supporting functions had to be implemented. Under this section, the implementation of features that are not strictly part of the terrain classification procedure are listed while a more detailed description of the terrain classification and roughness estimation procedure is found in chapter 4. Where possible, functionality already existing in the software infrastructure for the robot was either reused or modified to suit the needs of this project. This section outlines the main supporting features that were either created or reused.
3.6.1 Robot Description, Coordinate Frames and Transformations

The physical model of the robot and its components is specified in Unified Robot Description Format (URDF). URDF files can be processed by ROS for calculations and by Director for visualisation. These were reused from previous projects as the robot is used in the same configuration. The robots coordinate frames are specified in configuration files which allow us to specify the coordinate frame of the different components relative to a parent component. Additionally, communication channels on which to listen for updates of this relative transformations can be specified and allow us to transform between the coordinate frames of moving components by publishing their pose updates on the relevant channels. In our context, all of the components are stationary in relation to the robots base except from the planar laser scanner rotating on an axis parallel to the Multisense’s facing direction. An existing framework called BotFrames is utilised which given these specifications, establishes and updates quaternions which allow us to easily transform between the different coordinate frames of the robot. Wherever possible, these transformations are used since they provide a very robust and convenient way to work with multiple coordinate frames.

3.6.2 LiDAR pointcloud accumulation

As specified in section 3.5, the required pointcloud should contain a single sweep of the terrain and can be generated every half rotation of the Multisense sensor. This is because the Hokuyo UTM-30LX-EW produces a planar scan with an FOV of 270° so it only has to rotate 180 degrees to scan the same area again.

Figure 3.2: The figure illustrates the FOV of the Hokuyo scanner in blue and the incident points in red. In the top left, the left image from the stereo sensor is provided as reference. As each scan covers a plane, the ground is scanned by both sides of the planar scanner when it rotates 360°. As a result we can generate a new sweep of the ground every 180° of rotation.

The Hokuyo sensor produces planar scans at 40Hz. This scan is published on LCM as a
vector of ranges, along with the initial angle in radians and the angular step from point to point. The state of the Multisense joint on which the Hokuyo sensor is mounted is published at 80Hz. The planar scan is transformed to a pointcloud using the equations described in section 2.2.6 using the existing laser utilities framework. The accumulation of the line pointclouds is performed using a circular buffer. The size of the circular buffer is calculated according to the rotation speed of the LiDAR. Since the rotation speed of the sensor is constant, this buffer is instantiated in the initialisation of the program. The angle of the multisense joint is also monitored and when its position is such that the scan plane is parallel to the ground, the accumulated pointclouds are joined in a single pointcloud.

This was implemented using two LCM message handlers which listen for messages from the Hokuyo planar scanner and the Multisense sensor. The planar scan handler continuously transforms planar scans to pointclouds containing single scan lines and pushes them at the end of the circular buffer while the other handler continuously monitors the position of the position of the relevant Multisense joint and when it crosses the parallel-to-ground position, raises a signal which indicates that the sweep of the sensor is complete and the single-line pointclouds are joined in a single pointcloud. Subsequently the terrain rating function is invoked.

### 3.6.3 Stereo Camera pointcloud reconstruction

The stereo reconstruction for the Multisense S7 sensor happens on-board the sensor and it is capable of outputting at 15Hz. The algorithm used for this reconstruction is Semi Global Matching [Hir]. The output received is a depth image is accompanied by the image of the left camera of the sensor in RGB colour space. This is then converted into a pointcloud using the provided utilities for the sensor. An example of a pointcloud produced by this sensor in shown in figure 2.3.

This is implemented using an LCM message handler which monitors the network for messages from the Multisense Stereo sensor. When the handler is invoked, the received depth image is converted into a pointcloud using the existing utilities. The resulting pointcloud is in the coordinate frame of the stereo sensor and needs to be transformed to the robots coordinate frame. For this purpose, the established quaternions are used. Once the transformation is complete, the terrain rating process is initiated.

### 3.6.4 Pointcloud cropping

As explained in chapter 3, the algorithm will perform classification locally, therefore the pointclouds generated are passed through a box filter. The box filter takes as parameters the size of the box and the robot’s pose to remove any points which do not fall in the box. The size of the box was experimentally determined. For our purposes a square box with side of 12 meters was chosen. In most cases, our pointclouds are not dense enough near the edges of this box. Also, the accuracy of the projected points of
3.7 Methodology and evolution of the system

For the completion of this project, an iterative approach was adopted. An initial version of the algorithm in which the labelling process above was first implemented and tested in simulation as detailed in section 3.8.1. The results and experience gained from the simulation were then used to adjust the system when implementing the final approach and for planning a second set of experiments on real data that was recorded by tele-operation of the robot. Further tests were undertaken in simulation which involved clear cases of uneven terrain.

Due to the difficulty in finding or generating labelled data which involves terrain roughness, calibration of the parameters of the system was performed by manually observing the output of the algorithm and recording its failures on experiments performed using real data. More specifically, the experiments were performed on data recorded by tele-operation, the algorithm was ran at different parameters for the terrain clearance threshold and subsequently produced Receiver Operating Characteristic Curves in order to establish a suitable value for the traversability parameter for our application. The experiment is detailed in section 3.8.2.

The results from these experiments displayed an ability to distinguish between different terrains but in the data that was collected the robot was mostly driven on relatively smooth paths that were also quite wide compared to the robots width and subsequently a more suitable dataset was required.

For the final evaluation of the algorithm, we tele-operated the UGV in various conditions, ranging from smooth paths surrounded by grass to locations in which we deemed as the maximum traversability capability of the robot. The resulting data was then
hand-labelled at the level of individual cells. Details of the labelling process will be outlined later in this chapter.

3.8 Experiment Design

In this section the different experiments that were run are listed along with the reasoning behind them, the method that was used and the objectives we had for each.

3.8.1 Experiments in Simulation

The initial experimentation for the algorithm was performed in simulation and had various purposes. The first batch of simulation tests was designed to check that all the inputs of the system are processed correctly, accumulation of data works as expected and that the basic building blocks of the proposed algorithm will produce the expected results.

The algorithm was tested on a few of artificial environments each created to evaluate some aspect of the algorithm.

The first tests were performed to test the ability of the labelling strategy adopted from PTA to label the terrain into the 3 discrete categories (Obstacle, Drivable and Unknown) as described in section 2.3.2. The figure shows the resulting labelling from 2 experiments performed on a curved turn and a sharp 90 degree turn.

Figure 3.3: Left: Simulated environment in Gazebo. Right: labelled Map using the rules from section 2.3.2. Grid-cell width is 13cm. Labels are 1) Green: drivable cells, 2) Blue: Obstacle cells with more than 5 points, 3) Red: Obstacle cells with 2 to 4 points, 4) Black: Unknown with less than 2 points

Further experiments in the same setting were conducted to see how the proposed system responds to artificial uneven terrain. To do this, small objects and obstacles of various shapes and sizes were placed in the simulation to examine that response of the proposed system.
3.8. Experiment Design

Figure 3.4: Simulated Rough Terrain. Left: Simulated environment in Gazebo with 3 simulated patches of rough terrain with rock-like structures 3-8cm height. Right: labelled Map as explained in section 2.4. Grid-cell width is 13cm. Labels are 1) Red: Obstacle cells 2) Yellow-Cyan-Blue: DI of cell ≃ probability of having 3cm average height difference with surrounding cells 50%, 75% and 100%. 3) Black: Unknown with less than 2 points

The simulated rough terrain in figure 3.4 is clearly outlined for the first patch. The middle section is marked as more drivable than the outline. For the second and third patches the patches appear as drivable but with lower certainty. The edge of the patches further from the robot appear rougher than the leading edge because of the lower number of points incident due to occlusion.

Figure 3.5: Testing using simulated objects. Left: Simulated environment in Gazebo with cans in the path of the robot. White cans are 7cm height, Red is 12cm). Right: labelled Map as explained in section 2.4. Grid-cell width is 13cm. Labels are 1) Red: Obstacle cells 2) Yellow-Cyan-Blue: DI of cell ≃ probability of having 3cm average height difference with surrounding cells 50%, 75% and 100%. 3) Black: Unknown with less than 2 points

Figure 3.5 shows the output of the approach when presented with stationary objects. The purpose of this experiment was to see how the system performs when presented with small objects which only cover part of the grid cells. The output shows that the system could recognise the taller can as an obstacle while the others were marked as rougher terrain.

For these tests the following outcomes were observed:
1. All barriers in the scene and in the range of the sensors were be marked as obstacles
2. All flat plane visible by the sensor was marked as drivable
3. The output value over the rough patches is indicative of the roughness of the terrain

The results of this experiment showed that the system was able to recognise the features that were created, however in the real world we expect the sensor input to be much noisier.

### 3.8.2 Testing and calibration on real data

The second experiment was performed on data logs collected from the area around the Informatics Forum, George Square and surrounding areas. The logs were collected by tele-operating the Husky UGV and capturing the output of all sensors. This round of experiments was designed to both assess the performance of the system on real data and to be used as calibration dataset for tuning the traversal height parameter of the algorithm for the UGV. These tests were performed for with the LiDAR spinning at $\pi$ radians/s and $\pi/2$ radians/s.

The results were aggregated based on the following methodology: For each sweep of the ground generated by the LiDAR 2 binary indicators were produced:

1. Whether all the visible road terrain in the scene around an area of 3 meters around the robot is classified as road and
2. Whether all non-road terrain in the aforementioned area is marked as untraversable.

The resulting data is then counted and divided by total number of scans we perform. This metric was implemented to approximate of the True Positive classification rate (TPR) also known as Sensitivity and the False Positive Classification rate (FPR) also known as Specificity of the algorithm. The aim of this was to calibrate the traversability height threshold $\delta$ at which the statistical tests will be evaluated at. This parameter should be related to the vehicles traversability capabilities but also to how harsh terrain we would like the robot to drive over.

The algorithm was also visually evaluated using the methods mentioned above by projecting the classification back to a pointcloud and also projecting it back to 2D and overlaid onto the images produced from the Multisense Camera in order to make misdetections and misclassifications more clear.

The following objectives were set for this experiments:

- Establish whether the algorithm performs as expected using real data
- Examine the effect of running the algorithm with LiDAR spinning at $\pi$ radians/s and $\pi/2$ radians/s.
• Determine an appropriate value for the vehicle traversability height $\delta$ for use with the Husky UGV

The tuning was performed based on the ROC curves presented in figure 3.6.

![ROC curves for traversability height at $\pi$ rad/s and $\pi/2$ rad/s](image)

Figure 3.6: ROC curves for traversal height between 1 and 6 cm with the LiDAR spinning at $\pi$ and $\pi/2$ rad/s

In the experiments above, a height of 3 cm provided the best trade-off for misclassifications of obstacles and recall of road. Also at 3 cm and higher, no significant differences were observed between spinning the LiDAR sensor at $\pi$ and $\pi/2$ rad/s. At 2 and 1 cm, when LiDAR spun at $\pi$ rad/s gave higher False Positive rate than when it spun at $\pi/2$ rad/s. Thus, a value of 3 cm was selected as the traversability height for the Husky UGV for this approach.

### 3.8.3 Design Evaluation

As the first set of experiments were run on datasets where the robot was only moving on relatively smooth and wide paths, roughness deviation in terrain roughness was minimal. The final experiments were run on data collected within George Square, which was collected to highlight the strengths and weaknesses of the algorithm. The data was recorded as raw messages from the sensors using the LCM message logger for processing offline. In this dataset the robot was driven in paths with boundaries (e.g., from grass onto asphalt) where the robot would need to slow down to gracefully traverse and paths that are bounded by either a high curb or vegetation. More specifically this area was chosen as it contains the following terrain features:

- Asphalt bounded by curb, high or low vegetation or grass
- Smooth gravel paths bounded by vegetation and grass
- Smooth dirt paths bounded by high vegetation and grass
- A dirt path with protruding roots and rocks

To evaluate the performance algorithm, in the final experiments, the following algorithms were ran and compared:
1. The proposed algorithm with the LiDAR pointcloud as input
2. The proposed algorithm with the stereo pointcloud as input
3. The Roughness Index the metric proposed by Wilson as described in 2.3.3

For these experiments, the terrain was manually labelled to provide ground truth against which the algorithms’ error is compared using the following metrics:

1. The percentage of the terrain cells labelled as traversable that the algorithm is able to recover.
2. The percentage of the terrain cells labelled as obstacles / untraversable that the algorithm marks as drivable
3. For all cells labelled drivable which the algorithms also mark as drivable, the average deviation from the labelled roughness.

3.9 Terrain Labelling and Process

To enable evaluation of the proposed approach, the data that was most recently collected was manually labelled. This data was collected from George Square and labelled to reflect our own perception of the roughness of the terrain, having in mind the capabilities of the robot as well as the speed at which we would opt to drive at if we were driving the robot while always being on the cautious side.

The labelling process was applied for a run around the paths of the southern entrance of George Square. This area was selected for the various terrain features that it covers. A description of the area as well as photographs for reference are provided alongside the evaluation in chapter 5.

The manual labelling of the terrain was performed by projecting the colour data from the pointcloud produced by the stereo reconstruction to the same 2D grid as we use for terrain analysis. The labelling was performed by observing the terrain and deciding at which speed we, when operating the vehicle, would traverse it. High-resolution pictures taken on the day of the collection and the left image of the stereo sensor were kept and used for visual reference. The grid was then annotated with the following values:

1. 0 if the terrain is unknown
2. 25 if the terrain is deemed untraversable
3. 125-255 if the terrain is deemed traversable, with higher values indicating smoother terrain.
3.10 Identified Difficulties in Evaluation

The main difficulty that was encountered was with creating ground truth for the purposes of evaluating the performance of terrain roughness estimation as absolute ground truth for terrain roughness is unattainable. The labelling process described above was performed with the intention of providing accurate cell-wise comparison of the algorithms. The comparison as explained in section 3.8.3 was made possible but the actual numerical roughness rating labelling was very hard to produce to a satisfactory and indicative level of accuracy and precision. As per the definition of Papadakis stated in section 2.1.1 it is also relative to the capabilities of the given vehicle which means that any comparison undertaken only reflects the specific implementation and our intuition of the capabilities of the vehicle in relation to the terrain. In order to account for these difficulties, we also evaluate the final roughness estimation visually as this is the most common approach taken in literature. An example graph produced showing the average deviation of the algorithms from the labelled values is provided as an appendix.

3.11 Summary

In this chapter we have summarised the software and hardware platform that is used for this project as well as the supporting infrastructure developed for the completion of the task. We also outline the major requirements for the implementation and evaluation of the proposed approach.
Chapter 4

Implementation

This chapter documents the implementation details of the proposed algorithm as designed in the previous chapter as well as the pipeline build to support its execution evaluation.

4.1 Prepossessing

The first step in performing terrain classification and roughness rating is applying steps 1 to 4 from section 3.5.

One of the main differences of our approach to the related work in section 2.3.3 is that in order to preserve as much detail of the terrain as possible and avoid reliance on aligning consecutive pointclouds, the classification is performed for single scans of the terrain. For the LiDAR sensor, a rotation speed of $\pi/2$ was selected. This gives a good balance of the interval of between terrain scan and the number of points in cells. Furthermore, the LiDAR input was filtered so that any of any points that were incident on the robot and the equipment are removed. For stereo input, each pointcloud contains a full scan of the upcoming terrain but for the purposes of evaluating the proposed algorithm as a measure of terrain Drivability, local classifications are performed at the rate that the LiDAR sensor can produce terrain scans.

The next step of prepossessing required is to extract the region of interest required from the local classification. For this implementation, a square region of 12m width with the robot in the centre of the frame was chosen. This was selected as an appropriate size for our application based on the accuracy and range of the sensors and the speed of the robot. This was implemented using the Point Cloud Library (PCL) [RCII] BoxFilter and pose estimation from wheel odometry.

The final prepossessing step is projecting the pointcloud to 2D grids as explained in section 3.5. The mean and standard deviation of points is calculated incrementally. Each of the points is measured as a sample with its own standard deviation and the total standard deviation of a cell is calculated using the incremental weighted standard
deviation formulae [Fin09]. For both sensors, the maximum accuracy deviation value state by the manufacturers of the sensors of 30mm at 10meters was used.

4.2 Terrain Classification and Drivability Index

The main contribution and novel aspect of this project is the implementation of simultaneous terrain classification and roughness estimation by applying the terrain labelling function from PTA and inverting the statistical tests used in the PTA algorithm explained in section 2.3.2 to establish a metric of terrain roughness. The proposed approach is described in this section and outlined in algorithm 1 which is composed of 2 main sections. Lines 3 to 22 describe the process by which the DI value of cells in calculated while lines 23 to 30 describe how the final grid is labelled using the results of lines 3 to 22.

Once the preprocessing steps have been applied, the first step in rating the terrain is applying the terrain labelling function as in section 2.3.2. Therefore, for each cell and each of its neighbouring cells (loop in lines 7-21) in the local frame, the statistical T-test is performed between the two cells in the local frame in order to establish the probability that the cell height means differ by more than the obstacle threshold height \( \delta_1 \) (Lines 12-17)

\[
\text{ObstacleWitnessProbability}_n(c_{x,y},c_{i,j}) = 1 - \Phi(T - \text{score}_n(x,y,i,j);0,1) \quad (4.1)
\]

\[
T - \text{score}_n(x,y,i,j) = \frac{\delta_1 - |Z_{x,y} - Z_{i,j}|}{\sqrt{s_{i,j}^2/N_{i,j} + s_{x,y}^2/N_{x,y}}} \quad (4.2)
\]

where \( \text{ObstacleWitnessProbability}_n(c_{x,y},c_{i,j}) \) is the probability that cell \( c \) at location \( x,y \) differs by more than \( \delta_1 \) in height with a neighbour, \( \Phi(s,m,\sigma) \) is the Cumulative Distribution Function up to \( s \) for the Normal Distribution with mean \( m \) and standard deviation \( \sigma \), \( \delta_1 \) is the obstacle height parameter for the given application, \( Z_{x,y} \), \( s_{x,y} \) and \( N_{x,y} \) is the mean height, standard deviation and Number of points for cell location \( x, y \) respectively.

The T-score above is converted into a probability using the Cumulative Distribution Function for a Normal Distribution with mean 0 and standard deviation of 1. An extra constraint placed, is that all cells have to have at least 2 points incident in order to be considered valid (line 11).

Furthermore, parameters \( \text{obstacle Acceptance Probability} \) (line 18) and \( \text{Obstacle Witness Threshold} \) (line 23) are used to specify the confidence requirement for admitting obstacle witnesses and the number of total obstacle witness cells needed to label the cell as an obstacle. For our application the \( \text{Obstacle Acceptance Probability} \) was set at 80% and \( \text{Obstacle Witness Threshold} \) was set at 2 such that 2 cells both having a
4.3. Updating the Global Map

The local drivability rating needs to be incorporated in the global map of the environment in order to be used effectively for navigation. We implement this by adding a point to the global Drivability Index pointcloud representing the map of our environment for each cell of the local 2D grid that has a valid classification or rating. Cells which do not have a valid rating due to not meeting the minimum number of points required are omitted from this process.

After analysing the local ground sweep and adding all the required points to the global Drivability Index pointcloud, a Voxel Grid filter is used to down-sample the pointcloud in order to approximate a time decay function for overlapping roughness estimations.

\[
\bar{s}^{\tau+1}_{\text{ground}}(x,y) = 0.5s^{\tau+1}_{\text{ground}}(x,y) + \bar{s}^\tau_{\text{ground}}(x,y)
\]  

(4.5)

where \(\bar{s}_{\text{ground}}(x,y)\) is the DI value of cell \(x,y\) in the global Drivability Index map and the \(s_{\text{ground}}(x,y)\) is the new DI value for cell \(x,y\).
Algorithm 1: Terrain Classification and Drivability Index

Result: Local Terrain Classification and Drivability Index (2D Grid)

Input: Local 2D Grid with Mean Heights (MH), Standard Deviation (STD) and Number of Points per cell (NP)
Obstacle Height $\delta_1$
Vehicle Traversal height $\delta_2$

Initialize:

$\text{OUT} = \text{out}_{x,y} = 0$  
$x - \text{dim} \leftarrow [0..\text{width}(MH)]$
$y - \text{dim} \leftarrow [0..\text{length}(MH)]$

for $x$ in $x\text{-dim}$ do
    for $y$ in $y\text{-dim}$ do
        $h \leftarrow \text{mh}(x,y)$
        $\text{neighbourCount} \leftarrow 0$
        $\text{obstacleWitnesses} \leftarrow 0$
        $\text{groundScore} \leftarrow 0$
        for $i$ in $[-1..1]$ do
            for $j$ in $[-1..1]$ do
                $nx \leftarrow x + i$
                $ny \leftarrow y + j$
                if $\text{np}(x,y) > 1$ and $\text{np}(nx,ny) > 1$ then
                    $nh \leftarrow \text{mh}(nx,ny)$
                    $s_1 \leftarrow \text{std}(x,y)/\text{np}(x,y)$
                    $s_2 \leftarrow \text{std}(nx,ny)/\text{np}(nx,ny)$
                    $z \leftarrow (\delta_1 - |h - nh|)/\sqrt{s_1 + s_2}$
                    $g \leftarrow (\delta_2 - |h - nh|)/\sqrt{s_1 + s_2}$
                    $\text{obstacleScore} \leftarrow 1 - \Phi(z;0,1)$
                    if $\text{obstacleScore} \geq \text{obstacleAcceptance}$ then
                        $\text{obstacleWitnesses} += 1$
                    end
                    $\text{groundScore} += \Phi(g;0,1)$
                    $\text{neighbourCount} += 1$
                end
        end
        $\text{groundScore} \leftarrow \text{groundScore}/\text{neighbourCount}$
        if $\text{obstacleWitnesses} \geq \text{obstacleWitnessThreshold}$ then
            $\text{out}(x,y) \leftarrow \text{MarkObstacle}$
        else if $\text{groundScore} > 0$ then
            if $\text{groundScore} > \text{GroundAcceptanceThreshold}$ then
                $\text{out}(x,y) \leftarrow \text{groundScore}$
            else
                $\text{out}(x,y) \leftarrow \text{MarkObstacle}$
            end
        else
            $\text{out}(x,y) \leftarrow \text{MarkUnknown}$
        end
    end
end

return $\text{OUT}$
The Voxel Grid Filter operates by segmenting the pointcloud into voxels for a chosen size and replaces all the points in the voxel by one point which has the average position and value of the points in the cell [PCL11]. In this implementation, the voxel grid filter is applied with a voxel size equal to the size of the 2D grid cells. As this operation is applied on every iteration of the algorithm, the final Drivability Index map has the same resolution as the 2D grid of choice.
Chapter 5

Results

In this chapter, results from running the algorithm on data collected in the George Square are presented and evaluated. The data was recorded in the form of LCM message logs and the logs were processed offline in real-time. For the purposes of this evaluation, as the output rates and modes of the stereo sensor and LiDAR differ, the 3 algorithms were run concurrently in order to obtain simultaneous and thus perfectly synchronised output. The Roughness and Drivability Index maps presented are visualised by colouring the original points of the respective pointclouds.

5.1 Evaluation around George Square south entrance

This experiment was performed on data collected around the south entrance of George Square. The terrain incorporates a representative sample of features present in outdoor environments including dirt and gravel paths with both steep and very shallow boundaries as well as a rough section with tree roots protruding from the ground.

The above area will be examined in sections since each has different features which highlight where the algorithm succeeds and where it fails. Figure 5.1 provides reference for the area as well as a subdivision into smaller areas that will be referred to in the analysis for each section.
Chapter 5. Results

5.1.1 Area 1

The section consists of a path approximately 1.4m wide and is bounded on the left side (as seen in figure 5.1) by vegetation that is in the range of 20cm high while the right side has a natural curb ranging from 10cm in the far distance fading to just a couple of centimetres where the bottom left photograph in figure 5.1 was taken. The right curb also eases into a slight slope with some low vegetation as we move away from the path. This section was selected in order to evaluate the capabilities of the algorithm in geometry-based path segmentation.
Figure 5.2 shows both algorithms identifying the variation in roughness for each of the features mentioned. Looking at the individual resulting Drivability Index maps in more detail, our proposed algorithm gives a very well defined outline for the path, identifying both boundaries of the path clearly from the rest of the areas. The RI metric seems to mark the boundaries and anything beyond them as untraversable while DI marks the areas as drivable but with very high roughness. DI gives the centre of the path a much lower roughness rating than than the RI approach and also labels the sloping grass bank on the right of the path (as seen in Fig. 5.1) as being drivable but much rougher than the path. More specifically for our approach, the middle of the path is given a value very close to one, the right bank which is comprised of low grass a couple centimetres high is marked at around 0.55 while the boundary between them in area 1.1 which stands at around 6-10 cm is labelled as untraversable (DI = 0) and in area 1.2 where the natural curb boundary smooths out to a few centimetres is marked.
with the lowest traversable rating of 0.5 at the highest curb going up to 0.7 at the lowest point.

The results are similar for both the LiDAR and Stereo but the latter provides a much denser point cloud which results to higher confidence for the flat parts while the rougher sections are not given a better rating since our confidence for the terrain being within the given height does not increase with the density of the points but just strengthens our belief that the area is rough.

RI labels the path with values around 0.5, the area on the right around 0.4 and the boundary between them around 0.1 but details of the features seem to be lost.

Figure 5.3: Road Recall rate for the individual local classifications and ratings for the path in the bottom left picture of figure 5.1. Lines (Left Axis): Road Recall Rate, Bars (Right Axis): Total Number of Cells that were labelled drivable. The interval between the classifications is approximately 2 seconds.

Figure 5.3 shows the performance of the individual local classifications at identifying cells we have labelled drivable before they are incorporated in the final drivability map. Across the 50 second stretch, 25 slices are labelled and at each one the number of cells we have labelled drivable that the algorithm was able to classify as correctly is reported. Our proposed approach consistently has a much higher recall rate with an average of 79.5% using Stereo, 58.0% using LiDAR and just 40.8% for RI using Stereo.
Figure 5.4: Obstacle Misclassification Rate for the individual local classifications and ratings for Area 1. Lines (Left Axis): Obstacle Misclassification Rate, Bars (Right Axis): Total Number of Cells that were labelled drivable. The interval between the local classifications is approximately 2 seconds. In this context, lower values represent better performance.

Figure 5.4 shows the number of cells we have labelled as obstacles and the percentage of those that the 3 different methods mark as drivable for the same section as figure 5.3 but with the last few seconds cut off as they do not have enough obstacle cells ($\leq 10$) to provide any significant statistic. Our approach has an average of 18.4% using LiDAR and 34% using stereo while RI has an average of 11%. Figure 5.5 shows instances of local classifications of the 3 algorithms at around 12,20 and 30 seconds in the path corresponding to the x-axis of figure 5.4.
Chapter 5. Results

Figure 5.5: Example local classifications at approximately 12, 20 and 30 seconds (top to bottom). From left to right: Image from left stereo camera, generated ground truth, RI, DI(Stereo), DI (LiDAR)

For our approach, figure 5.5 reveals that the high obstacle misclassification rate is partly due to the top of the vegetation of in the left side (as seen from figure 5.1) of the path being relatively flat and partly due our labelling since the grass on the right side is quite smooth beyond the path boundary. This is a problem with the chosen evaluation procedure rather than the algorithm itself.

For RI, we can see that the effective range of the algorithm is about half of that of our approach on the same input which can partly explain the difference between the two approaches in both figure 5.3 and 5.4.

Additionally, for our approach, we observe that areas we have labelled as obstacles which are marked as drivable are disconnected from the drivable areas as their boundaries are correctly identified as untraversable. Therefore, for the purposes of traversability, such misclassifications are unlikely to present a problem in navigation since these areas will be out of reach. However, since those areas are actually traversable, they only present a problem to the numerical evaluation of our system.

5.1.2 Area 2

Next we examine the results of the algorithms on a relatively flat section, labelled area 2 in figure 5.1. This area was chosen in order to evaluate how the system copes with terrain at a longer range. As it is mostly obstacle free, for this section, we omit obstacle misclassification rates.
5.1. Evaluation around George Square south entrance

Figure 5.6 shows the final Drivability Index map and RI ratings for area 2. The two images representing the Drivability Index maps show that our approach assigns relative roughness values which reflect the features of the terrain showing noticeable differences between the flat sections and the slope in areas 2.2 and 2.3 and the grass in area 2.1 and the top of area 2.2. The Drivability Index map also clearly indicates the metal fence and high vegetation on the left side of the images as two separate obstacles. For the RI approach all the features appear blurred with the surrounding cells and for most of this section all the area directly in front of the robot in a range of approximately 2 meters is labelled as traversable. The metal fence and vegetation are not distinguishable on the output.

Looking at the individual terrain features, highlights the capabilities of our approach. The sloped grass edge in area 2.2 is given a low but traversable DI of 0.6 both when using stereo and LiDAR inputs but as soon at the slope fades into the grass, DI is higher at around 0.75 but noticeably lower than the gravel path with a DI close to one. This is both expected and desirable as we expect the standard deviation of points to increase slightly on the grass. In reality, the grass might provide a smoother traversal for a UGV but it is very hard to reason about this using geometric properties. For RI, none of the terrain features are clearly distinguishable. Apart from the traversed path which is marked as traversable due to being near the UGV, most areas are marked as untraversable. The grass in area 2.1 and 2.2 is labelled as rough as the slope leading to it. This can be attributed to the fact that the standard deviation of points increasing as distance from the sensor increases as well as being higher for grass than flat terrain. DI suffers less from this effect as the statistical test used is performed on the height of cells.

The difference between the LiDAR and stereo input is also noticeable. While with both inputs most terrain features are rated appropriately, the higher density of the stereo input means that for the smoother sections our confidence in height differences is higher, reflected by almost a perfect score, without changing our belief for the roughness of the harsher terrain. Additionally, the physical properties of LiDAR sensor used imply that we have less points incident in cells not directly in front of the robot as the plane of the Hokuyo sensor moves from a vertical to a horizontal position which is reflected by
the lower average road recall rate. Figures 5.7 and 5.8 examine the local classifications across time in an effort to see why these differences are observed.

![Figure 5.7: Road Recall Rate for the individual local classifications for area 2. Lines (Left Axis): Road Recall Rate, Bars (Right Axis): Total Number of Cells that were labelled drivable. The interval between the local classifications is approximately 2 seconds. In this context, lower values represent better performance.](image)

In this flat section, RI has a much lower road recall rate than DI. This confirms that RI values are affected by distance from the sensor as there is more traversable area further from the UGV which it classifies as untraversable. Looking at the difference of road recall as we go from a $\delta$ of 3 and 4 cm, we observe that an increase in $\delta$ of 1cm has a relatively constant effect across the local classifications for RI. On the other hand, DI seems to be more robust to effects of distance from the sensor but also of the slight increase in traversability parameter.

![Figure 5.8: Example local classifications at approximately 63, 86 and 107 seconds (top to bottom). From left to right: Image from left stereo camera, generated ground truth, RI, DI(Stereo), DI (LiDAR)](image)
5.1. Evaluation around George Square south entrance

Figure 5.8 shows 3 examples of local classifications and ratings. The RI is only able to label drivable terrain in a range of approximately 2 meters while DI is able to do so for around 4 to 5 meters using either of the two inputs. The effect discussed above for the spinning LiDAR is also noticeable. The last column of images shows that the LiDAR does not generate enough points towards the sides so the algorithm can only classify and rate terrain in front of the robot.

5.1.3 Area 3

Next we examine the results of the algorithms on the roughest section of our dataset, indicated area 3 in figure 5.1. This area starts with a patch about 80cm long with tree roots protruding from the ground, labelled 3.1. Their height ranges from around 2 to 5 cm. On the right of the roots there is a slightly less rough section within area 3.1. After that, areas 3.2 and 3.4 are quite flat and accessible but the middle area (3.3) has some roots on the sides, as well as 2 bush stumps protruding around 8 cm from the ground. Area 3.5 is quite smooth in the middle and the vegetation on both sides is around 4 cm high at the boundary with the road but slightly higher as we move further.

![Figure 5.9: Final Classification and Roughness Indication in area 3 of figure 5.1. Top Left and Top Right: DI using Stereo and DI using LiDAR, Bottom: Wilson RI using Stereo data. Pink line in the first image is the path followed.](image)

For this section, the difference between DI and RI is even more pronounced. Figure 5.9 shows the final mapping from the 3 runs. On top, DI using stereo as input produces a very clear and easy to interpret roughness rating for the whole section. Area 3.1 is given a ratings of around 0.5 at the roughest points to around 0.7 at its smoother sections on the sides. For the stereo DI map we can also identify that the middle section of this area is the roughest with a rating of 0.5 and that the right side \((DI \approx 0.85)\) is smoother than the left \((DI \approx 0.70)\) which reflects what can be seen in figure 5.1. Areas 3.2 to 3.4 are also representative of terrain. At the boundary of area 3.3 and 3.4
the roughness increases as there are some roots and the slope of the terrain is changes near the tree. On the left side, roughness is also higher due to roots and the stump ($DI \simeq 0.5$), although we would ideally want to see the stump marked with a DI value below 0.5 as it stands at around 8 cm from the ground. However, with a diameter of less than 5 cm, this is not surprising as we expect a large number of points of the dense pointcloud to be on the ground. Area 3.5 appears mostly smooth as expected with DI close to one. The slightly rougher DI of around 0.5 at the top is due to a patch of grass, around 2 cm high and slightly convex. In general, the results for this part are very indicative of our perceived roughness of the terrain.

In the top right of figure 5.9, the DI map using LiDAR as input also produces results indicative of the terrain but appears to be slightly less detailed as the whole area is marked as slightly rougher when compared to using stereo input. All features can still be identified but are not as clearly defined. Area 3.1 still appears rougher in the middle and smoother on the left rather than the right with the roughest sections in the very centre of the path being marked as untraversable. Area 3.3 appears to be smoother on the right with values around 0.5 and the left side which contains a stump labelled with quite low values ($DI \simeq 0.35$) with a few cells marked untraversable around the stump that should have a higher DI. Area 3.2 is classified as untraversable which is not representative as apart from some thin vegetation it is quite flat and within the abilities of the UGV. This is probably due to the low side density of LiDAR pointclouds combined with the thin vegetation. Areas 3.4 and 3.5 are rated with representative values. The stump in the right side of area 3.4 which on stereo input is marked traversable with DI of 0.5 is now marked correctly as an obstacle. In general, the results using LiDAR as input are representative of the terrain but compared to stereo input, some of the detail appears to be lost.

For RI, the map produced for this section is not very representative of the terrain. The features mentioned in the description of the area are not distinguishable on the RI map. As with area 2, the RI approach only labels terrain directly in front of the robot as traversable, marking everything else as untraversable.

![Figure 5.10](image)

**Figure 5.10:** Road Recall and Obstacle Misclassification Rates for the individual local classifications and ratings for area 3. Lines (Left Axis): Road Recall and Obstacle Misclassification Rate respectively, Bars (Right Axis): Total Number of Cells that were labelled drivable and obstacles Respectively. The interval between the local classifications is approximately 2 seconds.
5.2 Summary

In this chapter, we have seen the performance of the proposed method in terrain that has gravel and asphalt paths, grass, large obstacles, vegetation and traversable rough terrain such as the roots of trees. The proposed approach can reliably identify the roughness of the features that were found in the terrain tested. Large obstacles were marked correctly as untraversable throughout the tests while more subtle features which the UGV can traverse are marked with appropriate values for Drivability. On some rare occasions some traversable areas were marked as untraversable.

The performance of the algorithm when using pointclouds produced by the stereo sensor is generally more representative than when using LiDAR input. This is true in both smooth and rough terrain and can be partly attributed to the fact that the claimed accuracies of the sensors are similar at close ranges while the stereo sensor produces much
denser pointclouds. The differences observed between the Drivability Maps when using Stereo and LiDAR are reasonable and both outputs can be considered good representations of the roughness of the terrain.

When compared to RI, the maps produced by the proposed approach relating the capabilities of the UGV to the roughness of the terrain are much easier to interpret and seem to reflect our intuition regarding roughness in a much clearer way. Additionally, as seen in the local classifications in figures 5.5, 5.8 and 5.11, Drivability index has a bigger effective range meaning that its output can be used while moving at greater speed. It is worth mentioning that the method used for updating the global map for RI was the same as used for our approach and thus the final drivability maps show a result that is not very representative of the approach. However, the individual local classifications are indicative of the approach as described by Wilson [WR14] and thus the final maps could be considered indicative of the results when using this approach in a real-time application as the metric marks everything as obstacle as the distance increases.

Unfortunately, the absence and unattainability of absolute ground truth means that numerical evaluation relies on an instinctive and subjective labelling process which only partly reflects the real performance of the algorithms. More specifically, the ratings given to terrain, as well as what is marked as obstacles is based on intuition which does not reflect the true geometric properties of the terrain.

Despite the difficulties in evaluation, the proposed approach produces results which are easy to explain and interpret while the maps produced using the Roughness Indicator that was chosen as a benchmark are not a representative map of the terrain.
Chapter 6

Conclusion

6.1 Conclusions

6.1.1 Summary of work undertaken

Several approaches for terrain classification and roughness estimation are examined. Proprioceptive methods which use the frequency domain of the vibrations of the vehicle such as the Frequency Response method by DuPont et al. [DuP+08] and the speed-independent vibration-based terrain classification by Ward and Iagnemma [WI09] were considered but deemed unsuitable for the purposes of this project as we would like to act proactively to changes in the terrain. Furthermore, 2 exteroceptive approaches were identified and explored which propose a metric for estimating and summarising the roughness of the terrain. A simple metric (RI) proposed by Wilson [WRI4] that relates the standard deviation of the points incident in a cell of a 2D grid-map of the environment with a height parameter representing the traversability capabilities of the vehicle. A roughness mapping approach created for the NASA K-Rex rover by Jonathan Loh, Elkaim. and Curry [LEC13] presents a more involved approach in estimating the roughness of terrain also relating it to vehicle capabilities. This approach uses the normals of planes fitted to the individual grid-map cells as the main metric for roughness but also models explicitly for the slope, coverage and plane fit of cells to account for the distribution of points within cells.

Inspired by the 2 approaches outlined above, an alternative metric for terrain roughness which extends the terrain labelling function created by Thrun et al. [TMA06] in PTA is proposed. The proposed approach re-purposes the statistical tests used for the purpose of labelling obstacles over a 2D grid, in order to establish a probability that the cell falls within a given height difference from its immediate neighbours. This approach is formalised and subsequently adapted and implemented for the Husky UGV and the Multisense SL stereo and LiDAR sensor combination. The RI metric is also implemented for comparison.
6.1.2 Summary of evaluation

Several experiments were subsequently designed and conducted to assess the ability of the proposed method to rate the terrain roughness. Firstly, the approach as tested in simulation using the Gazebo Physics Simulator to test the response of the system to some artificial obstacles and rough terrain. The system was then tested on real data to confirm the results of the simulations and calibrate the parameters for the system in order to prepare for evaluation.

The final step of this project was evaluating the system on appropriate scenarios. This involved collecting new data which incorporates various features of unstructured terrain and processing it offline in a real-time manner to assess the performance of the proposed approach. Drivability maps which show the performance of the system were produced and discussed in chapter 5. In general the system produces intuitive and easy to explain results in most cases. The Drivability maps produced using stereo point-clouds as input provide a much clearer and representative view of the terrain roughness compared to the equivalent maps from LiDAR inputs. In addition to the drivability map comparison, the road recall and obstacle misclassification rates are examined for the area around the southern entrance of George Square.

One of the main difficulties identified in developing terrain roughness rating approaches for drivability is that absolute ground truth is unattainable as it is dependent on the capabilities of the given vehicle. The attempt to manually label terrain roughness had several serious drawbacks which hindered the ability of the designed evaluation approach to produce indicative results for the roughness ratings. The labelling was performed at instances of the local classifications in order to capture the performance of the approach at the lowest level. The labelling process used was not accurate enough to produce indicative results. This was partly due to lack in precision when labelling the data resulting to location mismathces and partly due to our intuition in labelling the terrain not matching the geometric properties of the terrain. An example of this is that the low vegetation in the topmost section of area 3 was labelled as untraversable as a human driver would not choose to go over it while if we consider its geometric properties it could be considered as drivable.

6.1.3 Summary of achievements and lessons learned

Through the completion of the project the following were achieved:

- A clear understanding of the principles required for robotic navigation and terrain estimation.
- A thorough investigation and critical evaluation of relevant terrain classification and roughness estimation methods.
- Familiarisation with industry standard tools such as ROS [Bad+16] and LCM [HOM09].
- Familiarisation with state-of-the-art sensors such as the Multisense SL.
6.2 Further Development

A novel method for rating terrain using a probabilistic approach was introduced.

An algorithm using the proposed approach as implemented and tested.

The proposed algorithm showed very promising results which are easy to interpret and match our intuitive perception of the terrain.

In addition to above, the following lessons were learned:

- The implementations of approaches for specific platforms need to be heavily modified to suit a particular application.
- Application of simple theories and mathematical principles become much more difficult when applied to real robots especially with regards to noisy measurements and undefined, unstructured environments.
- Ground Truth for Robotics applications is not always easy to obtain. Although nowadays some datasets exist (such as the KITTY [Gei+13] and RobotCar [Mad+] dataset for autonomous driving), none were found which include terrain roughness estimation.
- Terrain roughness is highly subjective and difficult to generalise.

6.2 Further Development

This part of the project focuses on terrain classification and roughness estimation. To improve the performance of the proposed approach and evolve the capabilities of the robot with the final goal of creating an autonomous vehicle, the following goals have been identified:

- One of the problems encountered is that big vibrations were still found to affect the local classifications. In order to improve the performance of the local classifications that the proposed approach makes, a way of incorporating the momentary noise in point accumulation from the LiDAR sensor should be explored. One way to do this is to look at the IMU data during the sweep of the terrain and in order to compensate of the vibrations that the vehicle experiences.
- In order to further improve the performance of local classifications, it is worth investigating a way to incorporate the timing difference between points in the terrain scans for the LiDAR sensor. One solution is to adapt the approach used in PTA [TMA06] for use in a spinning LiDAR rather than a bush-broom configuration such that the time between the points is explicitly modelled in the statistical tests.
- To improve the incorporation of local drivability ratings into the global map, 2 areas need to be improved upon. First, alternative odometry methods which incorporates the roll and pitch of the robot should be implemented and tested for the given robot in order to improve the alignment of consecutive classifications. Secondly, alternative approaches for updating the global drivability index map should be explored. One way to approach this would be to adopt a rule based
approach which considers the value of the new and old value for a given point in the update decision. One example would be to require the new drivability rating to exceed some threshold if it is updating a location previously marked as obstacle.

- In order to improve the final performance of the system, fusion of LiDAR and Stereo based results should be considered. Such an approach is documented in Probabilistic Traversability Map Generation Using 3D-LIDAR and Camera by [Soc+16]. In this approach, the stereo based system and the LiDAR based system assess the terrain individually and the results are probabilistically fused.

- A different approach that could be considered for improving the performance of the system is to fuse the LiDAR and Stereo data before applying the local classification. One such method is Real-time probabilistic fusion of sparse 3D LiDAR and dense stereo by Maddern and Newman [MN16]. In this approach LiDAR data is used to improve disparity calculation in stereo reconstruction which result in improved accuracy and range in the final pointcloud.

- In order to better evaluate the performance of the proposed approach testing under more conditions should be undertaken. Some examples which were not encountered in our tests are cliffs and rocky but traversable terrain which are common features for outdoor terrains.

- In order to utilise the Drivability Index maps created by this method for navigation, suitable a path planning approach has to be implemented. One possibility is to use Potential Fields for Local Path Planning [HA92] in order to make the robot traverse the path of least resistance in a reactive manner in the short term and use a global path planning method which is less susceptible to being stuck in local minima such the A* algorithm.

Due to unforeseeable circumstances, the second part of this project will continue with the ANYmal quadruped robot, also equipped with a spinning Hokuyo Laser sensor and stereo cameras. Due to the different dynamics of the system, the second part of this project will focus on the problem of terrain mapping for quadruped robots. More specifically, we will first focus on implementing Robot Centric Elevation Mapping by Faknhauser et al. [Fan+14] which builds a local 2D terrain map with uncertainty estimates in order to aid the step-planning process of the legged robot. On success of the this part and time-permitting, the project will continue to explore the Visual Teach and Repeat approach by McManus et al. [McM11] which will be adapted and implemented for the ANYmal robot. In order to realise this the project will require the following:

- Familiarisation with the new robotic hardware. Many of the concepts and software as well as the libraries used for the Husky UGV are also used for ANYmal but the kinematics for a legged robot dictate further familiarisation.

- Adapt and implement Robot-centric elevation mapping approach.

- Examine and evaluate the suitability of the Visual-Teach & Repeat approach for use with legged robot and the relevant sensors.
• Adapt and Implement the algorithm for the ANYmal robot
• Tele-operate the Robot for the Teach part of the algorithm
• Evaluate the performance of the system based on its ability to closely follow the taught path.
Bibliography


Appendix A

Statistical Tests

Statistical Significance and the T-Test

Statistical Significance is the term used to describe how important observations we have are given are according to their parameters. The particular Statistical Significance Test that is of interest in this paper is the T-Test which is designed to test how confident we are that the difference of two sample means, with some unknown population variance but known sample standard deviation, is greater than some threshold distance $\delta$.

The test is expressed as:

$$P(|Z_i - Z_j| > \delta) > \pi$$

The test is considered a success at $\pi$ level of confidence when the probability that this difference is greater than $\delta$ is greater than $\pi$.

In cases where we have unequal sample variances and sizes the T-Score of the two means is expressed as:

$$T = \frac{|Z_i - Z_j|}{\sqrt{\frac{s_i^2}{N_i} + \frac{s_j^2}{N_j}}}$$

where $Z_i$ is the mean height of sample $i$ and $s_i^2$ and $N_i$ are the sample variance and sample size of $i$ respectively. To convert this score into a probability by computing the Standard Normal Cumulative Distribution Function up to $T$. 

63
Appendix B

Supplementary Results

B.1 Average Deviation

In chapter 3 of this project, a method for numerically evaluating the terrain roughness estimates was proposed. The approach did not work well due to difficulty in the precision of the labelling both in terms of location and the roughness estimates themselves. As the results were not all indicative of the performance of the algorithms presented and the respective results, this is presented as a proof the concept and the work undertaken.
Appendix B. Supplementary Results

At a traversability height of 3cm, as presented in the evaluation, DI on stereo output has the highest deviation from the labelled value of all approaches which does not match the results shown in chapter 5. Furthermore, the standard deviation of the deviations at the same traversability height is around 0.20 which leads us to conclude that the roughness estimate labelling is highly inaccurate as the rest of the results are representative of the terrain.

B.2 George Square North

Data was collected from other parts of the George Square park which did not provide features as indicative as the ones in the area used in the evaluation section of the report. Therefore, the terrain classification and roughness estimation is for the north side of George Square is provided as supplementary proof the effectiveness of the method.
Figure B.2: Drivability Map for George Square North side using Stereo input

Figure B.3: Drivability Map for George Square North side using LiDAR input
Appendix B. Supplementary Results

B.3 Path Segmentation

Part of the experimentation for the project was on path segmentation but as it was not directly relate to the final approach presented, it is presented here as supplementary material. As we are dealing with unstructured environments, we’d ideally want to segment paths based on their geometric properties rather than using appearance based approaches. More specifically, a region growing path segmentation algorithm was implemented on top of the terrain classification and roughness estimation such that paths were segmented based on their roughness. The two figures below show the extracted paths when the region growing path segmentation rejects all cells below which are rated with DI below 0.5 for Area 1 from the evaluation chapter.

Figure B.4: Drivability Map for George Square North side using Stereo input

Figure B.5: Drivability Map for George Square North side using LiDAR input