Autonomous Soil Assessment System for Planetary Rovers

Dhara Kamleshkumar Shukla

A Thesis
In
The Department
Of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of

Master of Applied Science (Electrical & Computer Engineering)

at

Concordia University

Montreal, Quebec, Canada

November 2017

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Dhara Kamleshkumar Shukla

Entitled: Autonomous Soil Assessment System for Planetary Rovers

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Electrical and Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________________________ Chair
Dr. R. Raut

______________________________________ External Examiner
Dr. W.F. Xie (MIAE)

______________________________________ Internal Examiner
Dr. K. Khorasani

______________________________________ Internal Examiner
Dr. H. Rivaz

______________________________________ Supervisor
Dr. K. Skonieczny

Approved by: ______________________________
Dr. W.E. Lynch, Chair
Department of Electrical and Computer Engineering

______________ 20___

____
Dr. Amir Asif, Dean,
Faculty of Engineering and Computer Science

ABSTRACT

Autonomous Soil Assessment System for Planetary Rovers

Dhara Shukla

Planetary rovers face mobility hazards associated with various classes of terrains they traverse, and hence it is desirable to enable remote prediction of terrain trafficability (ability to traverse) properties. For that reason, the development of algorithms for assessing terrain type and mobility properties, as well as for coupling these data in an online learning framework, represent important capabilities for next-generation rovers. This work focuses mainly on 3-way terrain classification (classifying as one of the types: Sand, Bedrock and Gravel) as well as on the correlation of terrain types and their mobility properties in a framework that enables online learning. For terrain classification, visual descriptors are developed, which are primarily based on visual texture and are captured in form of histograms of edge filter responses at various scales and orientations. The descriptors investigated in this work are HOG (Histogram of Oriented Gradients), GIST, MR8 (Maximum Response) Textons and the classification techniques implemented here are nearest and k-nearest neighbors. Further, monochrome image intensity is used as an additional feature to further distinguish bedrock from the other terrain types. No major differences in performance are observed between the three descriptors, leading to the adoption of the HOG approach due to its lower computational complexity (over 3 orders of magnitude difference in complexity between HOG and Textons) and thus higher applicability to planetary missions. Tests demonstrate an accuracy between 70% and 93% (81% average) for the classification using the HOG descriptor, on images taken by NASA's Mars rovers.

To predict terrain trafficability ahead of the rover, exteroceptive data namely terrain type and slope, are correlated with the trafficability metrics namely slip, sinkage and roughness, in a learning framework. A queue based data structure has been implemented for the correlation, which keeps discarding the older data so as to avoid diminishing the effect of newer data samples, when there is a large amount of data. This also ensures that the rover will be able to adapt to changing terrains responses and predict the risk level (low, medium or high) accordingly.

Finally, all the algorithms developed in this work were tested and verified in a field test demo at the CSA (Canadian Space Agency) mars yard. The risk metric in combination with the queue based data structure, can achieve stable predictions in consistent terrains, while also being responsive to sudden changes in terrain trafficability.

ACKNOWLEDGEMENT

I would like to express my deep gratitude to Professor Krzysztof Skonieczny for his invaluable guidance and continuous support throughout this project. This could not have been accomplished without his thoughtfulness and constant encouragement. This work is a part of a larger program, in collaboration with other industries. I would like to thank all the members of our industrial partner ‘Mission Control’ for their sincere efforts and support. Their documentation for the project has also been a great help for this thesis.

I would like to express my gratitude to my committee members for their time and guidance through the review of the thesis. I would also like to thank my colleagues at Concordia University for always being there to help and share their valuable experience.

Last but not the least, I want to express my deepest appreciation to my husband and my parents for their endless love, support and motivation. Their unflinching confidence in me and positive energy helped me to overcome all the obstacles.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

The experiences of NASA's series of Mars rovers—Sojourner, Sprit, Opportunity, and Curiosity—provide a wealth of knowledge about rover mobility challenges in extra-terrestrial terrain. Spirit had experienced high slippage when crossing highly deformable sands [1]. Analysis of mobility challenges faced by Spirit showed that difficulties occurred while traversing locally low regions, where highly deformable sulfate-rich soils had accumulated. High sinkage events became particularly problematic when Spirit's right front-wheel-drive actuator failed and the rover was forced to drag the wheel, leading to a final embedding event in the Troy area. In March 2010, communications with Spirit were lost as the winter season approached, and the mission was terminated [1].

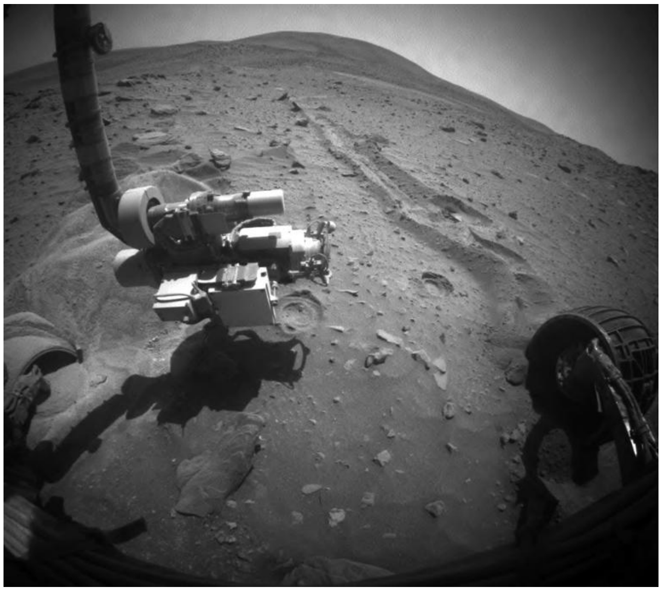


Figure 1-1: Image of Spirit embedded in Troy
Courtesy NASA/JPL-Caltech

The Opportunity rover has experienced high wheel sinkage and slippage on multiple occasions when traversing sandy crater walls or wind-blown ripples. In some instances wheel slip approached 100%, leading to scenarios where the rover could not reach the desired traverse target and was forced to re-route [2]. Such high slippage occurred when all six wheels were on the flanks of wind-blown ripples with mild slope. Un-compacted regolith was shown to be highly deformable, leading to high wheel sinkage and thus high motion resistance and increased slip. The most significant case of sandy ripples causing difficulties for Opportunity was the embedding event in what was dubbed “purgatory dune” that lasted from Sol 446 to Sol 484 [76].

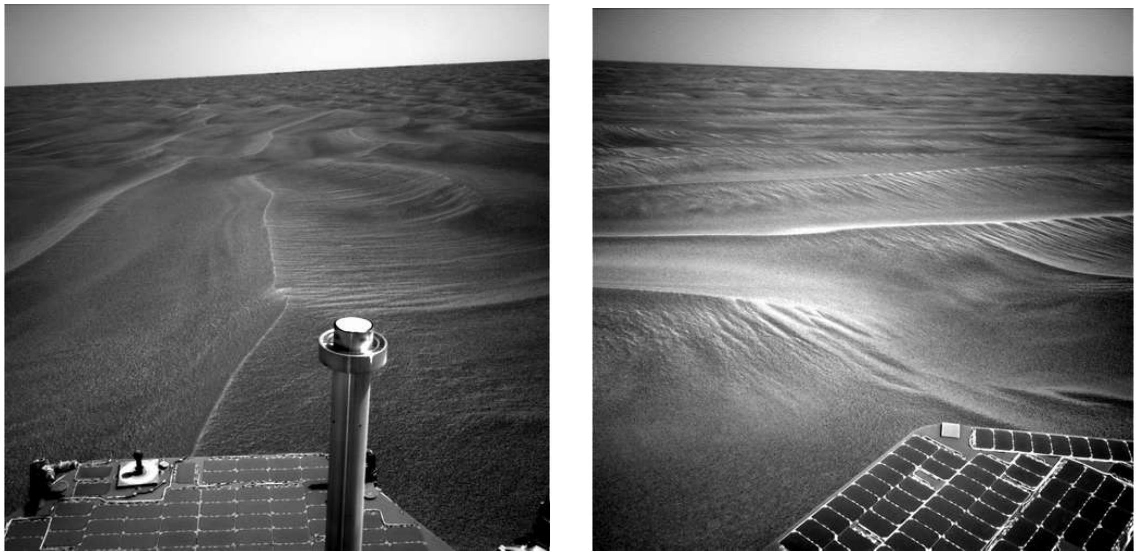


Figure 1-2: Front looking & Side looking views of "purgatory dune"
Courtesy NASA/JPL-Caltech

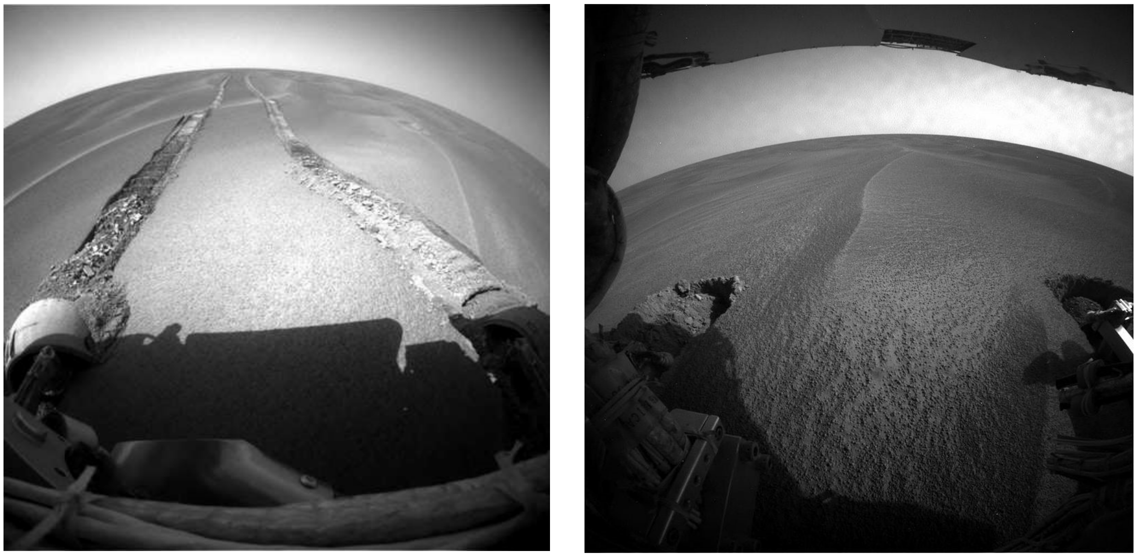


Figure 1-3: Images during Opportunity "purgatory dune" embedding event, showing wheel sinkage
Courtesy NASA/JPL-Caltech

The Curiosity rover has also experienced mobility difficulties when traveling over loose, wind-deposited soil, with the most extreme slippage events occurring when the rover attempted to travel over shallow slope formations ("ripples.") [77]. Further, Figure 4 below shows tears and punctures in the Mars Science Laboratory Curiosity rover's wheel. Curiosity's wheels are at highest risk in areas strewn with embedded jagged rocks [1].

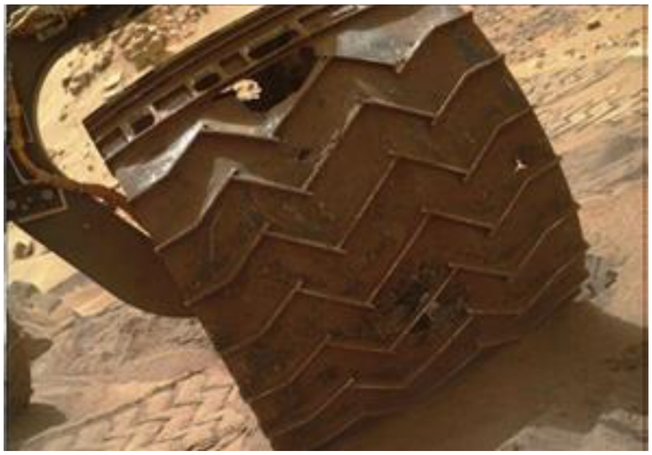


Figure 1-4: Tear and Punctures in Curiosity's wheel
Courtesy NASA/JPL-Caltech

1.1 Problem statement and proposed solution

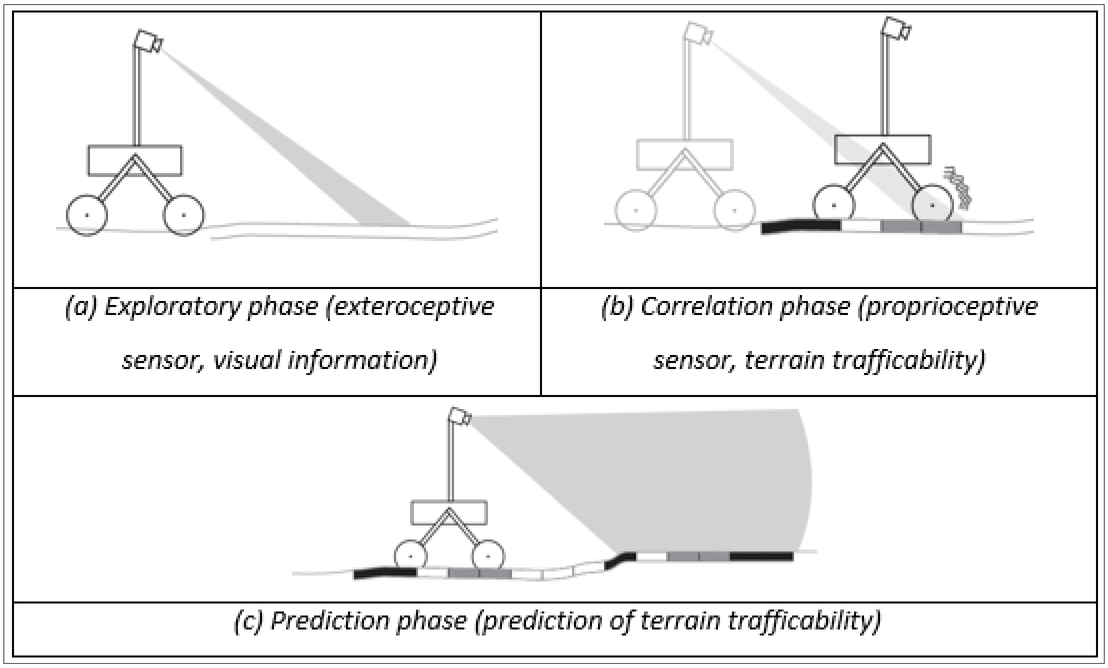
Analysis of data from NASA's series of rovers has thus clearly and repeatedly shown that non-geometric hazards, typically in the form of loose regolith deposits and often occurring in locally low or ripple regions, have posed by far the greatest challenge to safe rover navigation on the surface of Mars [1, 2, 76].

Current rover navigation algorithms rely primarily on the detection and avoidance of geometric hazards such as rocks and steep slopes. While some "terrain induced hazards" or "non-geometric hazards" can be avoided by painstakingly designing a safe route through terrain that has been carefully analyzed by a human expert, other hazards can only be detected through the use of on-board perception techniques. Those on-board perception techniques rely on two distinct classes of sensors: *proprioceptive* sensors and *exteroceptive* sensors. Proprioceptive sensors are those that measure the internal state of a system, e.g., quantities including vehicle body acceleration, angular rates, chassis vibration, and wheel/track drive torque, among others. Exteroceptive sensors are those that measure data related to the surrounding environment, e.g., vision sensors, LIDAR sensors, and multispectral sensors, among others.

To increase the likelihood of detecting terrain induced hazards, it would be desirable to enable a rover system to accurately estimate the physical properties of that terrain. Non-geometric hazards are highly dependent on wheel-terrain interaction properties. For that reason, the development of algorithms for assessing the terrain type, physical properties, and mobility properties represent important capabilities for next-generation rovers. Furthermore, to enable remote prediction of terrain trafficability, it is essential to introduce learning-based methods that couple proprioceptive and exteroceptive data in a correlation phase.

One aspect of this problem is terrain *classification,* i.e. assigning a terrain region of interest to one of a small number of discrete classes, such as compacted sand, loose sand, or rock. Another aspect is terrain *characterization,* i.e. capturing measured data from (typically) rover proprioceptive sensors when driving over a terrain region of interest, and analyzing this data to infer a quantitative measure of soil strength or some other physical property related to trafficability. These terrain classification and characterization techniques can be useful in identifying "terrain induced hazards" or "non-geometric hazards" that can pose severe mobility challenges to a rover.

Hence, the ultimate goal is to develop a system that will combine the best properties of terrain *classification* and *characterization* methods to allow a rover to accurately predict the physical properties of distant terrain and this system is named the "Autonomous Soil Assessment System (ASAS)". For classification techniques, the goal is to avoid the computational load, wherever possible owing to the fact that planetary vehicles have extremely limited computational resources due to high radiation level and large temperature changes of space [78]. Whereas, for the combination of both the approaches, the architecture has been inspired by the work of [3], which exploits both proprioceptive and exteroceptive sensor data to enable accurate prediction of distant terrain properties. In such an architecture, the rover captures exteroceptive sensor data from nearby terrain and performs terrain classification (Figure 1-5 (a)). Then, when the rover traverses that terrain, it captures proprioceptive sensor data and performs terrain characterization (Figure 1-5 (b)). A machine learning algorithm is then trained in order to couple the proprioceptive and exteroceptive data, in a correlation phase. During rover operation, a learned, data-driven model is then used to estimate trafficability properties of distant terrain using only exteroceptive data (Figure 1-5 (c)



(a) Exploratory phase (exteroceptive sensor, visual information)

(b) Correlation phase (proprioceptive sensor, terrain trafficability)

(c) Prediction phase (prediction of terrain trafficability)

Figure 1-5: Method for correlating terrain classification and characterization based on proprioceptive and exteroceptive sensing systems, From [3]

1.2 Contributions of this thesis

This thesis presents the development of the exploratory phase explained above as well as the correlation of exteroceptive and proprioceptive data in a learning framework, which results into the development of a data driven model.

The main contributions of thesis are as follows:

1. Evaluating visual descriptors according to computational efficiency, with relevance to planetary rover constraints.
2. Introducing a simple and adaptive data driven non-parametric trafficability model.

Chapter 2. Literature Review

Work in the literature that is related to autonomous soil assessment includes image classification (both in general and for planetary terrains in particular), proprioceptive terrain classification, and trafficability modeling and prediction.

2.1 Image Classification

Exteroceptive sensors such as visible light cameras and laser range finders (LIDARs) have been widely employed in the field of mobile robotics and used extensively for terrain classification purposes. A common difference between methods lies in the choice of features or descriptors used to train the classifier. Common filters used for feature extraction include Gradient, Sobel, Prewitt, Gabor, LBP (Local Binary Patterns) [38] etc. For example, local gradient filters are used in HOG [36] whereas Gabor filters are used in GIST [7, 8] as well as Textons [5, 6 & 12] (in conjunction with Gaussian and Laplacian of Gaussian filters). These three descriptors have been implemented in this work and hence, they are described in detail in Chapter 3. Apart from them, local gradients are also used in higher order descriptor such as SIFT [35] while Haar wavelets are used in SURF [37]. Haar wavelets are also used for feature extraction in the work performed in [3, 20 & 34].

SURF consists of recognizing the interest points in an image using a Hessian matrix at different scales. An NxN matrix is formed around every interest point, which is further filtered to extract features and form a descriptor. The Haar wavelets are used for extracting features from the matrices. Here, Haar wavelet response is derived in the x and y directions for the 4x4 sub-regions. The resultant matrix obtained in x-direction is then added to form a response, Dx. Similarly, Dy, |Dx| and |Dy| are, sum of y-response values and sum of absolute value of responses in x-direction as well as y-direction, respectively. This forms a 4 length vector. Hence for 4x4 sub-region, there will be 16x4=64 length vector, which is called the descriptor.

Figure 2-1 below examines the space of these techniques for capturing texture, plotted on axes of window size and orientation resolution. In terms of window size, gradient (3x1) is lowest in comparison with all the other techniques. Whereas, in terms of orientation resolution, gradient, Sobel, Prewitt, and Haar filters can only be oriented horizontally or vertically; LBPs additionally encode diagonal information; Gabor filters can be defined at arbitrarily high angular resolution, and co-occurrence matrices use information from all pixels regardless of their relative orientation.

From this comparison, it can be seen that gradient-based methods and Gabor-filter based methods represent opposite ends of a spectrum, from coarse to finely tunable, in terms of the metrics discussed. These differences motivate an exploration of whether such distinctions, at the extremes, lead to appreciable differences in performance or complexity. Hence, in this work, the performance of a gradient based method, essentially a simplified HOG descriptor, and Gabor-based methods including Textons and a simplified GIST descriptor has been explored and compared.

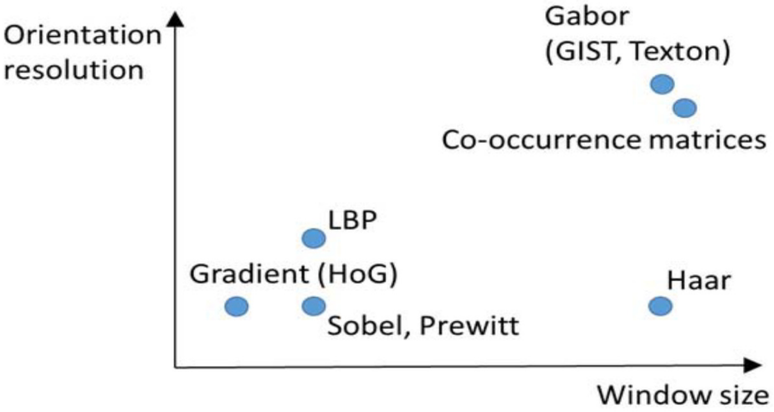


Figure 2-1: Qualitative complexity of various texture approaches in terms of their window size and orientation resolution

Moreover, from the point of view of quantitative analysis of computational complexity, the complexity of 2D convolution in spatial filtering, in terms of Big-O notation, is given by O(MN mn) [52]. Here image is of size M x N, whereas, the filter size is m x n. Such quantitative analysis is conducted for all three descriptors as well, in Section 3.4.

Most existing techniques for classifying terrain with exteroceptive sensors make use of machine learning algorithms (e.g., Support Vector Machine) [9, 10]. Some of the simplest classification method include nearest and k–nearest neighbor method. The work performed in [12, 35] exhibits the usage of nearest neighbor method for the image classification. The k -nearest neighbor classifier is a conventional nonparametric supervised classifier [53] and the work performed in [54] shows the comparison between K-NN and Neural network classification techniques for the classification of magnetic resonance images (MRI). Also, a neural network (NN) is used for multi-sensor (Image, Accelerometer, Encoder) road classification in the work of [22].

Convolutional neural networks (CNNs) came into the focus of attention with the ‘AlexNet’ developed by [55] who were able to achieve a top-5 test error rate (the rate at which, given an image, the model does not output the correct label with its top 5 predictions) of 15.4% for classifying images into 1000 different categories. This work illustrated the benefits of CNNs and backed them up with record breaking performance in 2012 ILSVRC (ImageNet Large-Scale Visual Recognition Challenge). Their network was trained over 15 million annotated images from a total of over 22,000 categories, for five to six days. After this work, several researchers have worked on CNN, trying to improve performance and reduce the complexity such as [56, 57, 58]. The current best CNN architecture is the one in [59] with 3.6 error rate, however, this network was trained on 8 GPU machines for two to three weeks. Thus, the ability of neural network techniques to outperform other classification techniques comes at a cost of very high computational complexity.

2.2 Planetary Terrain Classification

Some recent work has investigated images of Mars terrains directly [18, 19] or Martian analogue terrains consisting of sand, bedrock, and soil [20, 21]. Halatci et al [19] study the performance of a multi-sensor classification method in the context of Mars surface exploration. Features based on color, texture, range (i.e., surface geometry), and vibration arising from robot wheel–terrain interaction are utilized for classification. For the texture detection, a wavelet-based fractal dimension signature method, which yields robust results in natural texture segmentation as demonstrated by [60] has been employed. However, they still assume the availability of color images and are using data from the MER Panoramic Cameras, which are a science instrument rather than a dedicated navigation sensor. Similarly, Otsu et al [20] also assume the availability of color images. Moreover, Fujita et al [21] not only classify soil types but also the velocities of a rover relative to a terrain surface, thus identifying slippage arising from distinct terrain properties. This is accomplished by a classification method based on a dynamic texture analysis. The effectiveness of the combined distance measure for improving classification performance is demonstrated through experimental runs of two rover testbeds in sand pits. However, they utilize frequently sampled image sequences, another feature that may not be available to near-term planetary rovers.

Apart from that, other work in this area has focused on combining multiple exteroceptive sources. In [41], a method based on conditional random fields is trained using data from a 3D LIDAR and a monocular camera. A terrain classification approach for an autonomous robot based on Markov random fields is proposed in [42]. Here, the authors fuse 3D laser and camera image data. An approach based on combining the information from a downward-looking camera and a frontal camera is proposed in [9]. In this case, the classification algorithm is based on a GIST descriptor and color features. The work of [10] describes the use of three types of perception systems to classify the terrain, namely vision, depth and tactile sensors. In this study, the use of visual cues was shown to provide poorer results than the other sensing modalities.

2.3 Proprioceptive Data based Terrain Classification

A substantial body of work exists describing methods for terrain classification or characterization based on proprioceptive sensor data. Though this aspect is not directly explored in this thesis, it is an important related work. Proprioceptive terrain classification based on analysis of vibration in the rover structure is one of the most common approaches, and was introduced in [23] and later explored in [24,25]. Vibrations in the rover structure can be measured via a variety of sensors, including accelerometers mounted rigidly to the rover structure [26, 27], IMUs (Inertial Measurement Units) [28], and contact microphones mounted on the rover structure, ideally near the wheel-terrain interface [23, 29].

A comparative study of the utility of several different proprioceptive sensors is presented in [43]. That work describes experimentation with microphones, gyroscopes, accelerometers, encoders, motor current sensors, voltage sensors, and downward-facing ultrasonic and infrared cameras. It was reported that gyroscopes worked best for gravel, pavement and dirt; motor current worked best for sand; and microphones worked best for grass. This is likely due to the fact that distinct terrain types tend to yield distinct proprioceptive signatures, e.g., travel over gravelly terrain tends to induce vibration in the rover structure due to terrain roughness, and this vibration is best detected by a gyroscope; in contrast, sandy terrain is often relatively smooth, which does not result in significant rover vibration but rather leads to high motor current draw due to increased soil compaction resistance. Another example where a comprehensive sensor suite is tested, appears in [44]. In particular, accelerometers, rate gyroscopes, leg angle encoders, and motor current estimators are combined through a probabilistic framework for environment identification.

Vibration-based classifiers reported in the literature typically rely on machine learning algorithms to perform classification. Support vector machines (SVM) are a commonly used technique to enable on-line identification of terrain type [23, 29, 27, 32]. One of the practical challenges of vibration based classification is that such systems typically require training over a narrow range of operating conditions. As a specific example, it can yield poor results when the robot speed is significantly different from the training speeds. This effect is minimized in [30] by creating an estimate of the terrain profile from measured wheel accelerations and considering spatial (rather than temporal) frequency components of the estimated profile. Features extracted from the vibration data in various domains (time domain, frequency domain, and wavelet features) are analyzed in [45]. Another limitation of vibration-based methods is that while data is generally captured by a single sensor (mounted at the rover body, or near a wheel axle), each wheel is implicitly assumed to experience the same terrain. Thus, situations in which a vehicle simultaneously travels on two or more terrains may lead to poor results. In [31], a novel method called *Terrain Input Classification* improves traditional vibration-based techniques by eliminating or reducing speed dependency and by detecting multiple terrain traversal.

2.4 Trafficability/Terramechanics Models

The founding modeling paradigm of terramechanics (the study of vehicle-terrain interactions) relies on quasi-static pressure sinkage relationships to estimate compaction resistance [61] and/or empirical parameters, e.g. to estimate the location of maximum pressure beneath a wheel [62]. For rigid wheels on dry granular soil (a typical case for planetary rovers), the quasi-static assumption is contradicted by the data; Wong observed flowing granular soil, and pressure distributions more complex than predicted by quasi-static compaction [63]. Several researchers today highlight the insufficient predictive power of classical terramechanics models for planetary rovers [64-67].

Recently, data-driven approaches have become popular for assessing and modeling mobility. Some of the advantages of data-driven approaches over model-based approaches are summarized by [69]: adaptability to imperfect knowledge of model parameters, to heterogeneous combinations of materials, and to changing environmental conditions. Further, the data-driven modeling is increasingly non-parametric. Gaussian process regression is now commonly used to model relationships between terrain slope and rover slip [68-71]. This enables a data-driven model that is not forced to conform according to preconceived notions, and also provides estimates of uncertainty in the model. A key drawback of Gaussian process regression is its computational complexity, O(N^3) where N is the number of data-points. Karumanchi et al. provide a detailed discussion of Gaussian process regression in the context of slip vs. slope modeling [69]. They mention that use of a Relevance Vector Machine can reduce computation time, and also touch upon addressing multimodality in the data.

Despite slip vs. slope being one of the strongest and most broadly used relationships in mobility prediction, this relationship is nonetheless far from precisely predictable. Recent field tests [72] (and also this work) as well as reports from Mars rover data [71] highlight instances of huge variability in slip on similar slopes of sandy terrains. This variability and unpredictability is one of the key reasons for the popularity of non-parametric methods for this application.

2.5 Trafficability Prediction

The problem of predicting a planetary rover's mobility across terrain located ahead of it has received considerable attention in the literature in recent years. In [75], data-driven slip vs. slope models are learned for various terrain types and then they use exteroceptive measurements to classify terrain types and measure slope, thus predicting slip [12]. Predicted slip values are further used for goodness map generation and path planning, and simulation is used to resolve ambiguities in predicted traversability [16]. Ishigami et al. also use simulation for path planning, though their approach differs by focusing on terramechanics rather than machine learning; they minimize a cost function of rover instability, slip, traverse time and energy consumption [73]. Brooks and Iagnemma combine data-driven learning with terramechanics by using a self-supervised framework for classifying terrain types and ultimately predicting a terrain map classified according to achievable drawbar pull [3]. Krebs et al. present an unsupervised framework where the rover develops its own associations between terrain appearance and resulting mobility [74]. Peynot et al. develop stochastic mobility prediction based on features detected ahead of the rover related to terrain slope and roughness [70].

One trend worth noting in these approaches is that, slip is noted as a critical metric for rover mobility. Many researchers in the area focus on slip explicitly [3, 12, 16, 69, 75], and even researchers who do not do so recognize the implicit role it plays in other formulations of mobility prediction and note the value of investigating it directly [70, 73, 74]. When interpreting data-driven slip models, it is important to understand how the variability in slip data is related to mobility risk;

this can be informed by past work in terramechanics. In past literature studying vehicle-terrain interactions, 20% slip is frequently cited as an important threshold. Drawbar pull begins to plateau around 20% slip for many wheels (or tracks) while negative effects such as sinkage increase. Wheels can exhibit strongly nonlinear performance, with slip increasing very gradually with increasing load (induced by climbing an increasingly steep slope, for example) until it crosses above 20%, at which point it can rapidly increase to unsafe conditions of approximately 80% slip [46]; see Figure 2-2. Other researchers have noted wheels experiencing similar non-linearity at approximately 20% to 30% slip [47]. A non-dimensional quantity, P_{20}/W (Drawbar pull at 20% slip, normalized by vehicle weight), has been used as a benchmark metric for lunar wheel performance from the times of Apollo [47] to today [48, 49].

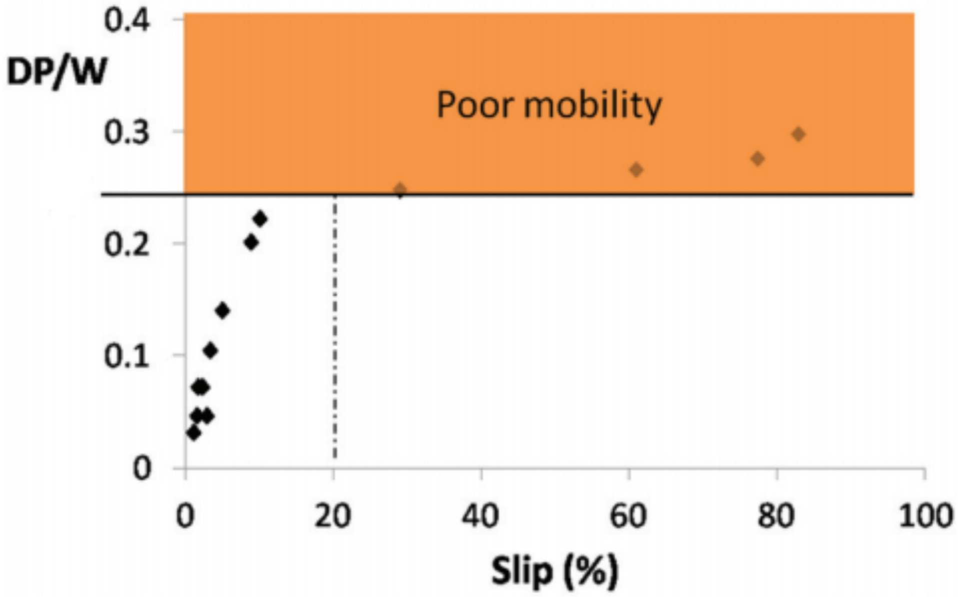


Figure 2-2: Drawbar pull plateaus around 20% slip and a vehicle enters a regime of poor mobility thereafter. This data, courtesy of NASA Glenn Research Center, characterizes the performance of a spring tire. From [46].

The work performed in [12, 75, 3] have been very closely followed by this work and hence a brief overview of both the work has been given here. In [12, 75], an approach to predict rover slip properties from a distance using stereo imagery is proposed. To address this issue, terrain classification is performed using Textons based method and nearest neighbor technique. Terrain appearance and geometry information are then correlated with empirically measured slip properties (which were measured as the difference of velocities estimated from the difference of wheel and visual odometry). This relationship is learned using a regression approach. In the end, slip is predicted online from remote visual information only (related to terrain type and slope).

The other work [3] follows a self-supervised learning classification paradigm. In a self-supervised approach, data collected on-line, which is identified by a “supervisory” classifier as belonging to a single terrain class, is used to incrementally train a “supervised” classifier that operates using a distinct sensor modality from the “supervisory” classifier. One approach based on self-supervised classification, presented in [14], fuses terrain predictions based on image data with predictions made by a vibration-based method. When the robot later traverses the classified area, it uses vibration data to verify its former prediction.

The work performed in [3] also shows the general utility of self-supervised learning. Here, a vibration-based terrain classifier takes the role of the supervisory classifier and the supervised classifier is the vision-based terrain classifier. Training data for the vision-based classifier are extracted from forward-looking stereo images stored in memory and recalled when the rover classifies a previously observed terrain patch using proprioceptive sensors.

2.6 Summary

The vast majority of related literature in visual terrain classification for off-road rovers [11-17, 3, 5] has two key differences from the present work: the use of color, and the inclusion of terrestrial-specific terrain classes like vegetation. These cited articles make very valuable contributions to the understanding and advancement in this field of study, but their assumptions make their goals somewhat different than those of this work to explicitly avoid additional sensing and computation load, whenever possible. Various visual descriptors, of varying complexity, have been proposed and used for related tasks in the literature.

In the field of trafficability prediction, this work focuses primarily on the development of a simple and adaptive trafficability model. Gaussian processes are typically used for trafficability (slip vs. slope) modeling. A key drawback of Gaussian Process Regression (GPR) is its computational complexity. The literature has discussed possible methods of addressing computation time and multimodal data, but at the time of the development of this thesis, no published work in trafficability prediction has implemented a simple and adaptive modeling approach.

Chapter 3. Exteroceptive Module

As mentioned in the introduction, the Mars rovers such as Spirit, Curiosity, and Opportunity have faced several mobility challenges on the planetary surface due to the hazards introduced by different terrain types. Different types of terrains introduce different levels of mobility hazard to the rovers. Hence, it is an important non-geometric feature, and being able to classify between different terrain types can inform the rover about the potential hazard level. The exteroceptive module is designed to use the image captured by stereo cameras (i.e. the terrain being seen in front of the rover), extract the features from these images and accordingly classify them into one of the three terrain types, viz. sand, bedrock and gravel.

Here, the feature being extracted and used for the classification is the texture information present in the image. There is no particular definition of texture within computer vision, however, the implicit meaning of texture in this work is the local changes/edges in image intensity, at various scales and orientations. This texture information from any image can be extracted by filtering it using oriented gradients as well as various filters such as Gabor, Gaussian, LoG etc., with different scales and orientations and it can be represented in form of visual descriptor, which is basically a low dimensional representation of a scene. There are various types of descriptors presented by other researchers in the past, however in this work, some of the computationally inexpensive descriptors (e.g. HOG) are used and compared with more expensive options like GIST and Textons. The details of these descriptors is given in the following sections.

Moreover for classification, simple techniques like nearest neighbor and K-nearest neighbor have been explored here in order to design a computationally inexpensive system. In this chapter, section 3.1 describes the selection and formation of the datasets, which will be used to validate the performance of different techniques. The implementation of various visual descriptors, tuning of their design parameters and their effect on the performance is described in section 3.2 as well as the classification techniques, their performance and the post processing technique has been explained in section 3.3. Lastly, the analysis of the results achieved for different datasets and some important observations are stated in section 3.4.

3.1 Data-sets

A classification system needs a Training data set to train the system, followed by a validation set to measure the performance of the system and execute the required improvements in the training method. Finally a test data set is used to determine the final performance of the system. In this application, these 3 types of data sets have been prepared by selecting images from two different sources. One is NASA JPL website [79,80], where the real Mars planetary surface images are available and the other is Canadian Space Agency Mars Yard, which is the emulation of the Mars planetary surface.

3.1.1 NASA JPL dataset

This website contains images captured by different rovers namely Curiosity, Spirit and Opportunity, from which images taken by Opportunity during following solutions are used to form different datasets.

Training dataset	**Validation Dataset**	**Testing Dataset**
Sol 4303	Sol 2315	Sol 2354
Sol 4334	Sol 1176	Sol 2330
Sol 603	Sol 2169	Sol 2320
Sol 2316	Sol 2173	Sol 2145
Sol 2317	Sol 2174	Sol 4483
Sol 2220	Sol 2324	

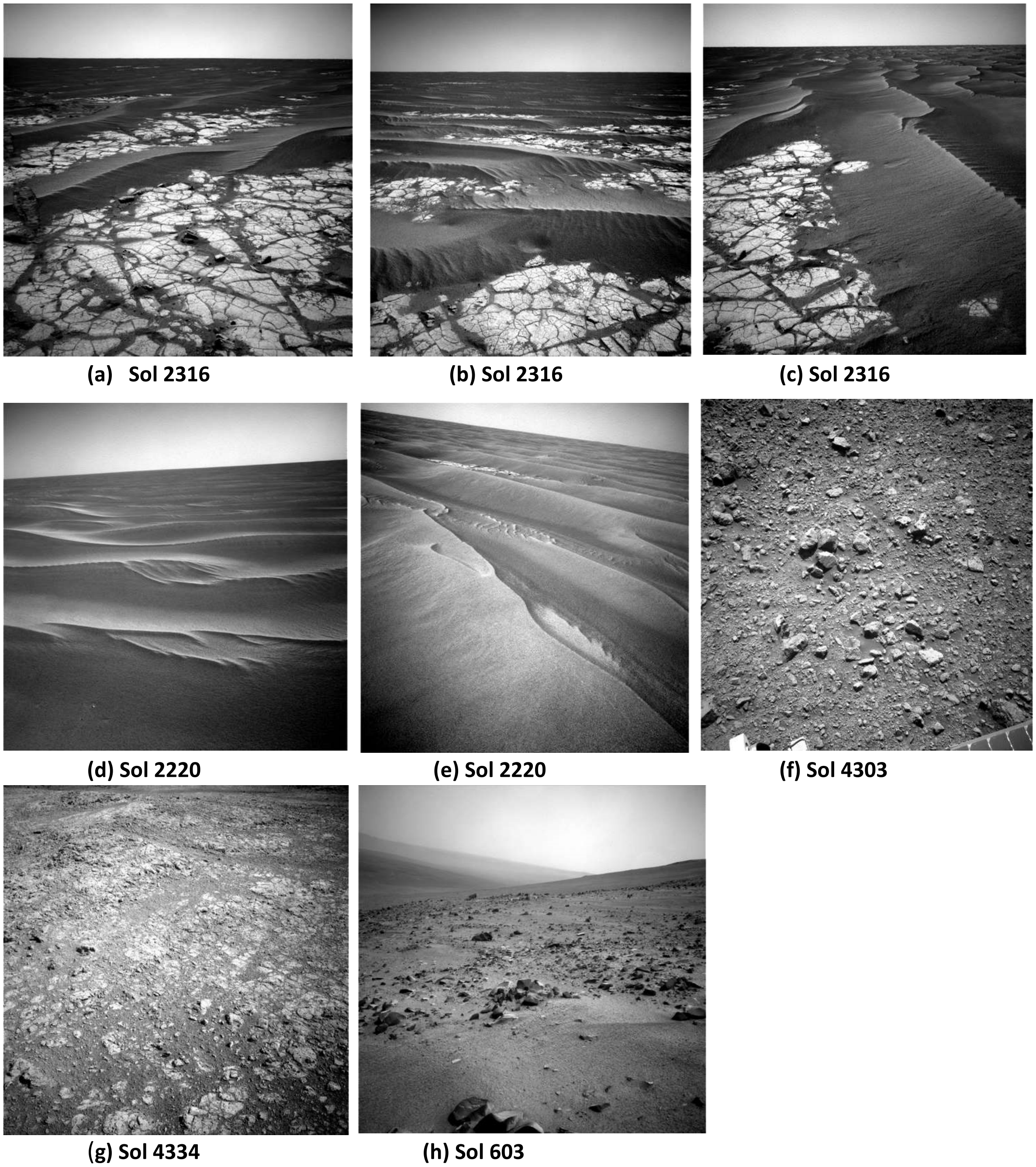
Table 1: Sol number of the images taken from NASA JPL website to form the datasets

The planetary surface images primarily consist of three terrain types namely bedrock, gravel and sand. Each image is of 500x500 pixels. The images mentioned in the above table, which are added to the datasets are shown in Figures 3-1, 3-2 & 3-3.

Further, pre-processing has been performed on these images before they gets added to any of the datasets. Firstly, the upper 40% of the images is discarded. The reason for this is that the terrains in this area are quite far from the rover in reality and it cannot provide with any reliable information. So, the effective image size is now 300x500 pixels instead of 500x500 and this cropped image gets added randomly to either of the three datasets. Secondly, each image is divided into small grid squares and each grid square is considered an individual image, to be added in the corresponding dataset. The reason for the same is that the images on the whole can heterogeneous, consisting of varied types of terrains and it is impossible to classify an image as one of the terrain type. Whereas in case of grid squares, they can be classified as one of the terrain type reliably as they cover a small area & are mostly homogeneous in nature. Here, the entire image is divided into 60 grid squares, leading to each grid size to be 50x50 pixels.

Apart from dividing the full image into grid squares, another common approach is to divide the grid square further into 4X4 = 16 sub-grids. Generating a histogram for each sub-grid separately and then concatenate all of them in a sequential manner to form the final descriptor, ensures that spatial information will also be preserved, up to some extent, enabling more advanced texture recognition. However, in this application, spatial information is not crucial enough. For example, there is a grid of the terrain type gravel and the system correctly recognizes its terrain type, then spatial arrangement or the orientation of the rocks in that grid are not important. In fact, two gravelly images with different arrangement of the rocks should be recognized as the same terrain type in this system. The prime importance here, is to simply differentiate its terrain type from the others. Moreover, not including spatial information also makes the system computationally less intensive as well as less memory consuming. Hence, the spatial information is not incorporated in any of the descriptor.

Another important factor to be considered, while preparing training library, is that it should not contain any bad examples or confusing images (ones which encompass more than one terrain types), as they can adversely affect the performance of the descriptor. Hence, such confusing images are eliminated (removed) from the existing training set, based on a personal judgement. Similarly, for the validation and the test set as well confusing image/bad examples have been removed. Such removal of confusing examples from validation and test datasets is considered only for avoiding ambiguity in performance evaluation. For example, when classifying heterogeneous image, containing both bedrock and sand, neither a classification as bedrock nor as sand can be considered unambiguously correct or incorrect.



(a) Sol 2316

(b) Sol 2316

(c) Sol 2316

(d) Sol 2220

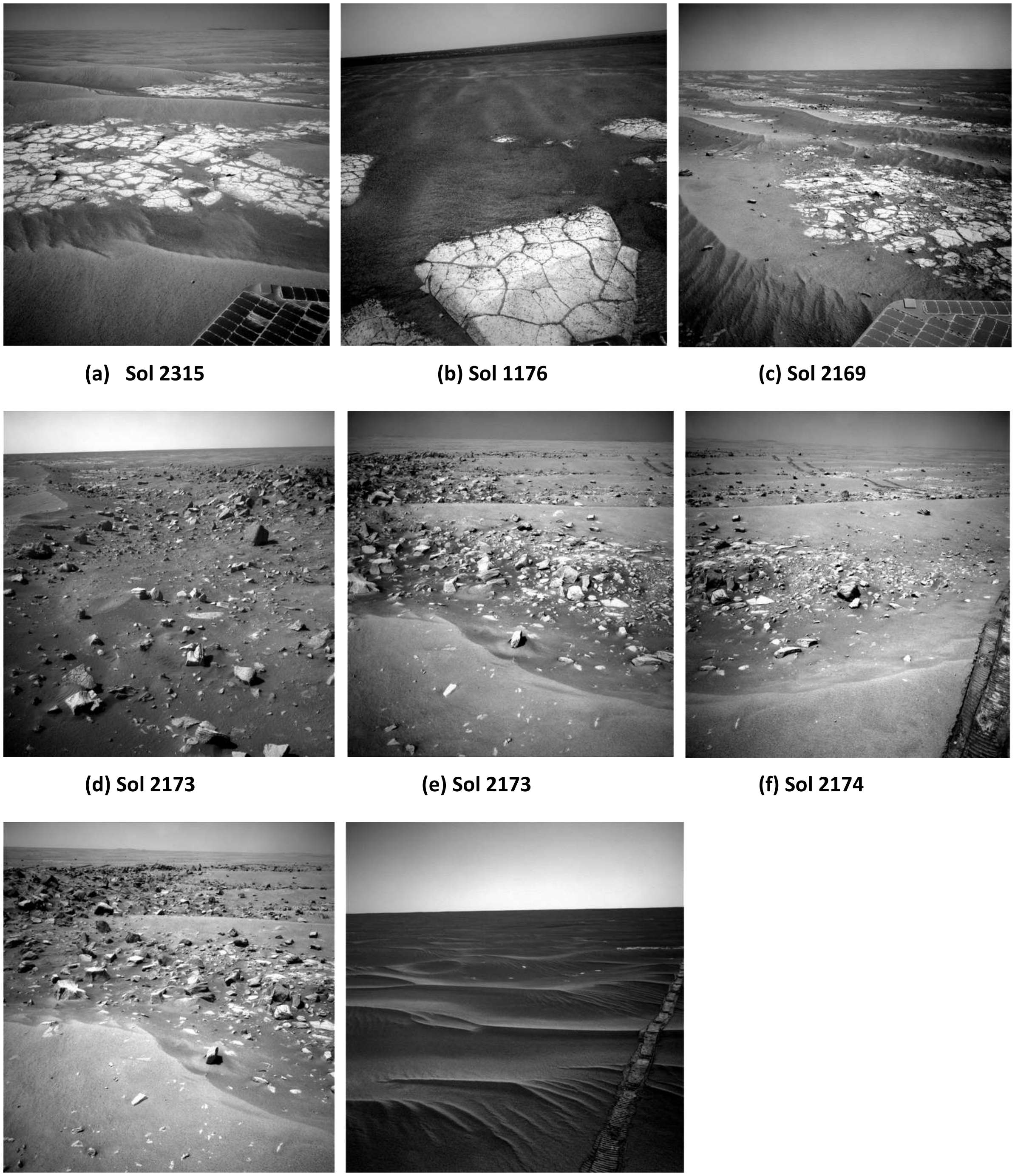
(e) Sol 2220

(f) Sol 4303

(g) Sol 4334

(h) Sol 603

Figure 3-1: Training Library Images
Courtesy NASA/JPL-Caltech



(a) Sol 2315

(b) Sol 1176

(c) Sol 2169

(d) Sol 2173

(e) Sol 2173

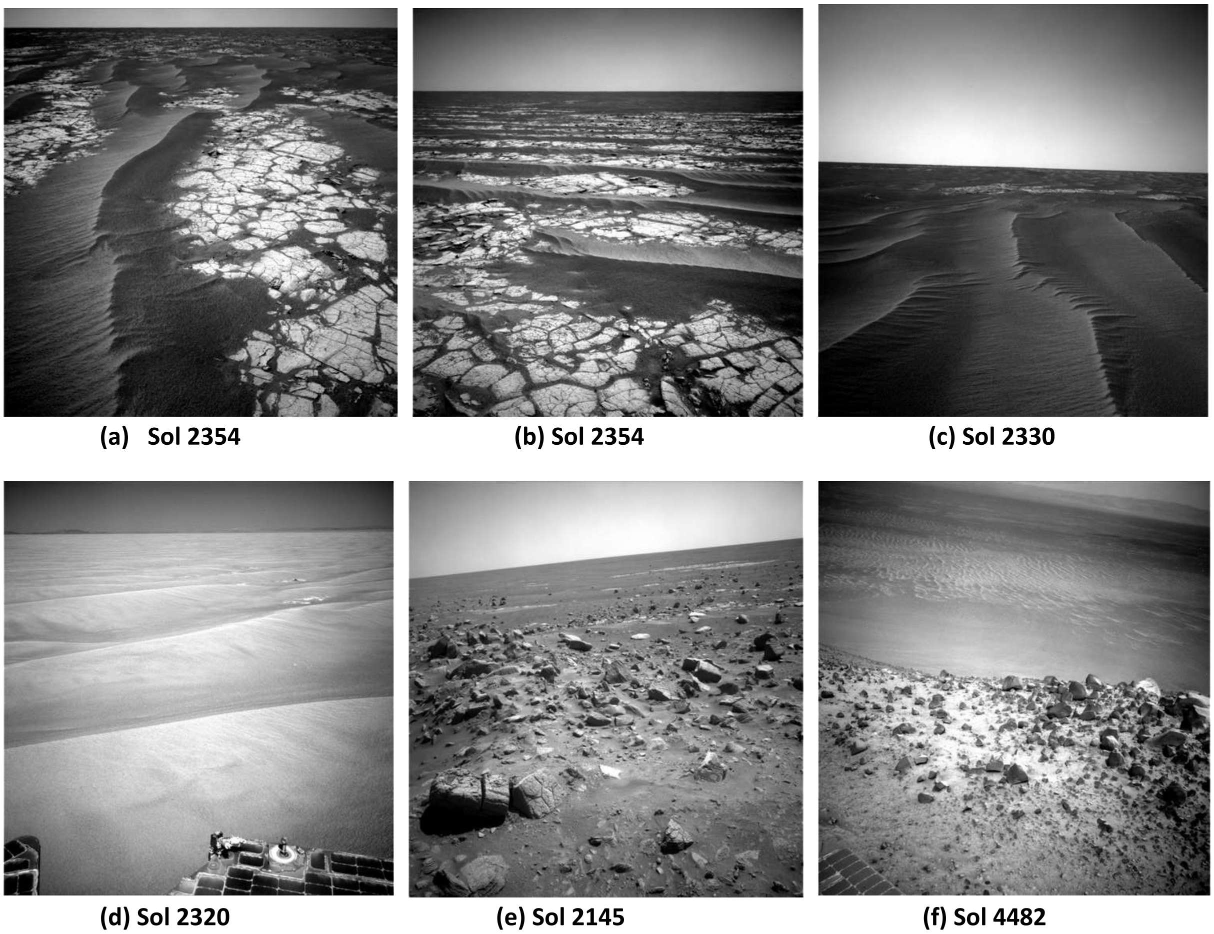
(f) Sol 2174

(g) Sol 2174

(h) Sol 2324

Figure 3-2: Validation Library Image

Courtesy NASA/JPL-Caltech



(a) Sol 2354

(b) Sol 2354

(c) Sol 2330

(d) Sol 2320

(e) Sol 2145

(f) Sol 4482

Figure 3-3: Test Library Images
Courtesy NASA/JPL-Caltech

At the end, providing intensity invariance to the algorithm is equally important. It is achieved by normalizing each image to a value, which is equivalent to the average intensity of all the images in the training set. After this step, the image finally gets added to either the training, validation or testing data sets and thus all the data sets are prepared.

The following Figure (Figure 3-4) illustrates the step by step process of adding an image to a dataset, which was explained in this section.

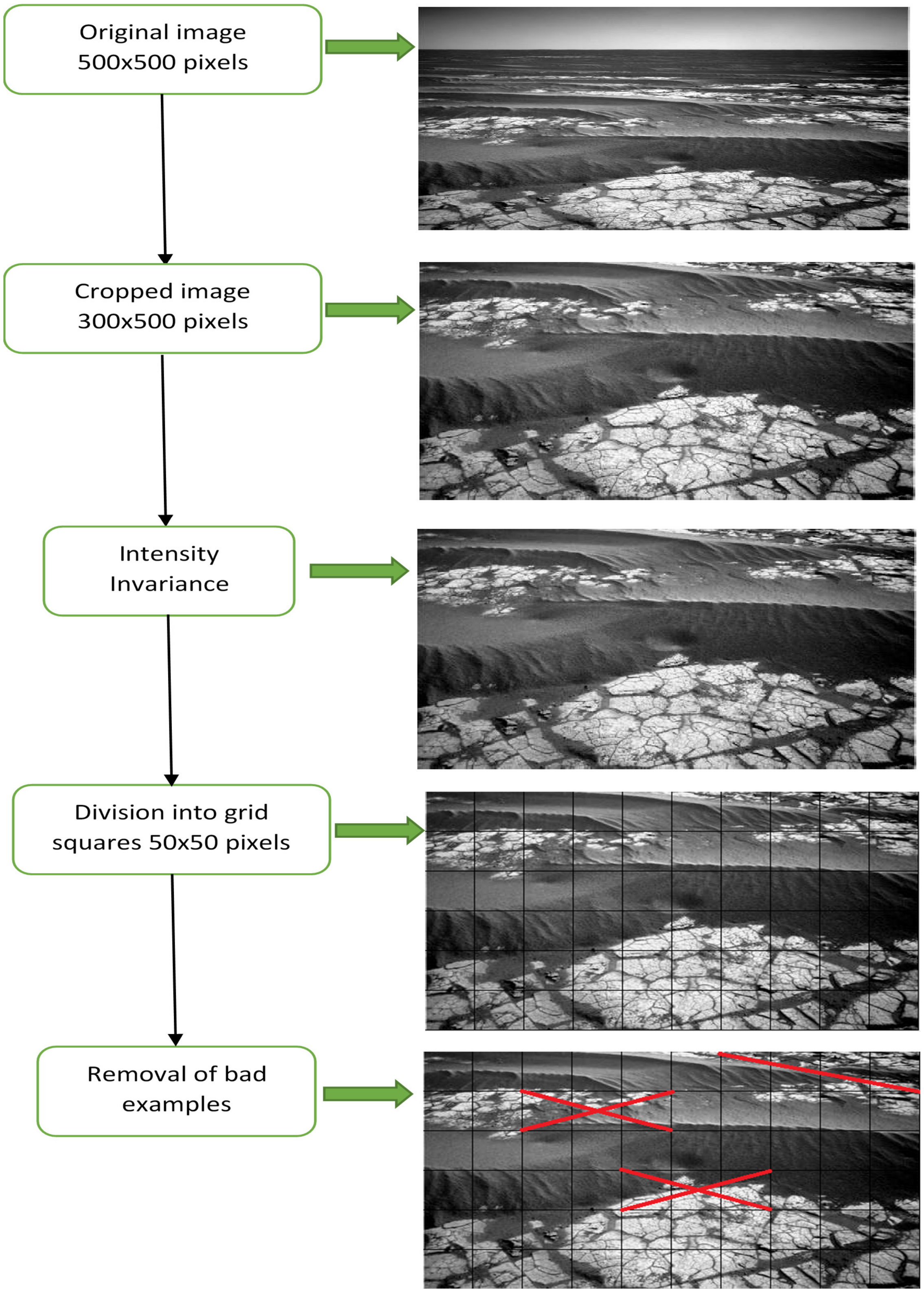


Figure 3-4: Illustration of step-by-step procedure of dataset formation

3.1.2 CSA (Canadian Space Agency) dataset

The Canadian Space Agency mars yard is an emulation of the planetary surface on Mars and it has a large-scale area consisting of all three terrain types (bedrock, gravel & sand) at various slopes. Due to its strong resemblance with the actual martial terrains, it has been selected for the on-site demonstration of the ASAS project and the system is trained with the visual examples from this site. The first step towards the formation of the datasets, is to collect the images from the yard and for the same, two test campaigns were held, where numerous test runs were performed with the rover. The rover captured plentiful images from the mars yard during these test runs. These raw images are then used to form the CSA data sets (training, validation & test sets), which are further used to train the rover for the demo and to validate the feature extraction as well as the classification algorithms. Figure 3-5 shows the examples of how different/similar the CSA mars yard terrains look from those on the actual martian surface.

After collecting the raw images, the same steps; as described in the NASA JPL dataset section; are followed for the pre-processing such as discarding the part of the image that covers the areas which are farther from the rover, intensity invariance, division into grids and removal of bad examples. Here, for the CSA images, the original image size is 720x1280 pixels, which reduces to 500x1000 pixels after discarding the non/less-informative parts of the image and this new image in then divided into 5x10 = 50 grid cells, resulting into the 100x100 pixels of grid squares. These pre-processed grid cells are then added to the corresponding data sets as the individual images. The images which are used to form the training and validation libraries are shown in Figures 3-6 & 3-7**.** These images are shown after all the pre-processing steps, in the form of grid cells. In the training library, the bad examples are removed, which is not the case with the validation library. The crossed grid cells in the training library images are the bad examples which have been removed.

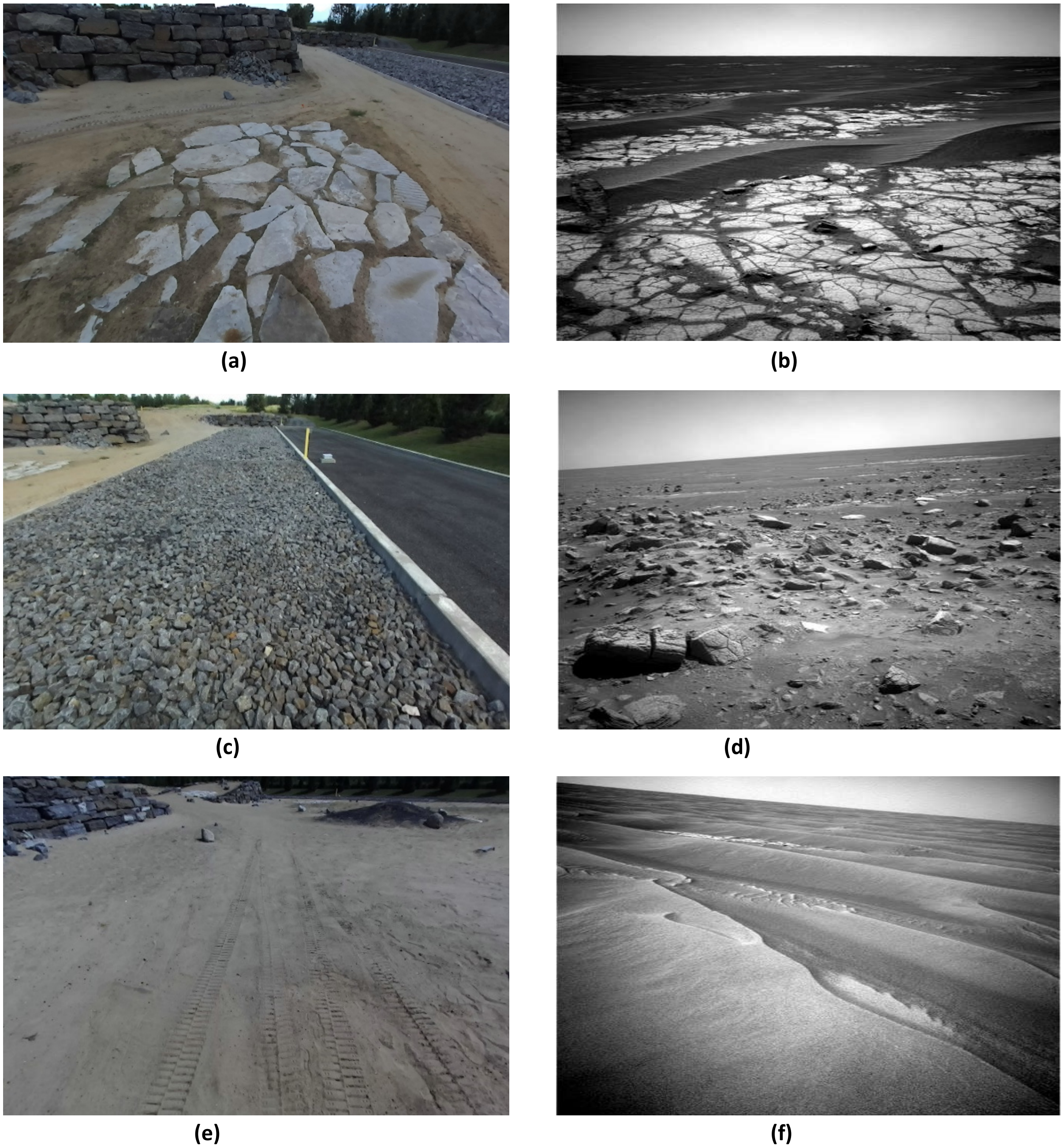
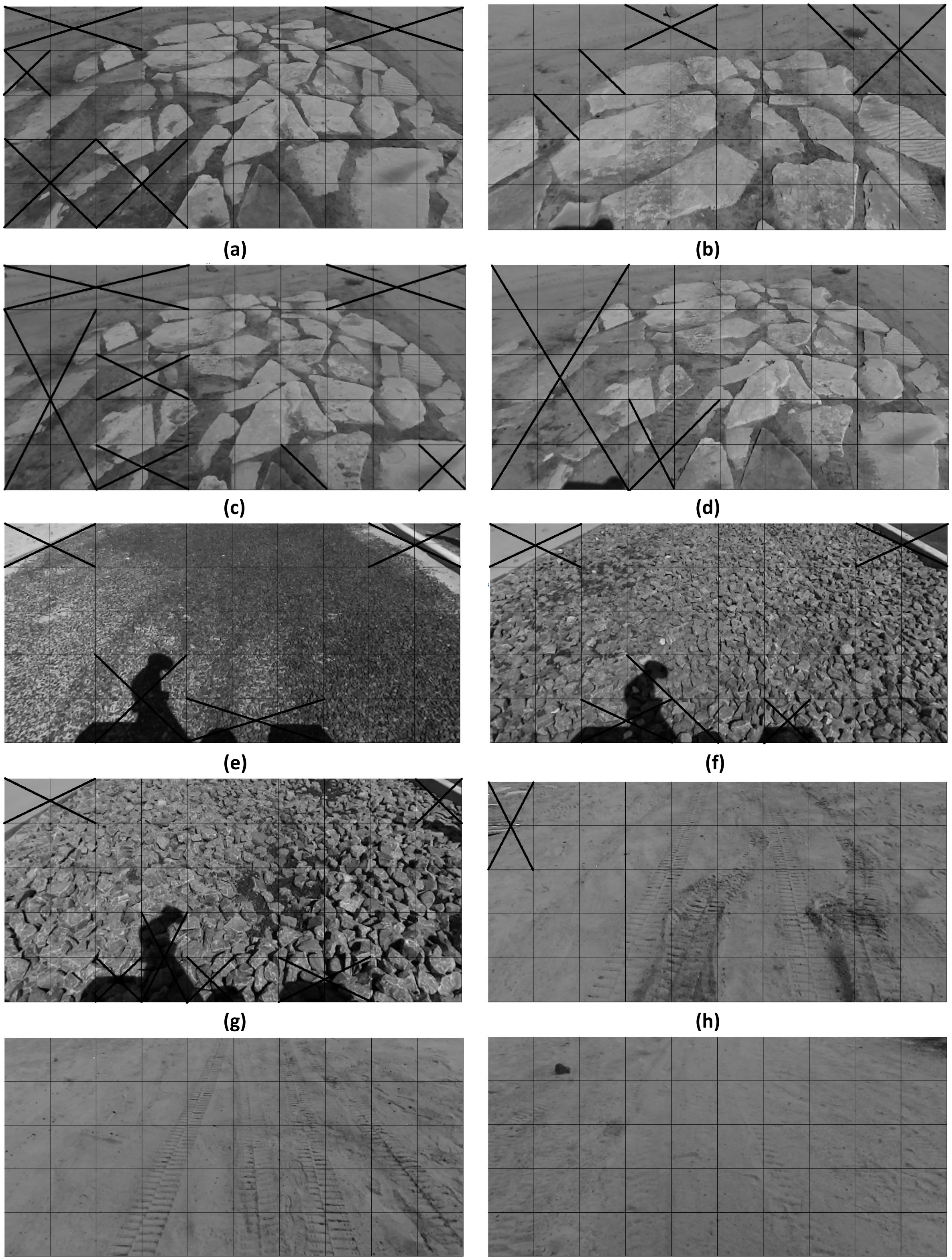


Figure 3-5: (a & b) Bedrock examples from CSA mars yard & JPL NASA website respectively, (c & d) Gravel examples from CSA mars yard & JPL NASA website respectively, (e & f) Sand examples from CSA mars yard & JPL NASA website respectively



(a)

(b)

(c)

(d)

(e)

(f)

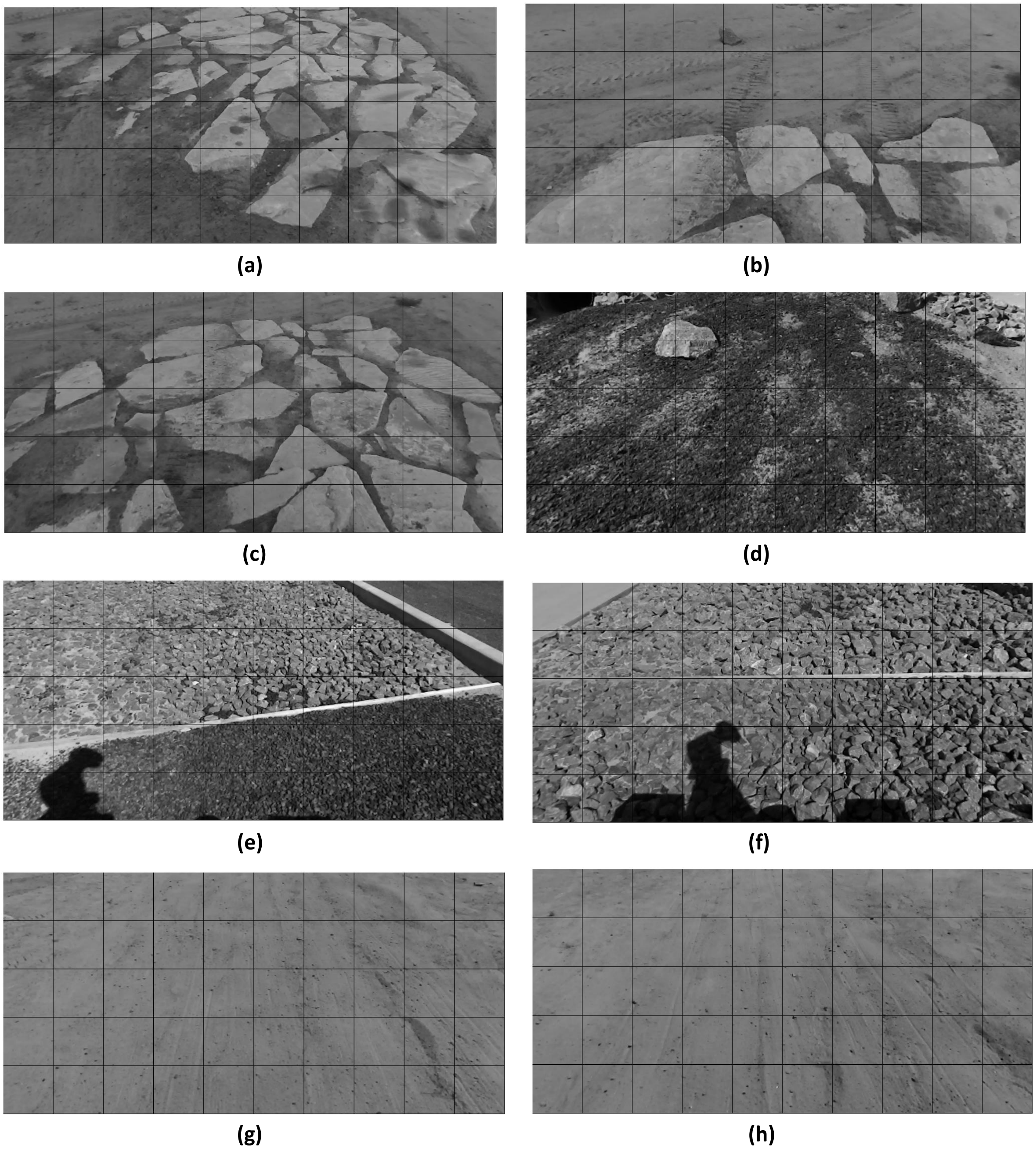
(g)

(h)

(i)

(j)

Figure 3-6: CSA Training library image



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 3-7: CSA Validation library images

3.2 Visual Descriptors

As mentioned above, a visual descriptor is a low dimensional representation of a scene. It formed by converting a full size image into a specific length vector, by extracting various features from the image. Here the features are extracted by filtering the image at various scales and orientations and the length of the vector (visual descriptor) depends on the number of pairs of scales and orientations, used for various filters.

3.2.1 GIST

For the feature extraction, GIST uses various Gabor filters, which are most commonly used for edge detection. Gabor filter is the modulation of a particular wavelength sine wave using a Gaussian filter of selected scale. Examples of Gabor filters with the scale values 4 and 8 as well as the orientations of 0, 45, 90 and 135° are presented in Figure 3-8**.**

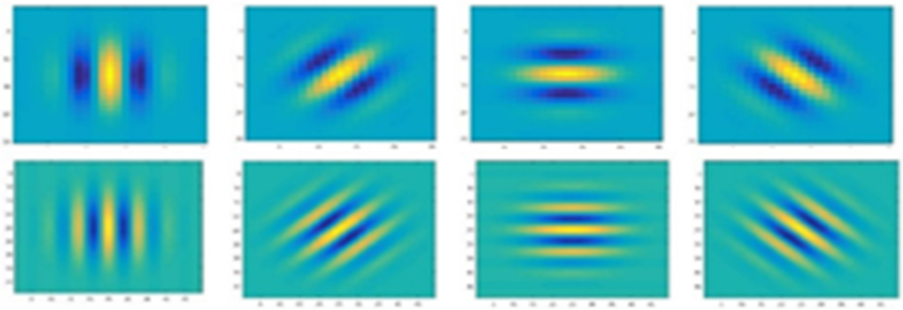


Figure 3-8: Gabor filters at 0, 45, 90 & 135° orientations with scale values 4 (upper line) and 8 (bottom line)

Filtering an image with such Gabor filters at various scales and orientation detects coarse as well as finer edges (depending on scale value) at given orientations from that image. Hence, a filter bank of total 32 Gabor filters, at 4 different scales (1, 2, 4 & 6) as well as 8 different orientations (0°, 22°, 44°, 66°, 88°, 110°, 132° & 154°) is designed. Such expanded selection of scales and orientations can extract wide range of textures from the image. The algorithm for generating GIST descriptor using this filter bank is explained below.

3.2.1.1 Algorithm

Applying this filter bank to any image, gives 32 filter responses for that image, which is being stored in an array named as ‘filter response array’. Further the average value of each response from the array is calculated and these 32 average values forms a 32-bin histogram. The average value is an indication of the strength of edges present in an image, at given scale & orientation. The histogram generated here is itself the GIST descriptor for an image and it incorporates 32 textures, formed by edges at 4 scales and 8 orientations, in the image. Similarly, GIST descriptors for all the images from the training as well as the validation dataset is generated, which can be used later for classification purpose.

3.2.1.2 Design Parameters and Result for JPL dataset

The Gabor filters have certain important parameters, whose selection can be crucial for texture detection, such as scales, orientations of filters, rotational invariance and wavelength selection for the Gabor filters. Hence, one can experiment with different choices for these parameters and analyze their effect on the texture detection, to select their best values. The detailed analysis of choices of parameters & their validation procedure is given below.

I. Scales/Sigma of Gabor filters:

Scale is the parameter associated with the gaussian filter, which is a part of the Gabor filter. Gaussian filter acts as a low pass filter, which removes the high frequency noise and generates a smooth image which can further be processed to extract the texture. Selection of a scale in form of sigma of the gaussian, basically decides the extent of high frequency content to be removed; the higher the scale value is, the more the higher frequencies gets removed. Secondly, it is very important to analyze the image at different scales, as some textures seem to occur at larger scales only and at the same time, it enables the detection of both, finer & coarse edges. Hence, to provide this kind of scale invariance, a set of 4 different values of sigma is being considered for the filter bank design. To select the appropriate set of sigma values, three different sets of sigma are being experimented with, which are:

Set 1: 1, 2, 4, 6 *Set 2:* 4, 6, 8, 10 *Set 3:* 0.5, 1, 1.5, 2.

Using all of these sets of sigma values, the classification results (using NN at this point of time) were acquired for the validation set images. The techniques for classification and their implementation will be explained in section 3.3. The results are summarized, in terms of percentage of correct detection out of total images of that terrain type, in Table 2, shown below.

Sr. No.	Scale Selection	Bedrock	Gravel	Sand
1	0.5 , 1 , 1.5 , 2	44%	46%	72%
2	4 , 6 , 8 , 10	64%	67%	97%
3	1 , 2 , 4 , 6	63%	64%	89%

Table 2: GIST Performance at various scales

As can be seen from the table, a set of "4, 6, 8, 10" sigma seems to work the best for each terrain type among the 3 experimental sets, closely followed by the third set of "1, 2, 4, 6". To select the optimum one between these two sets, further the gaussian response of a random image were analysed at each of these scale values. It can be observed here that the gaussian response at 8 & 10 sigma contains almost no information in the image due to excessive blurring, which is illustrated in Figure 3-9 below. Hence, finally the third set of "1, 2, 4, 6" sigma was selected for the design of Gabor filters to implement the GIST descriptor.

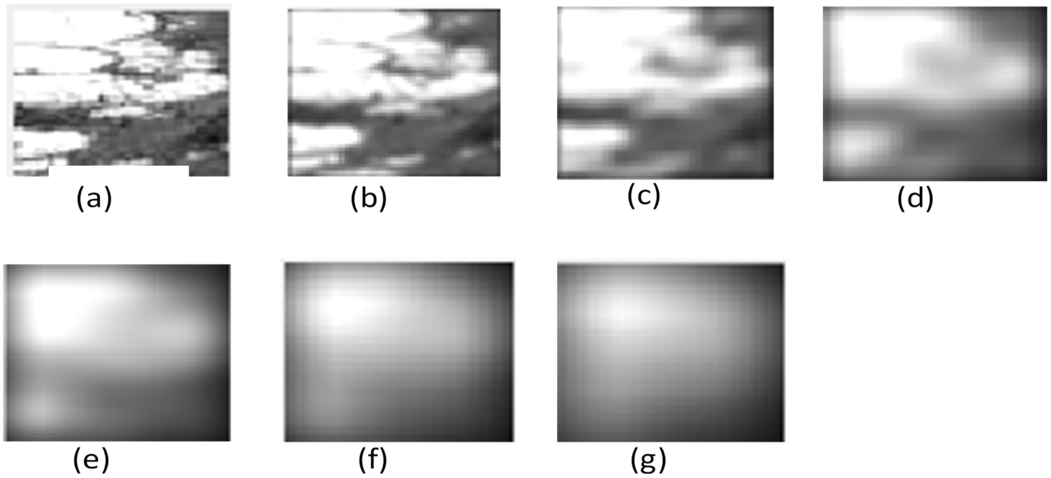


Figure 3-9: (a) Original image and Gaussian responses of the image with sigma = (b) 1, (c) 2, (d) 4, (e) 6, (f) 8 and (g) 10

II. Orientations of Gabor filters:

Gabor filters detect the edges in accordance with their orientations. For example, a filter with 45° orientation will be able to recognize edges in the image which are at 45°. Hence, in order to be able to detect almost all the edges present in the image, Gabor filters for the filter bank are designed with 8 different orientations and to make sure that all the edges get captured by the filter bank, these 8 orientation values are selected to be equidistant within the range of 0-180°. Thus the selected orientations for the filter bank are 0°, 22°, 44°, 66°, 88°, 110°, 132° and 154°.

The reason to restrict these values within the range of 0-180°, rather than 0-360° is that the edges detected at 45° is same as those detected at 225° i.e. -135° and this is true for all the corresponding values of positive and negative 0-180° ranges. This fact can also be verified from the illustration given below in Figure 3-10.

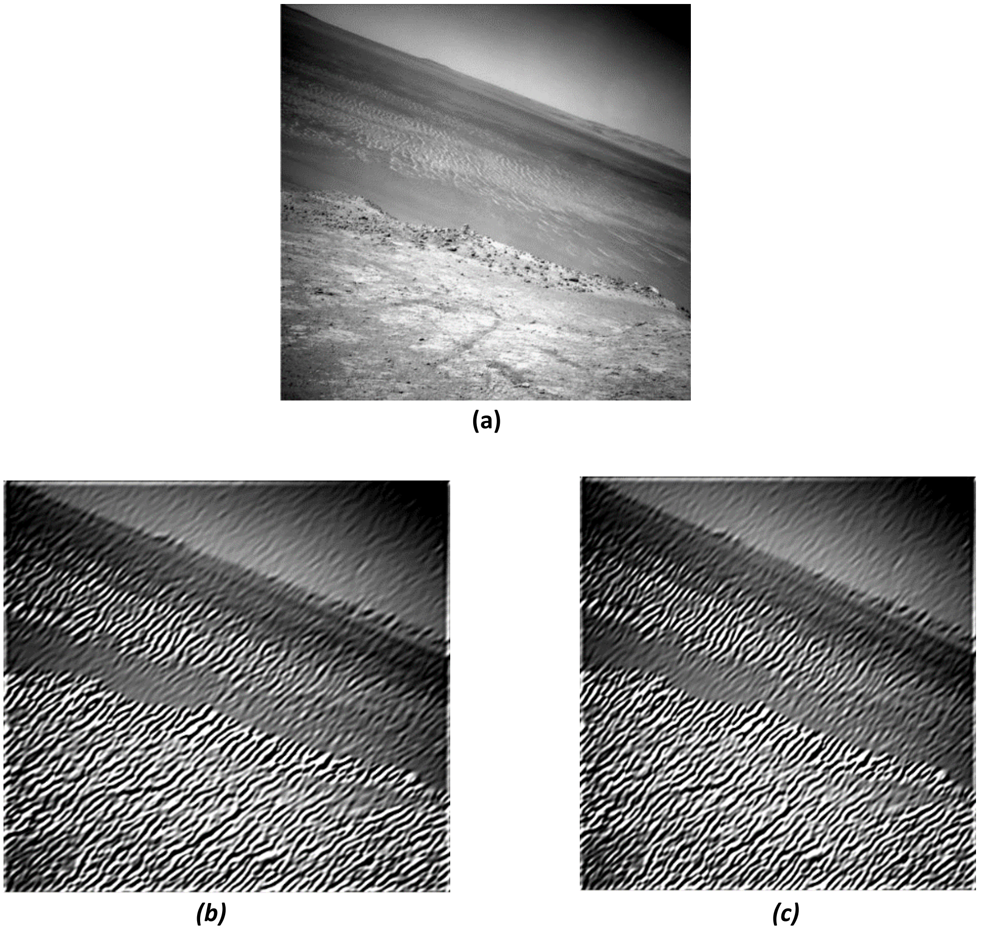


Figure 3-10: (a) Original Image and response of a Gabor filter at (b) 45 and (c) 225 (-135)

III. Gabor filter frequency selection (via wavelength ƛ):

As mentioned above, the sine wave super imposed on the Gaussian filter with a particular scale, forms a Gabor filter. Wavelength of this sine wave decides the frequency of the same and the frequency decides the thickness of the edges being selected by the given Gabor Filter. Thinner edges can be detected if the frequency is higher and thicker edges can be selected if the frequency is lower.

Moreover, for a given Gabor filter, the frequency of the sine wave is also related to the sigma selection for the gaussian filter. As already known, the gaussian filter removes high frequency content in accordance with its sigma selection and if the frequency, selected for sine wave, has already been removed by the gaussian filtering, then in such case, gaussian filter is basically smoothening out all the edges which are supposed to be captured by the Gabor filter and hence, the filter will not be able to extract any information from the image.

Keeping all these things in mind, it has been decided to experiment with 5 different wavelength selections which are ƛ = 4, 8, 16, 2σ, 4σ for scales σ = 1, 2, 4, 6. It can be clearly seen from the table 3 to table 7, that the best results are achieved when the wavelength is selected to be equal to twice of the sigma i.e. ƛ = 2σ.

IV. Rotational Invariance

Another important design parameter for GIST, is rotational invariance (RI). For this application, it is essential for the system to be able to detect all three terrain types, regardless of how the textures are oriented in space and hence the RI (Rotational Invariance) feature has been added to the algorithm. For example, if there is a grid square consisting of sand ripples, one doesn't need to consider the direction of the sand ripples, as long as it can be successfully differentiated from the other terrains. To implement such rotational invariance, firstly the bin with maximum magnitude is found from the histogram and after that taking that bin as a reference i.e. keeping that bin as the 1st bin, the whole histogram is rotated and reformed accordingly. Here also two different RI approaches are being experimented with, which are Local RI and Global RI.

In the first approach i.e. **local rotational invariance**, the histograms are considered separately for each scale and they are rotated by taking the largest bin as a reference. For example, let's say in the histogram for scale 1, 4th bin is the largest one. Here, the histogram bins will be rotated such that it will have the same bins in following order: "4 5 6 7 8 1 2 3". Now secondly let's say in the histogram for scale 2, the 7th bin is the largest one, then the histogram will be rotated to be in the "7 8 1 2 3 4 5 6" order. The same procedure is followed individually, for all four histograms corresponding to 4 different scales and then all of them are concatenated to form a GIST descriptor.

Whereas, in second approach i.e. **global rotational invariance**, firstly the addition vector is calculated, using all four histograms corresponding to 4 different scales and from the addition vector, the largest bin (bin with the maximum magnitude) is found. Now the index of this largest bin is used as a reference for all four scales to rotate their histograms individually. For example, in the addition vector, 4th bin is the largest one, hence, all four histograms are rotated such that their bins are in following order: "4 5 6 7 8 1 2 3". Lastly, by concatenating these rotated histograms, GIST descriptor is formed.

In order to decide the best approach among "No Rotation Invariance (RI)", "Local RI" and "Global RI", series of experiments have been performed for each approach, keeping the constant sigma selection i.e. σ = 1, 2, 4, 6 and varying ƛ to each of the above mentioned values i.e. 4, 8, 16, 2σ, 4σ. Thus, total 15 cases were observed for each terrain class i.e. 5 (ƛ selection) * 3 (RI approaches) = 15.

- **Result Summary for wavelength & RI selections:**

From the tables 3 - 7 shown below, it can be concluded that the Local Rotational Invariance approach along with σ = 1, 2, 4, 6 and ƛ = 2σ provides the best texture recognition.

ƛ / RI	4	8	16	2σ	4σ
No RI	64 %	63 %	59 %	70 %	46 %
Local RI	67 %	70 %	62 %	70 %	52 %
Global RI	70 %	69 %	63 %	68 %	51 %

Table 3: GIST performance for Bedrock

ƛ / RI	4	8	16	2σ	4σ
No RI	64 %	64 %	57 %	64 %	31 %
Local RI	57 %	68 %	56 %	70 %	42 %
Global RI	59 %	69 %	57 %	63 %	40 %

Table 4: GIST performance for Gravel

ƛ / RI	4	8	16	2σ	4σ
No RI	87 %	84 %	63 %	92 %	61 %
Local RI	80 %	94 %	64 %	91 %	54 %
Global RI	79 %	96 %	66 %	92 %	60 %

Table 5: GIST performance for Sand

ƛ / RI	4	8	16	2σ	4σ
No RI	72 %	70 %	60 %	75 %	46 %
Local RI	68 %	77 %	61 %	77 %	49 %
Global RI	69 %	78 %	62 %	74 %	50 %

Table 6: Average GIST performance for all three terrains

ƛ / RI	4	8	16	2σ	4σ
No RI	64 %	63 %	57 %	64 %	31 %
Local RI	57 %	68 %	56 %	70 %	42 %
Global RI	59 %	69 %	57 %	63 %	40 %

Table 7: Minimum GIST performance for all three terrains

V. Final Scale Validation:

The selection of the design parameters started with selecting the scale values for the Gabor filters and at that point of time, the other parameters were not selected. So, while performing initial experiments for scale selection, some of the other parameters viz. Gabor wavelength and RI approach were selected randomly and temporarily, their values were taken at ƛ = 8 and No RI approach. It is worth mentioning here that the values of orientations were always the same at above mentioned values. These temporary selections brought out the best results for the scale set of "1, 2, 4 & 6". However, after this initial scale selection, experiments were also performed for other parameters as well, to select between the different values and it turned out that the nest classification results can be achieved using ƛ = 2σ and Local RI approach, rather than the previously selected temporary values for these parameters. Hence, the finalized values for wavelength and RI approach are different than those taken during initial scale selection.

As a result, it is necessary to validate the scale selections with the finalized values and hence, again the experiments were performed for all three set of sigma, keeping other parameters at the finalized values (i.e. ƛ = 2σ, Local RI approach). From the results, shown in Table 8 below, it is evident that the initial selection for the set of sigma "1, 2, 4, 6", still works the best as it seems to be giving the best classification results.

Sr. No.	Scale Selection	Bedrock	Gravel	Sandy
1	0.5 , 1 , 1.5 , 2	64%	47%	74%
2	4 , 6 , 8 , 10	59%	64%	93%
3	1 , 2 , 4 , 6	70%	70%	91%

Table 8: Final Scale validation for GIST

3.2.1.3 Result summary for CSA dataset

In the efforts of achieving the best possible results for the NASA JPL dataset, there were number of experiments performed for selecting the appropriate values for the designing parameters and the methods. Further, the same selections for the designing parameters and the methods have been maintained with the CSA dataset as well, for the feature extraction using GIST. This way it is also possible to verify whether the same algorithm can work for two different datasets, which are similar to each other up to some extent. The success ratio in the detection of various terrains acquired using GIST for CSA dataset is given in the following table 9.

Terrain Types	Success ratio with GIST
Bedrock	76.5%
Gravel	98%
Sand	93%

Table 9: GIST Performance for CSA dataset (Local RI, $\lambda = 2\sigma$, σ set = 1, 2, 4, 6)

3.2.2 Histograms of Oriented Gradients (HOG)

HOG (Histograms of Oriented Gradients) [36], is a visual descriptor, which uses the gradient information for texture recognition, instead of filter responses. The gradient attributes are acquired at pixel level in any image and there are mainly two of them: Magnitude of the gradient and Orientation of the gradient. These attributes indicate the amount & direction of the intensity changes at given pixel, respectively and hence it is possible to extract texture information in form of edge detection using gradients.

3.2.2.1 Algorithm

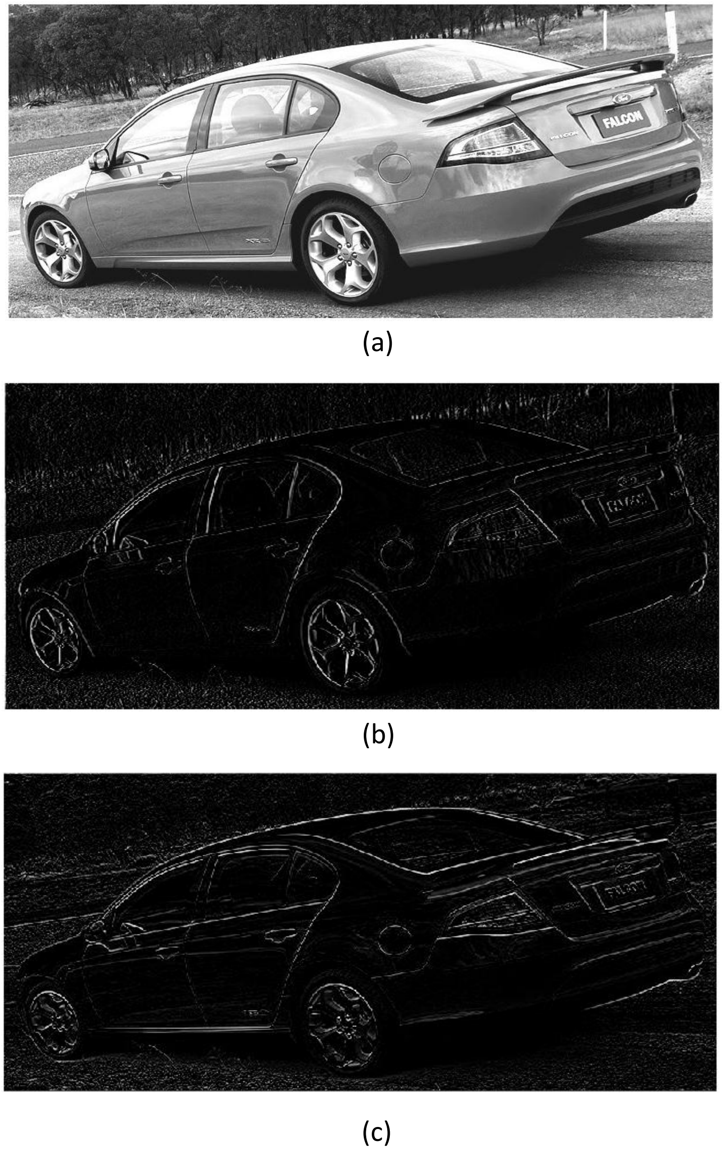
The first step for generating the HOG descriptor for any image is to calculate the gradient for the given image. To calculate the gradient, the image is filtered by the derivative masks, one in horizontal direction [-1 0 1] and the other in vertical direction [-1 0 1]'. This gives x & y derivatives of the image. Illustration of these X & Y derivatives of an exemplary image is given below in Figure 3-11. If the x & y derivatives of an image are called Dx and Dy respectively, then the magnitude & orientation of a gradient at each pixel of an image can be calculated as below:

$$\textbf{Magnitude} = \sqrt{Dx^2 + Dy^2}$$

$$\textbf{Orientation = atan2 (Dy, Dx)}$$

In the present system (ASAS), for a given image, the gradient is being found at four different scales, for the same reasons as considered in the case of GIST, i.e. to detect the finer as well as coarse edges and to capture the textures that occur only at certain scales. The experiments and the final selection for these scale values are explained below in the section 3.2.2.2. Moreover, to generate the 32 bins histogram (descriptor) and to cover the entire range between 0 to 180°, the similar orientations as GIST are selected for each of the scaled versions as well, i.e. 0°, 22°, 44°, 66°, 88°, 110°, 132° and 154°. Each bin of the histogram is considered to be for a particular orientation, for corresponding scaled version.

After calculating the gradients, for all four scaled versions of an image; their magnitude and orientations are scanned at pixel level. For each pixel of a scaled version, its gradient magnitude gets added to any one of the 8 histogram bins of that version, according to its gradient orientation. The distribution of the orientations in the histogram bins is explained in detail in the next section. The same scanning is performed for all four scaled versions and this way four different 8 bins histograms are generated, which can be concatenated to form a final 32 bins histogram (HOG descriptor). This flow of the algorithm is also illustrated in Figure 3-12



(a)

(b)

(c)

Figure 3-11: Example of x & y derivatives of an image (a) original image, (b) x-derivative, (c) y-derivative

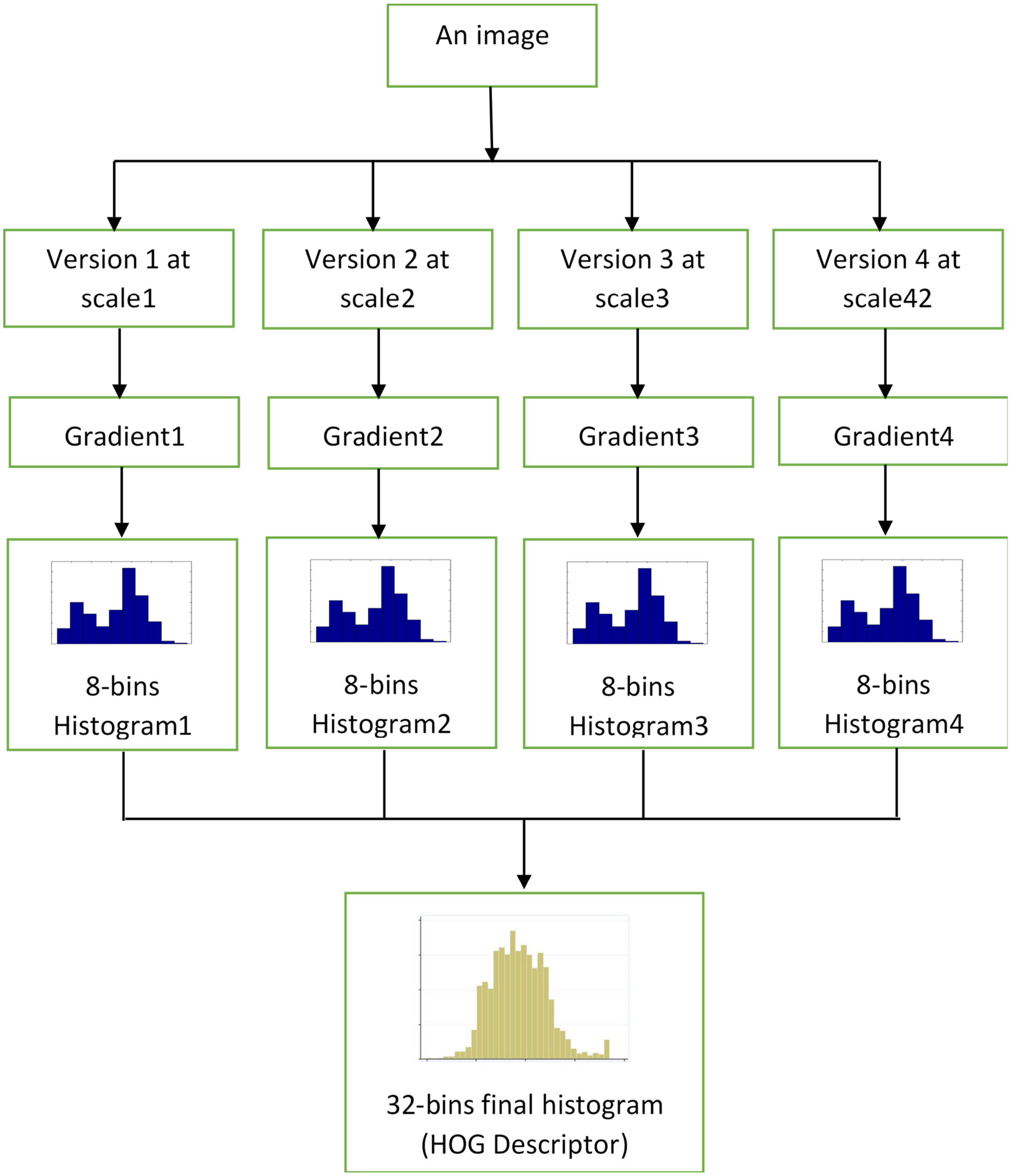


Figure 3-12: Illustration of the HOG algorithm

3.2.2.2 Designing Parameters and Result for JPL dataset

Similar to GIST, there are various design parameters in HOG as well, which can be tuned to achieve better feature extraction. These parameters are scale, orientations, rotational invariance etc. and by experimenting with different choices for these parameters and analyzing their effect on texture detection, one can select their best values for the application. The detailed analysis of choices of parameter & their validation procedure is given below.

I. Scale invariance:

To achieve scale invariance (i.e. if two images are at the different scale but having same texture information, then they can be recognized to be of the same texture class) and to be able to detect finer as well as coarse edges, the image needs to be analyzed at different scales, as discussed before. To make this comparable to GIST, the experiments were performed with the same set of sigma as GIST i.e. **i)** 1, 2, 4, 6 **ii)** 4, 6, 8, 10 **iii)** 0.5, 1, 1.5, 2.

However, unlike in the GIST algorithm, here there is a fixed vector as a derivative mask, whose window size cannot be increased. Hence, one option is to scale down the image proportionally. The original image size is considered to be equivalent to 0.5 scale and for other scales, image size is reduced with the base of 0.5. For example, if one needs to scale the image at 1 sigma, it is resized by multiplying with (0.5/1) i.e. dividing by 2 and so on for other scales.

The table below illustrates the result summary for the experiments using above mentioned sigma sets. It can be concluded from the results that the set “1, 2, 4, 6” provides best performance and the same is used for further experiments.

Scale Selection	Bedrock	Gravel	Sand
0.5 , 1 , 1.5 , 2	59 %	54 %	91 %
4 , 6 , 8 , 10	52 %	57 %	70 %
1 , 2 , 4 , 6	67 %	62 %	94 %

Table 10: HOG Performance at various scales

II. Gradient Orientation Range:

Calculating the orientation using ‘atan2’ function, gives the result in form of 0 to π and – π to π. However, in case of GIST, it was observed that the response for a particular angle and its counterpart in the opposite quadrant of Cartesian plane, is same i.e. response for 45 and -135 are same, hence – π to π range is not significant. The same is applicable for HOG as well, and hence all the orientations in the range of – π to π are converted into 0 to π, by adding π to them.

III. Centralized Histogram Bins:

For the formation of 8 histogram bins, the pixels are distributed according to their orientation, considering each bin to be a particular angle. Hence, one needs to divide the whole range of possible pixel orientations i.e. 0 to 180 degree, into 8 equal parts, giving a range of 22 degrees to each bin and one can take the same angles as in GIST i.e. 0°, 22°, 44°, 66°, 88°, 110°, 132° and 154°.

However, if the case of GIST is reconsidered, every histogram bin is actually a filter response at each of the following orientations, i.e. 0°, 22°, 44°, 66°, 88°, 110°, 132° and 154°. Thus considering the 2nd bin i.e. at 22°, it not only contains the outputs at 22° but also contains the outputs at the angles on both the sides of 22° up to certain extent. So, virtually one can see each bin is having center at one of the above mentioned angles in GIST. In order to implement the similar distribution in HOG, the centralized bins approach is used where, if the 2nd bin is at 22° then the gradient magnitude of all the pixels with gradient orientation from 11° to 33° gets added to the 2nd bin, so that it is centralized at $\frac{11+33}{2} = 22°$. Similarly, 33° to 55° is the range for 44° bin and so on.

IV. Rotational Invariance:

For the same reason as explained in GIST, rotational invariance is implemented in HOG as well. The experiments and analysis of the results for approaches “No RI”, “Local RI” and “Global RI” is also performed. The procedure is same as explained in GIST and the result summary is illustrated in table 11. Though there is no significant difference in the results for different RIs, Local RI seems to be achieving the best average performance among all three approaches, for all 3 terrains and hence it has been used, along with the sigma set “1, 2, 4, 6”.

RI Approach	Bedrock	Gravel	Sand	Average
No RI	67 %	63 %	94 %	74.66%
Local RI	67 %	71 %	90 %	76%
Global RI	62 %	65 %	96 %	74.33%

Table 11: HOG Performance for various RI approaches

V. Final Scale Validation:

Similar to GIST, here also, it is necessary to validate the scale selections with the finalized design parameters and procedures (Histogram Normalization) and hence, again the experiments were performed for all three set of sigma with histogram normalization and Local RI approach. From the results, shown in Table 12 below, it is evident that the initial selection for the set of sigma "1, 2, 4, 6", still works the best as it seems to be giving the most consistent classification results.

Sr. No.	Scale Selection	Bedrock	Rock Strewn	Sandy
1	0.5 , 1 , 1.5 , 2	56%	61%	92%
2	4 , 6 , 8 , 10	58%	54%	75%
3	1 , 2 , 4 , 6	67%	71%	90%

Table 12: Final Scale Validation for HOG

3.2.2.3 Result Summary for CSA dataset

Similar to what was done in the case of GIST, all the selections for the design parameters values and the methods have been maintained the same as they were finalized for the NASA JPL dataset and the results acquired using the same for the CSA dataset is shown below in Table 13.

Terrain Types	Success ratio with HOG
Bedrock	80%
Gravel	100%
Sand	97%

Table 13: HOG Performance for CSA dataset

3.2.3 TEXTONS

Texton based texture recognition aims to represent an image in form of a histogram that is a set of visual words, in other words a set of Textons. The Textons are basically the histograms, generated by clustering of the filter response vectors that were obtained by applying a filter bank to an image.

3.2.3.1 Algorithm

I. Filter Bank:

The implementation of this method (Texton based texture recognition) starts off with the design of a filter bank and two significantly essential factors to be considered for the same are:

A. To design a filter bank, which provides a rotationally invariant measure of textures. This makes our method compatible with HOG & GIST.

B. To design a filter bank, which does not have the property to not respond at all to oriented structures in the image.

Keeping these points in mind, the “Maximum Response (MR8) Filter bank” has been finalized, which includes one Gaussian & one LOG filter as well as edge and bar filters, at 3 different scales and 6 different orientation. Thus, the filter bank consists of total 38 filters [39]**.** Secondly, for being able to compare the performance of Textons with that of GIST and HOG, the scale and orientation selection should be comparable with GIST & HOG as well. However, here one has to select only 3 scales and 6 orientations, unlike HOG & GIST (which had 4 scales and 8 orientations). Therefore, it has been decided to proceed with the first 3 scale selections of the previous cases i.e. 1, 2, 4, whereas, for orientations, the 0 to π range is divided into 6 equal parts (instead of 8) i.e. 0°, 30°, 60°, 90°, 120° & 150°.

All the filters of the filter bank are illustrated below in Figure 3-13.

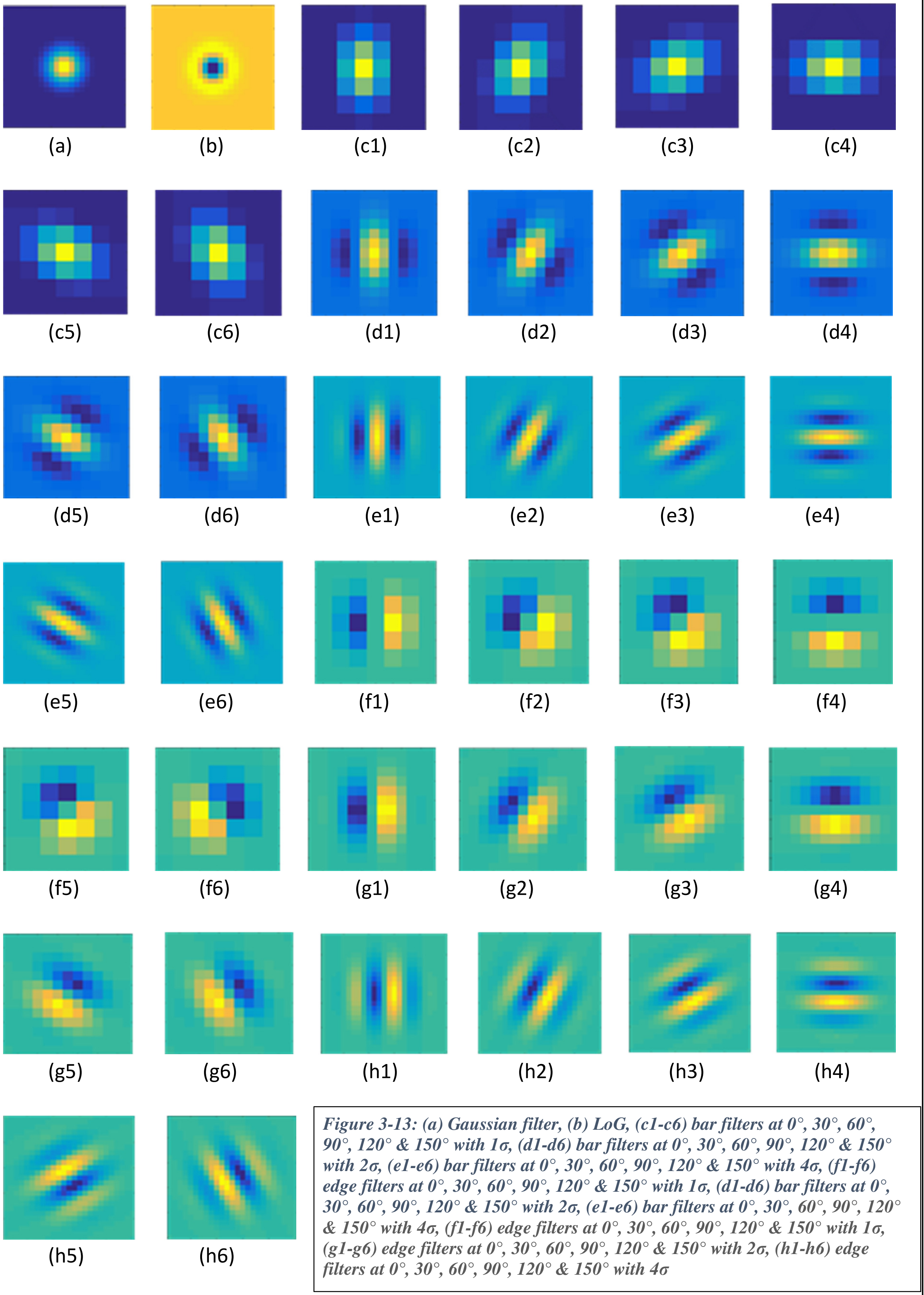


Figure 3-13: (a) Gaussian filter, (b) LoG, (c1-c6) bar filters at 0°, 30°, 60°, 90°, 120° & 150° with 1σ, (d1-d6) bar filters at 0°, 30°, 60°, 90°, 120° & 150° with 2σ, (e1-e6) bar filters at 0°, 30°, 60°, 90°, 120° & 150° with 4σ, (f1-f6) edge filters at 0°, 30°, 60°, 90°, 120° & 150° with 1σ, (d1-d6) bar filters at 0°, 30°, 60°, 90°, 120° & 150° with 2σ, (e1-e6) bar filters at 0°, 30°, 60°, 90°, 120° & 150° with 4σ, (f1-f6) edge filters at 0°, 30°, 60°, 90°, 120° & 150° with 1σ, (g1-g6) edge filters at 0°, 30°, 60°, 90°, 120° & 150° with 2σ, (h1-h6) edge filters at 0°, 30°, 60°, 90°, 120° & 150° with 4σ

II. Texton Library:

As mentioned above, Textons are basically the clusters of filter responses obtained from various images. In other words, each Texton represents a certain texture, which exists in one of the images whose filter response is used in clustering. Hence, as the name suggests, a set of such Textons forms a Texton library and this Texton library will further be used to generate the visual descriptor or the histogram, which is explained in the next section. In the following paragraph, the formation of Texton library is explained.

The first requirement was to induce rotational invariance in the descriptor and in order to achieve the same, from each set of edge and bar filters, which consists of filters at 6 different orientations with a particular scale value, only the maximum filter response over orientations is being used. Here is an example to explain the above statement. In a filter response vector, first two elements are the response of Gaussian & LOG filters. After that, there are 3 sets of filter responses for edge filters, each of which contains 6 response vectors pertaining to six orientations at a particular scale value (1σ, 2σ, 4σ). Again, there are 3 sets of filter responses for bar filters, similar to the edge ones. Now for the formation of the maximum response (MR) vector, first two responses is taken as it is i.e. 1st and 2nd element of the MR vector are the gaussian and LoG responses. Secondly, from each set of the edge filter responses, an orientation is found for which the response is maximum and that maximum response is added to the MR vector as 3rd, 4th & 5th elements. Similar process is executed with bar filter responses and only 3 maximum responses over orientations forms 6th, 7th & 8th elements of the MR vector. This forms the MR8 vector (formed using MR8 filter bank), that induces the rotational invariance property in the algorithm. The whole procedure of forming the MR vector is illustrated below, in the Figure 3-14.

This way an MR vector is generated for each and every pixel of an image, when filtered using the MR8 filter bank. As mentioned in the "Dataset" section, each grid square is considered as a training image, which is 50x50 pixels in size. If a matrix is formed using the MR vectors of all the pixels of an image, then each image in the training library gets a MR matrix of size 2500x8 and if there are total M images in the training library, an Mx1 cell will be generated for the entire library, where each element of the cell is 2500x8 MR matrix for each training image.

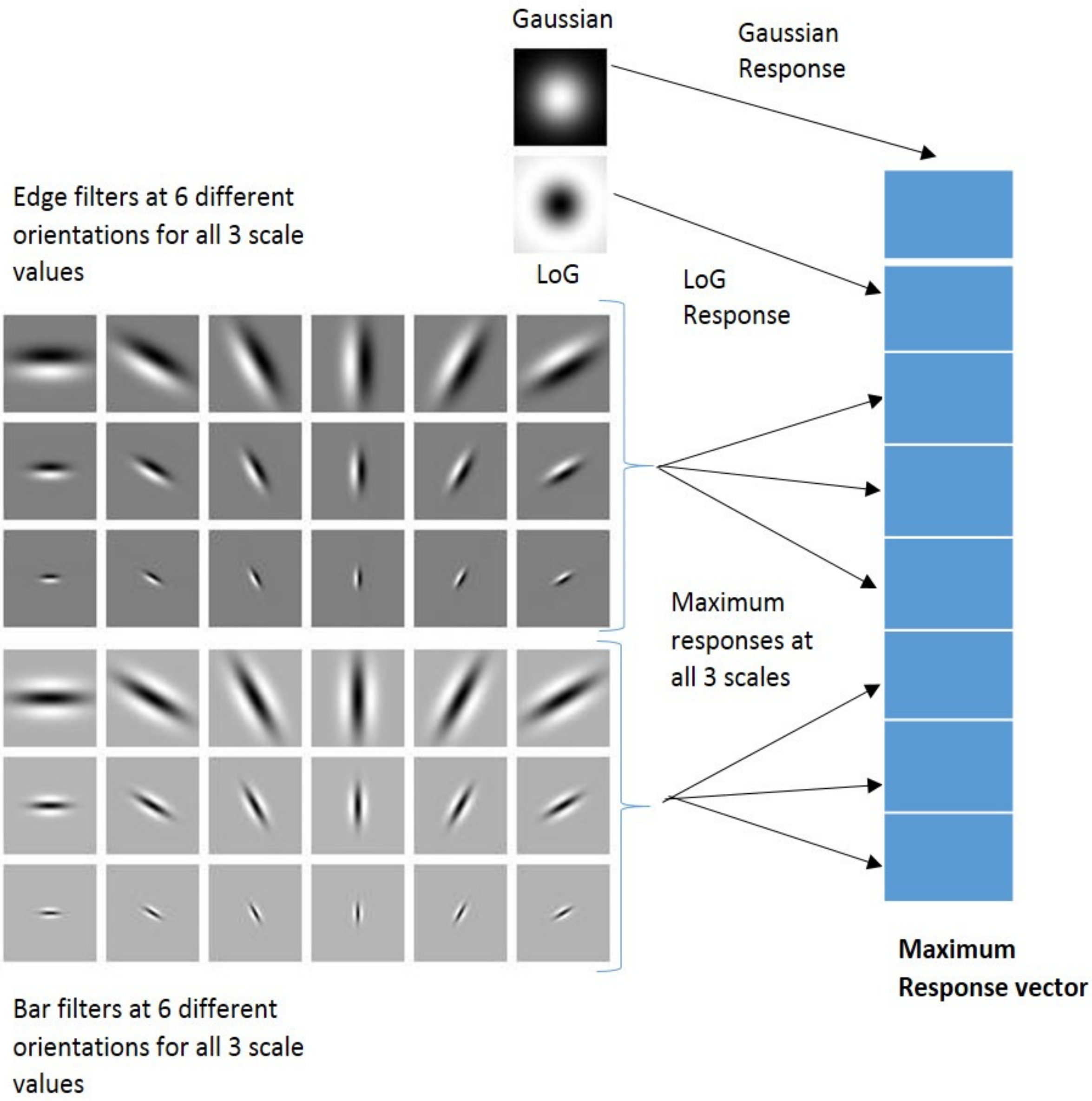

Gaussian
Gaussian Response
Edge filters at 6 different orientations for all 3 scale values
LoG
LoG Response
Maximum responses at all 3 scales
Maximum Response vector
Bar filters at 6 different orientations for all 3 scale values

Figure 3-14: Formation of maximum response (MR) vector using MR8 filter bank

Now, the final goal here is to generate the Texton library by clustering the filter responses of the training images. However, the point of the Texton library is to approximate all the variety of textures that occur in the real world with a finite-set of Textons and using all the filter responses from all the MR matrices of training library is tremendously expensive from the computational point of view. Hence, instead of using all the filter responses in the clustering process, the filter responses at 100 random pixels are extracted from each image in the library. In other words, in

Mx1 cell, from each MR matrix (2500x8 size), 100 rows are selected randomly, which are actually the filter responses for 100 random pixels. These randomly selected rows are concatenated into an enormous matrix called “GiantMatrix”, which results into a matrix of size (M*100) x 8. For example, if training library has 470 images, the size of the GiantMatrix will be of 47000X8 pixels. Further, in order to generate the Texton library, k-means clustering is performed on this GiantMtarix and 32 cluster points are generated, which are nothing but the actual Textons. A set of these 32 Texton vectors is called the “Texton Library”, which will be of the size 32x8 matrix and each row of this matrix represents one Texton. The reason for choosing to generate 32 Textons is the fact that the visual descriptor generated using Textons should be as comparable as possible with the other two descriptors (GIST & HOG), for the performance evaluation. The descriptors generated using GIST and HOG contain 32 bins and the histogram of the same number of bins can be generated by creating the Texton map of an image using 32 Textons. The process of generating the Texton map and in turn, the histogram of an image using the Textons is explained in the next paragraphs.

III. Texton Map and The Descriptor

In order to generate the Texton map, the first step is to derive the MR matrix for an image, by following the same procedure explained before. Once the 2500X8 MR matrix is generated, the MR vector of each pixel is compared with all 32 Textons from the Texton library. By finding Euclidian distance between the MR vector and the Texton vector, one can find the closest matching Texton for that pixel and then, the index of that Texton is assigned to that pixel. For example, if the closest matching Texton to a pixel is the 20th one, that pixel will get 20 as a value. This way all the pixels and every pixel gets a value between 1 & 32. This image with pixels having Texton numbers as their value is “Texton map”. At last, a 32 bin histogram is generated by counting the number of pixels associated with every Texton index and this histogram is called the “Texton Descriptor”. The entire algorithm is briefly illustrated in the Figures 3-15 and 3-16.

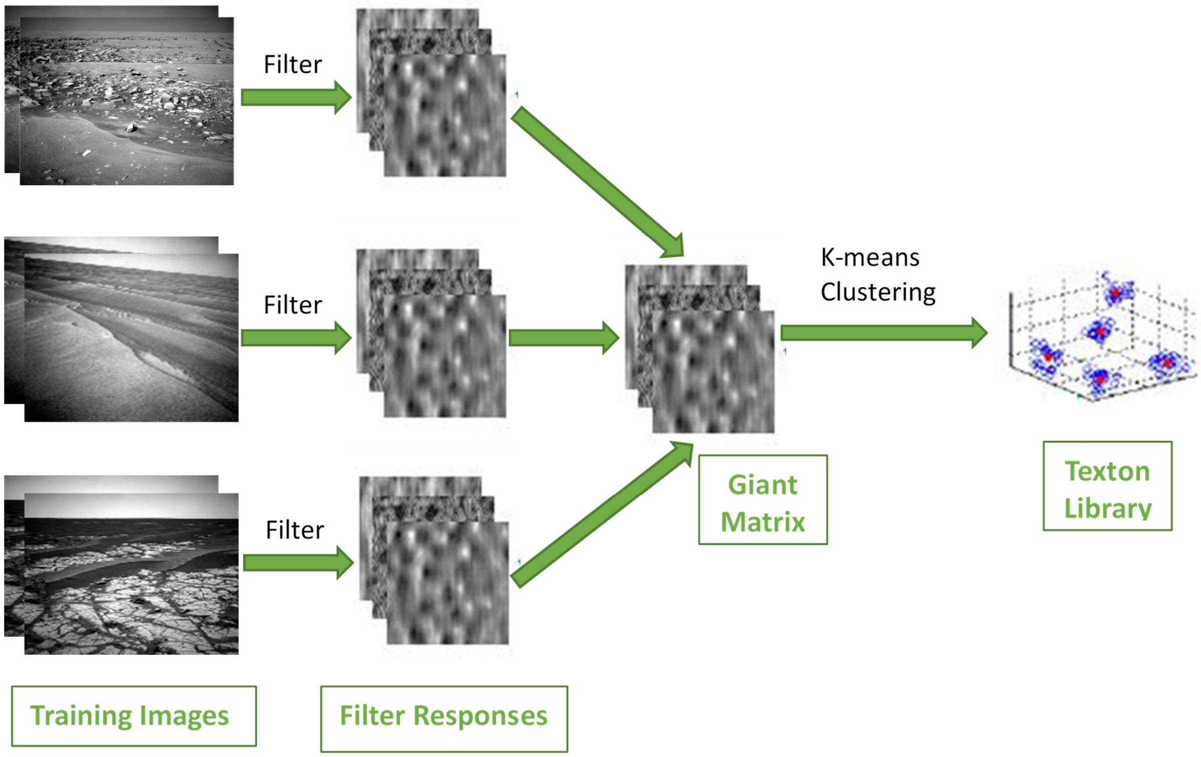


Figure 3-16: Illustration of Texton Library Generation

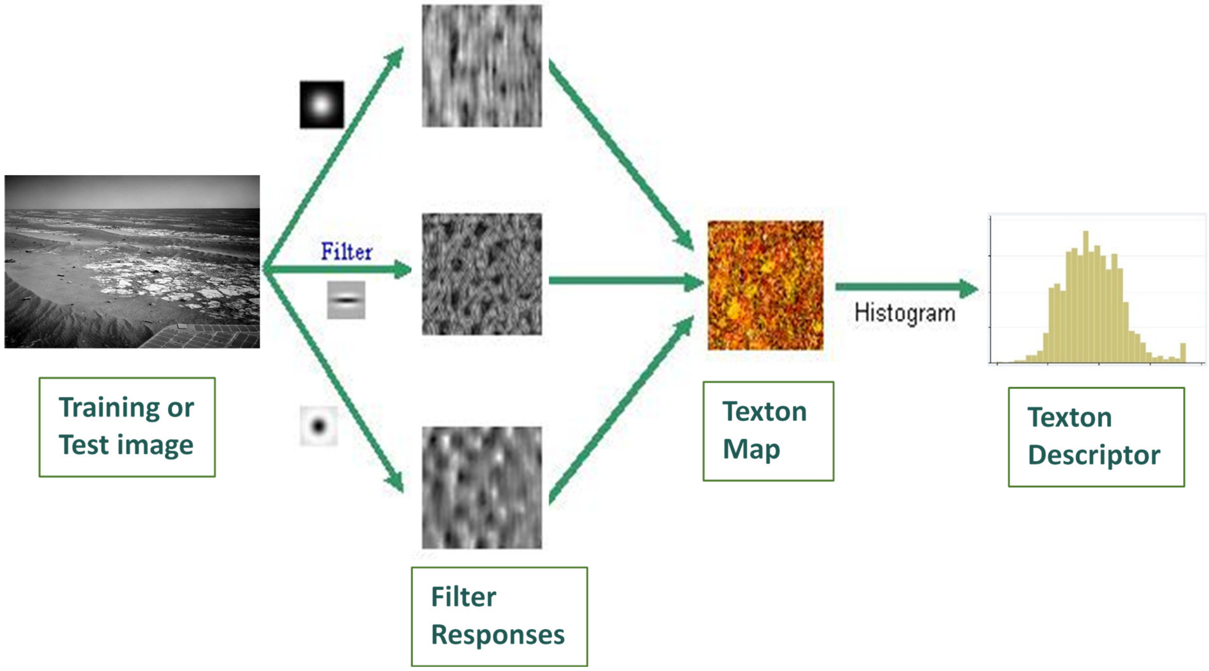


Figure 3-15: Illustration of Texton Descriptor Generation

3.3 Classification using Histogram Matching

Once the histograms for the training data set is obtained using any of the descriptors (explained in the previous section), the competency of that descriptor in recognizing different texture information can be determined by acquiring the ratio of correct terrain classification for that descriptor, using histogram matching. Such histogram matching can be performed between the histograms of the training set and the validation or the test data set.

There are several methods for histogram matching; one of which is "Minimum Intersection Method" [40]. In this method, two histogram vectors (one from the training library and the other for the test image) are compared and for each set of corresponding bins from these vectors, a minimum value is found out and added to an additional vector, which is called the minimum intersection vector. Once this intersection vector is formed, all of its 32 bins are added into a summation value. The higher is this summation value, the higher is the probability of matching for the compared histograms (images). The illustration of this method is shown in Figure 3-17**.**

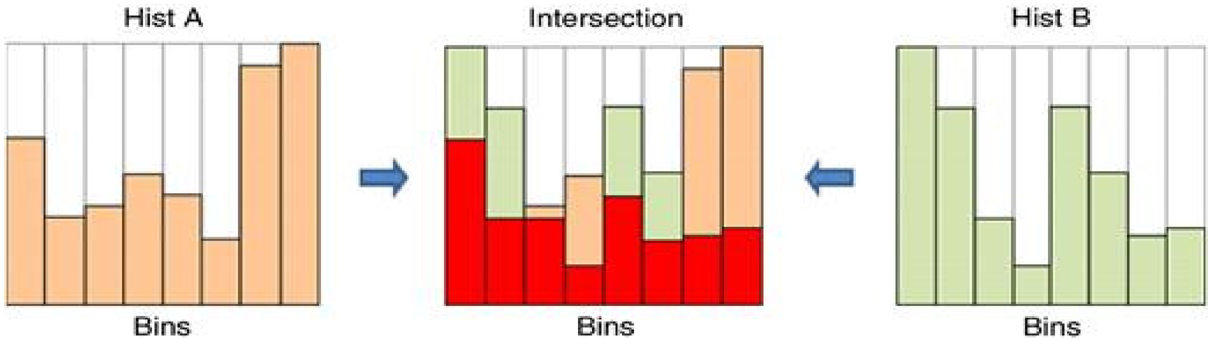


Figure 3-17: Illustration of Minimum Intersection Method, From [51]

However, while implementing and analysing the method for this application, one problem was recognized. For the explanation of the same, let's take an example of an 8 bin histogram. Let's say the histogram of a test image is as below:

3	2	5	1	4	2	4	1

Now, let’s say that the histograms of three different training library images are as below:

10	9	9	7	6	8	12	12

7	19	9	17	16	8	11	12

6	5	7	7	6	8	7	9

In this case, when the intersection histogram is found for the test image with respect to the training library images, it will be the same as that of the test image as the test image histogram has the minimum values in all the bins with respect to all three training histograms. Thus, in such cases, because of having the same summation value for more than one training images, it is not possible to find one best matched image for the test image and hence, the reliable terrain classification cannot be achieved using this method.

The second most common method for histogram matching is “Euclidean Distance Method”. As the name suggests, the euclidean distance is found between the two histogram vectors, being compared with each other. If the histogram of a test image is being compared with all the histograms of the training library images, then the one, with which the euclidean distance of the test histogram is the smallest, is counted as the best match for given test image. Such histogram matching method in turn implements the “Nearest Neighbour Method” of classification.

In this classification method, for any test histogram vector X, a training histogram vector Y_i is found, where the euclidean distance between X and Y_i is the smallest in comparison with all the other examples in the training data set Y. the class of this Y_i is assigned to the vector X. The same can be seen in Figure 3-18 (a) as well.

However, analysing the method in detail reveals that considering only 1–nearest neighbour might result in erroneous result in some cases. For example, if we consider 3-nearest neighbors instead of 1 and then take the majority voting of the class type among the nearest neighbors, the final result will be different from the one achieved using 1-nearest neighbor. This can be seen in Figure 3-18 (b).

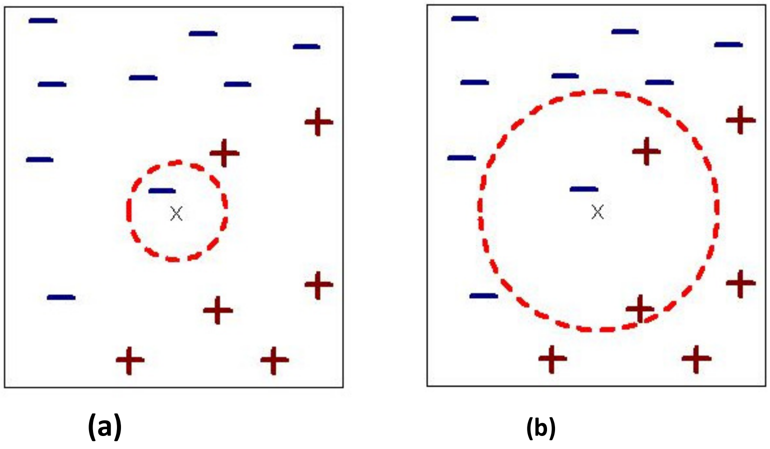


Figure 3-18: Illustration of (a) 1-nearest neighbor and (b) 3-nearest neighbor <https://www-users.cs.umn.edu/~kumar001/dmbook/dmslides/chap5_alternative_classification.pdf >

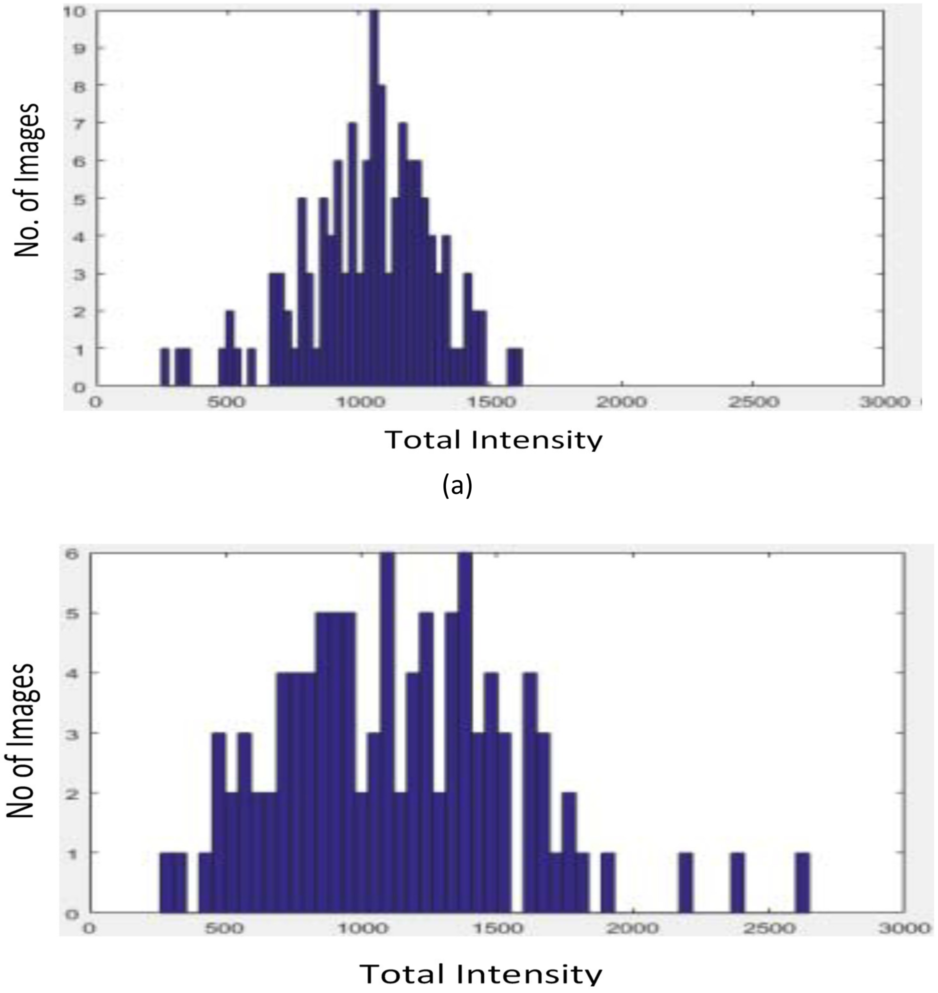
In other words, let's suppose there is a test vector for which the terrain class is known and is 'Sand'. When it is compared with the training vectors for the classification, there is a possibility to have the closest training vector of terrain class 'Bedrock', due to high resemblance between the two images. However, by considering several nearest neighbors might result into majority of nearest vectors having the class 'Sand' and this way the error can be resolved.

3.3.1 NASA JPL dataset

Hence, considering this point, the "k-Nearest Neighbor Method" is also implemented for the application, with k = 25. The reason for selecting such a large value for k is the fact that in this application the data set being dealt with are quite large. So in this method, for a given test image of class 'Gravel', among the 25-nearest neighbors, if 5 are 'Sand', 6 are 'Bedrock' and 14 are 'Gravel', then even if the closest neighbor is not 'Gravel', the system will assign 'Gravel' class to the test image owing to the majority voting result from the 25-nearest neighbors. Using k-nearest neighbor method for classification, the system could achieve a near perfect result of around 92% for the detection of 'Sand' terrain type among all the test or validation set images.

3.3.1.1 Intensity Thresholding

However, for the detection of 'Gravel' and 'Bedrock', the system did not exhibit the same success ratio as with 'Sand'. Analyzing various images of terrain type 'gravel' & 'Bedrock' reveals that there is a notable difference in the intensity of the images having these two terrain types. Bedrocks are white in color and hence, always brighter as compared to rocks in any image. This observation led to the conclusion that using the intensity information along with the texture information for classification can lead to a better differentiation between ‘Gravel’ and ‘Bedrock’ classes. For using the intensity information, it needs to be analysed first and for that the total intensity of all the training images of type ‘Gravel’ & ‘Bedrock’ are plotted in Figure 3-19.



(a)

(b)

Figure 3-19: Total Intensity Plots for (a) Gravel and (b) Bedrock

These plots have total intensity values plotted on x-axis and the number of images with given total intensity values are plotted on y-axis. It can be observed from the plots that, very few training images of terrain type ‘Gravel’ have the total intensity of more than 1400, whereas there are significant number of training images of type ‘Bedrock’, having more than 1400 total intensity value. Hence, an intensity threshold of 1400 can be derived from the training data set such that if a test image has been classified as the terrain type ‘Gravel’, then its total intensity is calculated and if it is more than 1400, then it is assumed that classification has some mistake and test image is assigned the class ‘Bedrock’ because ‘Gravel’ is not supposed to have the total intensity more than 1400. This technique is named as “Intensity Thresholding”.

Implementing this technique along with k-nearest neighbor (K-NN) classification method improved the result substantially from that using only 1-nearest neighbor (NN) technique, for all the descriptors. The classification results acquired for all three visual descriptors, using NN, K-NN as well as K-NN & IT (Intensity Thresholding) techniques, for different terrain types are illustrated and compared in the table below.

Terrain Type	**GIST**			**HOG**			**TEXTONS**		
	NN	**K-NN**	**K-NN & IT**	**NN**	**K-NN**	**K-NN & IT**	**NN**	**K-NN**	**K-NN & IT**
Bedrock	70%	65%	73%	67%	50%	71%	74%	65%	80%
Gravel	70%	81%	79%	71%	86%	81%	62%	78%	72%
Sand	91%	92%	92%	90%	96%	96%	92%	93%	93%
Average	77%	79%	81%	76%	77%	83%	76%	78%	82%

Table 14: Final Results (%) for NASA JPL dataset using various classification techniques

3.3.2 CSA dataset

For the CSA dataset also, the same technique as those used for NASA JPL dataset viz. Nearest Neighbour Method and k-Nearest Neighbour Method were implemented and experimented with, for the classification purpose. Here, due to the lower visual diversity in CSA dataset, NN alone already outperforms the best performance achieved on JPL dataset. However, analysing the results revealed that using k-nearest neighbour method (with k=25) either degrades the performance or does not improve performance, for this dataset. It seems here that the smaller CSA dataset may contribute to some confusion when k = 25. In none of the case, K-NN proves to be beneficial for the CSA dataset and hence, instead of increasing the algorithmic complexity (without any considerable gain), a simpler approach of the nearest neighbor is being used for the classification purpose.

3.3.2.1 Intensity Thresholding

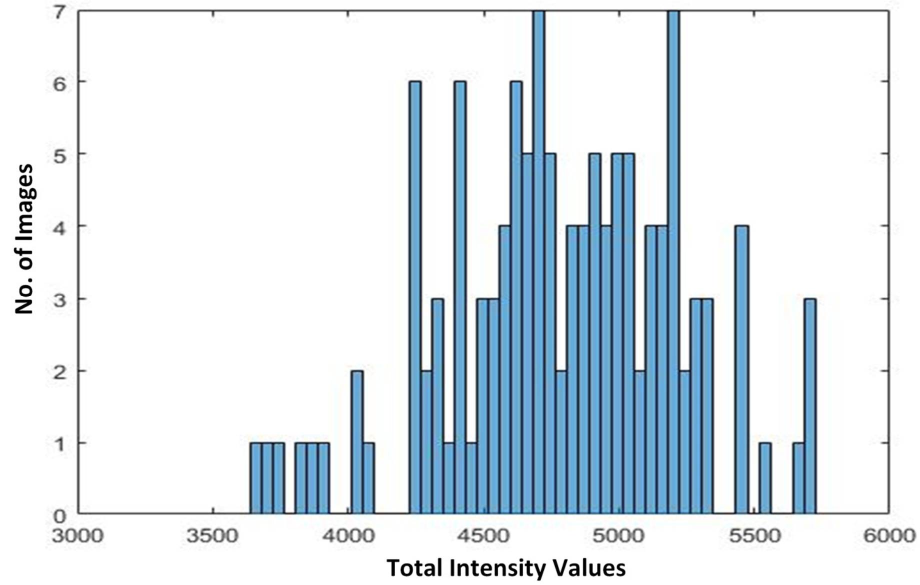


Figure 3-20: Total Intensity Plot for Bedrock (CSA)

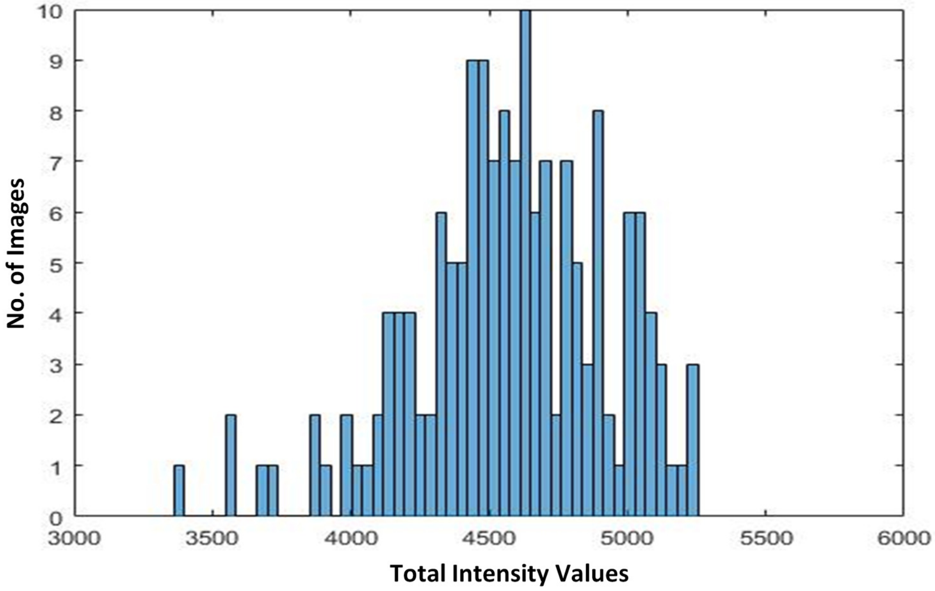


Figure 3-21: Total Intensity Plot for Sand (CSA)

Moreover, in case of the JPL dataset, there was a difficulty in differentiating between bedrock and gravel and as a solution intensity thresholding was used, which aided the classification further. However, in the CSA dataset, gravel type terrains are visually completely different from other terrains and hence, are easily recognizable and differentiable, as evident from the success ratio achieved by different descriptors. Here, bedrock and sandy terrains are confusing and hence, to verify whether intensity thresholding can be helpful for this case too, the intensity of all the training images comprising of bedrock and sand are plotted in Figures 3-20 & 3-21.

It can be observed from the plots that, "Intensity Thresholding" can be implemented here as well, similar to what was done with NASA dataset, but with the threshold of 5000. The reason for the same is that, for bedrock there are considerable number of images whose total intensity value reaches beyond 5000, whereas for sand there are negligible number of images with total intensity beyond 5000, as evident from the plots. Using this technique and analysing the results revealed that it actually helped improving the bedrock results. Thus, for the CSA dataset, nearest neighbor technique is used for classification and the comparison of the results acquired using this technique with & without intensity thresholding can be seen below in Table 15.

Terrain Type	**GIST**			**HOG**			**TEXTONS**		
	NN	**KNN**	**NN & IT**	**NN**	**KNN**	**NN & IT**	**NN**	**KNN**	**NN & IT**
Bedrock	76%	80%	83%	80%	70%	86%	90%	76%	96%
Gravel	98%	90%	98%	99%	99%	99%	99%	97%	99%
Sand	93%	99%	93%	97%	97%	97%	96%	99%	96%
Average	89%	89%	91%	92%	89%	94%	95%	93%	97%

Table 15: Final Result (%) for CSA dataset using various classification techniques

3.4 Results analysis

To check that the developed algorithms generalize well and have not become over fitted to the training and/or validation data, a final test is run on data that has been withheld throughout all development to this point. The table 16 below shows that the classification test results (in terms of percentage of correct detection out of total images of that terrain type), obtained using KNN & IT are comparable to those achieved at the culmination of development and validation.

Descriptor	Bedrock	Gravel	Sand	Average
GIST	77%	67%	92%	78%
HOG	80%	70%	93%	81%
TEXTONS	90%	70%	93%	84%

Table 16: Results (%) for Test set using KNN & IT (JPL dataset)

Also, two examples of labeled results are given below in Figure 3-22 that represents the terrain classification results achieved by the system. Red area is detected as sand, blue as bedrock and green as gravel.

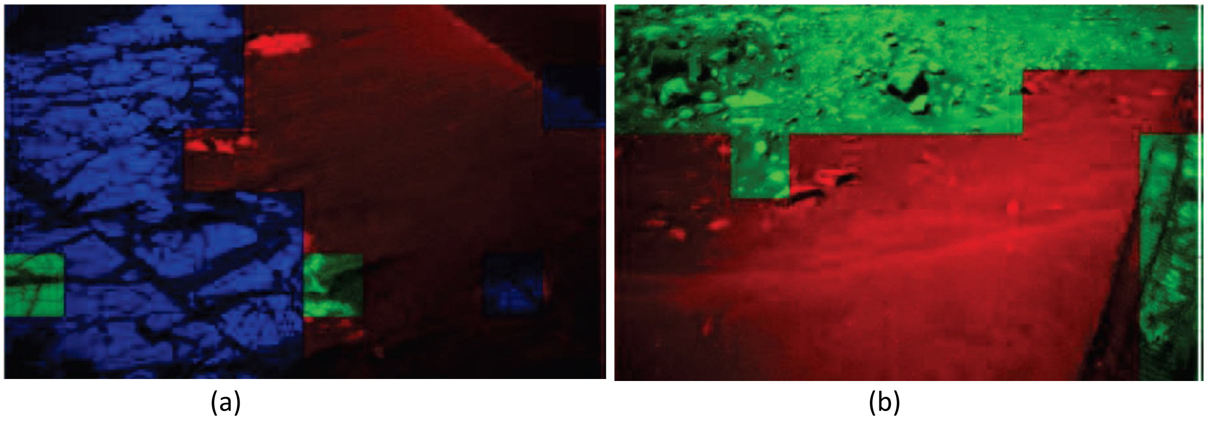


Figure 3-22: Representation of Terrain Classification Results

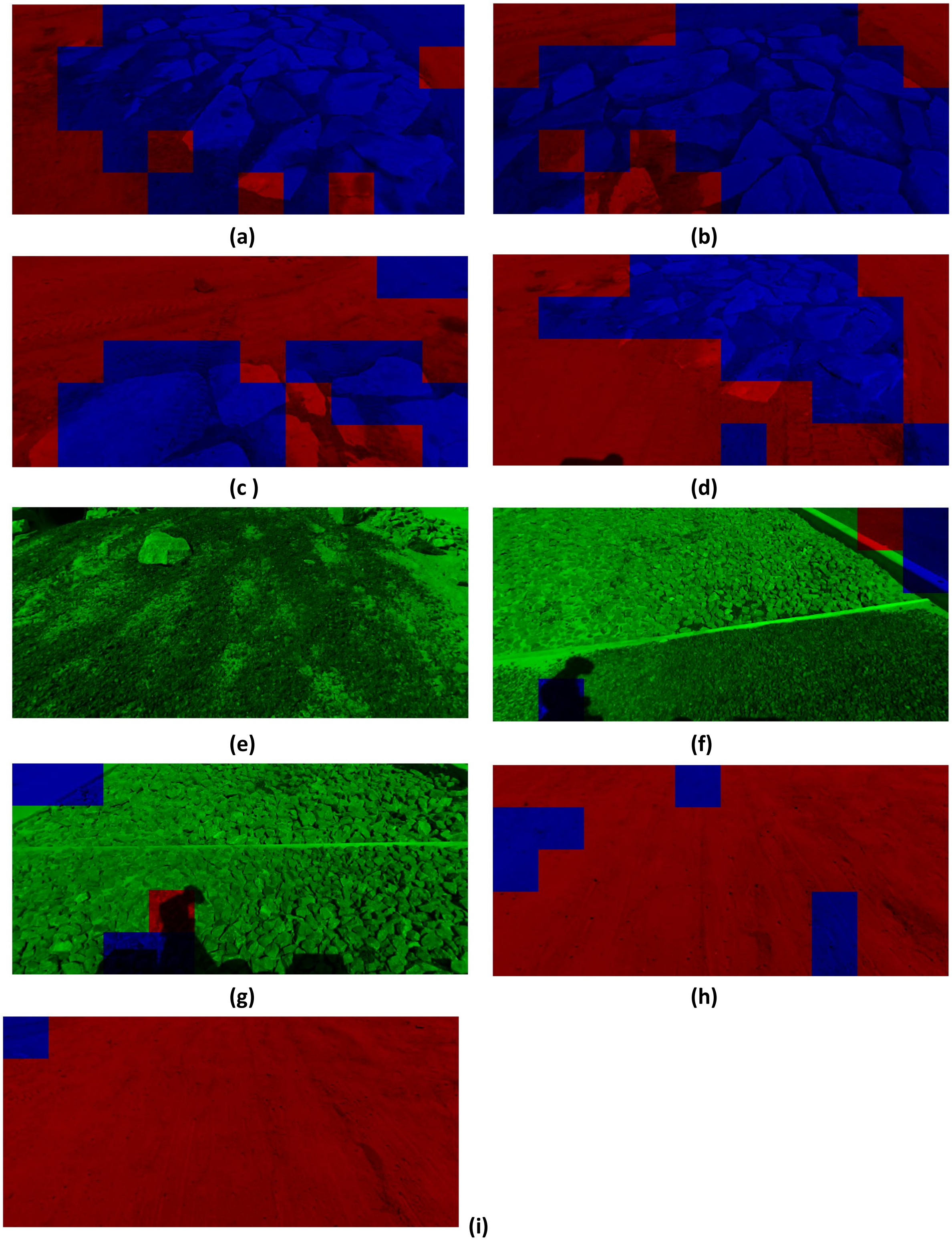
The overall results for CSA dataset, using the finalized classification approach (NN & IT) and all three descriptors can be seen in the following table (Table 17).

Descriptor	Bedrock	Gravel	Sand	Average
GIST	82%	98%	93%	91%
HOG	86%	99%	97%	94%
TEXTONS	96%	99%	96%	97%

Table 17: Final Results (%) for CSA dataset using NN & IT

As evident from the table the success ratio for gravel detection using all the descriptors seems high, unlike the ones for the NASA dataset. Despite using the same algorithm and design parameters as the other dataset, the reason for achieving such a high success ratio is the highly substandard diversity in the CSA dataset. As evident from various examples of this dataset's images (seen in the Figures 3-6 & 3-7), there are gravel beds made in the CSA mars yard, each of which consists of gravels of a particular size. Hence, all the images consisting of gravel are almost identical and so are the training and validation images, whereas there is a great diversity present in the images taken from the NASA JPL website. Thus, because of having negligible difference between training and validation images, the system is able to detect almost perfectly using these simpler descriptors. The same reason also explains the higher overall results than the other dataset.

Other than that, it is also evident from the table 17 that the highest success ratio is achieved by using Textons for the feature extraction. However, similar to the other dataset, it can be seen here as well that the results achieved by all three descriptor are comparable to each other and the algorithmic complexity does not seem to be providing substantial performance gain. Hence, as explained in that section, HOG seems to be the optimum choice for this dataset as well due to its lower complexity as well as runtime. The following figure (Figure 3-23) shows the representation of the terrain detection results using HOG for all the validation images of CSA dataset.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 3-23: Representation of results for CSA validation library

The important results observed here that merit a deeper discussion include, the importance of edge detection, criticality of the selection of intensity threshold value and the benefits of rotational invariance; in the context of planetary terrain texture classification, as well as the lack of significant performance difference despite varying algorithmic complexity.

3.4.1 Intensity Thresholding

As seen earlier, intensity thresholding has been implemented for both the datasets in combination with the classification algorithm. It is like utilizing an additional information from an image, along with the texture extraction. It was seen earlier that it helps improving the differentiation between certain terrain types. An important observation here is that the threshold values used for both the datasets are different and these values depend on all the images present in the training library for individual terrain types. It is notable here that adding or removing a few images from the training library might change the threshold value and could improve the capability of the system to differentiate between certain terrains. However, due to a hard coded intensity threshold value, this change would not be reflected in the system and continuously using the same threshold value can hamper the performance eventually. Thus, enabling an adaptable intensity threshold is proposed as potentially fruit full future work.

3.4.2 Algorithmic Complexity

As discussed in the Literature Review, the complexity of 2D convolution can be calculated by O(MN mn), where M x N is the size of the image and m x n is the filter window size. Based on this equation, the computational complexity of all three above mentioned descriptors are calculated for one image (100 x 100) below:

(a) HOG: It uses two derivative masks of window size 1 x 3 and 3 x 1 for the filtering. Hence the complexity for the same for scale equal to 1 (image size = 50 x 50) will be the summation of O (50 x 50 x 3 x 1) and O (50 x 50 x 1 x 3), which is equal to O (15000). Similarly, for scale equal to 2, 4 & 6 the image size is reduced proportionally and hence, the complexity for them will be O (3750), O (864) & O (384) respectively, leading to a total of O (19998).

(b) GIST: It uses Gabor filters at different scales (1, 2, 4, 6) and orientations (8 different values). Window size of the Gabor filter varies with the scale (window size = 6*scale x 6*scale). Thus for scale 1, there are 8 Gabor filters of size 6 x 6 at different orientations, giving the complexity of O (100 x 100 x 6 x 6) i.e. O (360000). Similarly filters with scale equals to 2, 4 & 6 gives the complexity of O (1440000), O (5760000) & O (12960000), respectively, leading to the total of O (20520000).

(c) Textons: it uses gaussian, LoG, 18 edge filters and 18 bar filters. Gaussian and LoG filters are of size 12 x 12, hence their computational complexity will be O (1440000) each. Edge and bar filters are at the scales of 1, 2 & 4 for 6 different orientations, thus there 12 filters at each scale value. Hence the complexity for scale 1, 2 & 4 will be O (4320000), O (17280000) & O (69120000), leading to a total of O (93600000). Apart from this, Texton based method also involves k-means clustering (O (kn) for 1 iteration; where k represents number of clusters and n is number of inputs). Here the number of iterations can vary and hence, the complexity of Textons can be thought of more than O (93600000).

From above discussion, Textons based method is the most complex one, whereas the HOG is far lesser complex than GIST and Textons.

An important observation from this work is that it demonstrates a lack of evidence that the increased complexity of GIST descriptors and Textons provide substantial performance gains over a relatively simpler gradient-based modified HOG descriptor. Runtime of the simplified HOG algorithm was faster than for the other options explored, consistent with its lower complexity. Runtime considerations are an important factor for planetary rovers, which usually have tight computational budgets. This is also part of the reason, why CNN (Computational Neural Networks) methods have not been implemented, in spite of their ability to outperform currently implemented approaches.

3.4.3 Edge Detection

Fundamentally, the various feature extraction filters most commonly used as bases of texture detection can be viewed as edge detectors. Textons use filter banks that include edge detectors, but also bar filters (i.e. ridge detectors), Gaussian, and Laplacian of Gaussian filters. An interesting aspect of the Texton results is that we can easily inspect the Texton library to see which of the filters in the filter bank contribute most prominently to the clustered textures.

The Figure 3-24 below shows the mean magnitude of each filter bank's contribution to the Textons extracted from the training data for NASA JPL dataset. The numbering of each bar along the x axis indicates 1: Gaussian filter; 2: Laplacian of Gaussian; 3-5: Highest edge response at 3 increasing scales; and 6-8: Highest bar response at 3 increasing scales. The highest contributions come from the 3 edge filters and the large scale bar filter, with the next highest contribution coming from the midscale bar filter. The other filters, notably the Gaussian and LoG, do not make important contributions to the Texton library.

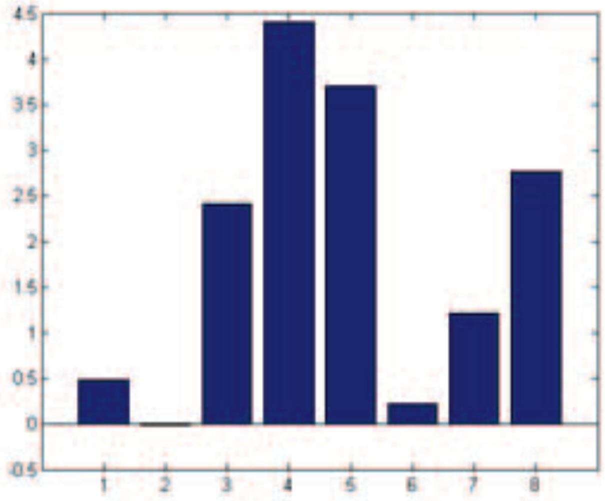


Figure 3-24: Average Texton response to each of the 8 filters in its filter bank. Edge filters and large scale bar filters figure most prominently

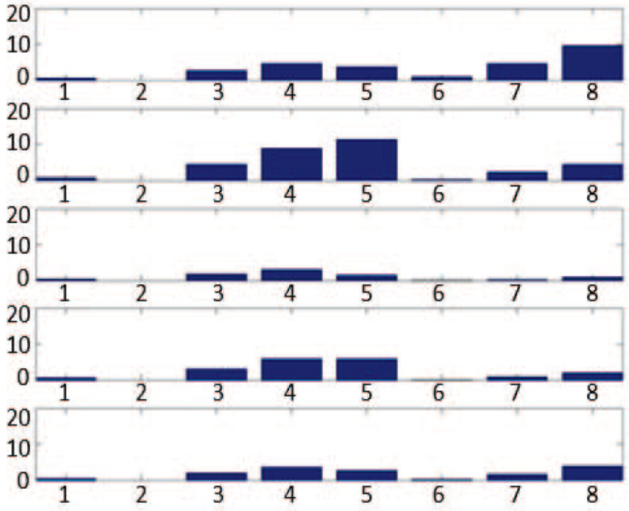


Figure 3-25: Top 5 magnitude textons, demonstrating consistency with average Texton result

Further, Figure 3-25 demonstrates that the above observations are not only true on average. It shows the top 5 magnitude Textons, all demonstrating distributions comparable to the average distribution. The other 27 Textons show similar results, just at lower magnitudes; they are omitted simply for brevity and clarity.

3.4.4 Rotational Invariance

As discussed previously, classification of terrain textures should not have to depend on encountering these texture from particular orientations. Instead, including a form of rotational invariance to the features being detected can avoid needlessly separating out similar textures seen from different vantage points. The results shown in tables (RI table numbers for GIST & HOG) appear to support this line of reasoning: the best performing HOG and GIST descriptors employ rotational invariance. In particular, local RI (performed independently at each scale) produced consistently good results. The fact that local RI appears to outperform global RI suggests that the relative prominence of edges at different scales is more predictive of terrain than the preservation of relative orientation between prominent edges at different scales (global RI preserves these relative orientations between different scales, local RI does not). A deeper investigation of these issues, studying particular examples from terrain images in detail, could be a fruitful direction for future work.

It is interesting to note that the MR8 Texton algorithm, in fact, implements an extreme version of local RI. Going beyond a reordering of edge filters such that the most prominent response consistently appears first for each scale (as described in the section of RI for GIST and HOG), the MR8 algorithm actually only includes the single most prominent response for each scale. In the modified HOG and GIST descriptors with local RI, this would be equivalent to only considering the 1st, 9th, 17th, and 25th elements of the histogram. If descriptor length is of concern for computational reasons, a shortening to a 4 element descriptor could be investigated in future work. However, a 32 element descriptor is already not long.

3.5 Summary

The exteroceptive module focuses mainly on 3-way terrain classification (classifying as one of the types: Sand, Bedrock and Gravel). For terrain classification, visual descriptors are adapted to this application and investigated, which are primarily based on visual texture and are captured in form of histograms of edge filter responses at various scales and orientations. The descriptors investigated in this work are HOG, GIST, MR8 Textons and the classification techniques implemented here are nearest and k-nearest neighbors. The performance and computational complexity of all three visual descriptors are also analyzed and compared with one another. Further, monochrome image intensity is used as an additional feature to further distinguish bedrock from the other terrain types. To validate and compare the performance of different techniques, two different datasets (NASA JPL and CSA datasets) have been used. No major differences in performance are observed between the three descriptors (81% vs. 84% performance on JPL dataset and 94% vs. 97% performance on CSA dataset for HOG and Textons, respectively), leading to the adoption of the HOG approach due to its lower computational complexity (over 3 orders of magnitude difference in complexity between HOG and Textons) and thus higher applicability to planetary missions.

Chapter 4. Proprioceptive Module

As mentioned in the Introduction, non-geometric hazards are highly dependent on wheel-terrain interaction properties. Terrain characterization is useful for deducing such wheel-terrain interaction properties, where the data measured by proprioceptive sensors is captured while driving the rover over a terrain region of interest and then this data is analyzed to infer some physical property related to trafficability. In ASAS, the proprioceptive data, namely linear acceleration & angular velocities of rover in X, Y and Z planes, visual & wheel odometry as well as pitch, roll & yaw of the rover (all with respect to time), is collected by driving the rover on different terrains at two different sites viz. CSA Mars Yard and Lafarge.

Another thing to recall the Introduction is that the ultimate goal of this system is to combine the properties of classification and characterization methods to enable an accurate prediction of the physical properties of a distant terrain. This is the main focus of this chapter. The proprioceptive module of ASAS focuses on correlating some important trafficability properties with the classified terrain types, which were obtained in the classification phase (as described in the previous chapter) and develops a supervised online learning framework, which performs trafficability prediction as well as enables online training of the rover, simultaneously.

The brief analysis of some of the data measured by proprioceptive sensors as well as some trafficability properties is given in the section 4.1. Section 4.2 gives an overview of design of the proprioceptive module so as to implement both the functionalities of online training and trafficability prediction. The detailed algorithm for implementation of these functionalities viz. online training and trafficability prediction is explained in the sections 4.3 and 4.4, respectively.

4.1 Data Analysis

As mentioned above, the proprioceptive data has been collected from two different sites i.e. CSA Mars Yard and Lafarge. This data is divided into different data logs for different test runs. Each test run has been designated by a number and its corresponding terrain type to identify which data log is for which terrain type. For example, Lafarge data logs for 22nd and 37th test runs are for Sand, while data logs for 60th, 61st and 19th test runs are for Gravel. This allows a user to analyze the data characteristics for different terrains. Another notable point here is that the terrain slope is being extracted from two different sources viz. IMU and stereo cameras. In comparison with stereo camera slope, IMU pitch data holds more reliability. Hence, stereo camera slope is used, only when the IMU pitch is not available i.e. when the slope of a distant terrain has to be used. Apart from that, IMU pitch has been used for all the operations.

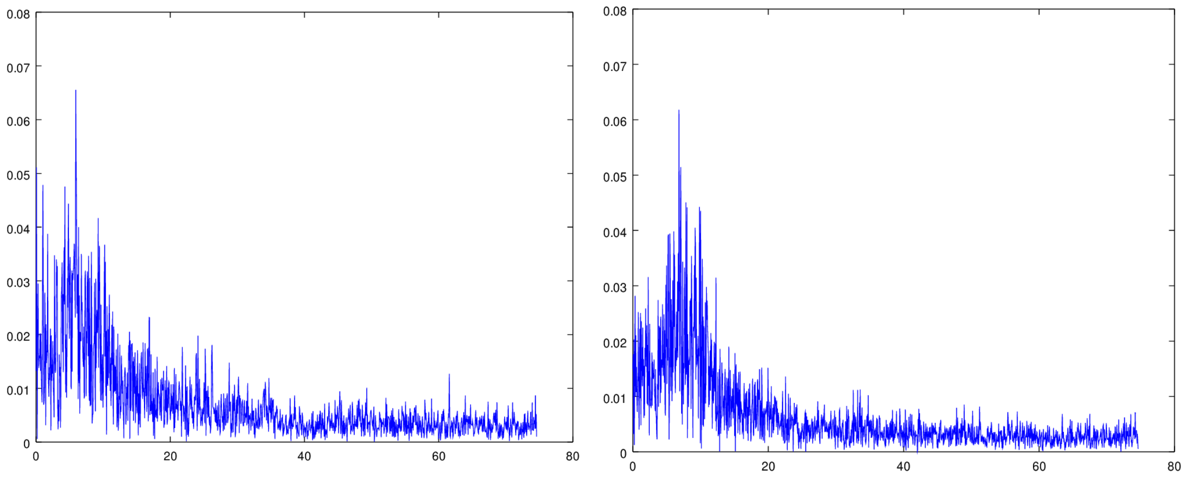
4.1.1 Vibration Analysis

The work performed in [23] shows that vibration data can be utilized for the classification between different terrains. Among the proprioceptive data that is being collected in ASAS, linear acceleration data is utilized to derive the vibrations that the rover experienced while traversing a specific terrain. The standard deviation in acceleration gives the strength of the vibrations, whereas the frequency spectrum of acceleration gives the frequency of the vibratory force. If any two terrain types have distinct vibration attributes (standard deviation or frequency or both), this feature can be used for classification. However, the standard deviation data for various terrain types does not possess very distinct values as can be seen below:

- Sand (tests 6, 7) : 0.11, 0.09
- Bedrock (tests 6, 7) : 0.25, 0.27
- Small rocks (tests 14, 16, 17) : 0.34, 0.18, 0.25
- Med rocks (tests 14, 16, 17) : 0.49, 0.36, 0.46

It is evident from these values that sand can be distinguished from other terrain types due to very small standard deviation, but bedrock and small rocks have an overlapping standard deviation values and they are not clearly distinguishable.

Further, another vibration attribute i.e. frequency spectrum is derived for bedrock and small rocks, as shown in the Figure 4-1 below. It is evident here as well that the two frequency spectrum are pretty much the same and hence, this feature does not have potential for classification for the given data.



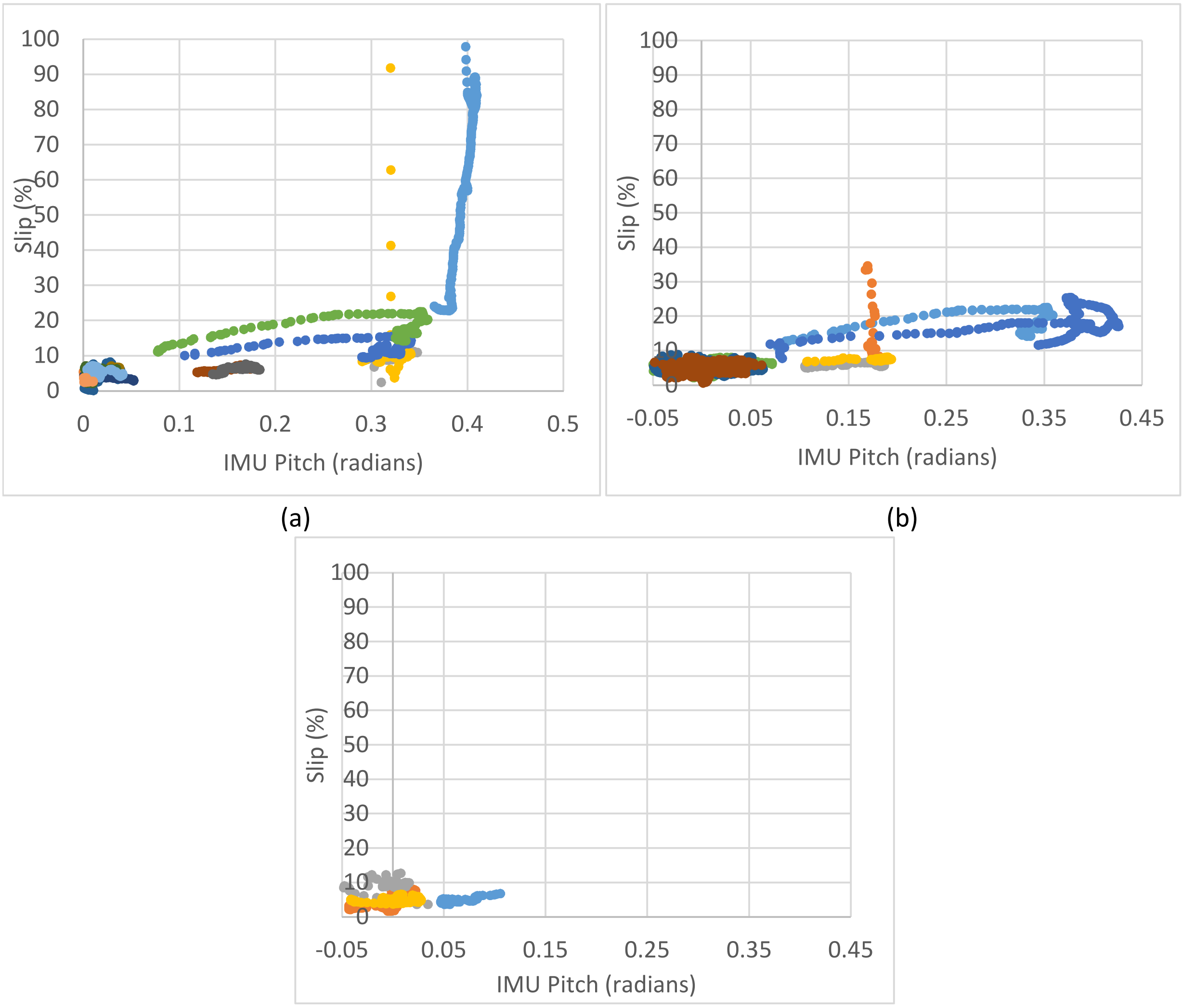
(a) (b)

Figure 4-1: Frequency spectrum of linear acceleration (a) Bedrock (b) Small rocks

4.1.2 Slip Analysis

Slip, experienced by the rover's wheels, is the most well-known trafficability metric, since it directly relates to rover forward progress. For a wide range of terrain physical types, wheel traction is known to increase monotonically with wheel slip until a peak is reached (typically at a slip range of 10-30%), at which point traction may decline or plateau. For this reason, wheel slip is an extremely desirable, and arguably sufficient, trafficability metric. Moreover, it changes as a function of slope/IMU pitch, on a particular terrain type. Hence, for its analysis, the terramechanics models for slip vs. IMU pitch is presented here.

The graphs shown in Figure 4-2 are plotted for the data collected from CSA mars yard. As can be seen, in case of Sand and Gravel, only one test run each has pitch values greater than 20 degrees (~0.35 rad.), whereas in the case of bedrock, there is almost no pitch value encountered beyond 5 degree (0.087 rad.), during any of the test runs. This is due to the lack of physical terrains at higher slopes, in CSA mars yard and as a result of this insufficiency of data, it is difficult to have a comprehensive analysis.



(a)

(b)

(c)

Figure 4-2: Slip vs. IMU Pitch plots for (a) Sand, (b) Gravel and (c) Bedrock (at CSA Mars Yard)

Hence, to solidify the observations, the proprioceptive data collected from the Lafarge site is also analyzed and the plots for the same are presented in Figure 4-3 below. As can be seen, contrarily to the CSA data, abundant data is available for pitch values higher than 20 degree. Moreover, it can be observed that as soon as the rover crosses the 20 degrees mark, the measured slip increases drastically for a number of test runs. This sudden rise in measured slip from 20% to almost 100%; for the increase in pitch values from 20 to 35 degree shows that it is a good measure of trafficability. Thus, 20% slip is considered a safety threshold for having a risk free trafficability.

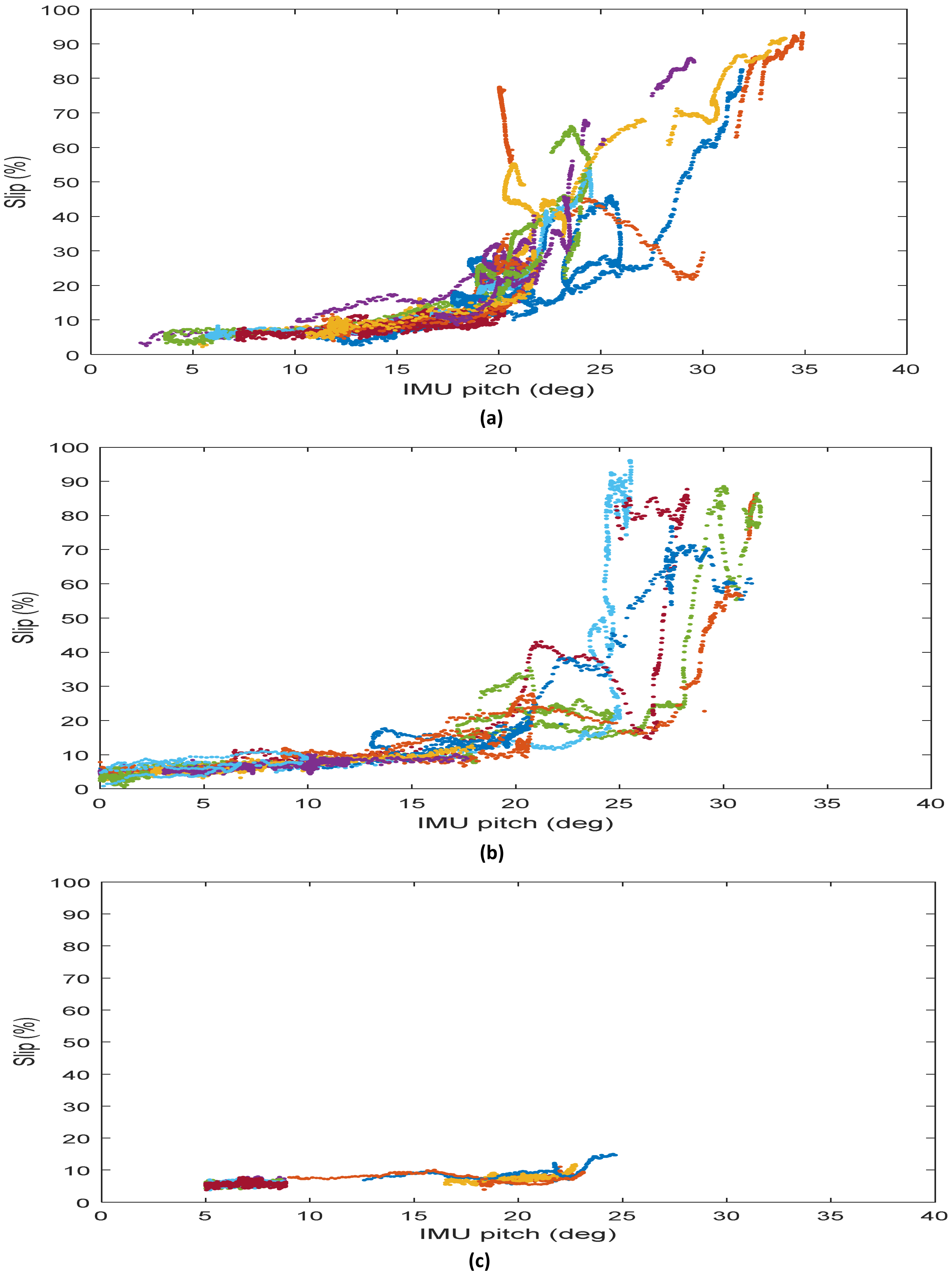


Figure 4-3: Slip vs. IMU pitch plots for (a) Sand, (b) Gravel, (c) Bedrock (for Lafarge data)

4.1.3 Sinkage Analysis

Similar to slip, sinkage is also a very important trafficability metric, owing to its decisive effect on rover's ability to traverse any terrain. Here, sinkage is derived using a camera on the underside of the rover. Also, similar to the analysis of slip, terramechanics models of sinkage vs. IMU pitch has been plotted in Figure 4-4 & 4-5.

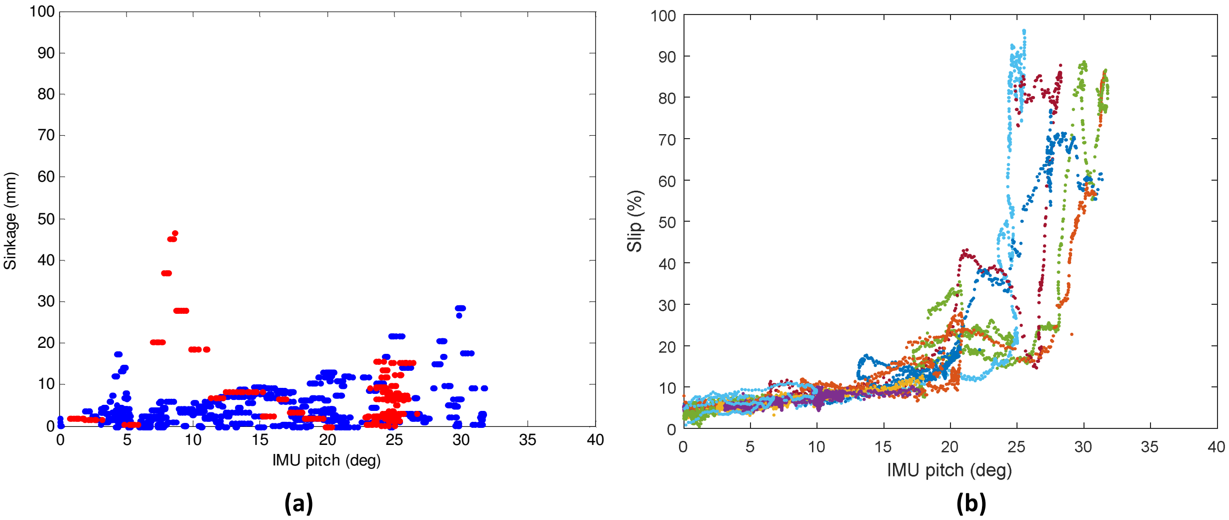


Figure 4-5: Comparision of (a) Sinkage vs. IMU Pitch and (b) Slip vs. IMU Pitch, for Gravel

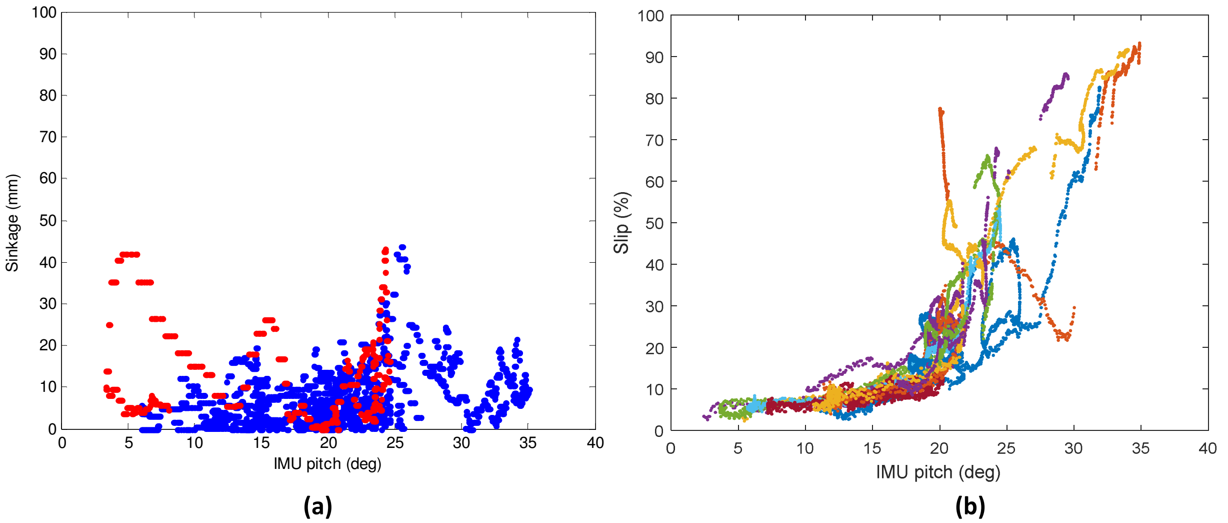
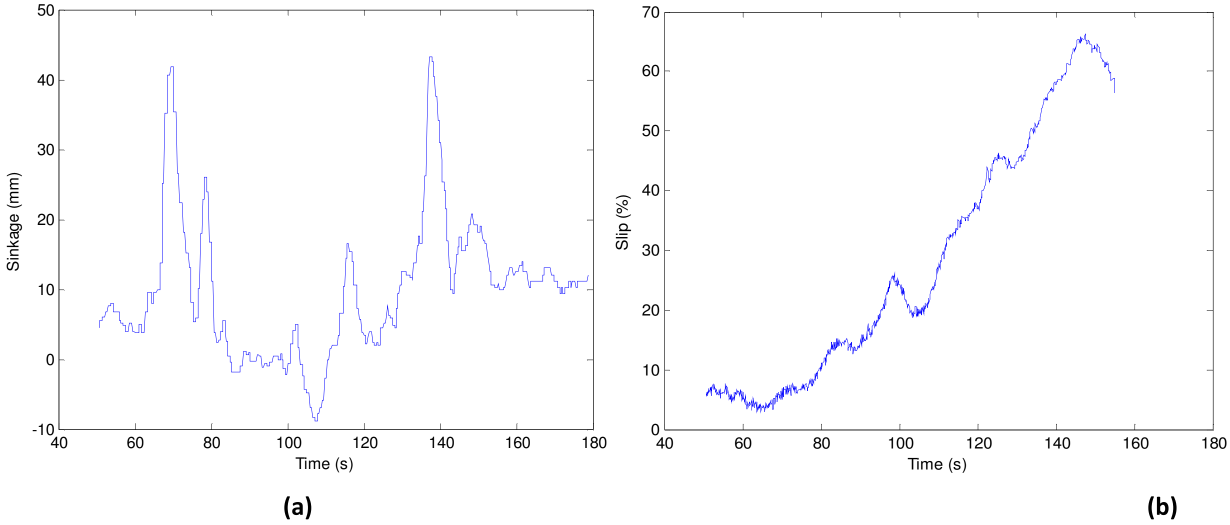


Figure 4-4: Comparison of (a) Sinkage vs. IMU Pitch and (b) Slip vs. IMU Pitch, for Sand

As can be observed, most of the sinkage measurements are very small values (in the range of 0 to 25 mm only) and unlike the slip measurements, they remain almost the same over the displayed range of pitch values. In other words, the trend of sinkage vs. IMU pitch is pretty much flat, with some random peaks. The random peaks are explainable by some occlusions observed in the test run videos at particular time stamps, whereas the very small sinkage values lead to the conclusion that the physical terrains, over which the rover is being traversed, are not causing the rover to sink enough to have the wide range of sinkage measurements.

Moreover, analyzing and comparing the plots of sinkage and slip with respect to time, for different test runs also provides some interesting observations such as higher sinkage value observations beyond 25° pitch, when the rover is experiencing the slip as high as around 70%. One of such example is shown in the Figure 4-6, where the occurrence of slightly higher sinkage value (>30mm) can be observed, at 140 second time stamp where slip measurement is almost 70%. Such late occurrences of higher sinkage values, where slip values have already reached way beyond the safety threshold (20%), do not provide additional information to the trafficability risk calculations, because slip measurements are already indicating very high trafficability risk.

Thus, in ASAS, due to the lack of wide range of sinkage values in the collected dataset and inability of the sinkage dataset to help with risk prediction, sinkage has not been used in the calculation of trafficability risk. However, sinkage being an important trafficability metric in general, it is being correlated with the exteroceptive data along with the slip measurements for future scope.



(a)

(b)

Figure 4-6: Comparison of (a) Sinkage vs. Time and (b) Slip vs. Time, for a specific test run

4.1.4 Roughness Analysis

The exteroceptive data provides slope and roughness as a result along with the classification result. Both slope and roughness are being calculated from the depth map provided by stereo cameras. Also, both of them can have a legitimate effect on the slip, experienced by the rover and hence may affect the trafficability of the rover. While the effect of slope is very much clear from the terramechanics models of slip vs. IMU pitch, the effect of roughness on slip is analyzed here. The slip vs. roughness plots shown below in Figure 4-7 exhibits the slip variation from 0% to 100% for a number of roughness values. This implies that here roughness is not having the anticipated effect on the slip measurements for the given dataset. Also, our industrial partner, being responsible for the calculation of roughness, found out that the data is noisy. Thus for ASAS, roughness is not having any effect in the calculation of trafficability risk. However, considering its ability to affect the slip measurements and in turn, the trafficability prediction, it has been correlated with exteroceptive data, along with slip & sinkage, for future scope.

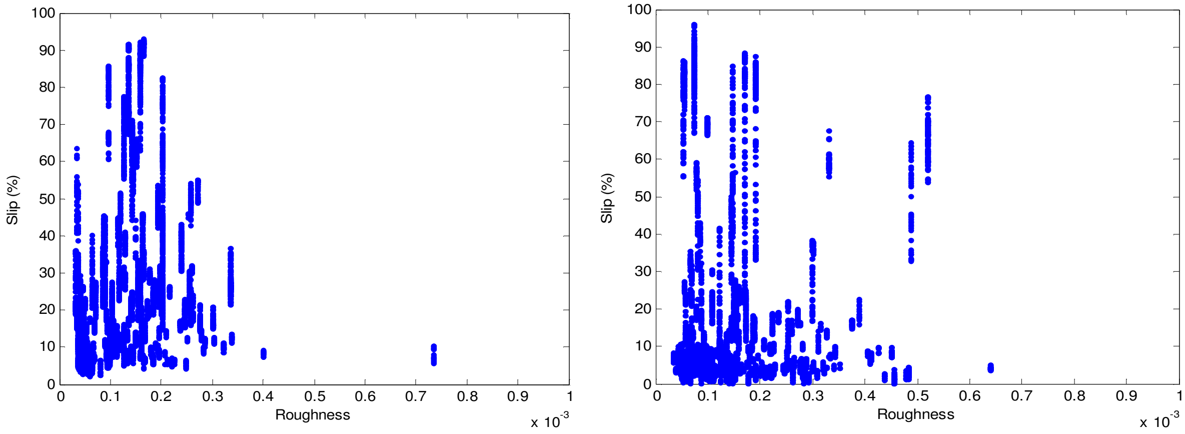


Figure 4-7: Roughness vs. Slip plots for sand and gravel

4.2 Design Overview

As explained in the data analysis section (4.1), there are three parameters that are collected because they may affect the trafficability risk prediction directly or indirectly viz. slip, sinkage and roughness. Also, exteroceptive data that affect the risk prediction are slope and the terrain type. To enable remote prediction of terrain trafficability, it is essential to develop a learning framework that correlates proprioceptive and exteroceptive data. Hence, the proprioceptive module of ASAS focuses mainly on two aspects. One of them is to develop a supervised learning framework that correlates the trafficability metrics viz. Slip, Sinkage and Roughness with the outputs from exteroceptive module viz. Slope and Terrain type online, as the rover traverses the terrains. This will develop an empirically learned data driven model, which is used to train the rover. The second aspect is to utilize this data driven model to estimate the trafficability properties of a distant terrains with the help of only exteroceptive data (slope & terrain type).

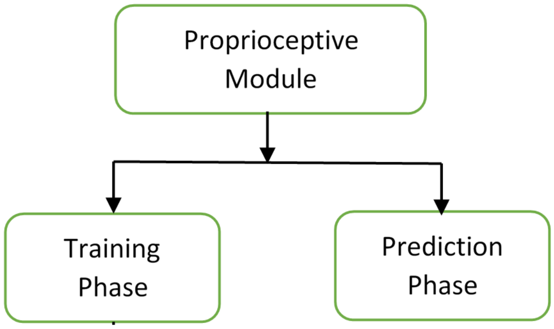


Figure 4-8: Operational phases of Proprioceptive Module

The proprioceptive module operates in two phases as shown in Figure 4-8: Training Phase and Prediction phase. In the training phase, initially the trafficability metrics are correlated with the exteroceptive data in the form of a structured data file, which we will be referring to as "Correlated Data Model**"** from now onwards. After that, an algorithm for a learning framework is designed by which the online update of the correlated data model will take place. Moreover, using the correlated data certain parameters are also calculated in this phase, which are explained in detail in 4.3.4. These parameters describe the trafficability risks of a terrain. Similar to trafficability metrics, these parameters are also correlated with the corresponding exteroceptive data (slope & terrain type) in form of a structured data file, which will be referring to as "Trafficability Model". This model also gets updated online along with the correlated data model and hence, turns into a data driven model during online operation. Thus, the training phase deals with the first aspect of the proprioceptive module.

In the prediction phase, only the exteroceptive data viz. slope and terrain type are taken as inputs and the parameters correlated with this exteroceptive inputs are fetched from the "Trafficability model". This parameters defines the trafficability risk of a distant terrain with given slope and terrain type. Thus, the prediction phase deals with the second aspect of the proprioceptive module.

4.2.1 Training Phase

As mentioned earlier, Training phase deals with development and update of the correlated data model in accordance with the learning framework. The inputs to this phase are: the output from exteroceptive & proprioceptive module i.e. the classified terrain type, IMU pitch as well as the trafficability metrics viz. slip, sinkage & roughness. While as an output, the correlated data model as well as trafficability model gets updated, which keeps rover's trafficability prediction ability up to date according to the latest measurements.

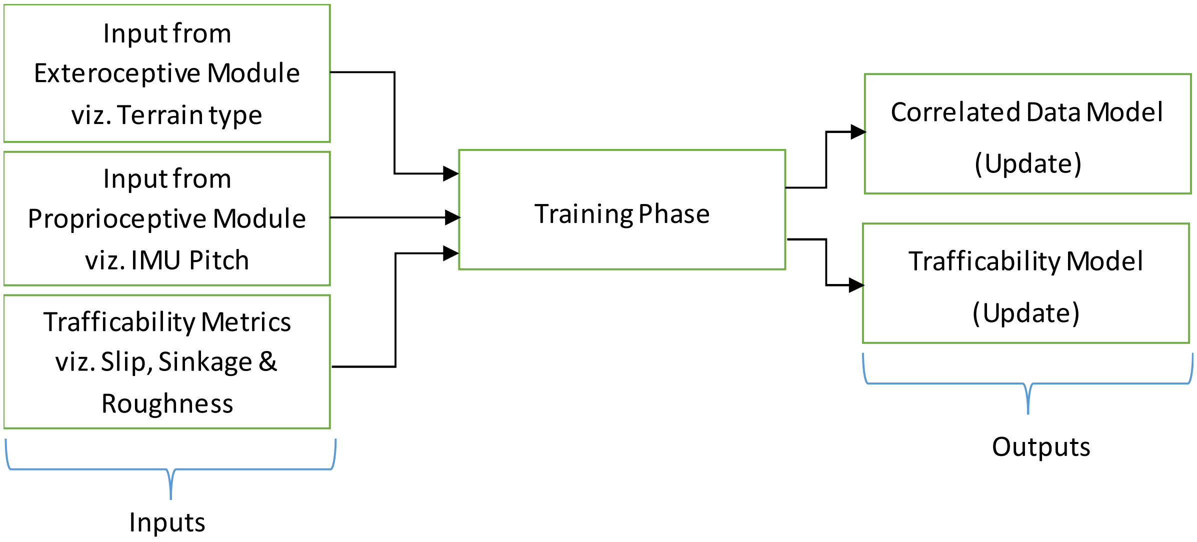


Figure 4-9: Block Diagram of Training Phase

The correlated data model is nothing but a structured data file and hence, designing a base architecture for the same means designing the structure of that data file. As can be seen from the Figure 4-9, this model gets 3 trafficability metrics as well as IMU pitch and terrain type as inputs. The requirement here is to correlate 3 trafficability metrics with the appropriate pitch value of a specific terrain type. Hence, each pitch value needs to have 3 lists of data samples associated with it corresponding to three trafficability metrics and all the terrain types needs to have such lists associated with each of their pitch values. Considering that the pitch value ranges from 0 to 45° and the resolution is taken to be 1°, each terrain type in the data file will end up having 135 lists of data samples corresponding all 45 pitch values (45 pitch values * 3 lists of data samples = 135). Each time a new data sample is added to the data model, the corresponding lists get updated for the given terrain type & pitch value. This explains the structure of the data file/correlated data model and by correlating several data samples of trafficability metrics to the appropriate exteroceptive data, which were obtained during on-earth test runs, a base architecture/base correlated data model is designed.

Moreover, using this correlated data model, the parameters defining trafficability properties are also calculated and the base architecture for the trafficability model is also designed in this phase. The outputs that summarize the trafficability properties are Average and Maximum Slip & sinkage, risk factor, number of times rover experienced >20% slip on given slope and risk level. The details of the calculation for these outputs are also given in section 4.3.4. This model also has the similar base architecture as that of the correlated data model, which means for all three terrain types, each pitch value needs one list of output parameters associated with it. Thus each terrain type needs to have 45 lists of data samples, each list corresponding to one pitch value. With this base architecture implemented, the models will keep updating while maintaining the original structure.

One important observation here is that even though the trafficability model is useful only in the prediction phase and not in the training phase, it is being generated and updated in the training phase. The reason for the same is the constraint of decision time while predicting the risk for a distant terrain. When a slope is queried in the prediction phase, rover needs to take a very quick decision about the trafficability risk and hence, all the calculations and pre-processing for the model is completed in the training phase itself and the system has to simply fetch the output parameters from the trafficability model.

For deciding on the learning framework for the online update of the models, certain special cases have been considered and taken care for, which might be encountered while online update of the correlated data model. Considering that during the test runs, the rover may not encounter all the pitch values for a terrain type leads to the fact that the rover might be getting large number of data samples for certain pitch values and no/very few data samples for the others, during the online update.

4.2.1.1 Implementation of Queue Data Structure

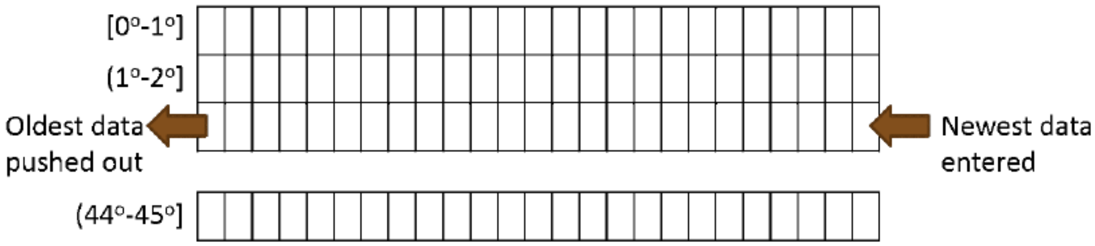


Figure 4-10: Illustration of FIFO queue like data structure

The first case is where large number of data gets added to a single list of data samples for any particular pitch value and terrain type. Here, after a certain extent, newer data won't be affecting the calculations of the output parameters and in turn, on the trafficability prediction. Their effect gets diminished by the older data. To avoid this situation, instead of having an accumulated lists of data samples for each pitch value, a FIFO (First In First Out) queue like structure (as shown in Figure 4-10) is used for adding the data samples in the lists. In such a structure, after the number of data samples present in the queues reaches a certain limit, the oldest data sample gets discarded with every addition of a new data sample. This ensures that rover's trafficability model does not remain rigidly fixed to the version that was created on earth. In future, when rover will start collecting and adding data samples from the planetary surface to the correlated data model, it will evolve and adapt to the conditions being experienced on the planetary surface and will eventually start predicting the trafficability risk according to the planetary conditions instead of that on earth; no matter how different the on-earth environment is from the actual planetary surface.

On the contrary, the second case is where there is no/very few data samples for certain pitch values. Here, it is not possible to have reliable and purely data driven prediction for such pitch values in the prediction phase. To handle such cases, interpolation or extrapolation is used for calculating the output parameters of the trafficability model. The selection between interpolation and extrapolation depends on the IMU pitch value. Let's say N data samples are allowed in one queue and minimum of N/4 data samples are required to make a reliable prediction. Suppose, pitch value of $\emptyset_b$ is being queried for prediction and n $(\emptyset_b) < N/4$, where n $(\emptyset_b)$ = number of data samples present in the queue of $\emptyset_b$. In this case if there exist the pitch values $\emptyset_i$ and $\emptyset_{j,}$ where $\emptyset_i < \emptyset_b$ & $\emptyset_j > \emptyset_b$and both of them have n > N/4, then interpolation is used for the calculation of output parameters. Whereas, if there is no $\emptyset_j$ $(\emptyset_j > \emptyset_b)$ that has n > N/4, then interpolation is not possible and hence, one has to extrapolate based on the data from other terrain types to consider the worst case scenarios. However, it is not advisable to update the trafficability model based on such extrapolated values. Thus, extrapolation is just used for prediction at a present moment in cases where interpolation is not feasible due to lack of data. The details of various cases and their solution is explained in the section 4.3.6.

Apart from these two cases, another type of unusual events observed while data analysis, were identified by a sudden rise in slip measurements for a very short span of time, during some of the test runs. Two of such events are illustrated below in Figure 4-11.

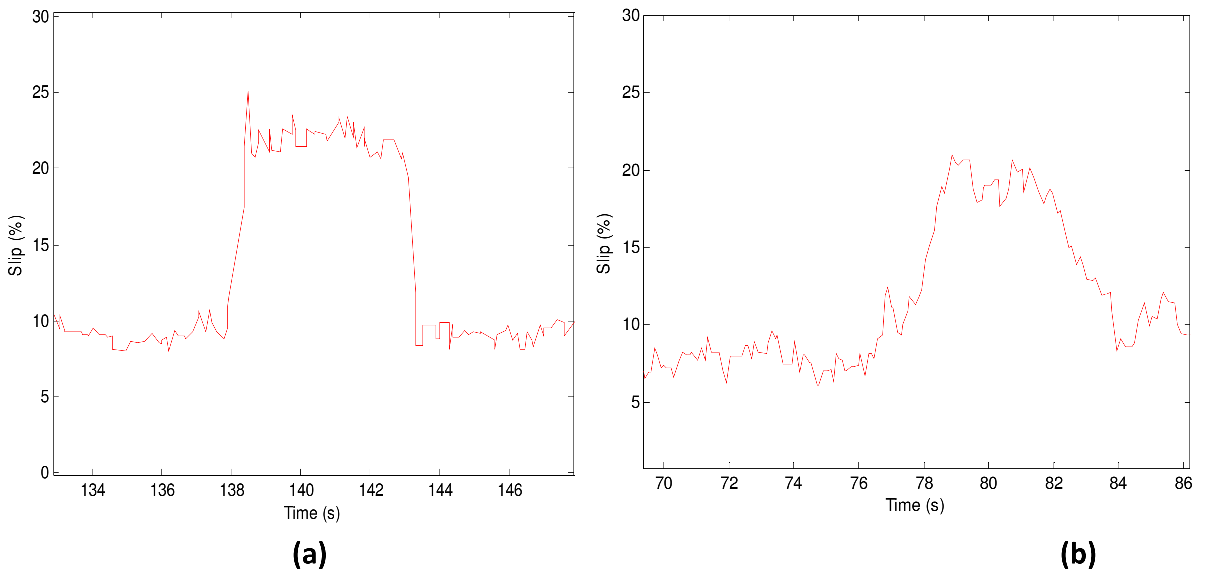


Figure 4-11: High Slip Events during two different test runs

In both the cases, these high slip events seem to last for around 5-6 seconds. Analyzing these events in detail, has revealed that sometimes they occur as a result of sharp turns or transition of the slope, while sometimes they are just due to noise. If such values are added to the correlated data model and used for the calculation of output parameters, without considering them as a special case, the system will learn the false model using these values. Such false learning by bad examples should be avoided as they can have several adverse effects.

The strategy to deal with such false High Slip Detection (HSD), is to observe whether this high slip data is continuous or is for a short time span. Whenever a sudden high slip data is observed for any particular slope & terrain type, a separate correlated data model begins to develop, consisting of only high slip data and an HSD mode gets activated for that terrain type. Hence, a new trafficability model is now formed using the said high slip data model and the prediction will also be based on this new trafficability model, as long as the HSD mode is active. If this is just a short event, that means if after a short time span, again slip measurements get back to normal values from high values, then it can be considered as one of the events mentioned above due to sharp turns or noise. In such case, the high slip data model will be deleted & HSD mode will be deactivated. This way original training model will be free from false high slip data and the trafficability model will keep functioning properly without being affected by such events. On the contrary, if such high slip measurements sustain for some time, the user expert can analyze the terrain and its characteristics and can deal with it accordingly. The details about the algorithm for this case is given in Appendix.

4.2.2 Prediction Phase:

As mentioned earlier, prediction phase deals with the trafficability risk prediction, for a distant terrain being visualized by the rover before traversing it. It uses the trafficability model for the prediction. The inputs to this phase are just the visual data i.e. the classified terrain type derived by the exteroceptive module and the slope derived from the depth map provided by stereo cameras. As mentioned, the output parameters correlated in the trafficability model have already been calculated and processed in the training phase. Hence, in this phase, the system just needs to fetch those parameters that are correlated with given terrain type and slope, from the model and output them to the system so that it can predict the trafficability risk for the rover.

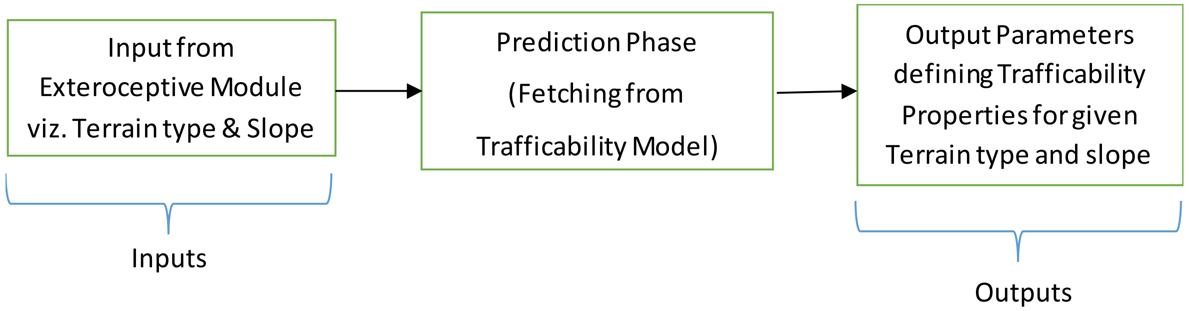
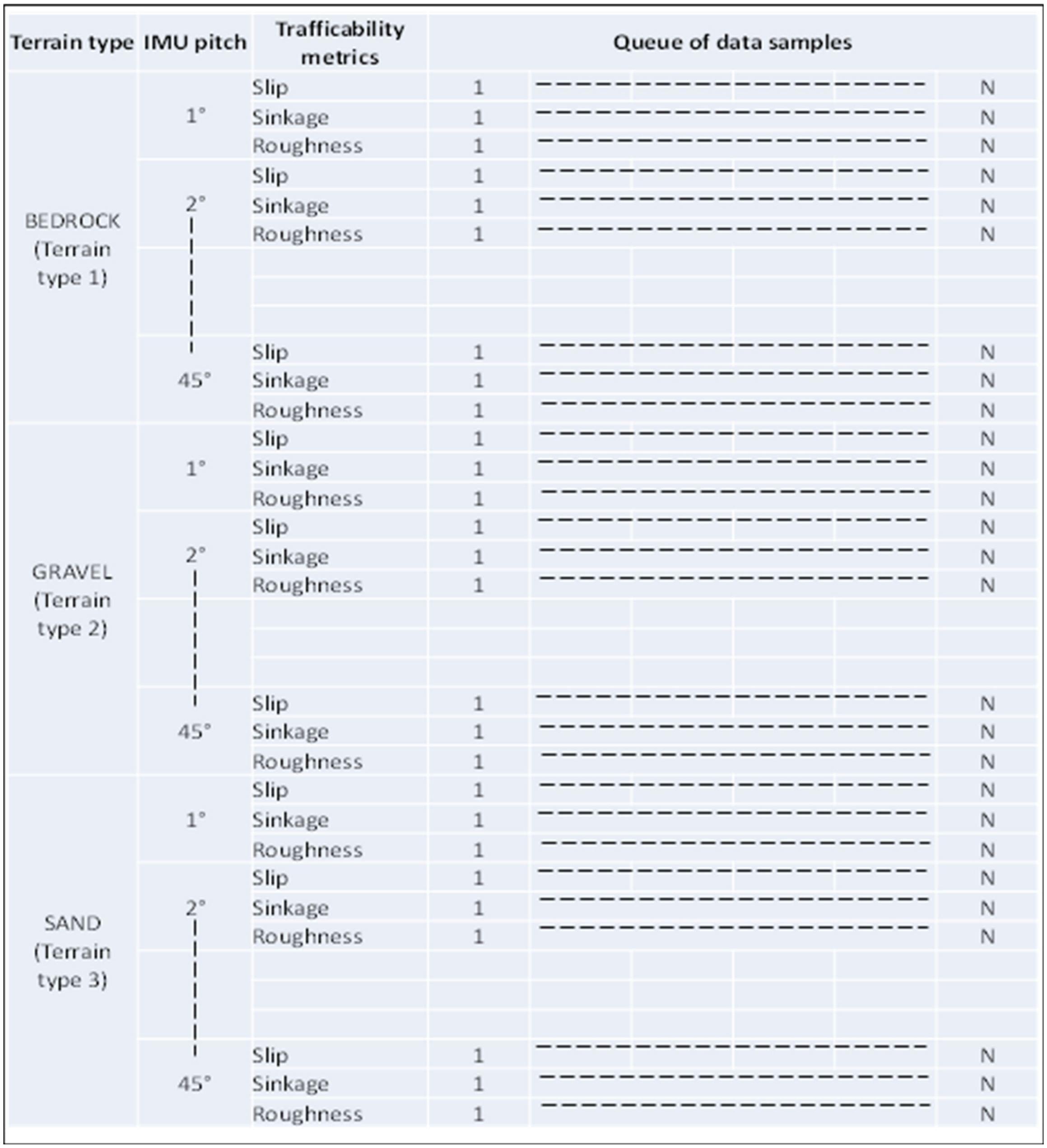


Figure 4-12: Block Diagram of Prediction Phase

4.3 Algorithm of Training Phase

As discussed above, in the training phase, three inputs namely terrain type, IMU Pitch and trafficability metrics are correlated with one another, which updates the correlated data model as well as the trafficability model.

4.3.1 Structure of Correlated Data Model



Terrain type	IMU pitch	Trafficability metrics	Queue of data samples		
BEDROCK (Terrain type 1)	1°	Slip	1	--------------------	N
		Sinkage	1	--------------------	N
		Roughness	1	--------------------	N
	2°	Slip	1	--------------------	N
		Sinkage	1	--------------------	N
		Roughness	1	--------------------	N
	45°	Slip	1	--------------------	N
		Sinkage	1	--------------------	N
		Roughness	1	--------------------	N
GRAVEL (Terrain type 2)	1°	Slip	1	--------------------	N
		Sinkage	1	--------------------	N
		Roughness	1	--------------------	N
	2°	Slip	1	--------------------	N
		Sinkage	1	--------------------	N
		Roughness	1	--------------------	N
	45°	Slip	1	--------------------	N
		Sinkage	1	--------------------	N
		Roughness	1	--------------------	N
SAND (Terrain type 3)	1°	Slip	1	--------------------	N
		Sinkage	1	--------------------	N
		Roughness	1	--------------------	N
	2°	Slip	1	--------------------	N
		Sinkage	1	--------------------	N
		Roughness	1	--------------------	N
	45°	Slip	1	--------------------	N
		Sinkage	1	--------------------	N
		Roughness	1	--------------------	N

Figure 4-13: Structure of the Correlated Data Model

As mentioned in the design overview of proprioceptive module, the trafficability metrics viz. slip, sinkage & roughness are correlated with the appropriate exteroceptive data (slope, terrain type) to form the base architecture of the correlated data model and the structure of this data model seems as shown below in Figure 4-13.

As can be seen here, each terrain type has 45 pitch values associated with it and each pitch value has 3 queues of data samples associated with it, which corresponds to 3 trafficability metrics (Slip, Sinkage & Roughness). These queues store the values of the trafficability metrics measured by the system in FIFO manner. Thus, each terrain type has 45 queues associated with itself. Moreover, each queue can store N data samples as shown in Figure 4-13. N decides the size of the queue i.e. number of latest data samples being held by the queue.

4.3.1.1 Importance of Queue Size (N)

The selection of this parameter N i.e. the queue size is crucial for the responsiveness and stability of the system. Selecting a very small size of queue means older data will keep getting replaced with newer ones quite frequently, which in turn means that system will adapt to the changing conditions instantly. This increases responsiveness of the system. However, this may lead a system's behavior to be highly unstable. Even if a specific terrain behaves differently for short time span, it will lead the system to change its trafficability prediction. On the contrary, a higher size of queue will make the system stable, but will reduce its responsiveness at the same time.

Thus, size of queue plays a decisive role in responsiveness and stability of the system and hence, selecting an adequate value for the same is very important. The effect of this factor on system's behavior has been discussed further with example in Field Test Results (Chapter 5).

Apart from that, some important aspects and the reasoning behind certain selections, regarding this structure, are discussed below:

- Even though the IMU is capturing the pitch values at higher resolution, in the correlated data model, the pitch values used for the correlation are taken at 1° resolution. This is because, given the fact that slip measurements on pitch values, very close to each other (having <1° difference) cannot be significantly different, it does not seem logical to use such high resolution for correlation and moreover, using high resolution also requires higher storage capacity.

  Hence, all the pitch values obtained from IMU are floored to their nearest integer values, thus separating the data samples into the bins based on their pitch values. This implies that all the values from 0° to 0.9° are considered to be 0° and all the samples measured at these pitch values will be added in the queue (bin) of 0° slope and so on.

- The pitch values are ranging from 0° to 45° for each terrain type. The reason for the same lies in the observations from data analysis in section 4.1.2. It can be seen from the terramechanics models of slip vs. IMU pitch that, by the time the pitch value reaches around 40°, the slip measurements are already crossing the 100% mark. Hence, one can conclude that trafficability risk for a slope of above 45° (for any terrain) can always be predicted as high. Moreover, it was also observed during the test runs that there is a possibility of rover tip-over, beyond 45° slope. These observations lead to the conclusion that whenever there is a query for a slope value higher than 45° for any terrain type, system should always predict 100% slip and the highest risk. There is no need, to record the measurements for such high slopes and depend on them for the prediction. Hence, the pitch values beyond 45° are not added to the correlated data model. In the training phase, if the pitch value beyond 45° is given as an input, the training will be skipped and that data sample will not be added.
- The queue size parameter N has been defined as a user configurable parameter. This ensures that, if the present value of N does not seem to be sufficient for the reliable prediction, the human expert in the loop can easily change it online.

4.3.2 Algorithm for the Correlated data model update

The correlated data model is stored in form of a structured data file (text file) in the memory of a computer, to make it easily accessible and also to ensure that the model does not get wiped out in case of power loss. This text file has a separate line for each queue. Since IMU pitch values range from 0 to 45°, this text fie has 45*3 = 135 lines for each terrain type, giving a total of 135*3 = 405 lines (for 3 terrains) in the entire file.

The flowchart shown in Figure 4-14 illustrates the algorithm for adding a new data sample in an appropriate queue. As can be seen, the inputs (trafficability metrics, IMU pitch and terrain type) are taken from exteroceptive as well as proprioceptive sensors and initial checks have been performed so as to accept only the appropriate pitch value. Secondly, the corresponding queues for given slope & terrain type (inputs) are scanned to retrieve the number of data samples present in them. This is accomplished by scanning a new line character while reading the text file into 2-D matrix. This step determines the end of line. A counter keeps increasing until new line character is found and then this count is stored in the (N+1)th column of the matrix. If the queue has less than N data samples, the new data samples can simply be added in the queue. However, if it is full, FIFO structure is implemented by discarding the oldest data and adding the latest one.

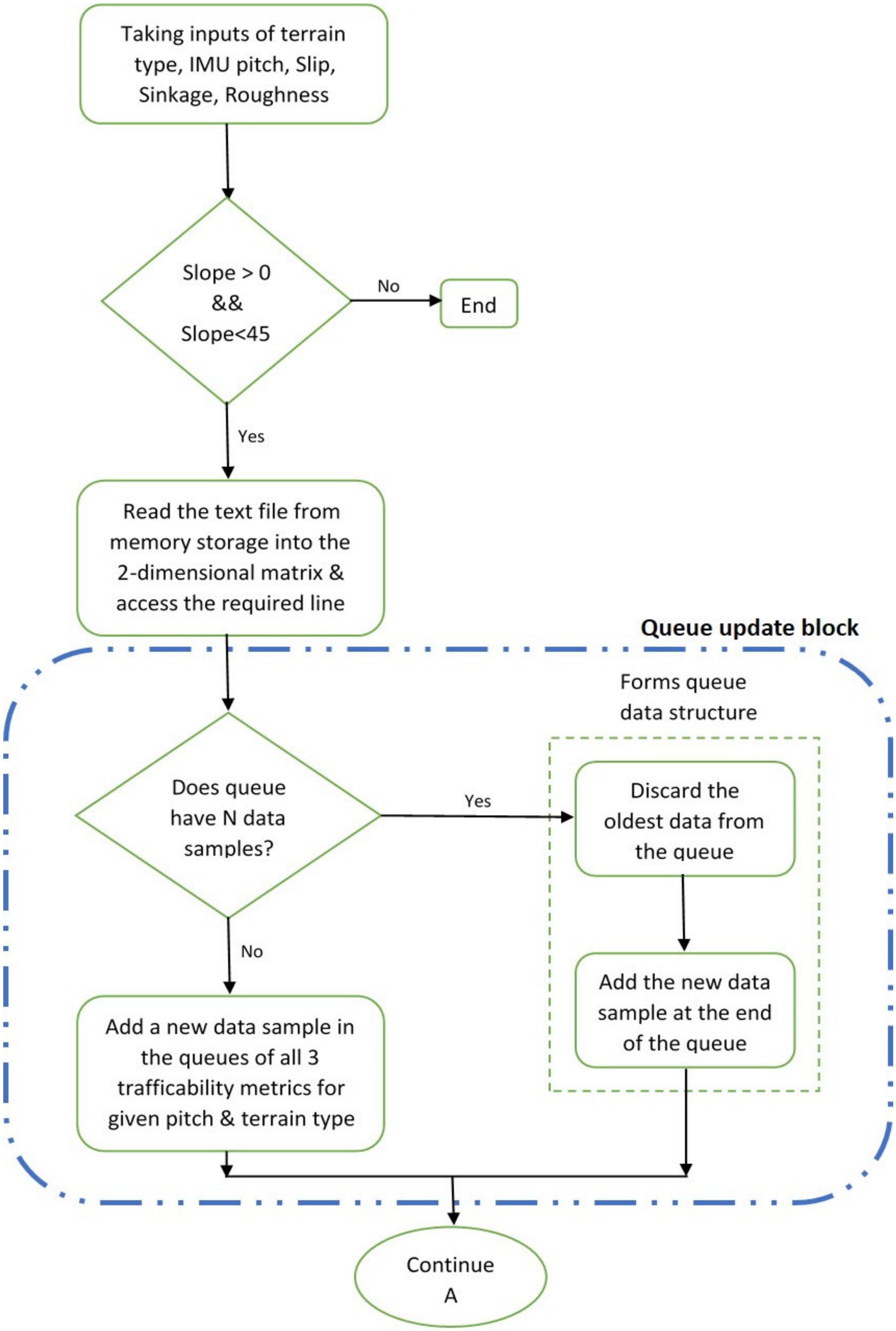
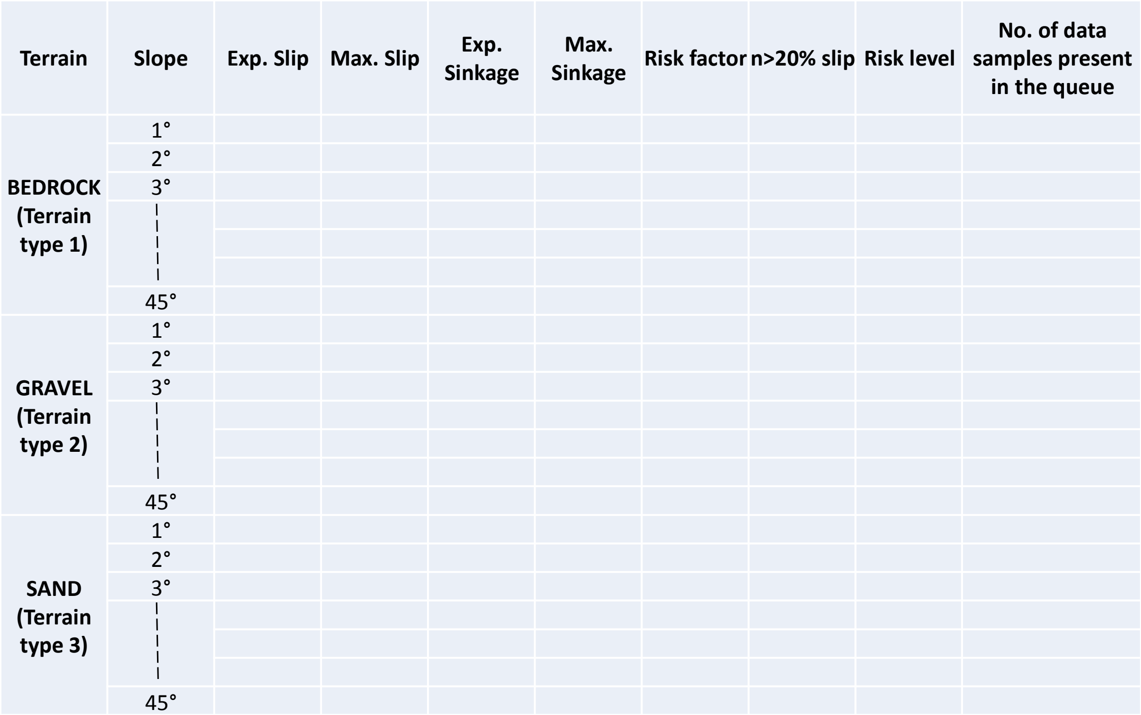

Taking inputs of terrain type, IMU pitch, Slip, Sinkage, Roughness
Slope > 0 && Slope<45
No
End
Yes
Read the text file from memory storage into the 2-dimensional matrix & access the required line
Queue update block
Forms queue data structure
Does queue have N data samples?
Yes
Discard the oldest data from the queue
Add the new data sample at the end of the queue
No
Add a new data sample in the queues of all 3 trafficability metrics for given pitch & terrain type
Continue A

Figure 4-14: Algorithm for adding a new data sample in a queue

4.3.3 Structure of the Trafficability Model

As mentioned previously in the design overview section 4.2, the output parameters, stored in the trafficability model are calculated and updated in the training phase itself. The structure of the trafficability model is given below in Figure 4-15.



Terrain	Slope	Exp. Slip	Max. Slip	Exp. Sinkage	Max. Sinkage	Risk factor	n>20% slip	Risk level	No. of data samples present in the queue
BEDROCK (Terrain type 1)	1°								
	2°								
	3°								
	¦								
	¦								
	¦								
	45°								
GRAVEL (Terrain type 2)	1°								
	2°								
	3°								
	¦								
	¦								
	¦								
	45°								
SAND (Terrain type 3)	1°								
	2°								
	3°								
	¦								
	¦								
	¦								
	45°								

Figure 4-15: Structure of the Trafficability Model

Similar to the correlated data model, the trafficability model is also stored in memory as a structured data file. As seen from Figure 4-15, this file enlists different output parameters for all the pitch values of each of the terrain types. Each line corresponds to a pitch value of a terrain type and hence, the text file has a total of 45*3 = 135 lines. Also there are 8 output parameters which are essential in deciding the trafficability properties. Thus, accessing this file as a 2-dimensional matrix requires 135 rows and 8 columns in it.

4.3.4 Algorithm for an initial trafficability model update

In the training phase, once a new data member is added in an appropriate queue, the system proceeds to updating the trafficability module so as to reflect the effects of the new data sample on the trafficability prediction. As can be seen from Figures 4-15 & 4-16, they are sequential, indicated by “Continue A”. The system simply calculates the output parameters using all the data samples present in the queue (corresponding to the queried pitch value) and updates the existing values by the new ones in the trafficability model. The details about the calculations of these parameters is given in the next section (section 4.3.5).

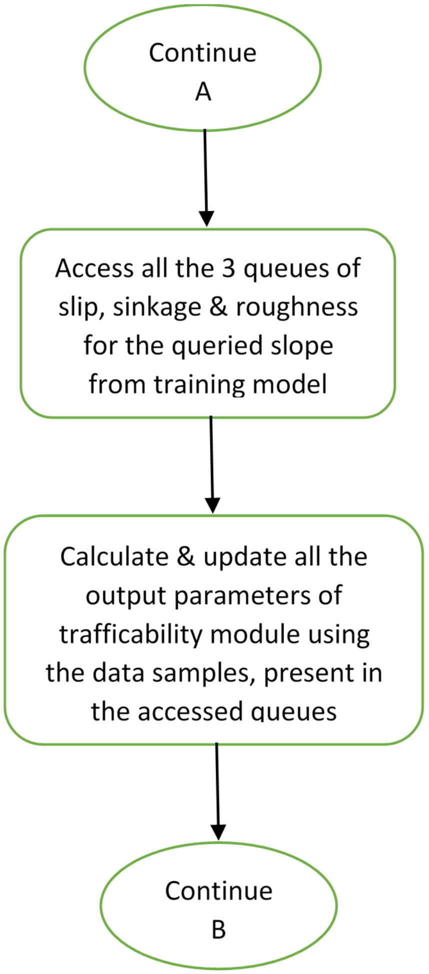

Continue
A
Access all the 3 queues of slip, sinkage & roughness for the queried slope from training model
Calculate & update all the output parameters of trafficability module using the data samples, present in the accessed queues
Continue
B

Figure 4-16: Algorithm for Initial Trafficability Model update

4.3.5 Calculation of output parameters

In the trafficability model, the output parameters listed for each pitch value are: Expected Slip, Maximum Slip, Expected Sinkage, Maximum Sinkage, Risk Factor, count of greater than 20% slip readings, Risk Level and Number of samples present in the queue of the corresponding pitch value. The Expected Slip and the maximum slip are calculated as average and maximum of all the data samples present in the slip-queue for the given slope, whereas, maximum slip is the maximum of all the readings in the same queue. Similarly, the Expected Sinkage and Maximum sinkage is the average and the maximum of all data samples present in the sinkage-queue of the given pitch. Risk factor is calculated as the ratio of number of times the slip has been greater than 20% with respect to the total number of data samples present in the queue.

Risk level is the result that decides whether the terrain is high risk, medium risk or Low risk in terms of trafficability. It is basically decided by threshold levels of the risk factor value and the expected slip value. When these two values cross their corresponding thresholds, the risk level transitions from Low to medium and then to high. In this system, the Risk factor value lower than 5% and expected slip lower than 20% is termed as 'Low Risk' level. Similarly, 'Medium Risk' level is when the risk factor is in between 5% and 65% and expected slip value is in between 20% and 50%. For the risk factor of more than 65% or the expected slip of greater than 50%, the risk level is termed to be 'High'. Using the expected slip value along with the risk factor for the final trafficability risk prediction ensures that, if the risk factor value is lower than a certain threshold just because of lack of data samples for the given slope, then it does not diminish the effect of higher expected slip resulted from interpolation or extrapolation (explained in section 4.3.7).

4.3.6 Algorithm for the Learning Framework/Online update

As discussed earlier in section 4.2, online update of the correlated data and trafficability model takes place according to the learning framework and requires considering and handling some special cases. Moreover, certain cases were also discussed there (in section 4.2.1) one of them being, where there are no/very few data samples in the queues of given pitch and terrain type and hence, it is not possible to make reliable prediction in such cases. These kinds of cases can be handled by interpolation or extrapolation.

For extrapolation, it has been also mentioned in the same section that the extrapolated values should not be used for updating the trafficability model. This is because the extrapolated values are calculated using the data samples of the other terrain types and it is not advisable to train the rover to predict the trafficability of a specific terrain type based on the other terrain type's data samples. As the extrapolated values are used only to predict the trafficability for the current terrain, and not to train the rover, it is not required to execute the extrapolation in the training phase. It is executed in the prediction phase only and hence, the extrapolation algorithm is explained in the prediction phase (section 4.4).

Apart from that, the algorithm and implementation for interpolation has been explained next in section 4.3.6.1. Also, one more case has also been considered here, which explains what if a user expert wants to introduce a new terrain type to the system. The details about how to handle this case are discussed in section 4.3.6.2.

4.3.6.1 Interpolation

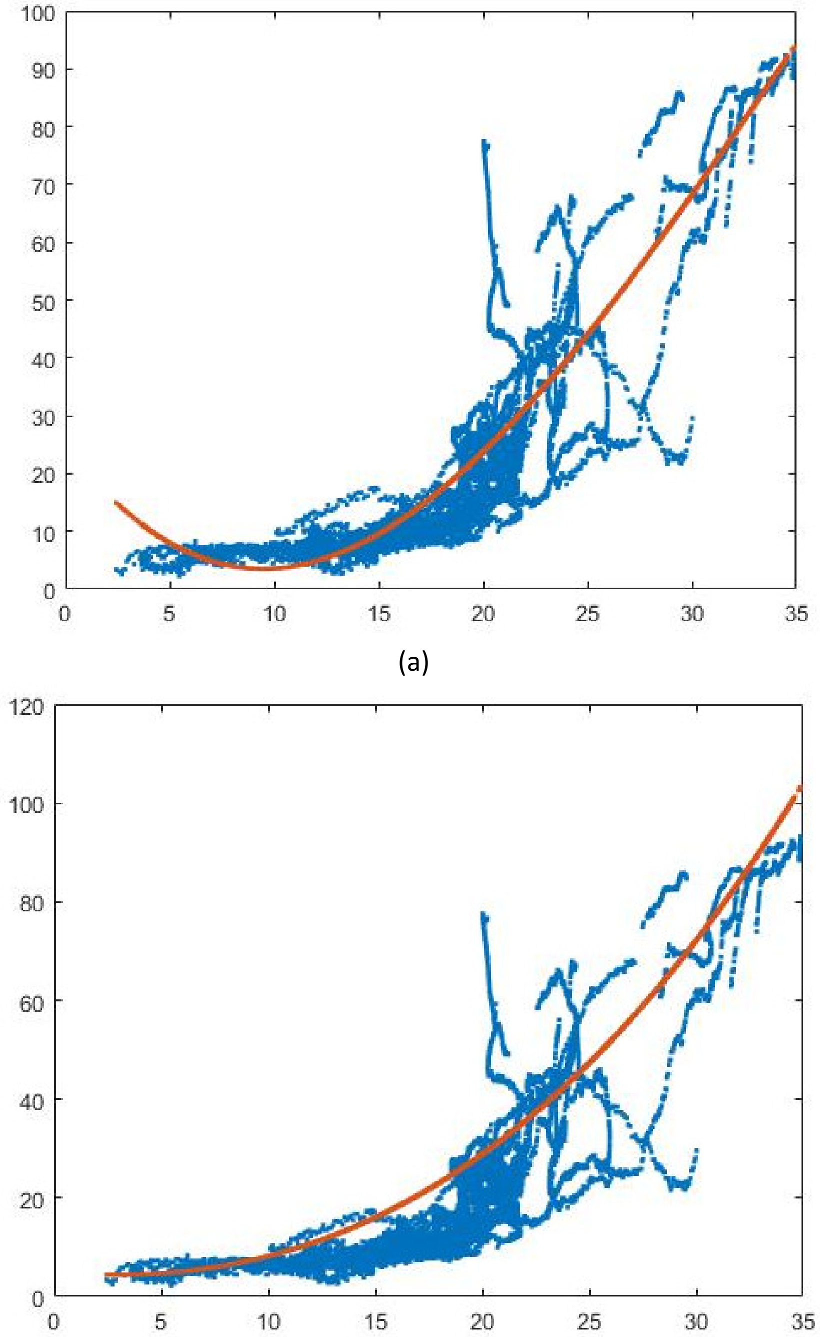
Interpolation is generally used to estimate a value in between two known values. It is imperative to have at least one slope on both the sides of the queried one, with a queue having an adequate number of data samples (N/4 in this case) and hence, in the cases where this condition is satisfied, interpolation becomes the obvious choice for the estimation of expected slip.

In this system, linear regression model using cubic polynomial has been used for interpolation. This cubic regression is performed keeping the fixed y-intercept. The reason behind the same is that while collecting the data during test runs, rover may not encounter all the slope values for all the terrains and hence, the data set may not have slip measurements at 0° slope for some terrains. For example, if there is no slip measurements for slope values from 0 to 10°, and interpolated slip needs to be calculated at 11°, there would be no slope below the queried one with a queue having an adequate data samples and hence, interpolation would not have been possible. Thus, having a fixed y-intercept will ensure that there is always a data point at 0° slope and it is possible to estimate an interpolated slip for the queried slope as long as at least one higher pitch value has an adequate data samples. Also, using a fixed y-intercept overcomes the occurrence of an up-turning in the cubic regression at lower slope values, which is undesirable. In Figure 4-17 (a)**,** the fitting curve using cubic regression without a fixed y-intercept is shown, where it can be observed that for the slope value lesser than around 7°, the corresponding slip values increases with the decreasing slope value. Whereas in Figure 4-17 (b), the fixed y-intercept is avoiding the occurrence of up-turning in the curve. The value of y-intercept is fixed at the average of first N data samples from the queues of lowest slope values.

Secondly, it is notable here that all the checks and updates regarding interpolation are taking place in the training phase itself. This is because every data sample being added to the Correlated data model affects the curve-fit obtained for calculating interpolated values. Hence, in every training iteration, the necessity to update the interpolation coefficients is checked as shown in the flowchart (Figure 4-18). For the same, the number of training data samples present for all the slope values is scanned sequentially. If any slope queue does not have enough data samples (i.e. n<N/4), the average slip calculated for that slope value is not reliable and the same has to be updated using interpolation. In this case, further slope values are scanned to determine if any of them has enough data samples (i.e. n>N/4), implying whether interpolation is feasible or not. If so, the corresponding coefficients for the interpolation curve are updated using cubic regression and are stored in the memory so as to be used while updating the expected slip in the trafficability model.

4.3.6.1.1 Updating the Trafficability Model using Interpolation

The next step in the algorithm is to update the trafficability model using the interpolation curve, wherever necessary. For the same, as shown in the flowchart below (Figure 4-19), again the number of data samples present in the queues of all the slope values are scanned sequentially and wherever there aren't enough data samples (n<N/4), the average slip is replaced, by the one which is calculated using interpolation. Further, it is also important to ensure that the expected slip values in the trafficability model should increase monotonically with the slope and the same is being taken care of as well, after calculating interpolated slip, as shown in the same flowchart.



(a)

Figure 4-17: Cubic regression (a) without fixed Y-intercept (b) with Y-intercept

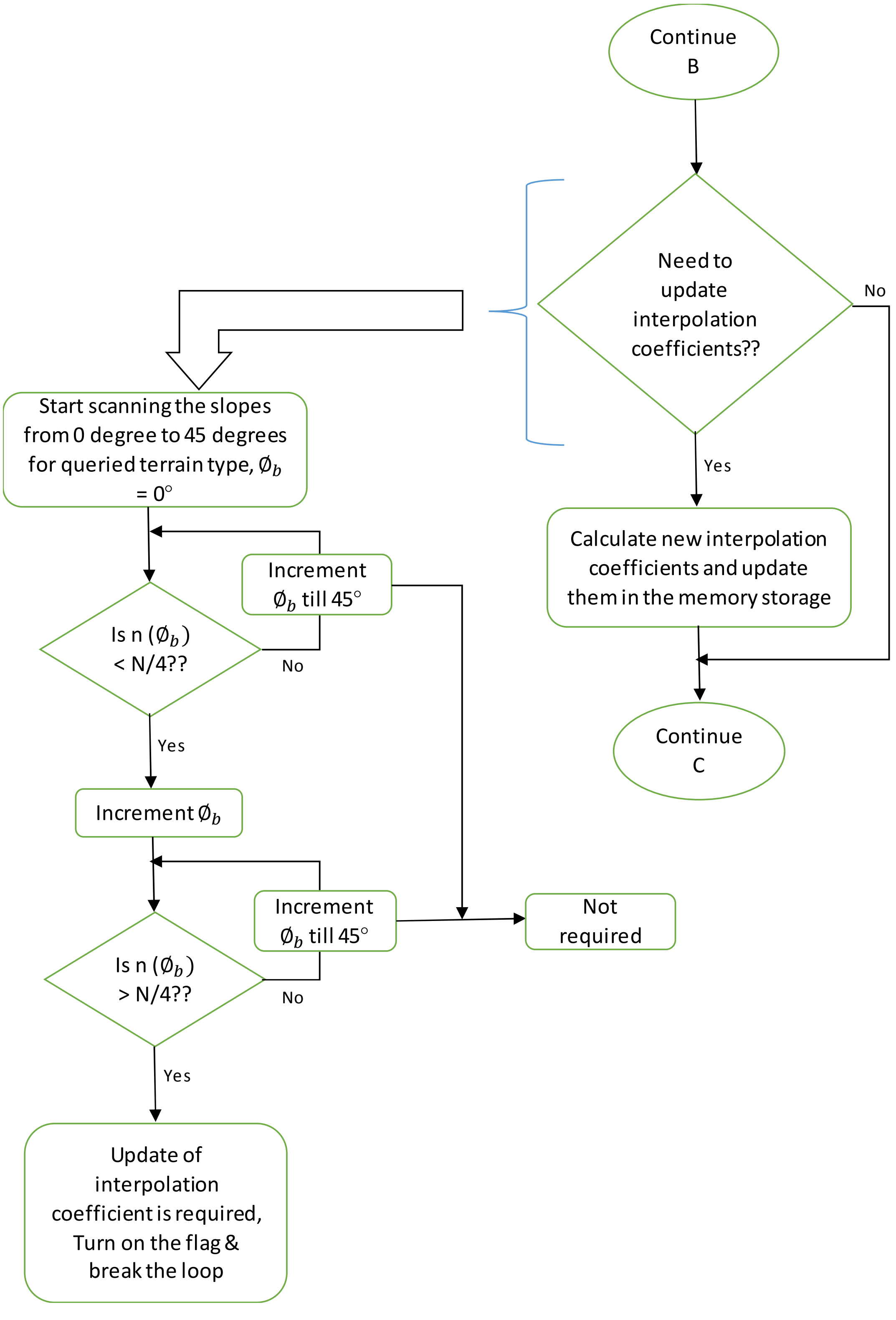

Continue B
Need to update interpolation coefficients??
No
Yes
Calculate new interpolation coefficients and update them in the memory storage
Continue C
Start scanning the slopes from 0 degree to 45 degrees for queried terrain type, $\emptyset_b$ = 0°
Increment $\emptyset_b$ till 45°
Is n ($\emptyset_b$) < N/4??
No
Yes
Increment $\emptyset_b$
Increment $\emptyset_b$ till 45°
Not required
Is n ($\emptyset_b$) > N/4??
No
Yes
Update of interpolation coefficient is required, Turn on the flag & break the loop

Figure 4-18: Algorithm for Interpolation check in Training phase

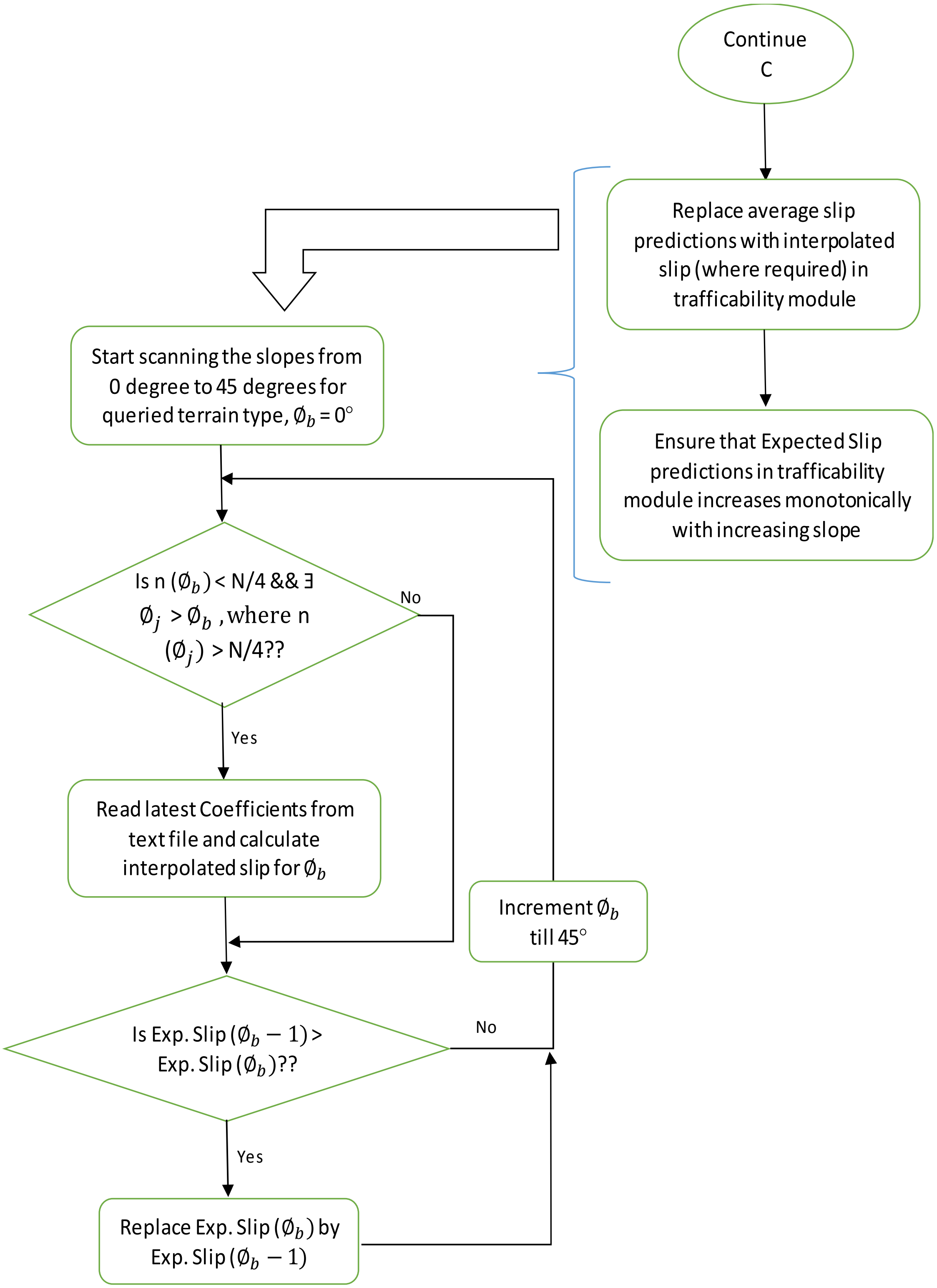

Continue
C
Replace average slip predictions with interpolated slip (where required) in trafficability module
Ensure that Expected Slip predictions in trafficability module increases monotonically with increasing slope
Start scanning the slopes from 0 degree to 45 degrees for queried terrain type, $\emptyset_b$ = 0°
Is n ($\emptyset_b$) < N/4 && ∃ $\emptyset_j$ > $\emptyset_b$, where n ($\emptyset_j$) > N/4??
No
Yes
Read latest Coefficients from text file and calculate interpolated slip for $\emptyset_b$
Increment $\emptyset_b$ till 45°
Is Exp. Slip ($\emptyset_b - 1$) > Exp. Slip ($\emptyset_b$)??
No
Yes
Replace Exp. Slip ($\emptyset_b$) by Exp. Slip ($\emptyset_b - 1$)

Figure 4-19: Updating the Trafficability Model using Interpolation

4.4 Algorithm of Prediction Phase

As discussed in section 4.2.2**,** the inputs to this phase are the classified terrain type and slope from the stereo cameras. The system simply fetches the output parameters from the trafficability model, corresponding to the terrain type and slope passed as inputs and displays the trafficability properties as outputs. However, there are few special cases and considerations which are taken care of in this phase.

4.4.1 Penalty for higher degree slopes

As discussed in section 4.3.1**,** the correlated data model holds the slope values from 0° to 45° only, for each terrain type. The reason for the same lies in the analysis of the raw data collected in the test runs, where the slip measurements touch the 90% mark at the 35° itself. Considering this reason as well as the mechanical design, which might cause the rover to flip at higher than 45° slope values, it has been decided to penalize the output parameters and trafficability risk, if the queried slope is higher than 45°. In such cases, the expected slip is set to 100% and the predicted risk level is always high.

4.4.2 Extrapolation

As discussed in section 4.2, for the cases where enough data samples are not present to predict the trafficability reliably, interpolated or extrapolated slip is calculated as an expected slip. Also, as seen in section 4.3.6, for interpolation the queried slope needs to be between two slopes which have enough data samples. In other cases, where none of the slopes higher than the queried one have enough data samples, interpolation is not possible. Hence, the expected slip for such slopes is determined using extrapolation.

First of all, the necessity of extrapolation is checked for the queried slope and terrain type. If it is required, that means it is not possible to predict the reliable expected slip & risk level using the data from the same terrain. In such a situation, one solution can be to calculate different slip values for each terrain type other than the queried one, using their corresponding data samples and to consider the maximum slip value among all of them as the expected slip, which prepares the system for the worst case scenario. However, for any two terrains Slip vs. Slope characteristics may not be inherently similar and this fact should be considered. Let's take an example to understand this. For example, one needs to calculate an extrapolated slip at 35° for gravel. Calculating the slip values using the data samples of sand and bedrock reveals that sand gives higher prediction at 35°

than bedrock. In this case, the obvious choice might be to use the slip derived from sand data samples as an expected slip. However, a very important factor to be considered here is the ratio between the slip measurements of different terrain types. Let's say gravel has an expected slip of 10% at 0° slope, whereas sand has that of 5% and both of them have monotonically increasing curve. Looking at the fact that gravel has higher expected slip than sand at 0°, one would expect the same slip measurements ratio at 35° as well, considering that the expected slip curves for all the terrains are monotonically increasing. Hence, instead of directly using the sand prediction for extrapolation, the slip0 (average of the N slip measurements at the lowest slope values) is calculated first, for all the terrain types. This gives the idea of the inherent difference between the expected slip curves for different terrain types. Next, the slip0 ratio is calculated for both sand and bedrock with respect to gravel, which are then multiplied (only if it is greater than 1) with the respective predictions (from sand & bedrock respectively) to calculate the extrapolated slips. From these two extrapolated values, the one which is the higher than the other is used as a final extrapolated value. This explanation to calculate extrapolated slip can be summed up by the logical expression given below:

$$Expected\ Slip(\varphi)_i = \max(Expected\ Slip(\varphi)_j * \max\left(\frac{Slip_{0,i}}{Slip_{0,j}}, 1\right))$$

Where, φ = queried slope

i = queried terrain type, j = other terrain types

$Slip_0$ = average of N slip measurements at the lowest slope values

4.5 Summary

The proprioceptive module focuses on the correlation of terrain types and their mobility properties in a framework that enables online learning. To predict terrain trafficability ahead of the rover, exteroceptive data viz. terrain type and slope, are correlated with the trafficability metrics viz. slip, sinkage and roughness. The implementation of training and prediction phases was described in this chapter. A queue based data structure has been implemented for the correlation, which keeps discarding the older data so as to avoid diminishing the effect of newer data samples, when there is a large amount of data. This also ensures that the rover will be able to adapt to changing terrains responses and predict the risk level (low, medium or high) accordingly.

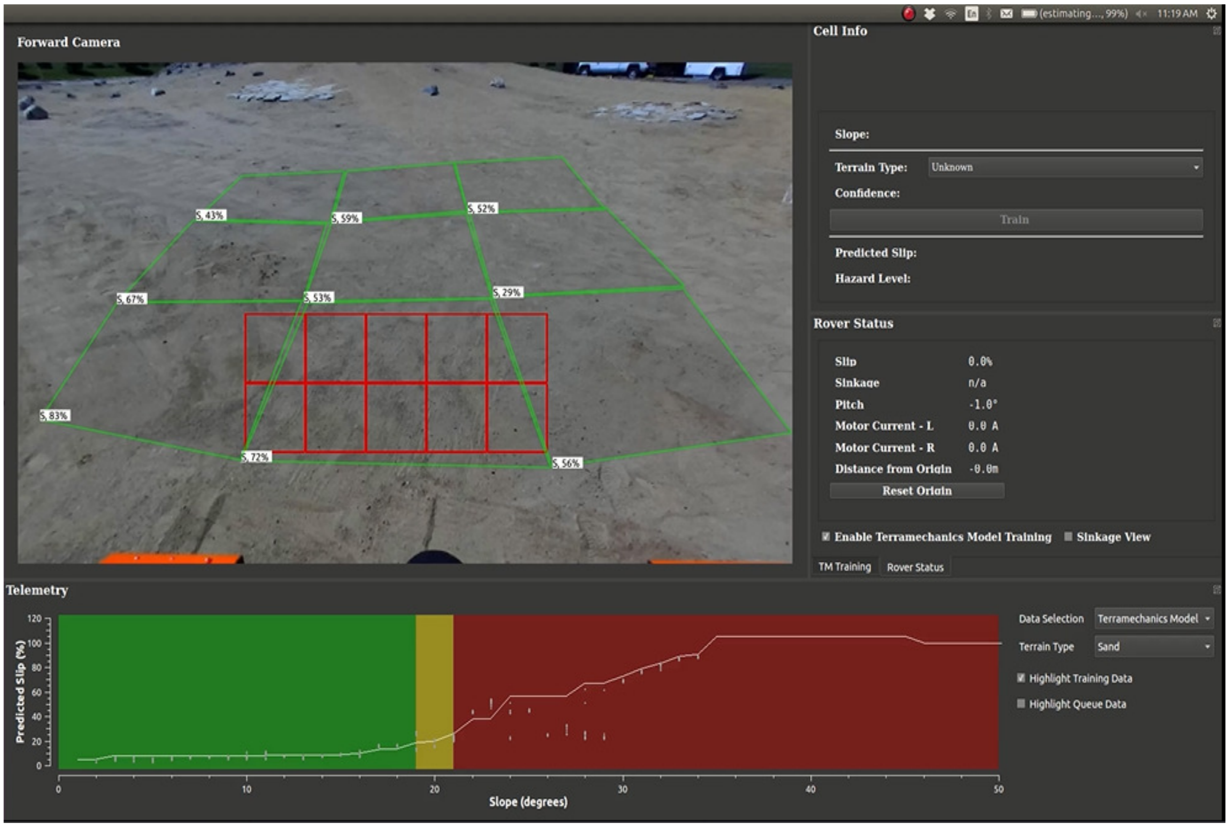
Chapter 5. Field Test Results Analysis

For the verification of the proposed method, a number of test runs were performed on the MARS yard developed by the CSA. For these field test runs, Ontario Gear Drive's J5 rover was used. This rover had been pre-trained with the training datasets, which were formed as described in chapter 3 & 4. For the proprioceptive module, the training library was comprising of the data measured during the test runs, from the site called "Lafarge". Whereas, for the exteroceptive module, the training library comprised of the data from CSA mars yard as well as Lafarge (explained in detail later). Recall that the goal here is to demonstrate the performance of terrain classification as well as a responsive, yet stable trafficability module, where the proprioceptive training model gets updated online using the data being measured, the system classifies the terrains which are in front of the rover and also, when approved by the user, the exteroceptive training model gets updated by adding new classified images to the dataset.

Thus, there were two aspects of the above mentioned goal which were mainly exhibited during these test runs. One is that rover runs on different terrains at different slopes present in the Mars Yard and it exhibits terrain classification based on the exteroceptive module and predicts the slip as a function of slope based on the proprioceptive module. Whereas, the other aspect is that, when the user enables the training of the proprioceptive module using the measured slip and slope values, the training model gets updated online, while the rover is traversing and collecting the data. This aspect exhibits the "Data Driven Model" of the proprioceptive module.

The various parameters measured, and the results derived by the rover using the proposed algorithm were recorded in a GUI, developed by our industrial partner 'Mission Control' and the snapshots of the same has been presented here. The snapshot in Figure 5-1 is presented here just to illustrate the appearance of the GUI.

As can be seen, the camera feed is in the center of the screen. The image is further divided into 9 big grid squares and the small text at lower left corner of each grid square shows the classified terrain type (sand, bedrock or gravel) for that grid. Also, the color of these grid squares shows the risk level predicted for the corresponding grid by the proprioceptive module, based on its slope value and terrain type.



Forward Camera

Cell Info

Slope:

Terrain Type: Unknown

Confidence:

Train

Predicted Slip:

Hazard Level:

Rover Status

Slip	0.0%
Sinkage	n/a
Pitch	-1.0°
Motor Current - L	0.0 A
Motor Current - R	0.0 A
Distance from Origin	-0.0m

Reset Origin

☑ Enable Terramechanics Model Training ☐ Sinkage View

TM Training Rover Status

Telemetry

Data Selection: Terramechanics Model

Terrain Type: Sand

☑ Highlight Training Data

☐ Highlight Queue Data

Figure 5-1: Snapshot of GUI

Further, in the bottom middle grid square, smaller sub-squares can be seen. Each bigger grid square is made of such small sub-grids and the classification is performed at the sub-grids level. All the sub-grids forming a bigger grid square get classified individually as one of the terrain types and then, based on the combined result of these classifications, that particular grid square gets classified in to one of the terrain types. The color of these sub-grids exhibits the terrain type in which they have been classified to. The orange color is for representing the terrain type sand, blue is for bedrock, green is for gravel and black is for unknown/non-classified terrain type.

At last, there is a Predicted Slip vs. Slope plot in the bottom horizontal section, which exhibits the output trafficability model for the given terrain type. The white line shown in the plot displays the expected slip at each slope value, fetched from the output trafficability model and the white dots around this line exhibits the slip measurements, present in the queue of each slope value. The areas covered with different colors exhibit the predicted risk levels, again fetched from the present output model. Here the green color shows low risk terrain, yellow color shows medium risk terrain and red color shows the risk terrain. As the terrain is traversed, the measured data gets added to the training model, the output model also gets updated and so do the risk level predictions.

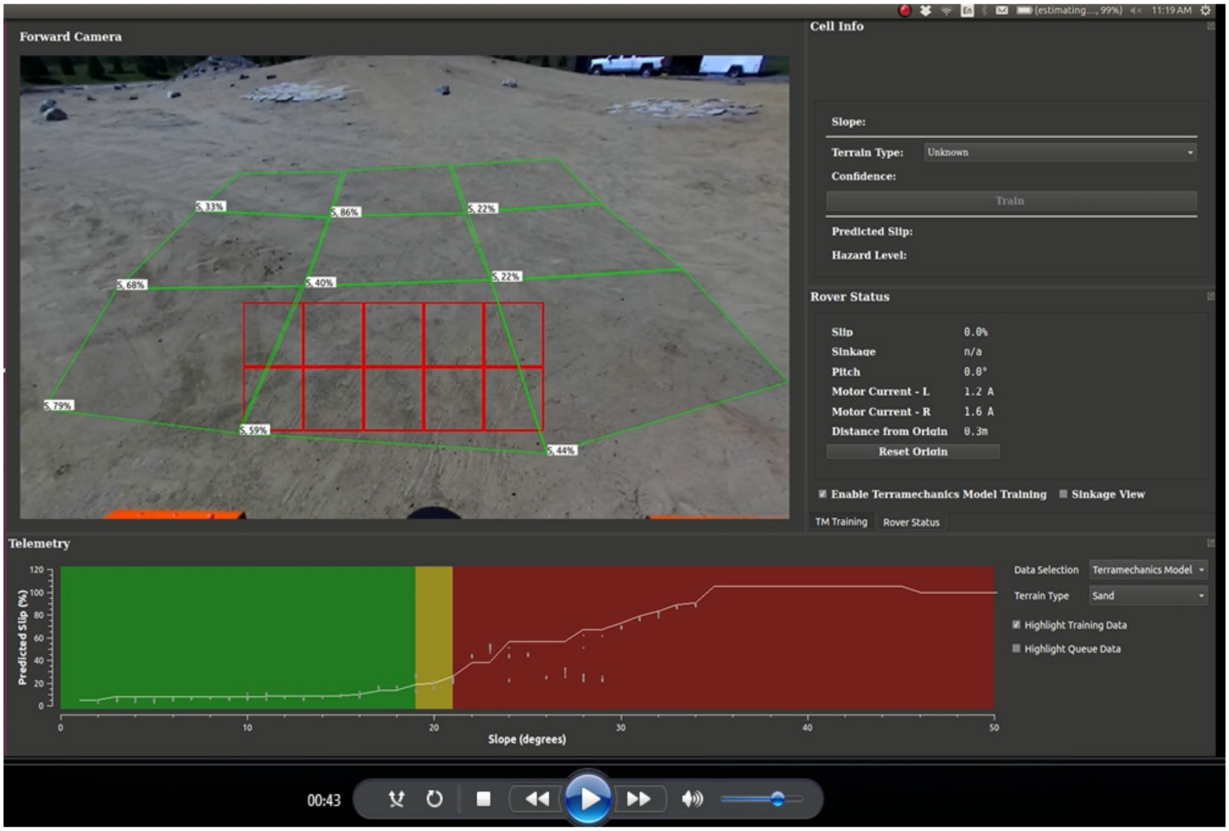
5.1 Important Proprioceptive Results

There were several test runs performed and the results were recorded for all of them, but an important test run has been discussed and analyzed here that significantly and very clearly displays the updates of the terramechanics model by adding new measurements in the training data set and exhibits the responsiveness of the “Data Driven model”. The update and learning of the terramechanics model was observed in other test run as well, however, this test run was specially designed to exhibit the quick responsive nature of the system and its capability of online learning.

This test run is referred to as the “Sled Test”, as a sled carrying weights, was tied behind the rover and the rover was made to run on normal sandy terrain with not-so-high slope. Due to having to pull the sled, rover was experiencing higher slip measurements than it would normally have at the given slope on a sandy terrain. Here, the terramechanics model was observed learning and changing according to the new measurements added to the library. This experiment and the results are illustrated in the figures below which are actually the snapshots taken at certain time stamps from the test run video.

The image here (Figure 5-2) shows the GUI at the origin. The predicted slip vs. slope plot at the bottom shows the trafficability model at the origin i.e. before starting the sled test run. It can be seen here that originally the low to medium risk level threshold is at 19° slope value, whereas the medium to high risk level threshold is at 21°.

Also, in the center of the GUI screen, the front camera feed can be seen. In which, as explained earlier in this section, the classified terrain type and the confidence in the classification is shown for each bigger grid square. Thus, along with the data driven terramechanics model, the classification results is also being validated here. However, one thing to note here is that the exteroceptive module is not being trained by the classified examples. In addition to that, below the rover status table block, the check box for enabling the training of the terramechanics model is checked, which means user has enabled the proprioceptive module's training.



Cell Info

Slope:

Terrain Type: Unknown

Confidence:

Predicted Slip:

Hazard Level:

Rover Status

Slip	0.0%
Sinkage	n/a
Pitch	0.0°
Motor Current - L	1.2 A
Motor Current - R	1.6 A
Distance from Origin	0.3m

☑ Enable Terramechanics Model Training ☐ Sinkage View

Data Selection: Terramechanics Model

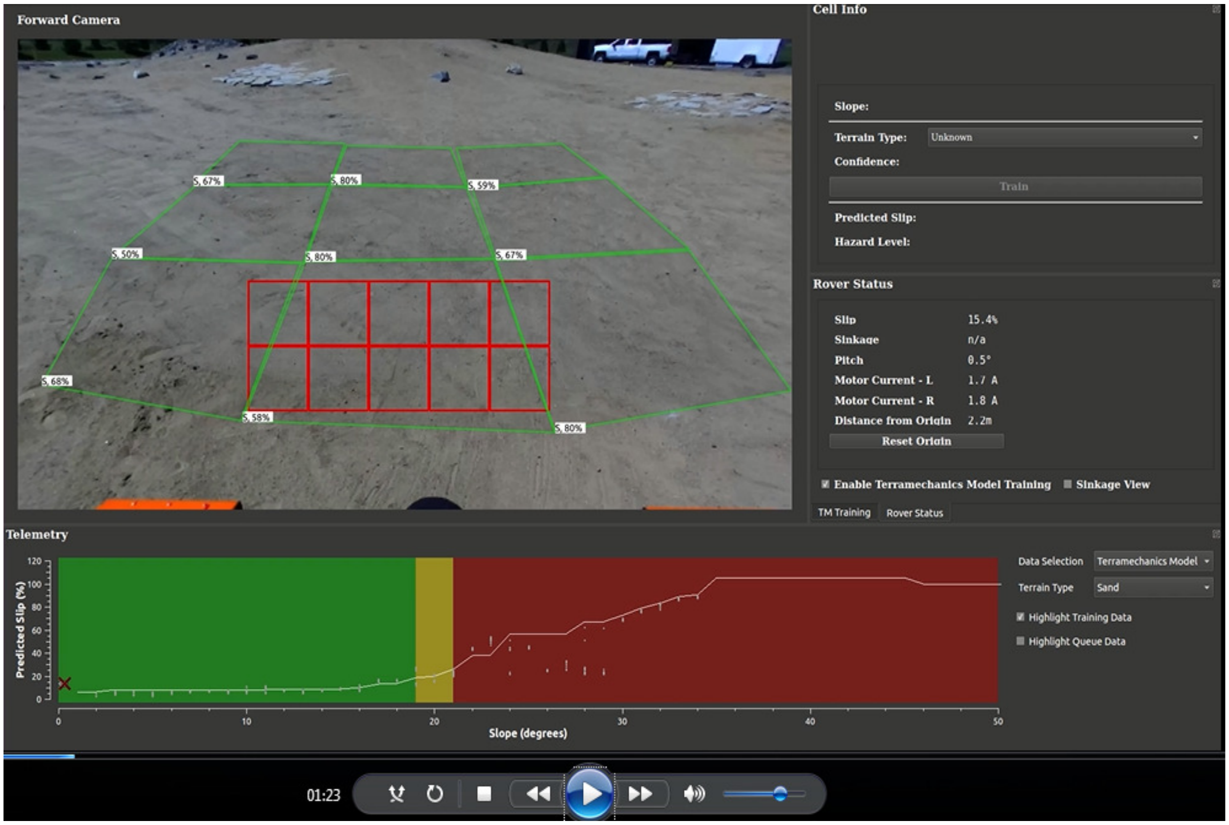
Terrain Type: Sand

☑ Highlight Training Data

☐ Highlight Queue Data

Figure 5-2: GUI at origin

In the next snapshot in Figure 5-3**,** which is taken after some seconds, it can be seen that the recording of new slip measurements has been started. The red colored cross exhibits the current slip measurement that is being added to the training library. However, as evident, the measurements have not yet reached the value that is higher enough to change the predicted slip curve and hence the low to medium as well as medium to high risk level thresholds are still the same as that in the initial plot. Also, in the rover status table block, the current slip and slope measurements can be observed.



Slope:

Terrain Type: Unknown

Confidence:

Predicted Slip:

Hazard Level:

Slip 15.4%
Sinkage n/a
Pitch 0.5°
Motor Current - L 1.7 A
Motor Current - R 1.8 A
Distance from Origin 2.2m

☑ Enable Terramechanics Model Training ☐ Sinkage View

Data Selection Terramechanics Model

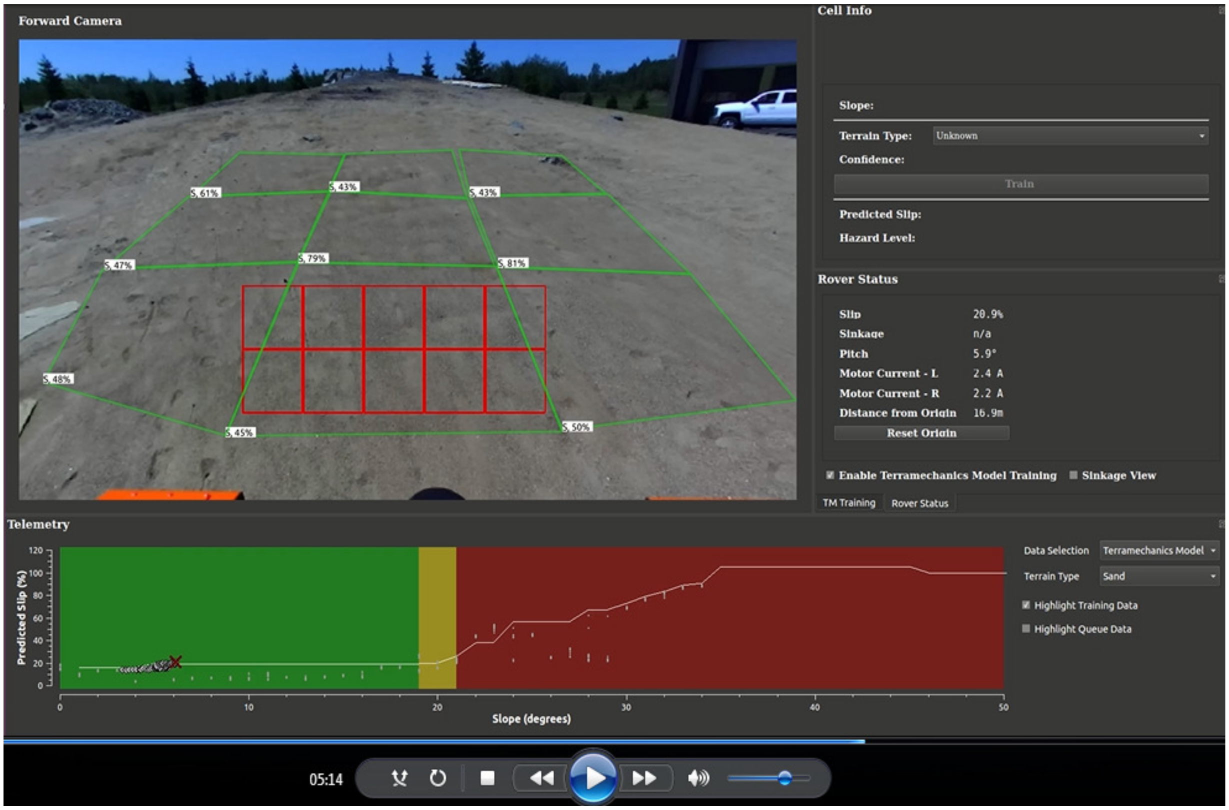
Terrain Type Sand

☑ Highlight Training Data

☐ Highlight Queue Data

Figure 5-3: GUI at 1:23 time stamp

The next snapshot shown in the Figure 5-4 is taken at 5:14 time stamp. As can be seen from the GUI, there is no slip measurement recorded above 20% before this point of time. The very first reading above 20% is recorded at 5:14 seconds (20.9%) and hence, the low-to-medium and med-to-high risk level thresholds are still the same i.e. at 19 and 21°, respectively. However, a notable observation is that the curve for the expected slip is modified from that in the 1:23 seconds snapshot, due to the addition of high slip measurements in the training library. As evident from Figure 5-3, the curve was at around 10% expected slip value initially for the slope values ranging from 1 to 16°, which is now around 20% (Figure 5-4)**.**



Slope:

Terrain Type: Unknown

Confidence:

Predicted Slip:

Hazard Level:

Slip 20.9%
Sinkage n/a
Pitch 5.9°
Motor Current - L 2.4 A
Motor Current - R 2.2 A
Distance from Origin 16.9m

☑ Enable Terramechanics Model Training □ Sinkage View

Data Selection Terramechanics Model

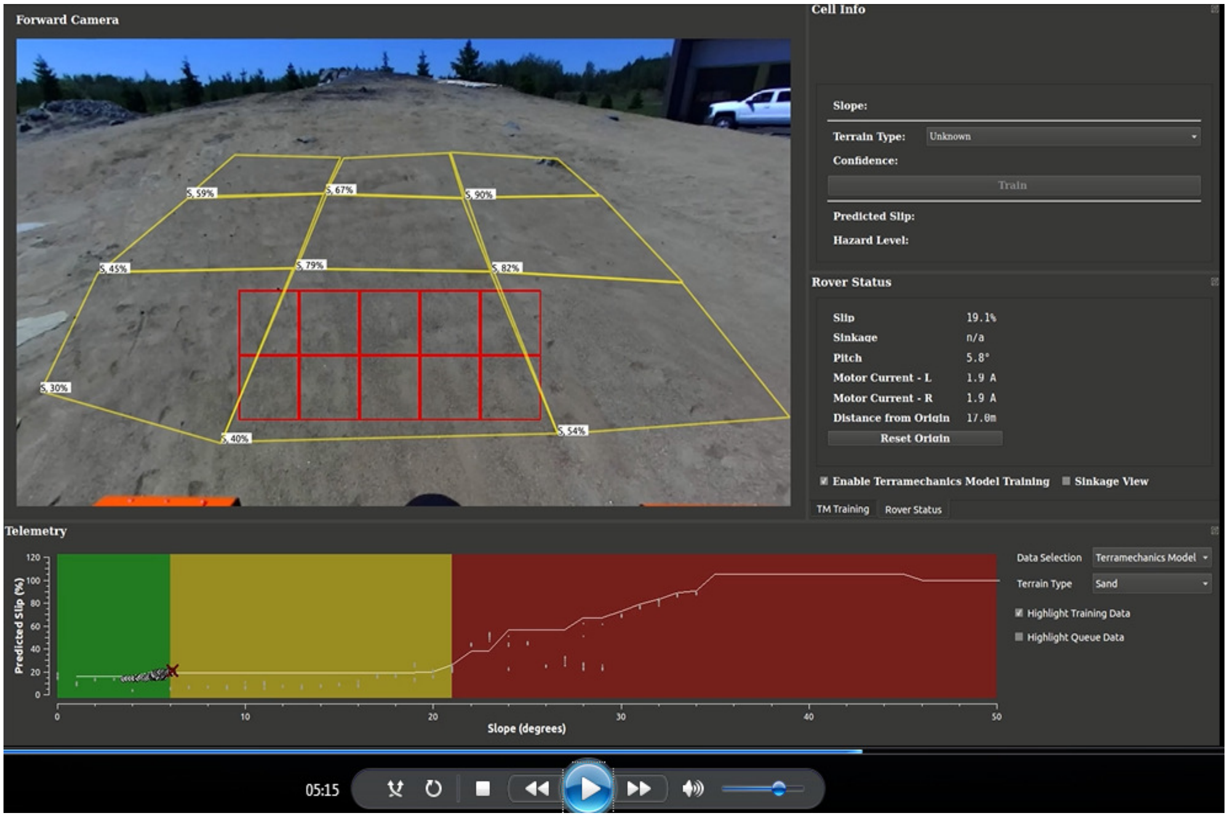
Terrain Type Sand

☑ Highlight Training Data

□ Highlight Queue Data

Figure 5-4: GUI at 5:14 time stamp

Now, as soon as a slip measurement of 20.9% is added to the training library in the queue of 6° slope, in the very next second (5:15 time stamp) the low-to-medium risk level threshold shifts to 6° from 19° (Figure 5-5). This is explainable by recalling the risk level calculation explained in the proprioceptive module's section, the risk level transitions from low to medium, as soon as the expected slip reaches beyond 20% or 5% of the readings in any slope's queue are more than 20%. Here the queue length has been selected to be 20, whose 5% is 1 and hence, as long as one reading is recorded above safety slip threshold value (20%), the risk level transitions from low to medium. The risk level prediction is also changed for the classified terrains which are ahead of the rover, as can be seen from the color of the bigger grids in the camera feed section of the GUI.



Slope:
Terrain Type: Unknown
Confidence:
Predicted Slip:
Hazard Level:

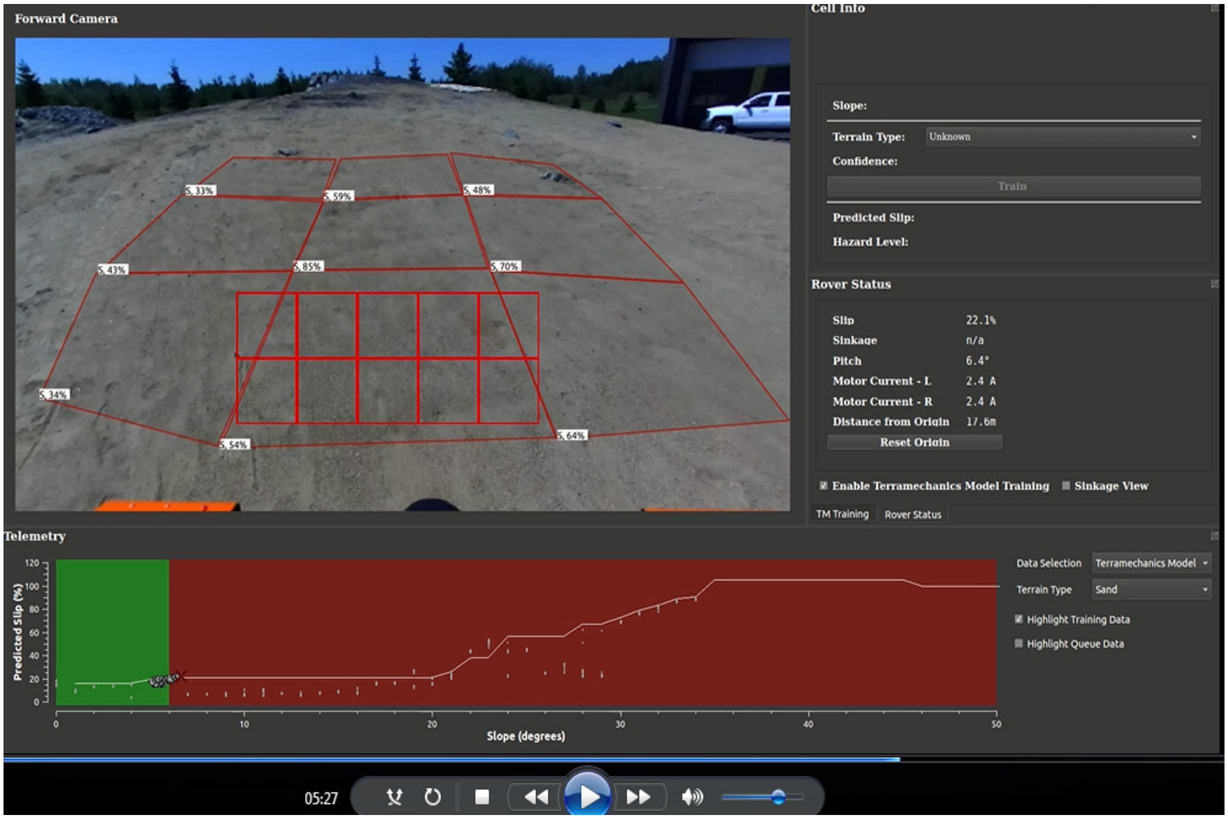
Slip 19.1%
Sinkage n/a
Pitch 5.8°
Motor Current - L 1.9 A
Motor Current - R 1.9 A
Distance from Origin 17.0m

☑ Enable Terramechanics Model Training ■ Sinkage View

Data Selection Terramechanics Model
Terrain Type Sand
☑ Highlight Training Data
■ Highlight Queue Data

Figure 5-5: GUI at 5:15 time stamp

Similar to the low-to-medium risk level transition, medium-to-high transition is also explained in the same section, which says that as soon as expected slip goes beyond 50% or more than 65% of the readings in the queue are recorded above 20% threshold, the risk level will transition from medium to high. In the "Sled Test Run", after the low-to-medium risk level transition took place at 6°, almost all the readings were recorded above 20% slip value and all were for the slope values ranging from 5.8 to 6.4. Hence, as soon as 13 readings were recorded above 20% after some seconds (at 5:27 seconds time stamp), the medium risk level at 6% transitioned into high risk level. So the high risk level which was earlier being predicted for the slope range of 21 to 45°, is now being predicted for all the slopes from 6 to 45° as evident from the Figure 5-6 and the modified risk level prediction for the terrains ahead of the rover is visible from the color of the grids here as well.



Slope:

Terrain Type: Unknown

Confidence:

Predicted Slip:

Hazard Level:

Slip 22.1%

Sinkage n/a

Pitch 6.4°

Motor Current - L 2.4 A

Motor Current - R 2.4 A

Distance from Origin 17.6m

☑ Enable Terramechanics Model Training ■ Sinkage View

Data Selection Terramechanics Model

Terrain Type Sand

☑ Highlight Training Data

■ Highlight Queue Data

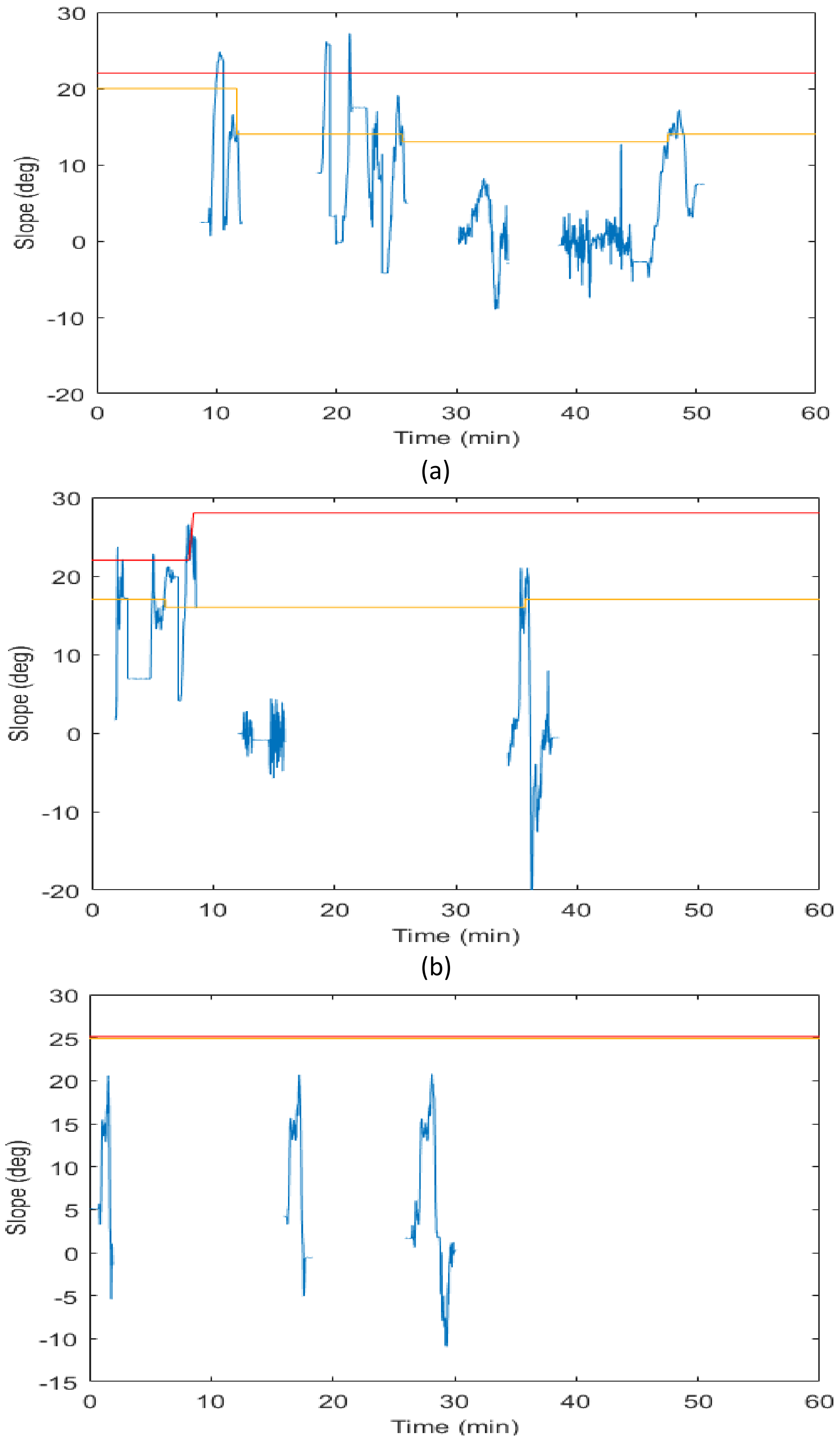
Figure 5-6: GUI at 5:27 time stamp

This whole experiment was set up to exhibit the responsiveness of the data driven model. However, along with being responsive, system should possess certain level of stability as well. As explained earlier in section 4.3.1, the decisive factor for responsiveness and stability of the system is the queue size. For example, the above mentioned experiment (sled test) was performed with two different queue sizes. The one discussed above in this section is performed with the queue size of 20 and the other one was also performed with the queue size of 100. It was observed in the first one that the system is responsive, whereas in the latter, one the risk level thresholds did not change at all because of insufficiency of high slip data samples in a queue (stable but not responsive). Thus, queue size is responsible to maintain the balance between responsive and stable behavior of the system and we have two extreme values of queue size. Hence, an analysis was carried out to find an adequate value of queue size that balances the responsiveness and stability. It revealed that

32 is an adequate queue size and to exhibit the same, following plots have been derived. These plots show the changes in risk level thresholds with time, as different tests were performed.

Yellow line in the plots indicates the low to medium (L2M) risk level threshold, whereas red line indicates the medium to high (M2H) risk level thresholds. Moreover, the blue points on the plot are the actual slope value measurements, those were encountered by the rover while the traversal. As can be seen in the plot for sand, there is a large displacement in L2M risk level threshold around the first time when the rover started driving on slopes near that threshold. A similar large displacement can also be observed in M2H risk level threshold, in the plot for gravel. Such large displacements in thresholds during the initial rover runs can be justified by the fact that the proprioceptive module has been trained with the data from Lafarge, where the terrains behaved differently than CSA and hence, initially rover was predicting the risks in accordance with the Lafarge terrains. But once it drove on the similar slopes at CSA mars yard and gathered the data from there, the system learned the behavior of the terrains at CSA mars yard and hence, changed its prediction. This exhibits the responsiveness of the system.

However, after this initial significant displacement, a quite stable prediction can be observed in both the cases, with only 2 displacements of 1° each, in the L2M risk threshold in sand's plot. Similar small displacements can also be observed in L2M risk threshold in case of gravel. These small displacements verifies that system is responsive at all the time and is able to detect if suddenly a terrain starts behaving differently. At the same time, maintaining almost a constant risk threshold for a specific terrain type ensures that system possess stability as well, along with the responsiveness.



(a)

(b)

(c)

Figure 5-7: Risk level thresholds analysis with N=32 and (a) Sand, (b) Gravel and (c) Bedrock

5.2 Exteroceptive Results

The overall performance of the exteroceptive module observed during the test runs is presented and discussed in this section. The test runs were performed with 3 different training data sets named as “Original Dataset”, “New_CSA Dataset” and “Pure_CSA Dataset”. The results obtained with all of them is shown below in Table 18.

Terrain Type	**Original Dataset**	**New_CSA Dataset**	**Pure_CSA Dataset**
Bedrock	43.28%	21.37%	47.80%
Sand	35.33%	94.92%	97.11%
Gravel	99.6%	81.51%	90.10%

Table 18: Field Test Results for Exteroceptive Module

As discussed in the “CSA dataset” section, the training data collected from the CSA mars yard was not diverse enough and hence, to start with the test runs, it was decided to collect some more images from another site in Lafarge and add them to the existing training data set. So for the initial test runs, the training dataset was having the images from CSA and Lafarge sites, which is referred to as “Original Training Set”. As can be seen from the table, the classifier performance for bedrock and sand is not satisfactory and quite low compared to the results obtained during the algorithm development and verification stage. This could be attributed to several reasons such as; different weather conditions, some practical implementation restrictions and the mismatch between the training libraries used during the field test and development stage. For the weather conditions, when the images were collected from the CSA mars yard for developing the training library, the weather beautiful and sunny. Whereas, before the field tests took place, it was raining continuously for two days. Hence, the texture for the same terrain, in the training library and on field, was visibly different from each other to a great extent. Also, the lighting conditions (which are quite different in sunny and cloudy weather) may have affected the performance. Moreover, for the practical implementation, it was required to compress the images before communicating them from the rover to the system and this obviously degrades the texture information of any image adversely. Lastly, there were new diverse terrains added in the mars yard by CSA, which were not present while collecting the images for training set.

5.2.1 Queue data structure for Exteroceptive Training Library

Further, to show case that if the training library has the examples from the same site where the rover is traversing, it can help improve the performance to a great extent, all examples of the sandy terrain were removed from the training library and plentiful sandy terrain images, taken on that very day by the rover, were added to the library and this library was given a name "New_CSA Dataset". As can be seen from the table, the performance for the sand improved dramatically, with the success ratio of 94%. This proves that if the classifier is continuously being trained with the new images from the current site, the performance can be enhanced a lot. This leads to a conclusion that if a queue like structure is implemented for the exteroceptive training library as well (similar to proprioceptive training library), then the rover will be able to adapt to the new planetary visual data, no matter how different it is from the on-earth training.

Moreover, it can be seen that the performance for the bedrock has degraded to a great extent. This owes majorly to the compression of the images. As discussed previously, the bedrock and sand images from the CSA mars yard are visually quite similar to each other and hence, intensity thresholding is implemented to differentiate between them. However, adding the sand images from the current site, compression of the images and the poor weather conditions, caused more confusion between both of them. Moreover, adding new images to the existing training library must have changed the effective intensity threshold, but having a hard coded threshold value didn't allow this change to reflect in the system. Thus, it seems that lots of bedrock images were confused as sand images. Hence, following the same example of sandy terrain explained in previous paragraph, more bedrock images from the current site were added to the training set and an improvement in the performance was observed, which can be seen in the table under the name of "Pure_CSA datset". Thus, the final field test results for exteroceptive were comparable to what was achieved in the in-lab analysis with a few well-explained exceptions.

5.3 Summary

This chapter was a discussion of the field test demo, which was held at CSA Mars yard to test and verify all the algorithms developed in this work. A 'Sled Test' (representing a sudden change in trafficability performance), demonstrates the responsiveness and online learning ability of the system. Moreover, the classification performance for the terrain classification was also analyzed here. All in all, the proposed risk metric (i.e. proportion of data points over 20% slip) in combination with the queue based data structure, could achieve stable predictions in consistent terrains, while also being responsive to sudden changes in terrain trafficability.

Chapter 6. Conclusion

The work presented here is a part of a larger program to develop an autonomous system for Mars rovers to make them intelligent enough to predict the trafficability properties of a distant terrain, just by knowing its appearance (image) and geometry (depth map) information. This work focuses mainly on the exteroceptive classification as well as on the correlation of proprioceptive and exteroceptive data in such a framework that enables the online learning.

On the exteroceptive classification front, the aim is to be able to classify the terrains correctly as one of three types viz. Sand, Bedrock and Gravel and the primary objective here is to achieve the satisfactory performance using the least complex method (due to the tight computational budgets for planetary rovers [78]). As discussed in section 2.4, the gradient based methods and Gabor filter based methods represent the opposite ends of the spectrum, in terms of computational complexity. Hence, this work demonstrates the implementation along with tuning of design parameters and the classification performance for three different visual descriptors namely, HOG (gradient based), GIST and Texton (both Gabor based). It shows that for monochrome images of planetary terrains, a good description of terrain texture can be achieved by quantifying edge detections at various scales and orientations. In addition to that it can also be observed that the classification between two confusing terrain types (gravel and bedrock in case of NASA JPL dataset, whereas sand and bedrock in case of CSA dataset) can be further enhanced by combining the texture based classification results with the intensity thresholding results. For example, the simplest descriptor HOG (along with Intensity Thresholding) achieves the accuracy of 81% for NASA JPL dataset (more diverse dataset) and 94% for CSA dataset (dataset with low diversity), for 3-way classification. Similarly, GIST achieves 78% and 91% as well as Textons achieve 84% and 97% accuracy, for the respective datasets.

One of the most important conclusions, derived from these experiments and results, is that the increased complexity of GIST and Textons does not seem to be giving significant performance gain over the simple HOG descriptor. The algorithm of the HOG descriptors is computationally least expensive (over 3 orders of magnitude difference in complexity between HOG and Textons) as well as its runtime is the fastest, among all the options explored in this work. As discussed earlier, for planetary rovers, runtime consideration is a very important factor as they usually have tight computational budgets. Hence, the ability to use simpler algorithms with little to no loss in performance is an important achievement for this specific application. Such constraints over the runtime and the algorithmic complexity, is also part of the reasons why higher order methods like CNN (Computational Neural Networks) methods have not been implemented for this work. A more detailed analysis of runtime savings can be a part of future work.

Another important observation here is the enhancement of the classification performance with the use of intensity thresholding. As seen in the sections 3.3.1.1 and 3.3.2.1, two different values of intensity threshold have been used for two different datasets. This leads to the fact that the threshold value depends on the training library of the corresponding dataset. If the training library is being updated for the concerned terrain type, then the threshold value should also be changed to reflect the addition or removal of images from training library. Similarly, using the hard coded threshold value will lead to the false thresholding and might degrade the performance instead of enhancing. This fact can also be verified from the exteroceptive field test results illustrated in section 5.2. As discussed in that section, new_CSA dataset has the sand images taken from on-site and Pure_CSA dataset has the sand as well as bedrock images taken from on-site. Despite adding new bedrock images from the new test site, the performance for the same is not satisfactory (the classification performance for bedrock degraded from 86% to around 48%) and one of the reasons might be the hard coded intensity threshold value. In the original dataset, the bedrock images present in the library were taken in sunny weather during various test campaigns. The bedrocks in this dataset were far brighter than the ones which were being added in the Pure_CSA dataset, because the images being added into the Pure_CSA dataset were taken in cloudy and rainy weather. Due to this difference of brightness of bedrock images, the threshold for both the datasets must be different, and hence if ever the training dataset is updated, it is a must to update the threshold value as well.

Another very important conclusion derived from the field test results is that implementing queue like data structure for the exteroceptive training library may improve the ability of the system to adapt to the changed appearance of the same terrain type. This will ensure that even if the on-earth terrains looks different from actual planetary terrains, the rover will be able to overcome the on-earth training and will train itself with the new planetary images.

On the other hand, regarding the correlation of proprioceptive and exteroceptive data, a learning framework has been designed and implemented in this work. It correlates the trafficability metrics

(Slip, Sinkage and Roughness) with the appropriate visual data (Terrain type and Slope) in a queue like data structured model; while the rover is traversing. It also updates the trafficability prediction model online according to the latest correlated data. In the field test results for the ‘Sled Test’ (Section 5.1), it was observed that adding the latest high slip measurements to the already existing correlated data model did change the trafficability prediction model online. While traversing the terrain, the rover was able to learn the new trafficability properties and the prediction for the distant terrains also updates immediately according to the new trafficability model. This demonstrates the successful development of a ‘Data Driven Model’, where queue structure enables the rover to adapt to the changed terrain behaviors.

Apart from that, importance of the queue size is discussed in section 4.3 and its importance in maintaining balance between the responsiveness and stability of the system has also been exhibited in the field test results. It was observed that, a small value of queue size (20) leads the system to be quickly responsive to the changing terrain behaviors and with an adequate value of queue size, the system can be both; responsive and stable at the same time. This leads to the conclusion that queue size should be selected responsibly and tweaking its value provides the freedom to tune the nature of the system (responsive vs. stable).

Future Work

In this work, very simple classification techniques have been explored such as nearest and k-nearest neighbors. This owes to the algorithmic complexity constraint. However, SVM is another simple classification technique that could have improved the performance further as compared to the explored ones. Thus one of the potential future work can be the implementation of multi class SVM, involving the tuning of several designing parameters. Another interesting future work can be a separate experimentation campaign, including cross-validation, to examine the effects of different values of K in the K-NN algorithm, which is selected to be 25 (chosen arbitrarily) at present. Also, as mentioned earlier, a more detailed analysis of runtime saving can be a part of future work.

Regarding the development of learning framework, the High Slip Detection (HSD) scenarios have been analyzed and a solution, to differentiate between false and true positives of such detection as well as to take care of both the cases, has been proposed in this work. However, due to lack of time, this solution could not be tested during the field tests. Hence, it is an interesting and well-needed future work to test this solution on-field to confirm whether it properly detects the high slip scenarios, while avoiding false positives.

Apart from that, this work can be used in further research work related to the path planning for planetary rovers. Moreover, this work explored some of the simplest techniques and the results achieved by them was analyzed. Another approach for looking at the same problem might be the exploration of some advanced techniques like CNN (Convolutional Neural Networks) and their performance. Analyzing the trade-off between the performance and complexity of these two different approaches for the same problem, can be a very interesting future work.

References

1. Arvidson, R. E., et al. "Spirit Mars Rover Mission: Overview and selected results from the northern Home Plate Winter Haven to the side of Scamander crater." *Journal of Geophysical Research: Planets (1991–2012)* 115.E7 (2010).

2. Sullivan, R.J., Anderson, R., Biesiadecki, J., Bond, T., Stewart, H., 2011. Cohesions, friction angles, and other physical properties of Martian regolith from Mars Exploration Rover wheel trenches and wheel scuffs. J Geophys Res 116 (E2), 1991–2012. <http://dx.doi.org/10.1029/2010JE003625>.

3. C. Brooks and K. Iagnemma, "Self-supervised Terrain Classification for Planetary Surface Exploration Rovers", Journal of Field Robotics, vol. 29(3):445-468, 2012.

4. I. Gheorghe , W. Li, T. Popham, A. Gaszczak, K.J. Burnham, "Key Learning Features as Means for Terrain Classification", Chapter in Advances in Systems Science, pp 273-282, 2013.

5. M.R. Blas, M. Agrawal, A. Sundaresan, K. Konolige, "Fast color/texture segmentation for outdoor robots," in *IROS 2008,* pp.4078-4085, 22-26 Sept. 2008

6. T. Leung and J. Malik, "Representing and Recognizing the Visual Appearance of Materials using Three-Dimensional Textons," Int. Journal of Computer Vision, vol. 43, no. 1, pp. 29-44, 2001.

7. A. Murillo, G. Singh, J. Kosecka, and J. Guerrero, "Localization in Urban Environments using a Panoramic GIST Descriptor," IEEE Transactions on Robotics, vol. 29, no. 1, pp. 146-160, 2013.

8. A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," International Journal of Computer Vision, vol. 42, no. 3, pp. 145-175, 2001.

9. R. Gonzalez, A. Rituerto, J.J. Guerrero, "Vision-based Terrain Classification for Off-Road Mobile Robots Combining Downward-Looking and Frontal Cameras", Journal of Robotics. [Under review, 2015].

10. K. Walas, "Terrain Classification and Negotiation with a Walking Robot" Journal of Intelligent & Robotic Systems, Volume 78, Issue 3, pp 401-423, 2015.

11. Manduchi, Roberto, et al. "Obstacle detection and terrain classification for autonomous off-road navigation." *Autonomous robots* 18.1 (2005): 81-102.

12. Angelova, Anelia, et al. "Learning and prediction of slip from visual information." *Journal of Field Robotics* 24.3 (2007): 205 231.

13. Karlsen, Robert E., and Gary Witus. "Terrain understanding for robot navigation." *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007.

14. Weiss, Christian, Hashem Tamimi, and Andreas Zell. "A combination of vision-and vibration-based terrain classification." *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008.

15. Bajracharya, Max, et al. "Autonomous off□ road navigation with end□ to□ end learning for the LAGR program." *Journal of Field Robotics* 26.1 (2009): 3-25.

16. Helmick, Daniel, Anelia Angelova, and Larry Matthies. "Terrain adaptive navigation for planetary rovers." *Journal of Field Robotics* 26.4 (2009): 391-410.

17. Bellutta, Paul, et al. "Terrain perception for DEMO III." *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*. IEEE, 2000.

18. Ono, Masahiro, et al. "Risk-aware planetary rover operation: Autonomous terrain classification and path planning." *2015 IEEE Aerospace Conference*. IEEE, 2015.

19. Halatci, Ibrahim, Christopher A. Brooks, and Karl Iagnemma. "Terrain classification and classifier fusion for planetary exploration rovers." *2007 IEEE Aerospace Conference*. IEEE, 2007.

20. Otsu, Kyohei, et al. "Autonomous Terrain Classification With Co-and Self-Training Approach." *IEEE Robotics and Automation Letters* 1.2 (2016): 814-819.

21. Fujita, Koki, and Naoyuki Ichimura. "Classifying terrain properties for planetary exploration rovers based on a combined distance measure with dynamic texture model." *Robotics and Autonomous Systems* 70 (2015): 156-165.

22. S. Wang, S. Kodagoda, Z. Wang, G. Dissanayake, "Multiple sensor based terrain classification", Australasian Conference on Robotics and Automation, 2011.

23. C. Brooks and K. Iagnemma, "Vibration-based terrain classification for planetary exploration rovers," in IEEE Transactions on Robotics, vol.21, no.6, pp.1185-1191, Dec. 2005.

24. E.J. Coyle, "Fundamentals and Methods of Terrain Classification Using Proprioceptive Sensors", PhD thesis, The Floridad State University, 2010.

25. C. Weiss, N. Fechner, M. Stark, A. Zell, "Comparison of Different Approaches to Vibration-based Terrain Classification", 2007.

26. P. Giguere and G. Dudek, "Clustering Sensor Data for Autonomous Terrain Identification using Time-Dependency", Autonomous Robots, 2009.

27. K.A. Mahadhir, S.C. Tan, C.Y. Low, R. Dumitrescu, A.T.M. Amin, A. Jaffar, "Terrain Classification for Track-driven Agricultural Robots", Procedia Technology, Volume 15, 2014, Pages 775-782.

28. M. Reinstein, V. Kubelka, K. Zimmermann, "Terrain adaptive odometry for mobile skid-steer robots," in ICRA, pp.4706-4711, 6-10 May 2013.

29. J. Libby, A.J. Stentz, "Using sound to classify vehicle-terrain interactions in outdoor environments", *ICRA 2012*, May, 2012.

30. C.C. Ward, K. Iagnemma, "Speed-independent vibration-based terrain classification for passenger vehicles," Vehicle System Dynamics, 47(9):1095-1113, 2009.

31. E.G. Collins, E.J. Coyle, "Vibration-based terrain classification using surface profile input frequency responses," in *ICRA 2008.* pp. 3276-3283, 19-23 May 2008.

32. C. Weiss, H. Frohlich, A. Zell, "Vibration-based Terrain Classification Using Support Vector Machines," in IROS, pp.4429-4434, 9-15 Oct. 2006.

33. W. Mou, A. Kleiner, "Online learning terrain classification for adaptive velocity control," in IEEE International Workshop on Safety Security and Rescue Robotics, pp.1-7, 2010.

34. Halatci, C. Brooks, K. Iagnemma, "A study of visual and tactile terrain classification and classifier fusion for planetary exploration rovers", Journal of Robotics, vol. 26(6):767-779, 2008.

35. Lowe, David G. "Object recognition from local scale-invariant features."*Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Vol. 2. Ieee, 1999.

36. Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05).* Vol. 1. IEEE, 2005.

37. Bay, Herbert, et al. "Speeded-up robust features (SURF)." *Computer vision and image understanding* 110.3 (2008): 346- 359.

38. Gonzalez, Rafael C., and Richard E. Woods. "Digital image processing."*Nueva Jersey* (2008).

39. Prince, Simon JD. *Computer vision: models, learning, and inference*. Cambridge University Press, 2012.

40. Lazebnik, Svetlana, Cordelia Schmid, and Jean Ponce. "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories."*2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06).* Vol. 2. IEEE, 2006.

41. S. Laible, Y.N. Khan, A. Zell, "Terrain classification with conditional random fields on fused 3D LIDAR and camera data," in European Conference on Mobile Robots, pp.172-177, 25-27 Sept. 2013.

42. M. Haselich, M. Arends, N. Wojke, F. Neuhaus, D. Paulus, "Probabilistic terrain classification in unstructured environments", Robotics and Autonomous Systems 61 (10), 1051-1059, 2013.

43. L. Ojeda, J. Borenstein, G. Witus, and R. Karlsen, "Terrain Characterization and Classification with a Mobile Robot," Journal of Field Robotics, vol. 23, no. 2, pp. 103-122, 2006.

44. P. Giguere and G. Dudek, "Clustering Sensor Data for Autonomous Terrain Identification using Time-Dependency", Autonomous Robots, 2009.

45. M. Reinstein, V. Kubelka, K. Zimmermann, “Terrain adaptive odometry for mobile skid-steer robots,” in ICRA, pp.4706-4711, 6-10 May 2013.

46. Skonieczny, K., Wettergreen, D., and Whittaker, W. R. (2016). Advantages of continuous excavation in lightweight planetary robotic operations. The International Journal of Robotics Research, 35(9):1121-1139.

47. Freitag, D. R., Green, A. J., and Melzer, K.-J. (1970). Performance evaluation of wheels for lunar vehicles. Technical Report AEWES-TR-M-70-2, ARMY ENGINEER WATERWAYS EXPERIMENT STATION VICKSBURG MS.

48. Wong, J. and Asnani, V. (2008). Study of the correlation between the performances of lunar vehicle wheels predicted by the nepean wheeled vehicle performance model and test data. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 222(11):1939-1954.

49. Wettergreen, D., Moreland, S., Skonieczny, K., Jonak, D., Kohanbash, D., and Teza, J. (2010). Design and field experimentation of a prototype lunar prospector. The International Journal of Robotics Research, 29(12):1550-1564.

50. Shukla, D. K. and Skonieczny, K. (2017). Simple texture descriptors for classifying monochrome planetary rover terrains. In Robotics and Automation (ICRA), 2017 IEEE International Conference on, pages 5495-5502. IEEE.

51. Hwang A., Higgins E. and Pomplun M., “A model of top-down attentional control during visual search in complex scenes”. In Journal of Vision (May 2009), Vol.9, 25. doi:10.1167/9.5.25

52. O. Fialka, M. Cadik, “FFT and Convolution Performance in Image Filtering and GPU”. Tenth International Conference on Information Visualization (July 2006), doi: 10.1109/IV.2006.53

53. R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, New York:Wiley, 2001.

54. N.H. Rajini, R. Bhavani, "Classification of MRI brain images using k-nearest neighbor and artificial neural network", *2011 International Conference on Recent Trends in Information Technology (ICRTIT)*, pp. 563-568, 2011.

55. A. Krizhevsky, I. Sutskever, G.E. Hinton, F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger, "Imagenet classification with deep convolutional neural networks", *Advances in Neural Information Processing Systems 25*, Curran Associates, Inc., pp. 1097-1105, 2012.

56. M. Zeiler and R. Fergus. “Visualizing and Understanding Convolutional Networks”, 2013.

57. K. Simonyan and A. Zisserman. “Very Deep Convolutional Networks For Large-Scale Image Recognition”, ICLR 2015.

58. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, "Going deeper with convolutions", Computer Vision and Pattern Recognition (CVPR), 2015.

59. K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition", CVPR, pp. 770-778, 2016

60. Espinal, F., Huntsberger, T.L., Jawerth, B., and Kubota T. (1998). "Wavelet-based fractal signature analysis for automatic target recognition," Optical Engineering, Special Section on Advances in Pattern Recognition, 37(1), 166-174.

61. Bekker, M. G. (1956). Theory of land locomotion. University of Michigan.

62. Wong, J. Y., & Reece, A. R. (1967). "Prediction of rigid wheel performance based on the analysis of soil-wheel stresses part I. Performance of driven rigid wheels". Journal of Terramechanics, 4(1), 81-98.

63. Wong, J. Y. (1967). Behaviour of soil beneath rigid wheels. Journal of Agricultural Engineering Research, 12(4), 257-269.

64. Meirion-Griffith, G., & Spenko, M. (2011). A modified pressure?sinkage model for small, rigid wheels on deformable terrains. Journal of Terramechanics, 48(2), 149-155.

65. Ding, L., Deng, Z., Gao, H., Tao, J., Iagnemma, K. D., & Liu, G. (2014). Interaction Mechanics Model for Rigid Driving Wheels of Planetary Rovers Moving on Sandy Terrain with Consideration of Multiple Physical Effects. Journal of Field Robotics. In press.

66. Irani, RA et al (2011). A dynamic terramechanic model ? J Terramechanics, 48(4), 307-318.

67. Senatore, C., & Iagnemma, K. (2014). Analysis of stress distributions under lightweight wheeled vehicles. Journal of Terramechanics, 51, 1-17.

68. Ho, K., Peynot, T., and Sukkarieh, S. (2016). "Nonparametric traversability estimation in partially occluded and deformable terrain". Journal of Field Robotics, 33(8):1131-1158.

69. Karumanchi, S., Allen, T., Bailey, T., and Scheding, S. (2010). "Non-parametric learning to aid path planning over slopes". The International Journal of Robotics Research, 29(8):997-1018.

70. Peynot, T., Lui, S.-T., McAllister, R., Fitch, R., and Sukkarieh, S. (2014). Learned stochastic mobility prediction for planning with control uncertainty on unstructured terrain. Journal of Field Robotics, 31(6):969-995.

71. Rothrock, B., Papon, J., Kennedy, R., Ono, M., Heverly, M., and Cunningham, C. (2016). Spoc: Deep learning-based terrain classification for mars rover missions. In AIAA SPACE 2016, page 5539.

72. Heverly, M., Matthews, J., Lin, J., Fuller, D., Maimone, M., Biesiadecki, J., and Leichty, J. (2013). Traverse performance characterization for the mars science laboratory rover. Journal of Field Robotics, 30(6):835-846.

73. Ishigami, G., Nagatani, K., and Yoshida, K. (2011). “Path planning and evaluation for planetary rovers based on dynamic mobility index”. In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pages 601-606. IEEE.

74. Krebs, A., Pradalier, C., and Siegwart, R. (2010). Adaptive rover behavior based on online empirical evaluation: Rover-terrain interaction and near-to-far learning. Journal of Field Robotics, 27(2):158-180.

75. Angelova, A., Matthies, L., Helmick, D., and Perona, P. (2007b). “Learning slip behavior using automatic mechanical supervision. In Robotics and Automation”, 2007 IEEE International Conference on, pages 1741-1748. IEEE.

76. Arvidson, Raymond E., et al. "Opportunity Mars Rover mission: Overview and selected results from Purgatory ripple to traverses to Endeavour crater."Journal of Geophysical Research: Planets (1991–2012) 116.E7 (2011).

77. Arvidson, R. E., Iagnemma, K. D., Maimone, M., Fraeman, A. A., Zhou, F., Heverly, M. C., Bellutta, P., Rubin, D., Stein, N. T., Grotzinger, J. P., et al. (2017). Mars science laboratory curiosity rover megaripple crossings up to sol 710 in gale crater. Journal of Field Robotics, 34(3):495-518.

78. M. Bajracharya, M.W.Maimone, D. Helmick. Autonomy for Mars Rovers: Past, Present, and Future. Computer, vol. 41, Dec. 2008, pp.44 - 50.

79. NASA Jet Propulsion Laboratory, 'Mars Exploration Rover Mission: Multimedia: All Raw Images: Opportunity', 2014-2017. [Online]. Available: https://mars.nasa.gov/mer/gallery/all/opportunity.html [Accessed: 01- May- 2016].

80. NASA Jet Propulsion Laboratory, 'Mars Exploration Rover Mission: Multimedia: All Raw Images: Spirit', 2014-2017. [Online]. Available: https://mars.nasa.gov/mer/gallery/all/spirit.html [Accessed: 01- May- 2016].

Appendix

As discussed in section 4.2**,** a high slip mode operation is used to handle unusual events like turning or noisy output, where the slip value for the given slope increases drastically, relative to the normal operation. Due to lack of time, this operation could not be verified in the field test demo. The details about the implementation of the same has been explained below.

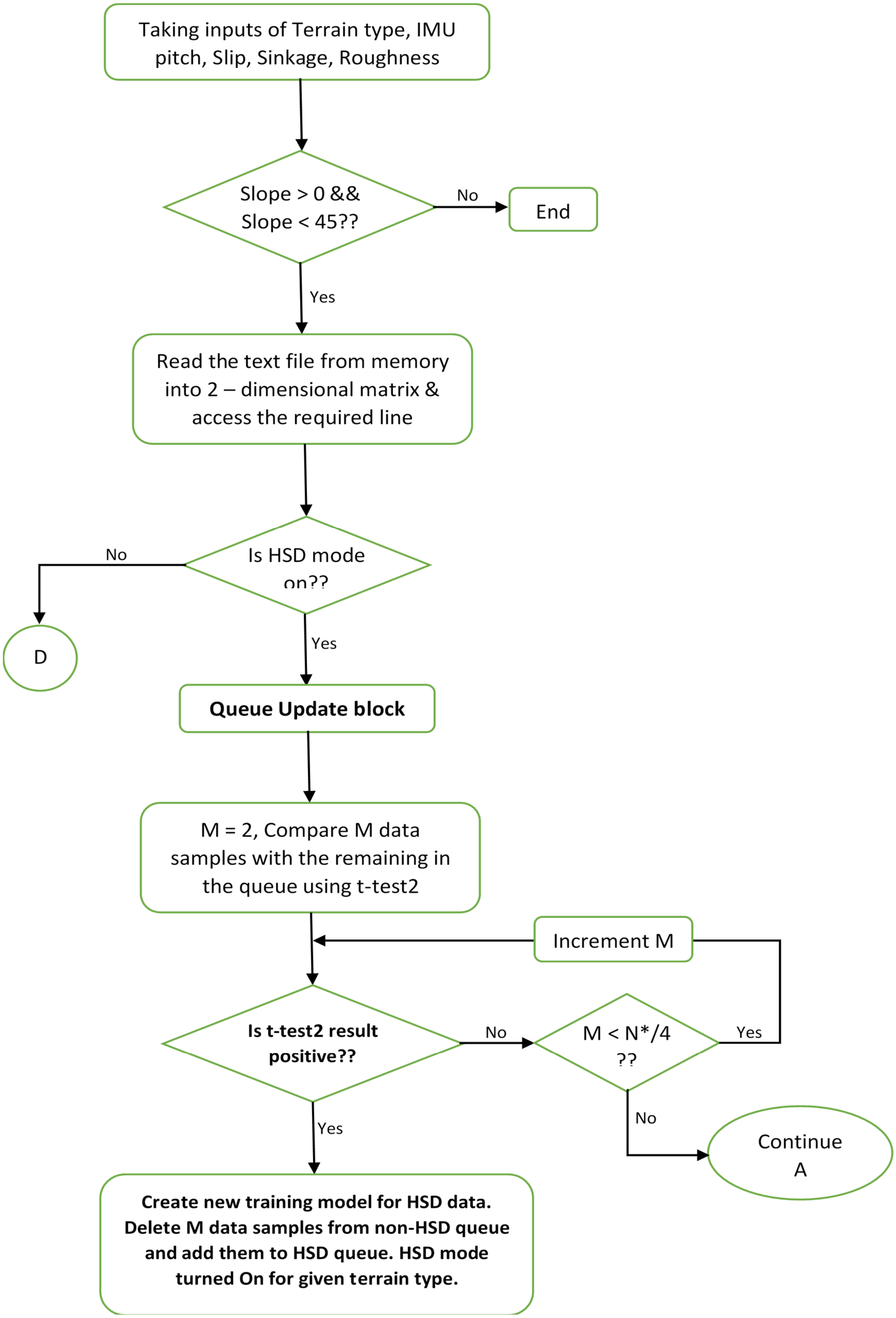
High Slip Detection

To check for the occurrence of these events a t-test2 is performed at every training iteration, wherein the latest values in the queue of the queried slope is compared with the older values, to decide if they are showing a different kind of distribution or not. Using a t-test2 ensures that the HSD mode is only detected, when the significantly high slip values are recorded consecutively in the training model.

The flowchart for detection and implementation of this mode is shown in Figure 8-1. As shown, in between the initial steps of the training phase (shown in the flowchart in Figure 4-14), an additional check is performed for high slip detection, after reading the training model in the 2-D matrix. If the system is running in HSD mode for given terrain type, due to previously detected high slip events, HSD flag for that terrain type would be equal to 1 and otherwise 0. If the flag is 0, the queue update block is executed first and then, t-test2 comparisons are performed on the queue of given slope value and terrain type. Let's say given queue has N* data samples, then t-test2 runs will be performed between latest M and the remaining older data samples, where M changes from 2 to N*/4, iteratively.

As soon as t-test2 result is positive, the loop ends and the HSD flag for given terrain is turned on to denote that the rover is experiencing relatively higher slip measurements on the same terrain. Also, the M high slip data samples are removed from the regular training model and a separate HSD training model is created, where these M data samples are added in the corresponding queue. At the same time, HSD trafficability model is also updated with the average value using M data samples. On the contrary, if t-test2 result is never positive till M = N*/4, it denotes that no high slip events are detected and in that case, normal operations continue after the "Continue A" point in the flowcharts discussed before.

Now, let's say, in the next training iteration, the system is already in the HSD mode for given terrain type (i.e. HSD flag = 1), then the new data sample will be added to the HSD training model and also, the HSD trafficability model will be updated using updated queue data. Further, to determine whether the slip measurements are returned back to the normal value, again t-test2 run is performed between the HSD and non-HSD queue data samples. If the result is positive, the HSD mode will continue for given terrain and if not, HSD flag will be turned off and the HSD models (training & trafficability models) for a given terrain type are reinitialized.


Taking inputs of Terrain type, IMU pitch, Slip, Sinkage, Roughness
Slope > 0 && Slope < 45??
No
End
Yes
Read the text file from memory into 2 – dimensional matrix & access the required line
No
Is HSD mode on??
D
Yes
Queue Update block
M = 2, Compare M data samples with the remaining in the queue using t-test2
Increment M
Is t-test2 result positive??
No
M < N*/4 ??
Yes
No
Continue A
Yes
Create new training model for HSD data. Delete M data samples from non-HSD queue and add them to HSD queue. HSD mode turned On for given terrain type.

Algorithm for HSD