A Near-To-Far Learning Framework for Terrain Characterization Using an Aerial / Ground-Vehicle Team

Ashkan Hajjam
University of Denver

Follow this and additional works at: https://digitalcommons.du.edu/etd
Part of the Computer Sciences Commons

Recommended Citation
https://digitalcommons.du.edu/etd/1202

This Thesis is brought to you for free and open access by the Graduate Studies at Digital Commons @ DU. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Digital Commons @ DU. For more information, please contact jennifer.cox@du.edu.
A NEAR-TO-FAR LEARNING FRAMEWORK FOR TERRAIN
CHARACTERIZATION USING AN AERIAL / GROUND-VEHICLE TEAM

A Thesis
Presented to
the Faculty of the Daniel Felix Ritchie School of Engineering and Computer Science
University of Denver

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
Ashkan Hajjam
August 2016
Advisor: Matthew J. Rutherford, Ph.D.
Abstract

In this thesis, a novel framework for adaptive terrain characterization of untraversed far terrain in a natural outdoor setting is presented. The system learns the association between visual appearance of different terrain and the proprioceptive characteristics of that terrain in a self-supervised framework. The proprioceptive characteristics of the terrain are acquired by inertial sensors recording measurements of one second traversals that are mapped into the frequency domain and later through a clustering technique classified into discrete proprioceptive classes. Later, these labels are used as training inputs to the adaptive visual classifier. The visual classifier uses images captured by an aerial vehicle scouting ahead of the ground vehicle and extracts local and global descriptors from image patches. An incremental SVM is utilized on the set of images and training sets as they are grabbed sequentially. The framework proposed in this thesis has been experimentally validated in an outdoor environment. We compare the results of the adaptive approach with the offline a priori classification approach and yield an average 12% increase in accuracy results on outdoor settings. The adaptive classifier gradually learns the association between characteristics and visual features of new terrain interactions and modifies the decision boundaries.
Acknowledgements

First I would like to sincerely thank my advisor, Prof. Matthew Rutherford, for all his help and support throughout my graduate studies. Matthew is extremely committed to the professional and academic development of his students and builds a strong bond with his students. He always emphasized the importance of finding a research topic which is in my interest and then tackling it with passion in a systematic way. I am sure learning his approach of tackling hard and big problems in a systematic way along with creativity will be very valuable to me later on in my career. Finally, he helped me a lot in improving my presentation and writing skills, for which I will always be grateful. Also I would like to thank Prof. Kimon Valavanis, who has incredibly been supportive and whose comments and advice have been very helpful to me. With his careful guidance throughout my degree I learned the standard scientific way of doing research. He has always raised the bar for my research which I am grateful for. Additionally, I would like to sincerely thank Dr. Nikolaos Vitzilaios whom without his dedication and cooperation to tackle challenges along the path, this research would not have been possible. Nikos has always been very kind and approachable and I certainly benefited a lot from his guidance and help. Also, I would like to thank Dr. Mahoor for his Pattern Recognition and Computer Vision courses which both are the laying blocks of this project. I am absolutely grateful to all of the people at DU2SRI who have worked with me during the past two years: Steve Conyers, Konstantinos Kanistras, Jessica Alvarenga-Veyna, Saka Pranith. I have learned a lot from each of them on how to perform quality research.

Finally, I would like to thank my parents Mahboob and Sohrab and my brother Arash who have always shown unwavering and unconditional support for my every endeavor, and in every struggle of my life.
List of Figures

1.1 Terrain characterization for path finding applications on a grid map. Both the field of view of the aerial vehicle and the ground vehicle are shown with a shadow. The terrain characteristics are only completely known once the ground robot traverses them ............................................. 3
1.2 Near-to-far learning framework. The IMU measurements on the ground vehicle are used to label images captured from the aerial vehicle. This introduces a near-to-far learning framework. The image on the aerial vehicles are far while the IMU measurements are near. .................. 5
3.1 An example where k-means yields an invalid clustering ..................... 28
3.2 K-fold cross validation ....................................................... 32
3.3 Example of Overfitting ....................................................... 33
4.1 RTI metric. Each one-second traversal signal belongs to one of these bins . 36
4.2 Ground truth vibration characteristics of different terrain. .................. 37
4.3 IMU placed on the body of the vehicle .................................. 38
4.4 Ground robot with the attached coordinate frame. The x axes faces in the same direction of the robot. The z axes is perpendicular to the ground .... 39
4.5 Tire tread pattern ............................................................. 41
4.6 Wheel of Ground robot on different terrain ................................ 42
4.7 Suspension System of the ground Vehicle. From [1]. ..................... 43
4.8 XMOS microcontroller ....................................................... 44
4.9 The commercial IMU. We utilize the accelerometer and gyroscope and use the magnetometer only during the first initialization. .................... 45
4.10 Acceleration on Grass and Asphalt terrain in the time domain. Acceleration on asphalt has smaller and more frequent positive and negative peaks whereas grass terrain has peaks that are spread and larger amplitudes .... 49
4.11 Amplitude of positive peaks from 1-second vibration signals on a sequence of grass terrain ...................................................... 50
4.12 Manual feature extraction distribution from 1-second vibration signals on Asphalt, Dirt, Grass, Woodchips terrain .................................. 51
4.13 Periodogram representation of a sample asphalt and grass terrain ....... 53
4.14 Flow chart of processing events before categorizing 1-second vibration signals. ................................................................. 55
4.15 Ground truth Characteristics of different terrain traversed in a sequence of 5 minutes. Arrows indicate sections of the traversed terrain in which characteristics are different from the norm ........................................ 56
5.1 DJI Phantom used as the AV with a GoPro camera attached to a gimbal ........................................ 58
5.2 Overhead view captured from aerial vehicle with the squares on top of the ground vehicle for scale references ........................................ 59
5.3 Camera Gimbal that is attached to the DJI phantom ........................................ 61
5.4 SIFT features and RGB Histogram of different image patches .................. 67
5.5 Figure showing gradient orientation histogram descriptors (from [2]) .......... 68
5.6 Texton representation of texture ........................................ 72
5.7 Bag Of Visual Words model. From [3] ........................................ 73
5.8 A basic Local Binary Pattern operator ........................................ 73
5.9 LBP with circular $(8, 1), (16, 2)$, and $(8, 2)$ neighborhoods. .................. 74
5.10 A Grey Level Co-occurrence Matrix of a sample image ......................... 74
5.11 Accuracy results of images captured from 3-6 meters (Group 3) ............... 81
5.12 Accuracy results of images captured from 6-8 meters (Group 2) ............... 82
5.13 Accuracy results of images captured from 6-8 meters (Group 1) ............... 82
6.1 Self Supervised characterization module ........................................ 84
6.2 Examples that resonate the necessity for a self-learning classifier ............... 87
6.3 Self Supervised Framework algorithm ........................................ 88
6.4 Sliding window for terrain characteristics. This allows an unlearning mechanism for previous characteristics association ........................................ 90
6.5 An instance were different lighting conditions lead to different classification results of the terrain ........................................ 92
6.6 Cumulative misclassifications of the Supervised classifier vs the ensemble of Self supervised classifier and offline supervised classifier. ............... 95
6.7 Characterization results of the online supervised adaptive method vs the offline supervised method ........................................ 96
7.1 Failed characterization due to mixed terrain in image patch ....................... 99
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Ground truth characteristics of different terrain.</td>
<td>21</td>
</tr>
<tr>
<td>5.1</td>
<td>SVM classification results using HSV colorspace as features for group 3</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>images</td>
<td></td>
</tr>
<tr>
<td>5.2</td>
<td>SVM classifier results using LBP features on group 3 images</td>
<td>79</td>
</tr>
<tr>
<td>5.3</td>
<td>SVM classifier results using GLCM features on group 3 images</td>
<td>80</td>
</tr>
<tr>
<td>5.4</td>
<td>SVM Classification results using SIFTBOVW on group 3 images</td>
<td>80</td>
</tr>
<tr>
<td>5.5</td>
<td>SVM Classification results using SURFBOVW on group 3 images</td>
<td>81</td>
</tr>
<tr>
<td>6.1</td>
<td>Cumulative Error of both classifiers on different sequencing of terrain on</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>a set of 220 image patches</td>
<td></td>
</tr>
<tr>
<td>6.2</td>
<td>Number of misclassifications occurred during traversal of the terrain type</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>using different window sizes</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

The topic of this thesis is characterization and classification of remote terrain. There are various applications that can utilize this approach, for example, in the field of path planning. Terrain can be classified into traversable, non-traversable, hazardous or hazard-free terrain. Assuming that a terrain is traversable and hazard-free, the “characteristics” of “remote” terrain are still of importance to robot applications, something that has not been discussed in literature. The characteristics of a terrain is measured based on the behavior the terrain will induce on a robot while it interacts with that specific terrain. This chapter defines the problem statement, after giving a motivation and background of current terrain characterization / classification approaches. We present a brief terminology for understanding the framework of our proposed method and an approach overview along with a summary of results and discuss the contributions of this work.

1.1 Motivation and Problem Statement

In autonomous long-range navigation, a ground robot must move from point A to point B in a way that best achieves mission priorities such as traversing the shortest path or
traversal of the safest path. In some instances, a path must be selected so that the vehicle’s interaction with any terrain upon traversal meets a certain criteria. For example, if there is a limit on the amount of electrical current available to the motors of a robot, we would want to avoid terrain that requires high torque and hence high currents on the motors. Different responses of a robot on a given terrain depend not only on the physical properties of the terrain but also on the physical specifications of the robot. Therefore, a heavier, bigger robot responds differently in comparison to a lightweight, smaller robot while interacting with rough terrain. This implies that characteristics of terrain are relatively defined according to the specifications of any given robot and fully determined whenever the robot traverses the terrain. This is a constraint, since we are interested in having knowledge of the environment with the capability to predict the characteristics of terrain for a specific robot before traversal. Recent literature shows that by learning the association between exteroceptive and proprioceptive sensor information, the behavior of a vehicle on far terrain patches can be predicted [4]. This concept is known as near-to-far learning [5]. Therefore, a near-to-far learning approach eliminates the requirement for a ground vehicle to actually traverse all terrain patches for finding the characteristics. Recent research has utilized different combinations of sensors for the near-range and far-range sensors and different algorithms to learn their association. Recent research in the field has focused on improving performance in terms of accuracy and latency of the learning algorithms.

Vibrations induced on a robot upon different terrain traversals affect the longevity of the robot parts and also the reliability of the robot sensor measurements. Therefore, in this thesis the focus is on predicting vibration properties of far terrain. The first step is to measure vibration properties of any given terrain traversal. This is accomplished through accelerometer and gyroscope measurement readings at 200 Hz. After data acquisition, a processing step is required to determine a metric for characteristics of the terrain. This is
achieved by mapping the vibration measurements into the amplitude and frequency space. We later divide this space into subspaces to discretize vibration representation of terrain. This module is then used as our near-range sensor setting.

![Terrain Characterization Diagram](image)

Figure 1.1: Terrain characterization for path finding applications on a grid map. Both the field of view of the aerial vehicle and the ground vehicle are shown with a shadow. The terrain characteristics are only completely known once the ground robot traverses them.

For planning an optimal “long-range” path that meets mission priorities, we require “global” estimation of the characteristics on a grid-based map. On-board exteroceptive sensors such as cameras on the ground vehicle, prove to have a limited field of view which only provides local information as illustrated in Figure 1.1. This leads to the introduction of a collaborative aerial / ground robotic framework in which the aerial vehicle scouts farther terrain and relays visual information back to the ground robot for processing. In this framework, the exteroceptive sensor is the sensor attached to the aerial vehicle and plays the role of the far-range sensor in the near-to-far learning framework as illustrated in Figure 1.1. Due to cost, payload and power constraint factors, a monocular camera is the attractive sensor choice in comparison to LiDAR counterparts for the exteroceptive sensor.
on the aerial vehicle. Therefore, visual cues such as color and texture are used to associate with vibration measurements. Due to change in altitude and different poses of the aerial vehicle on different terrain, scale and rotation invariant descriptors are required to describe the terrain. Last but not least, due to inter-class variation in vibration properties of any given terrain, there is no strict rule for classifiers. Therefore, classifier boundaries need to be updated on the fly.

1.2 Approach Overview

As stated in Section 1.1, the purpose of this research is to predict terrain characteristics of remote terrain. Our proposal is to learn the association between visual appearance of terrain patches captured by an aerial vehicle and the terrain’s vibration properties measured through IMU sensors on a ground vehicle while traversing the terrain. One standard approach is to gather characteristics information from different terrain “classes” such as grass, asphalt, dirt, woodchips and etc. prior to operating in the field. Once gathered, this information could be used to determine the average response of the ground robot behavior on those type of terrain. Once the average and median characteristics of terrain classes is known prior to the experiments the system would simply have to train a classifier to visually detect these terrain classes, and thereby predict the likely (average) vibration characteristics. After seeing the shortcomings of this method an adaptive near-to-far learning method is introduced to learn the association between visual features and vibration characteristics of terrain on the fly. In this section an overview of our framework along with the terminology utilized throughout this thesis is presented. We also describe how each of the algorithmic components fit into the overall near-to-far self-supervised learning framework. A big picture overview of our framework is presented in Figure 1.2.
Figure 1.2: Near-to-far learning framework. The IMU measurements on the ground vehicle are used to label images captured from the aerial vehicle. This introduces a near-to-far learning framework. The image on the aerial vehicles are far while the IMU measurements are near.

**Terrain Patch Characteristics**

Each terrain patch is identified by the frame number and time it was detected from the aerial vehicle along with the height of the aerial vehicle. Also the relative position of the image patch to the ground vehicle in that given frame is also determined. As the ground vehicle travels along its path, the wheels come into contact with the terrain patch and a one-second IMU reading is also associated with the patch. In this thesis, our interest is in determining vibration properties of a terrain patch before traversal by the ground vehicle. Later when the ground robot traverses the terrain and the ground truth is known, we measure the accuracy of our predictions.
**Proprioceptive and Exteroceptive Sensors**

Proprioceptive sensors, such as accelerometers, sense terrain features, only upon physical interaction between the ground robot’s wheels and the terrain whereas exteroceptive sensors can sense features related to terrain not in contact with the vehicle. In this thesis, the exteroceptive sensor is a camera facing downward, attached to an aerial vehicle while the proprioceptive sensors are accelerometers and gyroscopes installed inside the body of the ground robot.

**Proprioceptive Terrain Classification**

Grouping terrain patches based on proprioceptive sensor data, such that terrain patches with similar vibration properties are associated with the same terrain class is called proprioceptive terrain classification. In [6], researchers present both a supervised and an unsupervised method for proprioceptive classification. The supervised method relies on training of a Support Vector Machine (SVM) classifier to identify terrain classes based on proprioceptive sensor data, where terrain classes are defined by a human supervisor during training. The unsupervised method uses a Gaussian Mixture Model (GMM) to group terrain patches into classes based on proprioceptive sensor data. Our work on proprioceptive classification is heavily inspired by their work with slight tuning of parameters and descriptors. We use a FFT representation of proprioceptive measurements as the descriptor for our 1-second traversals. Also a more simplistic heuristic descriptor is introduced in Chapter 4. After finding the right tuning of parameters in a one-time offline setting, the outputs of this module are fed into the exteroceptive classifier as labels to continuously train the visual classifier. This is shown in Figure 1.2.
Exteroceptive Terrain Classification

One way of predicting terrain vibration properties is to learn attributes of certain terrain classes such as grass, asphalt and dirt acquired by an exteroceptive sensor and assume any of the mentioned terrain classes has a fixed vibration property associated with it. In this approach, the problem of predicting terrain characteristics changes into visual classification of terrain. An exteroceptive terrain classifier processes terrain patches based on features derived from exteroceptive sensor data. In our framework, the exteroceptive sensor is a monocular camera and the raw features are color and texture. In literature there is a lot of work on terrain classification from on-board cameras attached to ground vehicles [7] in different settings. Some utilize their classifiers on images acquired by forward looking cameras for far ranges [8] to close ranges [9]. Some classify images taken in preconditioned settings with downward facing cameras [10]. There is a scarcity of literature on terrain classification for moving aerial vehicles. The majority of work in that realm focuses on steady images captured from high altitude aerial vehicles or satellite images [11]. The work of Khan [12] is most similar to our classifier module in that we both assess terrain classifiers on images acquired by a vehicle flying 3-12 meters above the ground and the challenge of moving aerial vehicles is that they introduce motion blur. A major decision step for autonomous mobile robots is to use proper features and classifiers to detect terrain characteristics. Selecting the correct features and classifier combination is dependent upon the application and setup of the robot. In Chapter 5 we assess the stand-alone visual classifier which is a supervised SVM classifier on a range of visual descriptors ranging from Local Binary Pattern (LBP), Grey Level Occurrence Matrices (GLCM), Bag of Visual Words (BOVW) of Scale Invariant Feature Transformations (SIFT) and Speeded Up Robust Features (SURF). Depending on the class of the terrain, the amount of motion blur and altitude the images were taken, different descriptors achieve different results. Our re-
results indicate that BOVW SIFT and BOVW SURF descriptors perform poorly on average in comparison to classical features on images which had less texture and sharpness due to height and motion.

**Adaptive Near-to-Far Learning**

Confronted with a variety of terrain properties in outdoor settings, an *a priori* classifier is certain to fail since there will always be terrain which has not been seen and trained our offline classifier beforehand. Hence, there needs to be a classifier that learns the association between new image features and vibration properties on the fly for future interaction. This is how the near-to-far learning method is proposed for our problem. The framework we utilize works such that vibration characteristic labels are paired up with the visual classifier training images online. Additionally, there are inter-class vibration characteristics variations among certain terrain classes such as grass. For example, some grass terrain patches have higher amplitude in their vibration profile while some other grass terrain patches have lower amplitudes but higher frequency attributes in their vibration profile. This observation requires the use of a classifier that also updates its decision boundaries for previously seen terrain classes - hence the name “adaptive” near-to-far learning. This is achieved through an incremental SVM classifier which is discussed in more detail in Chapter 6. In contrast to SVM supervised learning, where the efficiency of the classifier drops rapidly along with the increasing number of samples, incremental SVM does not require large amounts of labeled samples as prior knowledge for training. We finally compare the results of our incremental SVM classifier with the offline classification method on the same set of image sequence and discuss the scenarios with highest gains and lowest gains in terms of accuracy.
Assumptions

In the context of this thesis, we assume all terrain is traversable and different behaviors are induced on the robot upon traversal due to different terrain characteristics. It should also be mentioned that the framework is implemented offline but the data is collected and analyzed in a sequential manner to resemble an online method. One important assumption of this framework is that the classifier assumes any 1-second segment of a signal contains only one terrain class.

1.3 Contributions

The contribution of this thesis is the deployment of an aerial / ground vehicle in an adaptive near-to-far learning setting. To the best of our knowledge, no other work has utilized images captured from an aerial vehicle as the far-range sensor in the near-to-far setting. Our work can also be considered a survey on utilizing state of the art descriptors for images taken from a moving aerial vehicle. The challenge for images taken from an aerial vehicle is that due to motion blur and the distance from the ground, many details of the terrain are decayed in the image. In Chapter 5 where the results of the stand-alone visual classifier are presented, different settings and conditions are tested and compared to one another. In this thesis, the association of visual appearance of images taken from a moving aerial vehicle to vibration properties of a terrain is learned online. This is new since previous work focuses on frameworks where the camera is attached on the ground vehicle. The adaptive self-supervised framework enables predictions of vibration properties of far terrain by learning from experiences gained during traverses of similar terrain. The results of our work can be utilized for path planning algorithms. By utilizing the adaptive near-to-far algorithm and comparing it to the stand-alone supervised SVM classifier, we observed
an average 6% increase in accuracy results. We also must add that our framework is a general framework. For example, our terrain characteristics can integrate more proprioceptive (local) sensor measurements in its calculations such as current measurements to represent torque of each motor and also encoder readings to measure slippage of the wheels during terrain traversal and later find the association of these characteristics with visual features of those terrain but we leave the details of that work for future study.

1.4 Organization of Thesis

The remainder of the thesis is organized as follows: Chapter 2 gives an overview of literature around the topic of terrain traversability. Chapter 3 provides background knowledge on different learning algorithms and terminology used for evaluation of the algorithms; details of the algorithms are discussed later wherever they are used. Chapter 4 describes the first module of our framework which is the proprioceptive terrain classifier using accelerometers and gyroscopes as sensor inputs. The results of different feature selections and classification algorithms are discussed along with a parameter selection analysis. Next, we investigate the results of state of the art visual terrain classification techniques on images grabbed from our moving aerial vehicle in Chapter 5. Due to motion blur introduced by the moving aerial vehicle, a variety of visual features and image descriptors are put to test and the best configuration is reported. The results of Chapters 4 and 5 are then combined to achieve a hybrid terrain characterization framework which is discussed in Chapter 6. The classifier from Chapter 4 acts as the proprioceptive feedback to the classifier presented in Chapter 5. The results of our near-to-far learning framework are compared to a state of the art a priori supervised classifier undergoing different scenarios in Chapter ??Finally, the challenges to our approach are discussed and future recommendations are provided in Chapter 7.
Chapter 2

Background and Literature Review

In this thesis predicting characteristics of far terrain is of interest. As mentioned in the introduction, to accomplish this task two modules are used. A proprioceptive module to characterize terrain that the ground robot is traversing at any given time and also an exteroceptive module that learns the association between visual features of far terrain and characteristics labels. In literature, terrain characteristics prediction falls under the umbrella of terrain traversability analysis. Methods for terrain traversability analysis are identified by Papadakis [7] as follows:

- Proprioceptive
- Appearance-based
- Geometry-based

As discussed below, the work in this thesis falls under the proprioceptive and appearance-based categories. Proprioceptive methods are useful in learning the model that captures the difficulty encountered when a vehicle traverses a given terrain. Since in our work we are interested in the response of a robot upon traversal of any terrain, proprioceptive techniques
are utilized. Therefore, previous work on proprioceptive-based methods is discussed in Section 2.1.

Appearance-based methods project the problem of traversability analysis into the image-processing and classification realm and usually assess a discrete set of terrain classes rather than regressing traversability. In appearance-based methods, terrain “classification” is the act of identifying a specific terrain from among a list of terrain candidates using sensory data such as LiDAR and cameras. Several approaches for terrain classification have been proposed in literature but the main focus of this research is on the use of single camera imagery captured from a moving aerial vehicle. The visual classifier module referred interchangeably as the exteroceptive classifier, receives images acquired by the aerial vehicle and processes them to find a mapping between visual features and terrain classes. An overview of previous work concerning appearance-based traversability analysis is provided in Section 2.2. Geometry-based approaches build a terrain model based on geometric structure of both the terrain and the robot by taking into account kinematic and stability constraints such as steepness of a terrain. Geometry-based methods are not discussed in the scope of this research since all terrain is picked to be traversible based on a geometric point of view.

Our proposed framework is a self-supervised framework in which labels from the proprioceptive module are fed into the exteroceptive (appearance-based) module. Section 2.3 reviews literature on hybrid approaches usually involving near-to-far learning approaches. In near-to-far learning frameworks, an association between far sensor readings and the near reading sensor is established. Our far sensor is a monocular camera attached to an aerial vehicle, flying within 3 to 12 meters above the ground whereas our near sensors are accelerometers and gyroscopes.
2.1 Proprioceptive Terrain Classification/Characterization

Proprioceptive terrain classification methods focus on improving accuracy results and performance issues such as time performance by selecting the best combination of sensors, descriptors and classifiers. Among these proprioceptive measurements, some focus on vibration while some focus on slippage and stickiness of the terrain. There is also a distinction between classification and characterization. In classification methods, the purpose is to classify measurement readings into their representative terrain class. For example, by recording and processing a signal acquired by a proprioceptive sensor, the classifier predicts if the current reading is grass, dirt or asphalt. On the other hand, in characterization methods the emphasis is on defining metrics to properly measure the amount of slippage or stickiness of a terrain type based on sensor readings. In this work, predicting vibration properties of far terrain is of interest. To accomplish this, as the first step an IMU is used to measure vibrations induced on a robot upon traversal of terrain. To understand the implications of using an IMU to measure vibration properties of terrain, a review of literature on proprioceptive classification is presented. Vibration-based terrain classification was first suggested by Iagnemma and Dubowsky using only accelerometers [13] as the proprioceptive sensor. Later, in [10], researchers use the vibrations induced on a wheel to estimate cohesion and the internal friction angle in a very constrained testbed. The testbed consists of a single rigid wheel mounted on an undriven vertical axis. By driving the wheel and carriage at different rates, variable slip ratios are imposed. Our wheeled robot has four wheels and different loads are forced upon each of them depending on the terrain type. Also speed variations affects the accuracy of their classifier. Sadhukhan and Moore [14] use a neural network approach focusing on classification of terrain for a high-speed regular size vehicle. The ground robot used in this thesis is a small size robot where small physical objects on terrain induce different responses. Also low speeds are of interest in the framework of this
research. Therefore, their method is not adaptable to our application. In Ojeda et al. [15], a differentiation between classifying terrain vs. characterizing terrain is first discussed. A smaller sized vehicle and an Artificial Neural Network (ANN) are used to classify terrain into distinct classes such as gravel, pavement, dirt, sand and grass. Their classifier confuses dirt with sand and grass due to similar vibration profiles of those terrain classes. In Weiss et al. [16] a Support Vector Machine (SVM) is used as the classifier on fast fourier transform (FFT) and power spectral density (PSD) representaion of vibration readings collected from a hand pulled cart traversing seven different terrain types. The methods mentioned above tackle the same problem by trying to increase accuracy results of their prediction. An evaluation on a standard dataset was required to compare the results of the approaches. Therefore, Weiss et al. [17] shows a comparison of previous classification methods based on frequency representation of data collected from an ATRV-Jr outdoor robot. It is shown that SVM classifiers give better results than other previous classifiers. In his survey, constant speed scenarios are evaluated. Due to speed variations in our experiments, an invariance to speed factor was required. Velocity independent terrain classification is addressed in [18, 19]. The proprioceptive module of our framework is built upon the success of SVM classifiers described in [18] and from pre-experiments, it is decided to present a set of different simpler features to input into the classifiers. This modification is discussed in Chapter 4. Finally, it is important to mention that proprioceptive terrain classification and distinguishing different terrain types using accelerometers and gyroscopes is possible only as long as the terrain causes distinct terramechanical behavior in the robot.
2.2 Exteroceptive Terrain Classification/Characterization

Research in exteroceptive terrain classification falls within one of the categories below.

- Geometry-based
- Appearance-based

As mentioned in the introduction section, a monocular camera is selected as the exteroceptive sensor due to power and weight constraints. A monocular camera captures color and intensity in each pixel of an image frame and lacks geometric information in a single image frame. Therefore, we discuss the appearance-based methods of literature on exteroceptive classification. The appearance-based approach in literature projects the problem into the image processing and classification realm where a discrete set of terrain classes is usually given as an output rather than a traversability metric. Furthermore, another categorization of terrain classification methods is based on the scale, resolution and viewpoint the images are captured from. Terrain classification by on-board cameras on ground robots generally falls into a different category compared to data acquired from aerial vehicles. Since the underlying techniques in on-board camera image classification and aerial image classification share many commonalities between one another, the literature on both the categories are presented. This is done separately so a fair comparison to the approach in this work is given.

2.2.1 On-board Ground View Classification/Characterization

Visual terrain classification techniques require learning a mapping between extracted features from images and their associated class / characteristic. Angelova et al. [20] perform classification, starting by fast and simple classifiers and advance into finer and more complex classification algorithms using color statistics and textons. The on-board cam-
era classifies terrain into human labeled terrain such as soil, sand, gravel, asphalt, grass and woodchips. This classification method does not recognize the fact that within a certain type of terrain like grass, different types of behavior can be induced in the robot. For example, wet grass or dry grass possess different traversability metrics for a moving wheeled robot or flat grass or bumpy grass can imply different behaviors. Given this, Howard and Seraji [21] regress the terrain traversability instead of classifying terrain by measuring terrain roughness, slope, discontinuity and hardness through a fuzzy logic network. In their work, roughness is defined as a function of size, concentration and average separation distance of rocky regions within the observed area captured from the on-board camera and hardness is predicted as a measure of potential wheel slippage, by performing texture analysis. Their approach relies on detecting rocky regions. This is not robust since there is no learning involved and in many instances the rocky region is camouflaged with another type of terrain such as grass. The near-to-far method proposed in this research learns the behavioral properties of any given terrain on the fly while being more robust to camouflage and occlusions.

One of the challenges of our proposed framework is that due to wind, the aerial images are captured from different altitudes and angles. This introduces scale and rotation variation to captured images of similar terrain. Therefore, in the visual classifier, descriptors need to be invariant to rotation and scale, meaning that if a terrain is captured from a different angle, the descriptor will still be the same and also invariant to scale. Recent developments in computer vision allow this to happen. Recent visual descriptors such as Scale Invariant Feature Transforms (SIFT) [2] and Speeded Up Robust Features (SURF) [22] are used to extract more robust features from image patches. Filitchkin and Byl [23] use these robust features using a bag of visual words (BOVW) model [24], to distinguish a predefined set of natural terrain classes and subsequently select appropriate gait behaviors for a quadruped robot. Their images are captured from close range, with sharpness and details present in every picture. Inspired by the results of Filitchkin, the visual classifier in this research uses
the BOVW representation applied on SURF and SIFT descriptors. But as explained in Chapter 5, some results of the classifier using these descriptors are not satisfying due to introduction of motion blur and lower resolution of images at higher altitudes. Overall, in this work an assessment on a variation of textural descriptors is presented. Depending on the presence of motion blur in images, and altitude of the aerial vehicle, different descriptors such as LBP prove better than SIFT or SURF features.

2.2.2 Aerial View Classification

Aerial classification of terrain started with the DARPA PerceptOR program. In the DARPA PerceptOR program, aerial LiDAR data is used to predict ground vehicle orientation over remote terrain, as well as to detect vegetation, so it is both considered an appearance-based and also a geometric-based method. Sofman et al. [25] use the combination of camera and range data from a laser mounted on a Unmanned Aerial Vehicle (UAV), to classify terrain into road, grass, tree, and building semantic classes which later in [26] is used to aid in long range navigation for an Unmanned Ground Vehicle (UGV). The approach relies on the geometric and semantic interpretation of heterogeneous data sources to produce traversal cost maps while every class has a fixed a priori traversal cost. As discussed earlier, fixed a-priori traversal costs assigned to semantic classes, produce wrong results for traversals since there is no online learning involved. To tackle the shortcoming, a learning framework is proposed in this thesis which also captures images from lower altitudes. Hovering at lower altitudes provides the capability to classify terrain into a wider range of semantic groups due to higher details. Hudjakov et al. [11] classify terrain from static aerial images into “house, “road, “grass or “dirt categories using a convolutional neural network from a database of static images captured from a UAV. In [27] , the authors suggest a multi-class gaussian process classifier which provides probabilities of class mem-
bership at each location in the image instead of semantic labels. The research in this thesis is different from those mentioned above because the captured images from the aerial vehicle are not static, meaning that they are captured while the vehicle is in motion resulting in motion blur. Also the resolution of the images in this research are usually higher than 100 pixels per meter whereas the other work have resolutions less than 10 pixels per meter.

To find the association between appearance and vibration characteristics of different terrain, it is assumed that texture of the terrain plays a vital role. Many state of the art texture classification approaches use sharp images containing a single texture captured from a fixed camera angle under controlled conditions but this is not the case for images from the moving aerial vehicle. The images captured in our case contain blurred artifacts due to motion blur. The best image texture descriptors in this research are heavily inspired based on the work of Khan [12]. Khan, investigates the performance of different image descriptors at varying resolutions for images taken from a flying vehicle at motion. Along with two texture-based descriptors, Local Binary Pattern (LBP) and Local Ternary Pattern (LTP), he also investigates the SURF descriptor. In his work it is shown that LBP and LTP descriptors perform best at low resolutions compared to other texture descriptors. In addition to his list of descriptors, we also assess the performance of Grey Level Co-occurrence Matrix (GLCM) along with BOVWSIFT on the set of images. Just like the previous mentioned work, his work focuses on terrain classification instead of characterization. Also there are no self-learning modules present in his work.

2.3 Near-to-Far Learning and Hybrid Approaches

In any near-to-far learning framework, the far sensor needs training. In our case the camera module attached to an aerial vehicle is considered the far sensor and needs a proprioceptive feedback for training purposes. A literature review on sensor fusion methods,
In [28], the authors use a hybrid approach to address terrain classification. They “fuse” the predictions of their visual classifier with the vibration classifier predictions. The fusion uses terrain class probabilities that are estimated by an SVM algorithm. They show classification into semantic groups is improved by over 10% on average compared to the best of the individual methods. The proposed hybrid characterization method in our thesis is different in that terrain characterization of remote terrain is implemented. Also there is no on-the-fly learning involved in their approach whereas this research has an incremental learning algorithm. Approaches that also seek far terrain characterization from a near-to-far framework with different sensor configurations are investigated. In Stavens et al [29], they train a laser sensor by monitoring vertical accelerations caused by unevenness of the ground on a commercial vehicle. A simple surface roughness-based traversability metric is analyzed for adapting the speed of a mobile vehicle based on the terrain immediately in front of it. Due to the small size of the ground robot in this research, a laser sensor can not fit on the robot. Also the lower height of our ground robot, limits the field of view necessary for a laser sensor to catch far terrain. Therefore, to overcome the mentioned constraints, an aerial vehicle with a monocular camera attached is utilized in our framework. In [15], visual appearance is learned based on vibration data and is used to identify the distant terrain but again is not applicable to our framework because of the limited field of view of our ground robot. Very recently, In [30], a self-learning framework for statistical ground classification using radar and monocular vision is presented. Their approach classifies terrain into ground and non-ground by calculating the Mahalanobis distance between the feature vector of the visual patch and the closest component of the current ground model which is learned online. Their method learns “classification” of terrain types whereas we learn “characterization” of terrain. Also the sensor settings are different in that we use a vibra-
tion analysis module to label the visual classifier whereas they use a radar. The closest sensor configuration to the framework proposed in this thesis, is the work of Brooks et al. In [6]. In their work, two proprioceptive terrain classifiers, one based on wheel vibration and one based on estimated traction force, are used to train an exteroceptive vision-based classifier that identifies instances of terrain classes in the long range. Our work is different from them in the set of descriptors used for our visual classifier and also the fact that images are captured from a moving aerial vehicle instead of an on-board ground vehicle. Learning the association between images captured from an aerial vehicle and vibration characteristics recorded by a ground vehicle has two main challenges. One is the scale ambiguity inherent in the images captured from an aerial vehicle, the other is the motion blur present in the aerial images. Brooks et al. do not have to tackle this challenge, since their images are always captured from the on-board ground robot and therefore images are captured from a constant angle and position.

2.4 Summary of Contributions

Past literature mostly addresses binary classification in which terrain is identified as either traversable or non-traversable [8, 31, 30]. The suggested framework here presents a variation among terrain characteristics and the impact on robot behavior caused by the terrain assuming all terrain is traversable. Furthermore, some terrain sensing approaches [32, 33, 25] detect “remote” geometric hazards and avoid them, such as rocks and abrupt elevation changes, but lack robustness to remotely identify non-geometric hazards [34] which strongly influence the ground robot mobility. In this research, it is argued that the behavior of a ground vehicle on different terrain can be described by an application-specific model which in this research is referred to as a Robot-Terrain-Interaction (RTI) model. For example, in one application a terrain with small bumps but high repeatability is preferred
over a terrain with big bumps but less frequency in occurrence. Therefore, a RTI model is proposed to discretize characteristics of terrain based on wheel interactions. The terrain model is calculated by proprioceptive sensor measurements acquired by an Inertial Measurement Unit (IMU) over a time period of traversal of that terrain. In the work of researchers such as [9], remote terrain sensing using exteroceptive sensors typically assumes that the visual appearances of different characteristics of terrain, are known a priori. This requires certain terrain classes such as grass to always have the same mechanical properties. But initial experiments in this research show that terrain characteristics differ significantly within human labeled classes such as “grass”. This is illustrated more informatively in Table 2.1. As an example, grass terrain traversals are mostly composed of Less Frequent High Amplitude vibrations, (LFHA features) while also having a big share of Highly Frequent High Amplitude vibrations (HFHA features) too. As a result, a standalone terrain classification algorithm is not able to predict characteristics of image patches solely based on a priori labeled classes.

<table>
<thead>
<tr>
<th>Characteristics Ground Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Woodchips</td>
</tr>
<tr>
<td>Sidewalk</td>
</tr>
<tr>
<td>Grass</td>
</tr>
<tr>
<td>Dirt</td>
</tr>
<tr>
<td>Asphalt</td>
</tr>
</tbody>
</table>

Table 2.1: Ground truth characteristics of different terrain.

Given this, a learning framework is necessary but previous near-to-far learning methods lack robustness when interacting with new terrain characteristics [28]. To address this problem, a self-learning characterization approach is proposed for learning the visual appear-
ance of terrain characteristics extracted from on-board accelerometers and gyros recording vibrations induced on a robot by interaction between the robot and its environment. This approach is different from existing literature in that it is a method which does not require human intervention to train the supervising module hence the term “self-supervised”, nor does it require huge database of \textit{a priori} assumptions about different terrain. Furthermore, the learning module performance improves over time due to incrementally updated decision boundaries which continuously adapt to changes in the ground characteristics. In the following chapters the modules of this framework are discussed.
Chapter 3

Machine Learning Background

All data processing in this thesis is performed offline using Matlab. The research utilizes classification / characterization algorithms [35] in different modules of the framework. They are all well known algorithms that are included in popular open-source machine learning libraries which provide wrappers for Matlab. We use LIBSVM [36] and LIBLINEAR [37] for the SVM classification and Regression modules. They are two popular libraries written in C++ with a C API. LIBSVM implements the Sequential minimal optimization (SMO) [38] algorithm for kernelized SVMs, supporting classification and regression while LIBLINEAR implements linear SVMs [39] and logistic regression models trained using a coordinate descent algorithm [40]. Another popular library used throughout this research is the scikit-learn [41] open source machine learning library. It features various classification, regression and clustering algorithms largely written in Python. The remainder of this chapter is an overview of each classification algorithm used within this thesis. Also a taxonomy of the evaluation methods in machine learning is presented so that the reader understands the significant contribution of the research.
3.1 Feature Space

The fundamental problem of machine learning algorithms is to approximate the functional relationship \( Y = f(X) \) or mapping between an input vector \( X = \{x_1, x_2, \ldots, x_M\} \) known as the feature vector and an output vector \( Y \), based on a set of data points, \( \{X_i, Y_i\}, i = 1, \ldots, N \). In our proposed framework we have two functional relationships to learn. One is learning the mapping from proprioceptive sensor features to distinct vibration classes. The other is learning the functional relationship between visual features to vibration classes.

Feature space refers to the \( M \)-dimensional space in which the input vector variables are defined. For example, selecting the average value of each of the R, G, B channels of an image yields a 3-dimensional feature space.

3.1.1 Feature Selection

The variables composing the feature vector are selected in an artful manner. This is called feature extraction [42]. As mentioned above, we are interested in the mapping from raw visual data (a 64 × 64 pixel RGB image patch) to the roughness and vibration vector space. Each image patch can inherently be represented by a 64 × 64 × 3 - dimensional feature space but a wiser set of features [43, 44] are selected such as local / global image descriptors.

In the functional relationship \( Y = f(X) \) sometimes the output \( Y \) is not determined by the complete set of the input features \( \{x_1, x_2, \ldots, x_M\} \) and instead is decided only by a subset of them \( \{x_{(1)}, x_{(2)}, \ldots, x_{(m)}\} \), where \( m < M \). It must be mentioned that it is fine to use all the input features, including those irrelevant features, to approximate the underlying function between the input and the output (in our case the whole 64 × 64 × 3 feature space)
but in reality, there are two problems which usually arise by the irrelevant features involved in the learning process.

1. The irrelevant input features will induce greater computational cost.

2. The irrelevant input features may lead to overfitting [45].

Therefore, selecting the correct features determines the success or failure of any classifying algorithm. In this research, for the visual classifier, the accuracy of GLCM, LBP and also BOW representation of both SIFT and SURF descriptors of any image patch are evaluated. For the proprioceptive classifier a set of manually defined features along with a discretized FFT representation is extracted. The details of these descriptors are presented in later chapters.

### 3.2 Unsupervised Classification

Unsupervised learning is the machine learning task of inferring a function to describe hidden structure from unlabeled data [46]. It is also closely related to the problem of density estimation in statistics. In the following, a description of the unsupervised algorithms used in this research is given.

#### 3.2.1 Gaussian Mixture Model

Gaussian Mixture Models (GMMs) [47] can be used for clustering data. A GMM is a parametric probability density function represented as a weighted sum of Gaussian component densities as given by Equation 3.1.

\[
p(\theta|x) = \sum_{i=1}^{K} \phi_i \mathcal{N}(\mu_i, \Sigma_i) \tag{3.1}
\]
GMMs are commonly used as a parametric model of the probability distribution of continuous measurements or features. A GMM is used in the proprioceptive terrain characterization module to cluster similar vibration feature vectors into the same group and also detect novel terrain characteristics. It must be mentioned that in practice, with real data, and no \textit{a priori} knowledge of density of different terrain types the number of gaussians are selected heuristically by comparing the number of the components and then fitting the density. Expectation-maximization [48] is the statistical algorithm to fit density functions through an iterative process. Detailed explanation of the approach can be found in [49]. The main difficulty in learning GMMs from unlabeled data is that there is no knowledge of the latent component of each sample. In the case of the proprioceptive module, when vibration signals are recorded from each 1-second travel segment, a point is given in the feature space. Since in natural scenes and robot traversals, similar terrains tend to be clustered next to each other, this inherent information is used as a This means that two consecutive readings have a high probability of belonging to the same terrain group and therefore belonging in the same density of the GMMs.

\subsection*{3.2.2 K-means Algorithm}

K-means is another unsupervised learning algorithm that solves the clustering problem [50]. In this research it is used as a feature learning (dictionary learning) step [3] for a BOVW model in the exteroceptive classifier explained in section 5.2.2. The main idea of the algorithm is to define k one for each cluster. The next step is to take each point belonging to a given data set and associate it to the nearest center [51]. k-means clustering partitions n observations into k clusters in which each observation belongs to the cluster with the nearest mean. This results in a partitioning of the data space into Voronoi cells. The k-means algorithm proceeds as follows:
1. Initialize by choosing $k$ random centers within the sample space.

2. Each sample gets assigned to the cluster whose center it is closest to.

3. The centers are re-calculated as the mean of all the samples of a cluster.

4. Steps 2 and 3 are repeated until the centers stop changing by more than a defined epsilon threshold.

The disadvantages of k-Means clustering in our applications is as follows:

- Centers should be located cautiously since different locations might cause different results.

- It is strongly sensitive to outliers and noise.

- The number of clusters ($k$) is given to the algorithm and is pre-determined.

- As it is a heuristic algorithm, there is no guarantee that it will converge to the global optimum

- k-Means works well when the shape of clusters are hyper-spherical (or circular in 2-d). If the natural clusters occurring in the dataset are non-spherical then k-means fails.

In Figure 3.1 an example has been shown that k-means will not yield the right clustering.

### 3.3 Supervised Classification

In supervised learning, the output datasets are provided and used to train the model and get the desired outputs. Supervised learning problems are categorized into “regression” and “classification” problems. In a regression problem, prediction results are a continuous
range. This means that a mapping between input variables to some continuous function is found. In a classification problem, prediction results are discrete outputs. In other words, input variables are mapped into discrete categories. First a K-nearest neighbor algorithm is introduced and later the Support Vector Machine (SVM) is thoroughly discussed since it is the main classifier used in this thesis. In this thesis, the results of the proposed framework are compared to the results of the \textit{a priori} classifier. In the \textit{a priori} classifier, a mapping from input to output is learned by providing enough training data. After a cross validation step confirms that our algorithm has a high enough accuracy result, the learned algorithm is applied to determine mapping of any new inputs.

\subsection*{3.3.1 K-Nearest Neighbor}

K-nearest neighbor (k-NN) proposed by Cover and Hart [52] is a classification technique which classifies objects based on the closest training pattern in the feature space efficiently. The output of the k-NN is a class membership. A vector is classified by a majority vote of neighbors, with the vector being assigned to the class most common among its \( k \) nearest neighbors. For example, if \( k = 1 \), then the object is simply assigned to the class of
that single nearest neighbor. A good choice of k depends on the application. For k = 1, the algorithm reduces to a simple nearest neighbor algorithm that associates the sample with the class of the closest matching instance in the training data. Larger values of k reduce the effect of noise on classification [53] results. The advantages of the k-NN compared to other techniques is that there is no need for tuning complex parameters to build a model. Also, unlike SVM, no training is involved and new training data can be added easily.

### 3.3.2 Support Vector Machines

Support Vector Machines [54] are based on the concept of decision planes that define decision boundaries. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new samples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap as wide as possible. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class, since in general the larger the margin the lower the generalization error of the classifier. New instances are mapped into the same space and predicted to belong to a category based on which side of the plane they fall.

**Kernel Trick**

SVMs can efficiently perform a non-linear classification using what is called the “kernel trick”, implicitly mapping their inputs into high-dimensional feature spaces. In other words, it allows the classifier to learn a nonlinear classification rule which corresponds to a linear classification rule for transformed data points $\varphi(\vec{x}_i)$. Moreover, a kernel function $k$ which satisfies $k(\vec{x}_i, \vec{x}_j) = \varphi(\vec{x}_i) \cdot \varphi(\vec{x}_j)$ is given [55].
The classification vector $\vec{w}$ in the transformed space satisfies

$$\vec{w} = \sum_{i=1}^{n} c_i y_i \varphi(\vec{x}_i),$$

where the $c_i$ are obtained by solving the optimization problem

$$\text{maximize } f(c_1 \ldots c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i c_i (\varphi(\vec{x}_i) \cdot \varphi(\vec{x}_j)) y_j c_j$$

subject to $\sum_{i=1}^{n} c_i y_i = 0$, and $0 \leq c_i \leq \frac{1}{2n\lambda}$ for all $i$.

The kernel trick enables operation in high-dimensional implicit feature spaces without ever computing the coordinates of the data in that space, but rather by simply computing the inner products between the images of all pairs of data in the feature space.

Some common kernels include:

- **Polynomial (homogeneous):** $k(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j)^d$

- **Polynomial (inhomogeneous):** $k(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^d$

- **Gaussian Radial Basis Function:** $k(\vec{x}_i, \vec{x}_j) = \exp(-\gamma \|\vec{x}_i - \vec{x}_j\|^2)$, for $\gamma > 0$.

In the SVM classifier presented in Chapter 5, both homogeneous polynomial kernels and also Gaussian radial basis functions are utilized to determine which presents better results.
**Multiclass SVM**

An SVM is a binary classifier, meaning that the class labels can only take one of the two values: ±1. But in many real-world problems, a distinction between more than two classes is required. For example, the visual classifier is interested in classifying terrain into grass, asphalt, dirt, woodchips. To overcome this limitation, two approaches are used to achieve multiclass SVM classification are described.

**One-vs-All Classification**

The simplest approach is to reduce the problem of classifying among \( K \) classes into \( K \) binary problems, where each problem discriminates a given class from the other \( K - 1 \) classes [56]. For the \( i \)th classifier, the positive examples are all the points in class \( i \), and the negative examples are all the points not in class \( i \). Then if \( f_i \) is the \( i \)th classifier. New inputs are classified by

\[
\hat{y} = \arg \max_i f_i(x)
\]

Although this strategy is popular, it is a heuristic that suffers from several problems. First, the scale of the confidence values differ between binary classifiers. Secondly, even if class distribution is balanced, the binary classification learners see unbalanced distributions because typically the set of negatives they see is much larger than the set of positives.

**All-vs-All Classification**

In the all-vs-all approach, each class is compared to each one of the other classes [57]. A binary classifier is learned to discriminate between each pair of classes, while ignoring the rest of the classes. This requires building \( K(K - 1)/2 \) binary classifiers. When testing a new sample, a voting is performed among the classifiers and the class with the maximum number of votes gets assigned to the test sample. This approach is more robust to the one-
vs-all approach but also more computationally expensive. In our research, the all-vs-all method is used.

### 3.3.3 Cross Validation

Cross-Validation is a statistical method for evaluating and comparing learning boundaries by dividing data into two segments. One segment is used to train a model and the other is used to validate a model. The basic form of cross-validation is k-fold cross-validation which is used for evaluation of our SVM algorithm.

#### K-fold Cross-validation

In k-fold cross-validation, the data is first partitioned into k nearly equally sized folds. Subsequently k iterations of training and validation are performed such that within each iteration a different fold of the data is kept-out for validation while the remaining $k - 1$ folds are used for learning. Figure 3.2 demonstrates an example with $k = 5$. The white boxes are used for training while the orange boxes are used for validation. The performance measure reported by k-fold cross-validation is then the average of the accuracy values computed in the loop.

![K-fold cross validation](image)

*Figure 3.2: K-fold cross validation*
3.3.4 Overfitting and Robustness

Overfitting is the phenomenon whereby a statistical model describes random error or noise instead of the underlying relationship. When the model is excessively complex, such as having too many parameters relative to the number of observations, the chances of overfitting arises. Generally, a learning algorithm overfits relative to a simpler one if it is more accurate in fitting known data but less accurate in predicting new data. A learning algorithm that reduces the chance of fitting noise is called robust. Enhanced robustness is the objective of the classifiers introduced in Chapter 5.

![Figure 3.3: Example of Overfitting](image)

3.3.5 Parameter Selection

During the offline phase of an SVM classifier, parameters are selected by trial and error. In training, the $C$ parameter trades off misclassification of training examples against simplicity of decision surfaces. A low $C$ makes the decision surface smooth, while a higher $C$ emphasizes classifying all training examples correctly by giving the model freedom to select more samples as support vectors.

Also, the $\gamma$ parameter determines how far the effect of a single training example reaches. $\gamma$ determines the inverse of the radius of influence of samples selected by the model as
support vectors. In [58], a practical guideline to parameter tuning is presented. We follow the instructions of the guideline to select which $C$ and $\gamma$ are best for a given problem in the RBF kernel. It suggests a “grid-search” on $C$ and $\gamma$ using cross-validation. Various pairs of $(C, \gamma)$ values are tried and the one with the best cross-validation accuracy is picked. Exponentially growing sequences of $C$ and $\gamma$ are selected in the grid-search method (for example, $C = 2^{-5}, 2^3, \ldots, 2^{15}$, $\gamma = 2^{15}, 2^{13}, \ldots, 2^3$).

### 3.3.6 Confusion Matrix

In the following, a method for evaluation of learning algorithms is discussed. A confusion matrix [59] is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one. Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class. The name stems from the fact that it makes it easy to see if the system is confusing two classes.
Chapter 4

Proprioceptive Terrain Characterization

Proprioceptive sensors measure values internal to any system. The sensors for measurement can be encoders, gyroscopes, accelerometers, GPS and even current sensors for each motor in a robot. Proprioceptive terrain “classification” is the process of assigning class labels to terrain patches based on features derived from proprioceptive sensor data. On the other hand, proprioceptive terrain “characterization” assigns characteristics to terrain patches based on features derived from proprioceptive sensor data. This chapter first presents an approach for proprioceptive terrain classification in Section 4.3 and then a combined characterization / classification method is proposed. The focus of this thesis is on characterizing vibrations endured by a robot while traversing terrain for 1-second intervals. Vibration characteristics for a terrain patch $i$ traversed at time $t$ is defined as a discrete discontinuous function of accelerations on the X, Y and Z axes of the robot and the angular velocities recorded within that 1-second period. This metric is referred to as the Robot-Terrain-Interaction (RTI) metric model throughout the remainder of the thesis.

$$\text{RTI}_i(t) = k_1 F((\ddot{x}(t), \cdots , \ddot{x}(t-1)), (\ddot{y}(t), \cdots , \ddot{y}(t-1)), (\ddot{z}(t), \cdots , \ddot{z}(t-1)))$$

$$+ k_2 G((\dot{\alpha}(t), \cdots , \dot{\alpha}(t-1)), (\dot{\beta}(t), \cdots , \dot{\beta}(t-1)), (\dot{\gamma}(t), \cdots , \dot{\gamma}(t-1)))$$

35
To calculate the RTI metric of any given terrain, the 1-second acceleration and gyroscope signals are mapped into the frequency domain as explained in Section 4.3.3. Next, based on application-specific requirements, a pattern for discretizing the frequency space is suggested. For example, an average of the amplitudes of all the frequency components can be calculated and based on the value, is assigned to a bin. In this research, the frequency space is represented by a 2-dimensional space. One dimension represents the average amplitude of the dominant frequency components while the other dimension represents the average frequency of the dominant frequency components. Therefore, each 1-second traversal is presented by a point in the 2-dimensional frequency-amplitude space. Next, this space is divided into 2 × 2 regions. The amplitude dimension is divided into 2 bins and the frequency dimension is divided into two bins. This is illustrated in Figure 4.1

![Figure 4.1: RTI metric. Each one-second traversal signal belongs to one of these bins](image)

The reason classification is distinguished from discrete characterization is that the same human-labeled terrain classes affect the robot quite differently under different conditions. For example, as shown in Figure 4.2, if only classifiers are used, the driving characteristics are different on asphalt depending on whether it is smooth asphalt or rough asphalt. The implications of this result are significantly important for Chapter 6, in which characteristics of
far terrain are predicted. In Figure 4.2, the ground truth characteristics of traversed terrain are shown. The columns show the number of each RTI class associated with the traversal of the semantic terrain classes. It is seen that even with a visual classifier able to detect semantic classes perfectly, it is unable to predict characteristics of far terrain. Given that \textit{a-priori} knowledge of average “Grass” terrain characteristics is known, using a standalone state of the art visual classifiers yields a 76\% accuracy for the grass terrain characteristics.

<table>
<thead>
<tr>
<th>Characteristics Ground Truth</th>
<th>LFLA</th>
<th>LFHA</th>
<th>HFLA</th>
<th>HFHA</th>
<th>Prob %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Woodchips</td>
<td>2</td>
<td>44</td>
<td>3</td>
<td>20</td>
<td>64%</td>
</tr>
<tr>
<td>Sidewalk</td>
<td>49</td>
<td>1</td>
<td>13</td>
<td>0</td>
<td>78%</td>
</tr>
<tr>
<td>Grass</td>
<td>3</td>
<td>81</td>
<td>2</td>
<td>20</td>
<td>76%</td>
</tr>
<tr>
<td>Dirt</td>
<td>6</td>
<td>0</td>
<td>34</td>
<td>0</td>
<td>85%</td>
</tr>
<tr>
<td>Asphalt</td>
<td>48</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>81%</td>
</tr>
</tbody>
</table>

In this table, it is seen that when the ground robot encounters grass and woodchip terrain types, in many instances it responds similarly. Also, sidewalk and asphalt terrain, cause similar behavior in the robot. In the case of asphalt and sidewalks, the vibrations encountered are easily recognizable because of the lower amplitude vibrations.

### 4.1 Overview

This section presents a method for classifying terrain patches based on vibrations induced in the ground vehicle structure through wheel-terrain interaction. The assumption is that if mechanically distinct terrains induce distinct vibrations, then features derived from vibrations can be used to distinguish between them. The approach relies on measurements of vibrations using accelerometer and gyroscope readings from an IMU mounted in the
body of the ground robot as shown in Figure 4.3. In the remainder of this chapter, first the robot configuration is discussed. The configuration is categorized into a digital and mechanical setup. In the digital setup segment, sensors and the processor specifications are provided. In the mechanical setup section, physical properties of the robot are explained. Next, the feature extraction and feature description setup of the characterization module is described. Vibrations are described in terms of a periodogram [60] and also a self-defined heuristic feature representation is introduced. Finally, discrete characterization of the resulting features using the SVM classifier is discussed. The classifier uses a supervised framework, which relies on labeled vibration training data collected for each of the terrain classes during an offline learning phase.

Figure 4.3: IMU placed on the body of the vehicle
4.2 Ground Vehicle Platform

In this section, the setup configuration of the ground vehicle and also the sensors required to get a proprioceptive terrain characterization / classification for each terrain patch are discussed. The outdoor ground vehicle, shown in Figure 4.4, is a Traxxas E-maxx with an XMOS XK-1A microcontroller development board operating as the embedded controller for this project. The XMOS is responsible for reading proprioceptive data and recording data readings on a SD card for later offline analysis. Since the focus of this work is behavioral analysis of the vehicle on different terrain, the most important parts and parameters of the robot which affect this behavior are presented in the following section. The internal parameters that affect the proprioceptive measurements are categorized into mechanical and digital parameters.

Figure 4.4: Ground robot with the attached coordinate frame. The x axes faces in the same direction of the robot. The z axes is perpendicular to the ground
4.2.1 Mechanical Parameters

Dimensions and Weight

The dimensions of the ground vehicle are 51.8 x 42 x 24.9 centimeters and it weighs 4.4 kilograms without the added sensors and additional equipment. The added sensors and box on top of the ground vehicle add another 450 grams resulting in the total setup weight of 5 kilograms.

Wheels and Tires

The tires are the only form of physical interaction for the ground vehicle with the environment. The ground vehicle is equipped with four wheels each having tractor tires making it able to run on rough terrain while reducing slippage. However, the downside of these stripes on the tires is that they produce an increased amount of vibration while traversing smooth terrain. To minimize the effect of the tractor tires on vibration classification a pre-processing method is suggested and implemented as discussed in Section 4.3.2. In Figure 4.5, the tractor tread pattern of the tires along with the size of the tires of the vehicle is shown. The width of the tire is 56 mm, measured from sidewall to sidewall and the diameter of the tire is 80 mm, measured from one end to the other passing through the center of the tire. In Figure 4.6, images of the vehicle tires on 4 different terrain types is shown.

Suspension system

Suspension is the system of springs, shock absorbers and linkages that connect a vehicle to its wheels and allows relative motion between the two. Suspension systems usually serve a dual purpose [61].

1) Contributing to the vehicle’s roadholding ability.
2) Isolating the inner body from road noise, bumps and vibrations.

In Figure 4.7, the suspension system of the Emaxx vehicle is illustrated. The tuning parameters of the suspension system are the pivot ball caps, caster and the shock mounting positions.

The Emaxx user manual [1], suggests that big bumps and rough terrain require a softer suspension with maximum possible suspension travel and ride height while driving on-road requires a lower ride height and firmer, more progressive suspension settings. The more progressive suspension settings helps reduce the following behaviors of the vehicle:

1) Body roll - This is a term used to describe how the body of a vehicle feels based on the overall movement. Higher body rolls lead to more bounciness in a robot.

2) Nose dive during braking - Nose diving is caused when a deceleration results in the front
of the vehicle to point toward the ground.

3) Squat during acceleration - During acceleration, the momentum of the vehicle will be transferred to the rear end and causes the front end to rise slightly. For our experiments, the out-of-the-box configuration is used which allows for firm suspension and low ride height. This setting improves high-speed cornering on smoother terrain by lowering the center of gravity. Body roll, brake dive, and squat are also reduced according to the manual. As a result, the gyroscope readings in the IMU recorded lower amplitudes.

4.2.2 Digital Parameters

As stated earlier, the vehicle is equipped with an XMOS XK-1A [62] microcontroller development board as the embedded controller for this project. The XMOS is programmed to read proprioceptive data and record data readings onto a SD card for later offline analysis. The XMOS XK-1A is a low cost development board based on a single XS-L1 device. It consists of a single XCore, which comprises an event-driven multi-threaded processor with general purpose I/O pins and 64 KBytes of onchip RAM. Figure 4.8 shows the XK-1A microcontroller development board. Using the XMOS reduces complexity of handling...
Figure 4.7: Suspension System of the ground Vehicle. From [1].

multiple I/O streams while simultaneously performing complex computational tasks. [63]. Since exact timing of the sensor readings is required within this project, the XMOS microcontroller proves to be a great choice.

**Sensors**

The IMU and GPS sensors on the ground vehicle are the proprioceptive sensors used throughout this thesis. In the pre-experimental phase of the project, it was assumed that every one-second terrain vibration signature can be associated to a location extracted from the GPS reading but two issues arose with this approach. The first issue was the precision of the GPS localization readings. When the GPS outputs are overlaid on a map and compared
to ground truth data, a $\pm 3m$ error is consistently present. Since terrain IMU readings must be associated with the terrain patch of approximately a 50 cm x 50 cm size, this accuracy is not acceptable. The other issue with using GPS in association with IMU readings is that GPS readings are updated at 4Hz whereas the IMU readings are updated at 200Hz. This makes the association of IMU readings with the locus of the terrain patch impossible within the accuracy requirements. Given these issues, it is decided not to utilize the GPS readings for association purposes but still recorded the readings for global localization.

**IMU**

Inertial sensors are used in navigation. These sensors measure the second derivative of position. An accelerometer measures the inertia force generated when a mass is affected by a change in velocity and a gyroscope measures the rate of rotation. In this project, readings from the gyroscope and accelerometer are a representation of terrain characteristics as the robot traverses different terrain. Attached to the ground vehicle is the 9DOF Razor IMU that incorporates three sensors - an ITG-3200 (MEMS triple-axis gyro), ADXL345 (triple-axis accelerometer), and HMC5883L (triple-axis magnetometer). The outputs of
these sensors are processed by an on-board ATmega328 and output over a serial interface. The IMU board is shown in Figure 4.9. The resolution of the gyroscope on the X, Y, Z Axis is ±2000°/sec full-scale range.

The ADXL345 3-axis accelerometer gives high resolution (13-bit) measurement at up to ±16g. It measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. The accelerometer is used at the ±4g scale and 12-bit resolution which yields a typical 2mg/LSB at a 200 Hz output data rate. Later, it is described how to get a frequency representation of the sampled accelerometer data. The IMU also has a magnetometer which is not used throughout this research.

Figure 4.9: The commercial IMU. We utilize the accelerometer and gyroscope and use the magnetometer only during the first initialization.

4.3 Vibration Analysis on Ground Robot

Now that the digital parameters and mechanical parameters of the ground vehicle are explained, the method for vibration based classification / characterization is presented.
4.3.1 Approach

To capture the vibrations present within the body of a robot, an IMU sensor is mounted to the body. The 3-axis accelerometer measures accelerations up to $\pm 4g$ at the sampling rate of 200 Hz along each of the x, y, z axes. It measures both dynamic acceleration resulting from motion or shock and static acceleration, such as gravity. Each terrain type creates a vibration signal consisting of a series of acceleration values. The acceleration data is parsed into non-overlapping segments, where each segment represents 1 second of robot travel or 200 acceleration and gyroscope samples. Moreover, it is observed that variation in the speed of the vehicle modulates the frequency and amplitude of the measured accelerations over a given terrain, hence it is of utmost importance to train and test the classifier on data sets belonging to approximately the same speeds. Also, to make sure the samples representing a specific terrain are immune to noise, undesired samples get filtered out. To accomplish this, a band pass filter inspired by [18] is implemented to remove undesired low frequency signals associated with vehicle maneuvers such as changing speeds as well as the contribution of gravity on all axes; also high frequency signals, which are caused by the motors on the body of the vehicle are also removed by the band pass filter. The method described in [19] is also applied to remove terrain impulses from the profile that are caused by potholes and are not necessarily a characteristic of the terrain to improve classification accuracy. After pre-processing the signal in the time domain, each 1 second segment is individually transformed to the frequency domain using a Fast Fourier Transform (FFT) and the periodogram of a terrain traversal is derived. Next, each feature is normalized to a mean of 0 and standard deviation 1.
4.3.2 Pre-processing

Removing Inherent Body Vibration

By placing the ground vehicle on a box so that the wheels do not come in contact with the ground, the motors are turned on and the speed is set to 3 levels used for our experiments. IMU readings for 1-minute periods are recorded while the motors are turned on. The IMU readings in this setting show inherent body vibration caused by the motors alone which are totally independent of terrain characteristics. An average of the 1-minute readings is subtracted from the raw data recorded as the robot traverses different terrain at different speeds. This pre-processing step helps separate terrain-only IMU readings from the inherent body vibrations. It must be mentioned that removing inherent body vibrations does not affect terrain classification results discussed later in this chapter. The inherent vibrations are present in all terrain types. Hence, this results in a similar constant offset for all terrain vibrations.

Removing Impulses

One single pothole or a single gravel stone on a terrain are not necessarily representative of the whole underlying terrain type since they happen to affect IMU readings for only a very short period of the 1s segment and are not of a repeating nature. Therefore, the method in [19] is used to detect and remove impulses caused by single occurring events from the recorded 1s segments fed into the classifier. The steps of the method are described below.

1. Calculate the standard deviation of the vibration readings over a short moving window.

2. Calculate the moving average of the standard deviation calculated from step 1, over a longer window.
3. Compare the numbers calculated in step 1 and step 2 and determine whether the number calculated in step 1 is bigger than a heuristic multiple of the number calculated in step 2.

4. If the previous step is true, then this instance was an impulse and hence removed.

It should be noted that this impulse detection method labels transitions from relatively smooth to rough terrains as an impulse too; For example when transiting from dirt to grass or from asphalt to grass, this method detects an impulse. However, the impulse is only detected briefly at the initial terrain transition.

### 4.3.3 Feature Extraction

The raw data collected from robot traversal on different terrains is a sequence of 1-second segments of vibration data as a vector of acceleration and gyroscope measurements, where each segment is made up of 200 acceleration and gyroscope samples. Given a time series of vibration signals \( v = [v_0, v_1, \cdots, v_{199}] \) sampled at a frequency of \( 200Hz \), the first step is to extract features. Figure 4.10 shows some acceleration samples of different terrain in the time domain. As can be seen from a naked eye, asphalt has faster and smaller changes in acceleration compared to grass terrain which has slower and bigger changes in acceleration. In the next section, these behaviors are described by a self-defined parameter and also by a frequency representation of the recorded signals.

#### Self-defined Parameters

By looking at the samples in Figure 4.10, the first thing noticeable is different behavior on periods and amplitudes of positive and negative peaks on each distinct terrain. Acceleration on asphalt has smaller and more frequent positive and negative peaks whereas grass
Figure 4.10: Acceleration on Grass and Asphalt terrain in the time domain. Acceleration on asphalt has smaller and more frequent positive and negative peaks whereas grass terrain has peaks that are spread and larger amplitudes.

terrain has peaks that are spread and have larger amplitudes. To characterize these behaviors, positive and negative peaks are extracted from sensor data obtained over each 1 second window frame. Detecting positive and negative peaks is done by examining change of the sign of derivatives as below:

\[
\frac{X_k - X_{k-1}}{\delta t} > 0, \quad \frac{X_{k+1} - X_k}{\delta t} < 0, \quad \text{Positive peak}
\]

\[
\frac{X_k - X_{k-1}}{\delta t} < 0, \quad \frac{X_{k+1} - X_k}{\delta t} > 0, \quad \text{Negative peak}
\]
After detecting positive and negative peaks from the 1s segment, first and second order statistics of the peaks are calculated.

\[
\mu_{pp} = \frac{1}{n} \sum_{k} X_{pp}(k), \quad \mu_{np} = \frac{1}{n} \sum_{k} X_{np}(k) \\
\sigma^2_{pp} = \frac{1}{n} \sum_{k} (X_{pp}(k) - \mu_{pp})^2, \quad \sigma^2_{np} = \frac{1}{n} \sum_{k} (X_{np}(k) - \mu_{np})^2
\]

\(\mu_{pp}\) is an average of positive peaks and \(\mu_{np}\) is an average of negative peaks. \(\sigma^2_{pp}\) is a variance of positive peaks and \(\sigma^2_{np}\) is the variance of negative peaks. In Figure 4.11, the average of positive peaks of grass terrain, in a sequence of 80 second traversal of grass is shown. Figure 4.12 shows the variance and mean of peaks of the asphalt, dirt, grass and woodchip.

![Grass Sequence Positive peaks mean](image)

Figure 4.11: Amplitude of positive peaks from 1-second vibration signals on a sequence of grass terrain
terrain samples. By analyzing the peak mean and peak variances, repeated similar characteristics within each terrain class is observed and to some extent separated clusters are seen. By analyzing Figure 4.12, it is observed that acceleration data for dirt terrain has a small distribution of peak means and peak variances whereas grass has a larger distribution. Also the distribution of the dirt terrain group is overlapped with the asphalt terrain group. This shows that using only the mean and variance of the peaks, dirt and asphalt terrain are not distinguishable from each other as separate classes but also means according to our defined parameters, they have very similar characteristics. The distribution of woodchip and grass terrain, also overlap each other but are very well distinguishable from dirt and asphalt terrain by their amplitude of peaks. Woodchips also have a slightly higher amplitude than grass terrain as can be seen.

Figure 4.12: Manual feature extraction distribution from 1-second vibration signals on Asphalt, Dirt, Grass, Woodchips terrain
Spectral Analysis

Any signal that has an amplitude varying in time can be represented in the frequency spectrum. According to Fourier analysis any physical signal can be decomposed into a number of discrete frequencies [64]. The statistical average of a signal in terms of its frequency content is called its spectrum.

Suppose that a signal is sampled at N different times (in our case $N = 200$), with the samples uniformly spaced by $\Delta t$ (in our case $5 ms$), giving values $x_n$. The simplest technique to estimate the spectrum is the periodogram, given by the modulus squared of the discrete fourier transform.

\[
S(f) = \frac{\Delta t}{N} \left| \sum_{n=0}^{N-1} x_n e^{-i2\pi nf} \right|^2, \quad -\frac{1}{2\Delta t} < f \leq \frac{1}{2\Delta t}
\]

where $1/(2\Delta t)$ is the Nyquist frequency. The Nyquist frequency is half of the sampling rate of a discrete signal processing system [65]. According to Nyquist theorem, any analog waveform can be captured in digital values and then recreated in an analog form again. In order to achieve this, the sampling rate has to be at least twice the highest frequency inherent in the analog signal. In Figure 4.13, the periodogram of grass terrain and asphalt terrain for a 1-second segment is shown. As can be seen in the figure, the grass terrain has a spread distribution of frequency peaks whereas the asphalt has a narrower distribution of peaks. Also the amplitude of the asphalt frequency components is on average less than the grass frequency components. Now that the frequency representation of different signals is shown, feature extraction for the classifier is required. From the FFT representation of the signal, a feature vector $F$ with a size of 50 is extracted with each element of the vector corresponding to the amplitude at the given frequency. Next, the feature vectors are normalized in the training set such that each feature has mean 0 and standard deviation 1. These feature vectors define the basis of the characterization metrics which are grouped
Figure 4.13: Periodogram representation of a sample asphalt and grass terrain
into clusters meeting our application. For example, if we are sensitive to amplitude more than frequency these vectors are clustered into higher number of bins in the range of our amplitude data while we create lesser bins in the range of frequency changes. The discrete characterization module is used to train a visual classification module that is explained in the next chapter. And eventually in Chapter 6, this module characterizes each incoming image patch into different characteristic categories as will be illustrated. But before that, the goal is to characterize terrain under the ground robot. This is achieved fully autonomously by grouping the terrain into novel and distinct characteristics classes or by first manually hand-labeling a training set of these signals like most of the similar work [13] and then using the classifier to group signals.

4.4 Experiment and Categorization Results

The vehicle is driven around on different terrain such as asphalt, woodchips, dirt and grass at 3 different speed settings to collect 30 minutes worth of training data. The runs are repeated for three speeds (slow: \( \leq 0.5 \) m/s, normal: \( 0.5 - 1 \) m/s and fast: \( \geq 1 \) m/s). Each second of terrain traversal creates a \( 1 \times 200 \) amplitude vector of acceleration samples, equally spaced at 5 milliseconds from one another. Next, as shown in the flow chart of Figure 4.3.2, the pre-processing step is applied to the \( 1 \times 200 \) vectors. During this step, noise is removed and also an impulse detector removes terrain containing pot holes. As for the feature selection step discussed in Section 4.3.3, either a self-defined parameter or spectral analysis on the \( 1 \times 200 \) signal is presented and finally a feature vector is extracted. A label is assigned to each feature vector and feeds into the trainer of the supervised classifier.

This chapter is concluded by presenting the results of the discrete characterization method on a sequence run of 5 minutes on different terrain types. The purpose of this chapter was to assign labels to terrain traversals. These labels, are ground truth character-
Figure 4.14: Flow chart of processing events before categorizing 1-second vibration signals.

The benefits of knowing the ground truth characteristics of terrain, is twofold. First, it enables assigning characteristics labels to images acquired from the visual classifier, and as a result is a pre-training step for the visual classifier. Secondly, it allows evaluating the accuracy of the visual classifier by comparing the results to the ground truth.

Two important results can be drawn from Figure 4.15. First, it is seen that a certain terrain class does not possess only one type of terrain characteristic. The arrows in the figure show transitions within any terrain class. The other important implication is that these instances, where the terrain characteristic of the terrain class change, are not sparse instances. They happen in a group of consecutive samples so as a result, different terrain categories within a certain terrain class, are clustered next to each other. This finding is important since it gives room for the visual classifier discussed in Chapter 6, to learn association between visual features and terrain characteristics in a incremental sequential way.
Figure 4.15: Ground truth Characteristics of different terrain traversed in a sequence of 5 minutes. Arrows indicate sections of the traversed terrain in which characteristics are different from the norm.
Chapter 5

Exteroceptive Terrain Classification

5.1 Introduction

Exteroceptive terrain classification is the process of assigning class labels to terrain patches based on data collected from any exteroceptive sensor. The exteroceptive sensor in our case is a monocular GoPro camera attached beneath a flying DJI phantom shown in Figure 5.1. The goal is to deploy a vision-based classifier that when presented with visual features associated with a terrain patch, identifies which of the known classes (i.e. classes for which the classifier has \textit{a priori} training data) appears most similar to the newly observed patch. The terrain is represented as a set of feature vectors derived from the color and the visual texture of the acquired image. These features are all extracted from RGB color images captured on an aerial vehicle while in motion. Finally, the SVM classifier is evaluated with different sets of kernels and different sets of feature representations. Also, in situations where an unexpected terrain patch is captured and is not similar to any of the training set data, the image patch is classified as unknown terrain. The vision-based terrain classification operates on visual features derived from color and visual texture. SVM classifiers for each visual feature type are used to predict the likelihood a particular terrain
patch belongs to any given terrain class. In the following sections a detailed analysis of each module of the visual classifier is proposed.

5.2 Approach

The video recorded by a GoPro camera is later processed offline into separate images, but keeping in mind that the long term goal is to be able to perform terrain characterization in real-time. Offline analysis of the video frames along with proprioceptive sensor readings of the ground vehicle requires a reference time \( t = t_0 \) set for both the video and IMU measurements. The system is designed to learn the relationship between proprioceptive measurements of the terrain which are known only at the moment of traversal by the ground vehicle and the visual appearance of the terrain captured from an aerial vehicle. To accomplish this the first step is to find corresponding image patches and proprioceptive labels. Since the terrain underneath the ground robot can not be seen from an aerial vehicle at time of traversal, a method involving shifting in time is required assuming a ground vehicle traverses terrain a fixed distance in front of it \( \Delta t \) seconds later. In other words, if an image from the aerial vehicle is captured with the ground vehicle present in the frame, it is assumed the image patch in front of the ground vehicle will be traversed \( \Delta t \) seconds later;
$\Delta t$ is calculated based on the assumption that the ground vehicle is moving in a straight line at a certain constant velocity. Therefore, the image captured at time $t$ containing the terrain patch ahead of the ground vehicle, corresponds to the proprioceptive sensor recordings at time $t + \Delta t$. In the following section the detection of the ground vehicle from an image is described. Next, extracting the image patch in front of the vehicle that best corresponds to the proprioceptive measurements discussed in Chapter 4 is discussed. An overhead view of the ground vehicle captured from the aerial vehicle with the region of interest extracted for correspondence is shown in Figure 5.2.

![Figure 5.2: Overhead view captured from aerial vehicle with the squares on top of the ground vehicle for scale references](image_url)
5.2.1 Preprocessing

Blur Removal

Blurry pictures are a combination of motion (camera motion or subject motion) and slow shutter speeds. Shutter speed refers to the amount of time the sensor is exposed to the scene. When a camera creates an image, that image does not represent a single instant of time. Because of technological constraints, the image represents the scene over a period of time. Most often, this exposure time is short enough that the image captured by the camera appears as an instantaneous moment. Camera motion is due to shaking of the camera or due to translational motion of the camera itself (in this case, due to the motion of the aerial vehicle it is mounted to. To overcome the blur problem in this framework, both a hardware solution and a software solution are discussed.

Hardware Solution

Shutter Speed

A video is a collection of images captured at a certain frequency or frame rate. Frame rate refers to the number of individual frames that comprise each second of a video, in units of frames per second (FPS). As discussed earlier, one factor that increases image blur is the shutter speed of a camera. The GoPro has the following options for its video recording resolution and FPS:

1080p: 1920 x 1080, 30FPS
960p: 1280 x 960, 48FPS + 30FPS
720p: 1280 x 720, 60FPS + 30FPS

In commercial cameras if the video is captured at 30 FPS, the shutter speed will be approximately 1/60th of a second. On one hand images with higher resolution are preferred
to be captured by the camera since they capture details of the texture at higher resolutions but on the other hand less image blurriness is also important. As shown above, lower FPS results in higher resolution (1080p) but as a result longer shutter speeds and hence higher possible motion blur. Also, GoPro cameras automatically set their exposure based on the amount of light available to their sensor. The GoPro compensates for lack of light by slower shutter speeds to ensure a well balanced picture. This means that brighter scenes result in fast shutter speeds and hence less blur. Overall, in the pre-experiment analysis of this research it was shown that shutter speed and image blurriness play a more important role in classification results compared to the resolution of the images. Therefore, the 960p mode for the GoPro camera is selected due to higher shutter speeds compared to the 1080p resolution mode.

**Gimbal**

A quadcopter gimbal is a 3-axis camera stabilization and anti-vibration device. It uses brushless motors to adjust the position of the camera in three axes. In Figure 5.3, the gimbal and camera assembly for a DJI Phantom 3 quadcopter are shown. The gimbal has a controller encased in a circuit board. This controller sends out commands to the brushless motors that stabilize the camera. Modern aerial vehicles often include a gimbal such as this, and the blurriness of images is significantly reduced by using the DJI phantom with a gimbal.

![Gimbal and Camera](image)

Figure 5.3: Camera Gimbal that is attached to the DJI phantom
Software Solution

Image sharpness is an important characteristic when detecting texture and hence estimate roughness of a terrain image patch. Sharper images will have more texture compared to blurry images. Hence, removing blurred images from the training image set of the classifier increases accuracy results. One approach to detect blurriness in images is to compute the FFT of an image and analyze the result. As a rule of thumb, if there is a lower amount of high frequency components than usual in an image, this implies a blurry image. The shortcoming of this approach is that defining the threshold of “high” is heuristic and depends on the terrain type that is captured by the camera. For example, smoother terrains such as dirt, even without motion blur have lower frequency components compared to a woodchip-heavy terrain that has inherent motion blur. Therefore, detecting blurry images is a heuristic method that does not fit within the autonomous framework of this work. For example, if it is known a priori that an image of a grass terrain is captured, then it is possible to set a threshold for detecting whether the captured image is blurry or not. But since this isn’t the case for the proposed autonomous module, the approach is abandoned and only utilized for post experiment analysis.

Ground Vehicle Detection in a Frame

Since we assume the image patch in front of a ground vehicle is traversed later in its trajectory, a method to detect the ground vehicle in any image frame and then extract a box a certain distance in front of the vehicle is required. Therefore, as the first step, detection and localization of the ground robot is necessary in any cluttered scene. Given a reference image of the ground robot, a template matching algorithm is used to detect the vehicle in an image frame. Many template matching [66] approaches exist for object recognition but for this problem, the module must consider possible scale change and in-plane rotation of
the target image. This is due to the fact that the aerial vehicle changes altitude during its
course of flight and hence the scale of a ground vehicle changes in the captured images.
The system uses the SIFT descriptor which is immune to both scale and in plane rotations.
This means that if a feature is captured from different perspectives and different scales in
two images, their SIFT descriptor for salient points will be almost the same thus solving
the correspondence problem. A step by step algorithm for detecting a vehicle is presented
below:

Algorithm 1 Vehicle detection in a given frame
1. Detect feature points in both the target image and reference image
2. Extract feature descriptors at the points of interest in both images (SIFT features)
3. Find point matches including possible outliers
4. Locate the object in the scene using presumed matches. This is done by calculating
   the transformation relating the matched points, while eliminating outliers. The outliers are
   removed using a RANSAC [67] algorithm approach

By the transformation calculated in step 4, the algorithm is able to detect the orientation
of the vehicle in the image frame. This is vital since extracting the image patch in front
of the ground vehicle a certain distance ahead of the vehicle requires knowledge on the
orientation and location of the robot within the frame.

Image Patch Extraction

Assuming a constant velocity $V (m/s)$ for the ground vehicle, the vehicle travels a
certain distance $x (m)$ calculated by $x = V \times \Delta t$ in $\Delta t$ time. In order to extract the image
patch that will be traversed by the ground robot at time $\tau = t + \Delta t$, the image patch that
is $x$ meters in front of the vehicle at time $\tau = t$ is extracted. In a digital image, distance is
measured in terms of pixels. So a conversion between $x$ meters and the number of pixels in
an image is necessary. In order to achieve this relation, squares are attached on top of the ground vehicle with known dimensions as shown in Figure 5.2. This gives a reference for scale in the captured images. For example, if a patch $40\text{cm}$ ahead of the vehicle is required for extraction, knowing that the side of the square is $20\text{cm}$ long and is 30 pixels in our image, then a patch 60 pixels in front of the ground vehicle should be extracted. Once the terrain patches are extracted from any given frame, the options of descriptors and classifiers applied to them are discussed in the following sections.

### 5.2.2 Feature Extraction

As with any classification module, the first step is to extract a feature vector from the image of interest. A thorough discussion of different features is presented in this section. A feature vector from a $128 \times 128$ image can be as simple as the average R, G, B values of all the $128 \times 128$ pixels, represented as $[\bar{R}, \bar{G}, \bar{B}]$ after a scaling factor. This simplistic $1 \times 3$ vector representation of a terrain proves to actually yield good results on color distinguishable terrains like grass and asphalt during sunlight but fails significantly in distinguishing, (for example), a grass in shade from asphalt in shade. This example shows that feature descriptors for any classification algorithm need to be selected based on the requirement criteria. In the following section we discuss feature vectors that best describe both color and textural data in an image. The goal is to classify five separate terrain with the highest classification results. The terrain types are Asphalt, Gravel, Grass, Woodchips and Dirt as shown in Figure 5.4. Evaluation of the different feature vectors are described next.

**Color**

Color data is directly available from the GoPro camera as red, green, and blue (RGB) intensities for each pixel. In Figure 5.4, next to each terrain type the RGB color histogram
of the images is given. The color histogram is a representation of the distribution of colors in an image and represents the number of pixels that have colors in each of a fixed list of color ranges. However, illumination intensity affects all three values in a raw RGB representation, which leads to poor classification results. In an outdoor environment such as the experimental environment of this research, the system has to work under greatly varying light conditions. In [68], five types of light variations are described as summarized below:

- **Light intensity change**

\[
\begin{pmatrix}
R_L \\
G_L \\
B_L
\end{pmatrix}
= 
\begin{pmatrix}
a & 0 & 0 \\
0 & a & 0 \\
0 & 0 & a
\end{pmatrix}
\begin{pmatrix}
R \\
G \\
B
\end{pmatrix}
\]

- **Light intensity shift**

\[
\begin{pmatrix}
R_L \\
G_L \\
B_L
\end{pmatrix}
= 
\begin{pmatrix}
R \\
G \\
B
\end{pmatrix}
+ 
\begin{pmatrix}
c \\
c \\
c
\end{pmatrix}
\]

- **Light intensity change and shift**

\[
\begin{pmatrix}
R_L \\
G_L \\
B_L
\end{pmatrix}
= 
\begin{pmatrix}
a & 0 & 0 \\
0 & a & 0 \\
0 & 0 & a
\end{pmatrix}
\begin{pmatrix}
R \\
G \\
B
\end{pmatrix}
+ 
\begin{pmatrix}
c \\
c \\
c
\end{pmatrix}
\]

- **Light color change**

\[
\begin{pmatrix}
R_L \\
G_L \\
B_L
\end{pmatrix}
= 
\begin{pmatrix}
a & 0 & 0 \\
0 & b & 0 \\
0 & 0 & c
\end{pmatrix}
\begin{pmatrix}
R \\
G \\
B
\end{pmatrix}
\]
• Light color change and shift

\[
\begin{pmatrix}
R_L \\
G_L \\
B_L
\end{pmatrix} =
\begin{pmatrix}
a & 0 & 0 \\
0 & b & 0 \\
0 & 0 & b
\end{pmatrix}
\begin{pmatrix}
R \\
G \\
B
\end{pmatrix} +
\begin{pmatrix}
c_1 \\
c_2 \\
c_3
\end{pmatrix}
\]

To reduce the effects of illumination level on classification, a hue, saturation, and value (HSV) representation of color is used hereafter. HSV separates luma, or the image intensity, from chroma or the color information. This is very useful in many applications. For example, it provides robustness to lighting changes, and shadows.

Below is the HSV colorspace formula conversion as used in OpenCV [69] which is also implemented in Matlab. In case of 8-bit images, R, G, and B are converted to the floating-point format and scaled to fit the 0 to 1 range for later processing.

\[
V \leftarrow max(R, G, B)
\]

\[
S \leftarrow \begin{cases} 
\frac{V - min(R, G, B)}{V} & \text{if } V \neq 0 \\
0 & \text{otherwise}
\end{cases}
\]

\[
H \leftarrow \begin{cases} 
60(G - B)/(V - min(R, G, B)) & \text{if } V = R \\
120 + 60(B - R)/(V - min(R, G, B)) & \text{if } V = G \\
240 + 60(R - G)/(V - min(R, G, B)) & \text{if } V = B
\end{cases}
\]

\[0 \leq V \leq 1, \ 0 \leq S \leq 1, \ 0 \leq H \leq 360.\]
Figure 5.4: SIFT features and RGB Histogram of different image patches
SIFT Features

Scale Invariant Feature Transform [2] is a method that combines detection and description of points of interest for further use for object recognition and image classification tasks. SIFT creates a scale-space by progressively applying different degrees of Gaussian blur to the input image at various scales. The interesting keypoints are detected by the Laplacian of Gaussian (LoG) [70] which is approximated efficiently by computing the Difference of Gaussians (DoG) [71] at two consecutive scales. To calculate keypoint locations, SIFT detects the local extrema at these DoG images by comparing each pixel with all its neighbors including those at consecutive scales. To achieve rotation invariance, each keypoint is assigned a consistent orientation based on local image properties. Gradient magnitudes and orientations at a 4 x 4 subregion around each keypoint are calculated and weighted by their distance to further form a histogram of orientations with 8 bins. Hence, the SIFT descriptor becomes a vector in $128(4 \times 4 \times 8)$ dimensions as shown in Figure 5.5. The SIFT descriptor is an attractive choice in our classifier module since the aerial vehicle has change in altitude during its flight and also different orientations on different kind of terrain lead to scale changes and also rotations in the images. In Section 5.2.2 the combination of SIFT descriptors with another algorithm is discussed.

Figure 5.5: Figure showing gradient orientation histogram descriptors (from [2])
**SURF Features**

Speeded Up Robust Features (SURF) [22] is based on a similar concept as SIFT, but faster to calculate. SURF creates a scale-space by applying box filters of varying size to the input image. In SURF, square-shaped filters are used as an approximation of gaussian smoothing. SURF uses a blob detector based on the Hessian matrix to find points of interest. The determinant of the Hessian matrix is used as a measure of local change around the points. Points are chosen where this determinant is maximal. The interest points are found in different scales. Instead of gradients, a distribution of Haar-wavelet [72] responses around the neighborhood of keypoints is used in SURF. Similar to SIFT, a dominant orientation is assigned to each keypoint to achieve rotation invariance. The descriptor is calculated at a $4 \times 4$ subregion around keypoints. Within each subregion, Haar-wavelet responses are computed. The feature vector for the corresponding subregion is finally calculated by considering the sum of Haar-wavelet responses with their absolute values both in horizontal and vertical directions. This results in a feature vector with 64 dimensions. Just like the extracted SIFT features, the SURF descriptor is later combined with another algorithm explained in Section 5.2.2.

**Visual Texture Classification**

Haralick considers a texture as an “organised area phenomenon” which is decomposed into “primitives” having specific spatial distributions [73] or in simpler terms visual textures are related to local spatial variations of simple stimuli like color, orientation and intensity in an image. Texture classification assigns a given texture to some texture classes [74, 75]. The majority of classification methods involve a two-stage process. The first stage is feature extraction, which yields a characterization of each texture class in terms of feature measures. It is important to identify and select distinguishing features that are invariant to
irrelevant transformation of the image, such as translation, rotation, and scaling. Ideally, the quantitative measures of selected features should be very close for similar textures. Since we assume texture plays an important role in predicting roughness of a given terrain, therefore in this section we analyze different textural descriptors and their classification. In Leung and Malik [76] for feature extraction, the method applies a filter bank onto the training textures for each material with known viewpoints and illumination. A k-mean clustering algorithm is deployed to identify k clusters from the vector space concatenating all filter responses. Cluster centers are the representative textons of each material and act as feature descriptors. The textons of all materials together create a global texton dictionary, so that each material is represented by a particular probability density function. For any texture to be classified, the distribution of texton frequencies with respect to the texton dictionary is computed. This representation can be seen in Figure 5.6.

**Bag of Visual Words**

The idea of the bag of visual words technique is that an image can be treated as a document where it is made up of visual words. Just like with analyzing documents in which a histogram of word usage in a document is a fair descriptor of the document, histogram representation based on independent features is a good descriptor of an image. Defining words in images includes following these three steps [24]:

Feature detection
Feature description
Codebook generation.

To generate vocabulary or “codewords”, the k-means clustering algorithm explained in the machine learning background chapter is used. A codeword can be considered as a representative of several similar patches. The k-means problem is to find an integer number
of centers that best describe groupings of descriptor data. More formally: let $k \in \mathbb{Z}$ be the desired number of clusters and let $X \subset \mathbb{R}^M$ be the set of descriptors. The goal is to find $k$ centers (descriptors) in $C \subset \mathbb{R}^M$ that minimize $\phi$ in Equation 5.1.

$$\phi = \sum_{x \in X} \min_{c \in C} \|x - c\|^2 \quad (5.1)$$

One simple method to create codewords is performing k-means clustering over all the vectors. [76] Codewords are then defined as the centers of the learned clusters. Thus, each patch in an image is mapped to a certain codeword through the clustering process and the image can be represented by the histogram of the codewords. The problem in [77] with using this approach is that SIFT / SURF features do not record relative spatial placement of the features. Once a visual vocabulary has been created each image is described by a word frequency vector. First, a vocabulary word $v_i$ is assigned to each descriptor $d_j$ in the image by choosing the $i$ that minimizes $\|v_i - d_j\|$. This process essentially approximates each descriptor with the vocabulary word that has the nearest Euclidean distance. Each word is then counted and the frequency of each word is stored in a vector $q \in \mathbb{Z}^n$ where $n$ is the number of visual words. In Figure 5.6, a BOVW representation of simple textons is presented on simple texture patterns. Due to randomness of textural patterns present in outdoor settings, the texton representation is not the choice of this work and instead a BOVWSIFT and BOVWSURF representation is made. Images in the training set each have a corresponding frequency vector and are used to later train the SVM classifier in section 5.2.3 as shown by Figure 5.7.

**Local Binary Patterns**

Local Binary Patterns (LBP) [78] is found to be a powerful feature for texture classification. It is a local descriptor computed by thresholding the neighborhood of a pixel and then
using the bit pattern produced as a descriptor. The operator assigns a label to every pixel of an image by thresholding the $3 \times 3$ neighborhood of each pixel with the center pixel value and considering the result as a binary number. Then, the histogram of the labels are used as a texture descriptor. An illustration of the basic LBP operator is shown in Figure 5.8. To be able to deal with textures at different scales, illumination and orientation, the LBP operator is later extended [79]. Defining the local neighborhood as a set of sampling points evenly spaced on a circle centered at the pixel to be labeled allows any radius and number of sampling points. The notation $(P, R)$ is used for pixel neighborhoods which means $P$ sampling points on a circle of radius of $R$. Figure 5.9 shows an example of circular neighborhoods.

**Gray Level Co-occurrence Matrix**

The GLCM [80], is a square matrix that reveals certain properties about the spatial distribution of the gray-levels in the texture image. It shows how often a pixel value known
as the reference pixel with the intensity value $i$ occurs in a specific relationship to a pixel value known as the neighbor pixel with the intensity value $j$. Each element $(i, j)$ of the matrix is the number of occurrences of the pair of pixel with value $i$ and a pixel with value $j$ which are at a distance $d$ relative to each other. In this thesis, rotation invariant features are of interest. Therefore, four possible spatial relationships ($0^\circ$, $45^\circ$, $90^\circ$, $135^\circ$) are accommodated into the GLCM presentation of features.
Figure 5.9: LBP with circular \((8, 1), (16, 2),\) and \((8, 2)\) neighborhoods.

\[
C_{\Delta x, \Delta y}(i, j) = \sum_{p=1}^{n} \sum_{q=1}^{m} \begin{cases} 
1, & \text{if } I(p, q) = i \text{ and } I(p + \Delta x, q + \Delta y) = j \\
0, & \text{otherwise}
\end{cases}
\]

where \(i\) and \(j\) are the image intensity values of the image, \(p\) and \(q\) are the spatial positions in the image \(I\) and the offset \((\Delta x, \Delta y)\) depends on the direction used \(\theta\) and the distance at which the matrix is computed \(d\). An image along with its GLCM is presented in 5.10.

In order to estimate similarity between different GLCMs extracted from different images,
Haralick [82] proposes 14 statistical features to be extracted. In the context of this work, only four of these features are selected. The description of the four most relevant features is as follows.

Energy, also called Angular Second Moment (ASM) is a measure of textural uniformity of an image. It has a maximum where gray level distribution has either a constant or a periodic form. For example, if the P matrix contains a large number of small entries, the energy feature will have a smaller value.

\[
\sum \sum P_d^2(i, j)
\]

Entropy measures the disorder of an image.

\[
- \sum \sum P_d(i, j) \log P_d(i, j)
\]

When the image is not texturally uniform many GLCM elements have very small values and hence the Entropy is low.

Contrast measures the amount of local variations in an image.

\[
\sum \sum (i - j)^2 P_d(i, j)
\]

Inverse Difference Moment (IDM):

\[
\sum \sum \frac{P_d(i, j)}{|i - j|^2}, \ i \neq j
\]
This measures image homogeneity. When most of the occurrences in GLCM are concentrated near the main diagonal, this parameter reaches its highest value.

5.2.3 Classification

Now that the features are extracted from previous steps, they are fed into the training system of a classifier. The vision-based terrain classifier in this step is a supervised classifier using an SVM approach. The SVM is implemented using the open-source library LIBSVM [36], as was used for the vibration-based terrain classifier in the previous chapter. For vision-based classification, linear or low order polynomial kernels are appropriate since they enable fast classification as opposed to RBF kernels without undergoing any overfitting in the process. In this section, a linear kernel is used, with the cost factor C optimized by cross validation over a subset of images used for training.

SVM

Training

The training goal of a linear SVM is to find a hyperplane that provides the maximum margin of separation between classes. Let the training set consist of \( k \) points \((q_i, y_i)\) indexed by \( i \) where \( y_i \) determines if the descriptor vector belongs to the given class. If it belongs to the class then \( y_i = 1 \) otherwise \( y_i = 1 \). Any hyperplane that separates the data can be described by all vectors, \( h \), that satisfy \( w \cdot h - b = 0 \) where \( w \) is the vector normal to the hyperplane and \( b \) is the scalar bias. The solution for the optimal hyperplane is achieved through solving Equation 5.2. After feeding the labeled images of the training phase along with the corresponding descriptors, LIBSVM’s function SVMtrain is used to
find the hyperplane in Equation 5.2 which later will be used to predict the label of future descriptor vectors.

\[
\min J_1(w, \xi) = \frac{1}{2} w^T w + c \sum_{i=1}^{N} \xi_i
\]

Subject to
\[
\begin{cases}
    y_i [w^T \phi(x_i) + b] \geq 1 - \xi_i, & i = 1, \ldots, N \\
    \xi_i \geq 0, & i = 1, \ldots, N
\end{cases}
\]  \hspace{1cm} (5.2)

\(c < 0\) is the penalty parameter of the error term.

Labels

SVMs are inherently two-class classifiers but for classifying terrain into 6 groups with one being the unknown class requires a multiclass SVM. Also, the standalone visual terrain classifier is a supervised classification framework where the labels of the training data are hand labeled manually by us. During the course of data collection the altitude of the aerial vehicle changed and thus affected the resolution of the images captured. In order to have a fair evaluation of the classifier and descriptors, all the images captured from the aerial vehicle are grouped into 3 separate altitudes. Images captured from 8m - 10m or above are called group 1, images captured from 6m - 8m are called group 2 and finally images captured from 3m - 6m are referred to as group 3. For the training step, around 200 images per each terrain class and per each altitude group is captured. They are all hand-labeled and fed along with their corresponding descriptors explained in the previous sector into the trainer to find the separating hyperplanes. The descriptors assessed are the SIFTBOVW, SURFBOVW, LBP, GLCM and HSV.
Model Parameters

An SVM classifier only returns the predicted label of a test vector. However, LIBSVM [36] additionally offers the possibility to obtain class probabilities. So for each class the probability of the test vector belonging to that class is also returned and used in our analysis of descriptors. The performance of the SVM model depends on the parameters $C$ and $\epsilon$ and $\gamma$ in case of using a RBF kernel. The explanation of these parameters was given in the machine learning background section. Until now, there is no precise method to find the optimal solutions of the values of these parameters. The common method is to build a parameter space with all the alternative parameters of the model and finding the optimization in this space with the target of minimizing the error between the output results and the ground truth. The grid search along with cross validation is the method used for finding the appropriate parameters. After thorough testing the best $(C, \gamma)$ is $(2^3, 2^{-5})$ with the cross-validation rate of 78.5%.

5.3 Results

In this section, the results of different descriptors and best classifiers performed on the images captured from the aerial vehicle is presented. As mentioned in section 5.1, a DJI phantom fitted with a downward facing GoPro camera attached to a gimbal is used. The vehicle is flown around a park interacting with five visually different terrain types. The results from using the LBP, GLCM, SIFTBOVW, SURFBOVW and HSV colorspace descriptors along with the SVM classifier are discussed. The true positive accuracy rate of the entire dataset and also a confusion matrix is reported. The tables in the next pages present accuracy results of the five descriptor approaches on the five terrain type images captured from group 3 altitudes. Here, a 10-fold cross-validation is used to verify the
results. As can be seen from the tables below there is no certain descriptor that works best for all types of terrain. For example, the SIFTBOVW approach classified the images containing leaves on a dirt terrain as asphalt. After thorough scrutiny it is realized that this is due to the fact that major SIFT descriptors extracted from that specific type of image patch are extracted mostly from the leaves rather than the dirt and hence the classifier looks for finding the correspondence between the leaves SIFT features and the trained images. It appears that the training set for asphalt had more images with leaves and hence is selected as the predicted class for the dirt terrain. On the other hand, SIFTBOVW performed better than other descriptors to differentiate between woodchips and leaves on dirt and asphalt.

<table>
<thead>
<tr>
<th>SVM classifier using HSV on group 3 images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Woodchips</td>
</tr>
<tr>
<td>Woodchips</td>
</tr>
<tr>
<td>Gravel</td>
</tr>
<tr>
<td>Grass</td>
</tr>
<tr>
<td>Dirt</td>
</tr>
<tr>
<td>Asphalt</td>
</tr>
</tbody>
</table>

Table 5.1: SVM classification results using HSV colorspace as features for group 3 images

<table>
<thead>
<tr>
<th>SVM Classifier using LBP features on group 3 images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Woodchips</td>
</tr>
<tr>
<td>Woodchips</td>
</tr>
<tr>
<td>Gravel</td>
</tr>
<tr>
<td>Grass</td>
</tr>
<tr>
<td>Dirt</td>
</tr>
<tr>
<td>Asphalt</td>
</tr>
</tbody>
</table>

Table 5.2: SVM classifier results using LBP features on group 3 images

In the following graphs shown in Figure 5.11 to Figure 5.13, the accuracy results of performing classification using different descriptors on the three different image groups are shown. Unlike the confusion matrix, this representation does not show which terrain is misclassified with which terrain but it gives an overall accuracy result of the descriptors on
SVM Classifier using GLCM features on group 3 images

<table>
<thead>
<tr>
<th></th>
<th>Woodchips</th>
<th>Gravel</th>
<th>Grass</th>
<th>Dirt</th>
<th>Asphalt</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Woodchips</td>
<td>153</td>
<td>40</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>76%</td>
</tr>
<tr>
<td>Gravel</td>
<td>20</td>
<td>157</td>
<td>7</td>
<td>7</td>
<td>9</td>
<td>78%</td>
</tr>
<tr>
<td>Grass</td>
<td>5</td>
<td>2</td>
<td>147</td>
<td>28</td>
<td>18</td>
<td>73%</td>
</tr>
<tr>
<td>Dirt</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>140</td>
<td>59</td>
<td>70%</td>
</tr>
<tr>
<td>Asphalt</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>89</td>
<td>106</td>
<td>53%</td>
</tr>
</tbody>
</table>

Table 5.3: SVM classifier results using GLCM features on group 3 images

SVM Classifier using SIFTBOVW on group 3 images

<table>
<thead>
<tr>
<th></th>
<th>Woodchips</th>
<th>Gravel</th>
<th>Grass</th>
<th>Dirt</th>
<th>Asphalt</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Woodchips</td>
<td>185</td>
<td>4</td>
<td>11</td>
<td>0</td>
<td>1</td>
<td>92%</td>
</tr>
<tr>
<td>Gravel</td>
<td>18</td>
<td>162</td>
<td>2</td>
<td>7</td>
<td>11</td>
<td>81%</td>
</tr>
<tr>
<td>Grass</td>
<td>11</td>
<td>2</td>
<td>151</td>
<td>25</td>
<td>11</td>
<td>75%</td>
</tr>
<tr>
<td>Dirt</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>132</td>
<td>66</td>
<td>65%</td>
</tr>
<tr>
<td>Asphalt</td>
<td>3</td>
<td>6</td>
<td>0</td>
<td>70</td>
<td>121</td>
<td>60%</td>
</tr>
</tbody>
</table>

Table 5.4: SVM Classification results using SIFTBOVW on group 3 images

different terrain. By looking at Figure 5.11 to Figure 5.13 whereby the accuracy results of different descriptors are shown at different altitudes, it is shown that at lower resolutions the pure texture classifiers such as LBPs perform better than the SIFT and SURF counterpart but at higher resolutions, BOVWSIFT provides a better result. Also, as can be seen from the figures, altitude has small effect on classification results of images described by the HSV descriptor. The shortcoming of the HSV descriptor is that it fails to classify specific objects on a given terrain such as leafs on the grass. Moreover, it is seen from the figures that texture descriptors fail to give high classification results for asphalt and dirt in images taken at higher altitudes. This is mostly due to blurring effects and also loss of resolution of image patches with higher altitudes.
Table 5.5: SVM Classification results using SURFBOVW on group 3 images

<table>
<thead>
<tr>
<th></th>
<th>Woodchips</th>
<th>Gravel</th>
<th>Grass</th>
<th>Dirt</th>
<th>Asphalt</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Woodchips</td>
<td>175</td>
<td>6</td>
<td>18</td>
<td>1</td>
<td>0</td>
<td>87%</td>
</tr>
<tr>
<td>Gravel</td>
<td>28</td>
<td>142</td>
<td>2</td>
<td>12</td>
<td>16</td>
<td>71%</td>
</tr>
<tr>
<td>Grass</td>
<td>14</td>
<td>2</td>
<td>140</td>
<td>31</td>
<td>13</td>
<td>70%</td>
</tr>
<tr>
<td>Dirt</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>120</td>
<td>88</td>
<td>60%</td>
</tr>
<tr>
<td>Asphalt</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td>73</td>
<td>119</td>
<td>60%</td>
</tr>
</tbody>
</table>

Figure 5.11: Accuracy results of images captured from 3-6 meters (Group 3)
Figure 5.12: Accuracy results of images captured from 6-8 meters (Group 2)

Figure 5.13: Accuracy results of images captured from 6-8 meters (Group 1)
Chapter 6

Self-Supervised Framework

In this chapter, the findings of Chapters 4 and 5 are combined to make the self-supervised framework. In Chapter 4, terrain was categorized from every 1-second ground traversal into a discrete set of characteristic metrics. Later, in Chapter 5, terrain images were classified into semantic classes using a combination of descriptors with a supervised classifier. The classifier in the exteroceptive module was trained offline by human-labeled images. Now in this chapter, the characterization results of the proprioceptive module are used as labeling inputs to the exteroceptive classifier in an incremental fashion as shown in Figure 6.1. The advantage of this proposed framework is twofold. First, it eliminates the need for manual training, hence the name “self-supervised” stems from here. Secondly, in unknown environments which no a priori knowledge of the environment is known, the visual classifier learns the association between proprioceptive characteristics and visual features of any image patch as it traverses the unknown terrain. In the following sections an overview of the “self-supervised” online learning algorithm modules and parameters are presented. It is discussed how it differs from the standalone visual classifier discussed in Chapter 5. Then a comparison between the accuracy results of a traditional offline “supervised” classifier with the “self-supervised” classifier is demonstrated.
It must be mentioned that the result of the adaptive self-supervised framework is scene dependent, meaning that results vary based on a variety of parameters with one of them being the frequency of transitions among different terrain types. Overall, in natural outdoor settings, the results of the self-supervised online learning classification framework show improvements compared to the traditional supervised classifier. The scenarios which had the highest gains in terms of classification results using the self-supervised method compared to the standalone supervised method are analyzed. Also, the scenarios in which the self-supervised online learning approach performs poorer compared to the standalone supervised visual classifier counterpart are discussed.

Figure 6.1: Self Supervised characterization module

The standalone visual classifier presented in Chapter 5, lacks an online learning framework. The training phase of the exteroceptive standalone module is performed offline. This means that during an offline phase prior to experiments, the training set consisting of visual features with associated labels is fed into the classifier as \textit{a priori} knowledge for training the classifier. As discussed earlier, the proprioceptive characteristics of a certain
visual terrain class are not consistent. For example, a dead grass terrain upon traversal induces small amplitude vibrations with higher frequencies on the ground vehicle, while a few meters away another dead grass terrain patch induces larger amplitude vibrations with lower frequencies upon traversal. As a result, although both have similar visual cues and appearance, they have different proprioceptive characteristics labels. The image patches in Figure 6.2 show examples in which the traditional supervised classifier is unable to assign correct terrain characteristics or sometimes assigns an “unknown” label to the patches. The “unknown” label is due to the fact that there is not enough samples in the training set that have similar visual feature representations to these image patches. In the case of wrong characterization, similar visual features do not necessarily mean similar proprioceptive characteristics. This is the case for both sparse leaves and dense leaves on the terrain as shown in Figure 6.2c and Figure 6.2e. In these instances, the characteristics of the terrain are not changed from the underlying terrain which is grass but the standalone classifier does not recognize the underlying terrain. The same applies to situations in which there are leaves on asphalt as shown in Figure 6.2f. In all these situations although the visual features are similar since the leaves are the dominant feature in the image patches, the proprioceptive terrain characteristics are closer to the underlying terrain characteristics meaning that they should be close to the characteristics of asphalt and grass. From the observations in Figure 6.2, two observations are made.

1) An “online learning” module is required to learn the association between images and their characteristics “on-the-fly” since there are always instances in which the visual classifier has no prior knowledge about.

2) The learning module needs to be adaptive. This means that sometimes previous associations need to be unlearned and learned again. This is the case for dense leaves on grass and dense leaves on asphalt.
If there is no adaptive learning in the system, the framework associates any image patch with dense leaves to the characteristics of grass terrain since that is the characteristics learned from previous interactions. This causes wrong characterization results for instances with leaves on asphalt. To tackle the issues above, the adaptive self-supervised online learning framework is proposed. The framework modules are shown in Figure 6.3. As can be seen, the framework is a combination of the proprioceptive classifier and the exteroceptive classifier discussed in previous chapters.

6.1 Framework

6.1.1 Association of Vibrations to Images

The aerial vehicle flies above the ground vehicle during the training phase and captures time stamped images taken from a monocular camera consisting of the ground vehicle and the terrain ahead of it. The framework needs to assign characterization metrics calculated from Chapter 4 to each segment the ground robot traverses and label image patches acquired through the aerial vehicle as it goes by. The image patches taken at time $\tau = t$ are sectors the ground robot will traverse at time $\tau = t + \Delta t$ and log the IMU values along the x, y and z axis. Assuming the ground robot does not have variations in speed $v$ during its coarse then a $64 \times 64$ pixels image patch $\Delta x = \Delta t \times v$ meters ahead of the ground robot is extracted. Hence, from each image the ground robot is detected by a feature matching algorithm and its position is estimated by running RANSAC on the matched features. After locating the robot in the image, the orientation is determined and finally a square patch with the desired orientation and distance in terms of pixels is extracted in front of the ground robot. This patch feeds into a feature extraction module and later the feature vector is as-
Figure 6.2: Examples that resonate the necessity for a self-learning classifier
Figure 6.3: Self Supervised Framework algorithm - a) Pre-processing images of the visual classifier. b) Pre-processing vibration signals. c) Feature extraction of the visual classifier. d) Feature extraction for the proprioceptive module. e) Labeling vibration signals of the proprioceptive module. f) Self-supervised training of the visual classifier.
signed a characterization label acquired through the ground robot characterization module. The training set for the visual classifier is incremented with every 1-second interaction the ground vehicle has with the terrain and at the same time the learned model is applied to predict characteristic metrics of images taken from the aerial vehicle ahead of the robot.

6.1.2 Incremental Learning

Apart from combining the proprioceptive and exteroceptive modules, an adaptive online framework needs to be implemented. In online learning, data becomes available in a sequential order and is used to update the best predictor for future data at each step. This is different than batch learning in which the best predictor is learned on the entire training data set at once. Batch learning is the method for training the visual classifier in Chapter 5. To accomplish “online” learning using an SVM method, an incremental SVM presented in [83] for incremental learning and adaptation of SVM classifiers is utilized. This method is used since batch SVM training is computationally intensive on training data that continuously arrives since it is accumulated as the vehicle traverses terrain. To limit the training time for the visual classifier, each terrain characterization class is limited to a maximum of 20 sets of feature vectors in which some of the older data is discarded if new data arrives exceeding that maximum.

6.1.3 Sliding Window

Also, the proposed solution for unlearning and learning the association between image patches and the proprioceptive characteristics, is a sliding window. The sliding window method converts an online supervised learning problem into a classical supervised learning problem [84]. It constructs a window classifier that maps an input window of width w into
an individual output value. This is illustrated in Figure 6.4. In the results section, parameter tuning of the sliding window are discussed. The larger the sliding window, unlearning of previous traversed terrain becomes more difficult but also terrain characteristic estimations become more robust.

Figure 6.4: Sliding window for terrain characteristics. This allows an unlearning mechanism for previous characteristics association

### 6.2 Experiment and Results

This section is dedicated to the description of the tests performed, and the corresponding results. The experiments are performed using the “E-Maxx ground robot driving at speeds around 2 - 3 m/s and with an aerial vehicle hovering above the ground robot. The ground robot is manually controlled to drive over different terrain. The site was a flat region with no obstacles on the trajectory of the robot. The test terrain consists of dirt, grass,
asphalt, woodchips, gravel and sidewalks under different illumination levels and varying visual appearances. The aerial images are captured using a DJI phantom equipped with a GoPro color camera. The average altitude of the aerial robot was 8 m. Blurred images due to excessive vibration of the aerial vehicle are removed through a preprocessing step. The framework assigns RTI characterization metrics to each segment the ground robot traverses and labels image patches acquired through the aerial vehicle as it goes by. The training sliding window of the classifier is gradually filled with every 1-second interaction the robot has with the terrain and at the same time older associations are discarded.

6.2.1 Supervised Offline Classification

From Chapter 4, a probability distribution of terrain characteristics of different terrain classes was extracted. Given that distribution, a traversal of a grass terrain patch, imposes high amplitude low frequency vibrations with a 79% probability. So one way of predicting the characteristics of far terrain is to detect the “class” of image patches acquired from the aerial vehicle. This characterization approach is not adaptive and also relies on accurate class prediction of visual classes. Also, different lighting conditions and occlusions in outdoor settings make a successful “a priori” classification algorithm almost impossible unless all the corner cases are recorded prior to experiments. For example, a picture of the same image patch taken in different lighting conditions is shown in Figure 6.5.

6.2.2 Self-Supervised Online Classification

To demonstrate the advantage of the self-supervised online learning method with respect to a supervised system trained a priori, the performance of both the methods is shown. In order to have a fair comparison, the self-supervised online classifier starts with the same trained classifier as the offline supervised classifier and learns further associations in a slid-
Figure 6.5: An instance where different lighting conditions lead to different classification results of the terrain.

In Figure 6.7, the percentage error of the two methods over a sequence of 5 minutes of terrain traversal is shown at any given time. The ground robot starts from dirt and then transitions into grass, asphalt, woodchips and sidewalk terrain in an orderly sequence. The sliding window is also set to 20 which gives the best accuracy results on this sequence. As can be seen in the dirt segment of the traversal, both the supervised and ensemble classifier yield the same results and make misclassifications on the same terrain patches.
This is due to the fact that the ensemble classifier’s sliding window does not have enough instances to learn new associations between visual features and its corresponding characteristics. Therefore, the characterization results are similar to the standalone supervised classifier. The arrow on the grass terrain illustrates a gap between the supervised classifier and ensemble classifier which is explained. At first, the error of both classifiers grow due to misclassification of the two methods but once the sliding window of the ensemble classifier learns the new association between visual features and terrain characteristics, it predicts the right characteristics and this is were the gap grows. This is seen in Figure 6.6. Although, this proves to be a trade off, since a few instances later the ground truth characteristics of the terrain goes back to what it usually is, whereas the sliding window weighs the most recent associations more and hence wrongly classifies terrain. This results in the gap between the two methods to cancel out and hence both the supervised and ensemble classifier yield the same classification results on average. The most improvement in using the online terrain characterization algorithm is seen on asphalt and sidewalk terrain. The arrows indicate the instances where the sliding window algorithm yields better accuracy results compared to the offline supervised classifier. This is due to the fact that instances of asphalt and sidewalk terrain that had different characteristics are clustered next to each other. This allows the online learning method to have enough instances for learning the association. On the other hand, the transition of characteristics in woodchip terrain happen too quickly and sparsely from each other that the sliding window fails to learn new associations and hence no improvement on woodchip terrain can be seen.

From the accuracy results of the scenario above, it is revealed that the online learning classifier relies on both the sequence of terrain and the sliding window size of the algorithm. Therefore, Table 6.1 shows the effect of different sequence traversals on the cumulative misclassification results.
Table 6.1: Cumulative Error of both classifiers on different sequencing of terrain on a set of 220 image patches

<table>
<thead>
<tr>
<th>Terrain</th>
<th>Cumulative Offline Classification Error</th>
<th>Cumulative Ensemble of Offline and Online Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dirt-Grass-Asphalt-Wood-Sidewalk</td>
<td>41</td>
<td>27</td>
</tr>
<tr>
<td>Grass-Dirt-Asphalt-Wood-Sidewalk</td>
<td>41</td>
<td>29</td>
</tr>
<tr>
<td>Dirt-Grass-Sidewalk-Asphalt-Wood</td>
<td>41</td>
<td>31</td>
</tr>
</tbody>
</table>

Table 6.2: Number of misclassifications occurred during traversal of the terrain type using different window sizes

<table>
<thead>
<tr>
<th>Terrain</th>
<th>10 samples</th>
<th>20 samples</th>
<th>30 samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dirt</td>
<td>3</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Grass</td>
<td>10</td>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>Sidewalk-Asphalt</td>
<td>7</td>
<td>7</td>
<td>15</td>
</tr>
<tr>
<td>Woodchips</td>
<td>14</td>
<td>7</td>
<td>14</td>
</tr>
</tbody>
</table>

Also, Table 6.2 shows misclassifications occurred during different terrain traversals having different window sizes for the same sequence. As can be seen, a window size of 20 results in the least amount of misclassifications. Reducing the window size allows quick learning but also is less robust to noise. Therefore, the only case in which a window size of 10 is better than the size of 20, is at dirt terrain. On dirt terrain, visual features of the image patches look similar but in rare occasions the terrain characteristics are also different. Since these characteristic changes do not happen in clusters, a big sliding window is unable to catch the variation but a 10 sample window size learns and unlearns the association very quick.
(a) Ground truth characteristics side by side with the characteristic results of the supervised classifier and the ensemble classifier

(b) Number of misclassifications grows faster in the supervised classifier in comparison to the ensembled self-supervised classifier

Figure 6.6: Cumulative misclassifications of the Supervised classifier vs the ensemble of Self-supervised classifier and offline supervised classifier.
Figure 6.7: Largest improvements are seen on pavements and asphalt terrain whereas on woodchips terrain no improvement is seen.
Chapter 7

Conclusion and Future Work

In this thesis, a learning framework for finding association between visual features of images captured from an aerial vehicle and a terrain traversability metric is described. First, acceleration data is mapped on to the frequency space. Next, the frequency space representation of vibrations is associated with the image space for classifier training. The experiments in Chapter 6 show that the proposed method shows good overall adaptability and improvement in terrain characterization. The cumulative accuracy for the framework depends on a reliable training set, and also the robustness of the visual classifier, size of the rolling window and frequency of terrain characteristic changes.

The contributions of this work is an average 12 % improvement in accuracy results of characterization of terrain on the same set of scenarios. To accomplish this, a speed invariance factor up to 20 % is integrated in our algorithm and an online learning self supervised algorithm was incorporated. Also utilizing an aerial vehicle as a far sensor and a ground vehicle proprioceptive sensor as near sensors in the near-to-far learning setup is a novel combination.
7.1 Future Work

It is seen that the classifier fails in many instances because of mixed terrain in an image patch as shown in Figure 7.1. A solution for future work is to take the image patch of the exact path the wheels of the ground robot traverse instead of a rectangular image patch. This requires using a camera on board of the robot in addition to the camera on the aerial vehicle. Also, association between proprioceptive sensors and image patches is based on assuming a constant trajectory and speed between each image frame. For future work we suggest integrating state of the art visual odometry and mapping algorithms on both the aerial vehicle and the ground robot to have a more accurate association between proprioceptive measurements and exteroceptive measurements.

In this work it was shown that a fixed size rolling window is on average beneficial in improving characterization results but in some terrain instances it negatively contributes to misclassification. For example misclassification instances increased on woodchip terrain. Therefore, we suggest that for future work an adaptive rolling window size be implemented and investigated in our framework. Also, the framework discussed in this thesis is implemented offline due to expensive computation. For future work, we suggest using binary descriptors such as ORB descriptors to significantly reduce the latency of the back-end of our algorithm. Reducing the latency of the feature extraction module is the most important step to move towards real-time implementations. Although a thorough study needs to be done for robustness analysis.

In our work it is shown that different selection of descriptors yield different accuracy results on different terrain and there is no one single descriptor that always performs better than other descriptors on all scenes. Therefore we suggest utilizing implementing our
framework in parallel on a modern GPU using different descriptors and depending on the terrain type use an ensemble of the classification results. Due to parallelization of the process we expect no added latency but more robustness.

Finally, more proprioceptive sensors can be integrated in the framework described in this thesis. Motor currents are parameters related to the thrust of mobile robots and can be used as sensors in proprioceptive measurements. A broader proprioceptive metric can be defined to integrate current readings of the motor in the model.

![Figure 7.1](image.png)

**Figure 7.1:** The framework is unsuccessful in describing the terrain characteristics of this image patch due to failed recognition of the exact path that the wheels traversed

During this research we also realized that previous work in this field needs to be standardized based on a benchmark dataset. Therefore we suggest building a benchmark dataset for incorporating ground truth of characteristics vs terrain images for further research in this field and comparison purposes.
Bibliography


