Terrain Classification System for Power-Assist Wheelchairs

Ahn, Joel	Khorram, Austin	McConkey, Keenan
13998159	22105150	21337167

Project Sponsor

CARIS Laboratories

The University of British Columbia

Engineering Physics Project Laboratory

ENPH 459 - Capstone Project

Project Number: 1908

April 6, 2019

Executive Summary

Project Summary

- We are developing a reliable and low-cost terrain classification system for power-assist wheelchairs.
- Knowledge of a wheelchair's current terrain will help to optimize motor control algorithms that provide power-assistance to its wheels.
- We have utilized smartphones as data acquisition devices for training terrain classification algorithms. These algorithms will ultimately be used to determine the wheelchair's current terrain in real time.

Motivation for Project

- Trivial terrains such as grass and gravel can pose significant challenges to manual wheelchair users, especially over long periods of use.
- Motor control systems on current commercially available power assist wheelchairs provide an electromotive force to the wheels based solely on the user's input force.
- Power-assistance can aid users in traversing difficult terrains, however this technology could be improved to adapt these terrains in real time.
- By combining terrain-recognition and user intention-detection innovations, our sponsor CARIS Labs hopes to improve the lives of wheelchair users.

Key Objectives and Pivots

- We chose to use smartphones as our data-acquisition system due to their scalability and their powerful onboard sensors useful for terrain classification.
- Using smartphones provided us the ability to collect large amounts of data for training terrain classification algorithms. However transferring these algorithms to the smartphone for real-time classification has been pushed out of scope.
- The current scope of our project includes providing a convenient method of data collection for future teams, as well as PC-based classification algorithms and data pipelines for differentiating between grass, gravel, pavement, and indoor terrain surfaces.

Conclusions

- We are able to classify three of our four chosen suitable terrains by processing optical and IMU data through our classification algorithms.
- Our data acquisition Android app can concurrently collect data from three sources.
- Although excellent for collecting data, the Android environment may not be the best platform for creating a real-time terrain classification device.

Contents

List of Figures

1 Introduction

1.1 Sponsor & Background

1.1.1 CARIS Labs

CARIS Labs (2) is a research team based out of UBC dedicated to improving human-robot interactions through engineering innovations. Their overarching goal for this project has been to improve the lives of wheelchair users. Wheelchair users face many difficulties in their lives, even in highly human-tailored urban environments. Common terrains such as grass and gravel can require strenuous effort to traverse, necessitating assistance. By creating wheelchairs capable of both recognizing their users' intentions and adapting to the terrains they are traversing, power-assist wheelchair technology can be greatly improved to enhance the mobility of its users.

1.1.2 Power-Assist Wheelchairs

This work builds on current commercially-available power-assist wheelchair (PAWC) technology (see Figure 1). These wheelchairs function similarly to self-propelled manual wheelchairs but provide a boosting torque through onboard hub motors whenever a user applies a force to the push-rims of the two main wheels (3). PAWCs significantly reduce the effort required by their users and improve the range of their mobility.

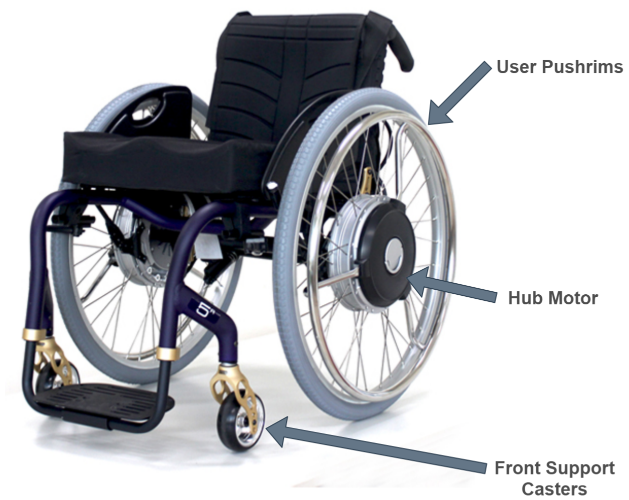

Figure 1: A conventional power-assist wheelchair with labelled push-rim, hub motor, and front support casters. Hub motors provide a boost to the users input torque when a push is applied to the push-rims.

1.2 Project Objective

The main objective of this capstone project has been to create a combined data acquisition and terrain classifier capable of identifying terrain states. Terrains of interest include grass, gravel, pavement, and indoor concrete. These four terrains were chosen as they are commonly encountered in an urban environment. The classifier is ultimately meant to integrate into a larger system and enable adaptive motor control, so integration has been factored into our design considerations. Additionally, a prominent design criterion has been scalability, as the chosen terrains to classify are somewhat arbitrary and the ability to add new terrains in the future is an essential priority to the sponsor.

1.3 Scope & Limitations

The scope of the project has been limited to only include the development of a data acquisition system and offline classification algorithms. Essentially, we have taken steps to train and test the algorithms on a PC, but have not included the final step of integrating those algorithms into a real-time classification system. This choice was made in the interest of time. However, this milestone is critical as it lays the foundations for a veritable translation of the Python PC algorithms to an Android mobile environment or a dedicated real-time embedded system.

1.4 Outline of Report

The rest of the report is split into three sections: Discussion, Conclusions, and Recommendations. The Discussion section will explain some of the fundamental concepts of the project and the design approach taken throughout its completion. It will also present project results along with the testing used to verify them. The Conclusion section will talk about the important takeaways of the project, especially in regards to both the sponsor's priorities. Finally, the Recommendations section will discuss the next steps we recommend the project take and give advice to future capstone teams who may inherit it.

2 Discussion

2.1 High Level Overview

The bulk of this project is split into two major components: the first being the data acquisition system and the second being the terrain classification algorithms. These components were developed independently of each other in separate environments. As discussed in Section 1.3, these components are to eventually be integrated into a single system, but that is outside the current scope of this project.

One critical design aspect in which these two modules overlap is in the terrain states they are designed to classify. Through discussion with CARIS, we came up with a list of which terrain states their smart power-assist wheelchair should be able to recognize and adapt to. These terrains meet three criteria: (1) they are often encountered by wheelchair users in daily activities, (2) they have unique characteristics that the motorization of the wheelchair should adjust to, (3) they are distinctly recognizable using a corroboration of easily accessible sensors. The four terrains that best met these criteria include: (1) grass, (2) gravel, (3) pavement, (4) indoor polished surface. We also considered other terrains such as sand and ice. However these surfaces ended up being left out of the scope of this project due to lack of available testing area. A system level diagram of the end result of our project can be referred to in Figure 2.

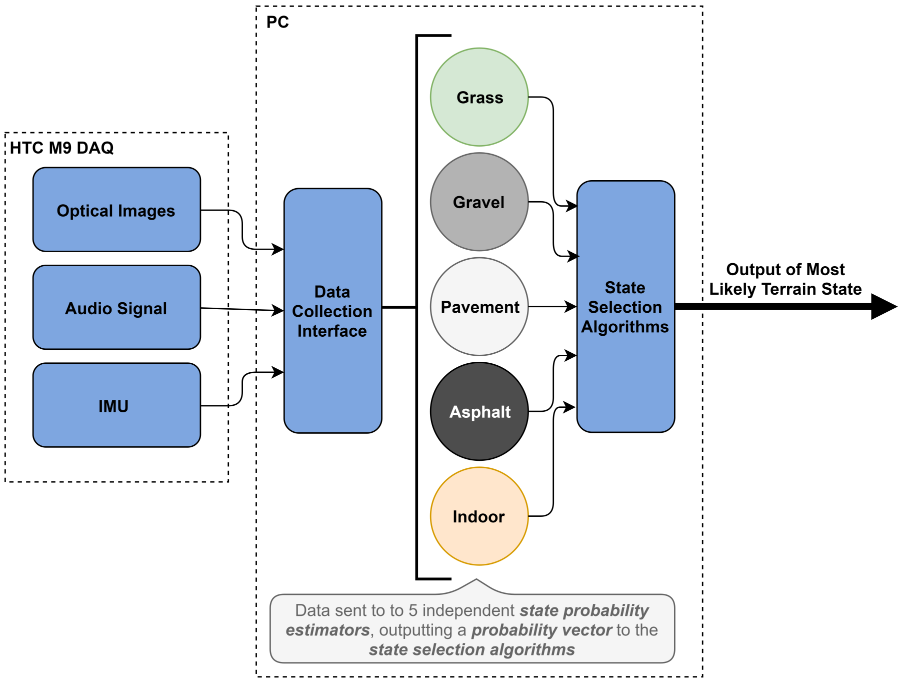

Figure 2: System level diagram for the final classification process.

2.2 Theory & Fundamentals

Current literature recommends, barring expensive high-precision methods, sensors that are best suited to classification of terrain environments are inertial measurement units (IMU), audio, and optical cameras, hereafter referred to as acceleration, audio, and optical classifiers. We ruled out early on other forms of sensors, such as soil penetrometers because they introduced problems of attaching to the wheelchair; and lidar (light detection and ranging) systems, as they required hefty capital when cheaper sensors were available.

2.2.1 Acceleration Classifier

By analyzing the linear and gyroscopic accelerations of a wheelchair in motion, one can characterize the roughness of and ultimately identify a terrain (4). We explored multiple strategies for analyzing the data: peak-variance analysis and a Fast-Fourier Transform (FFT) analysis.

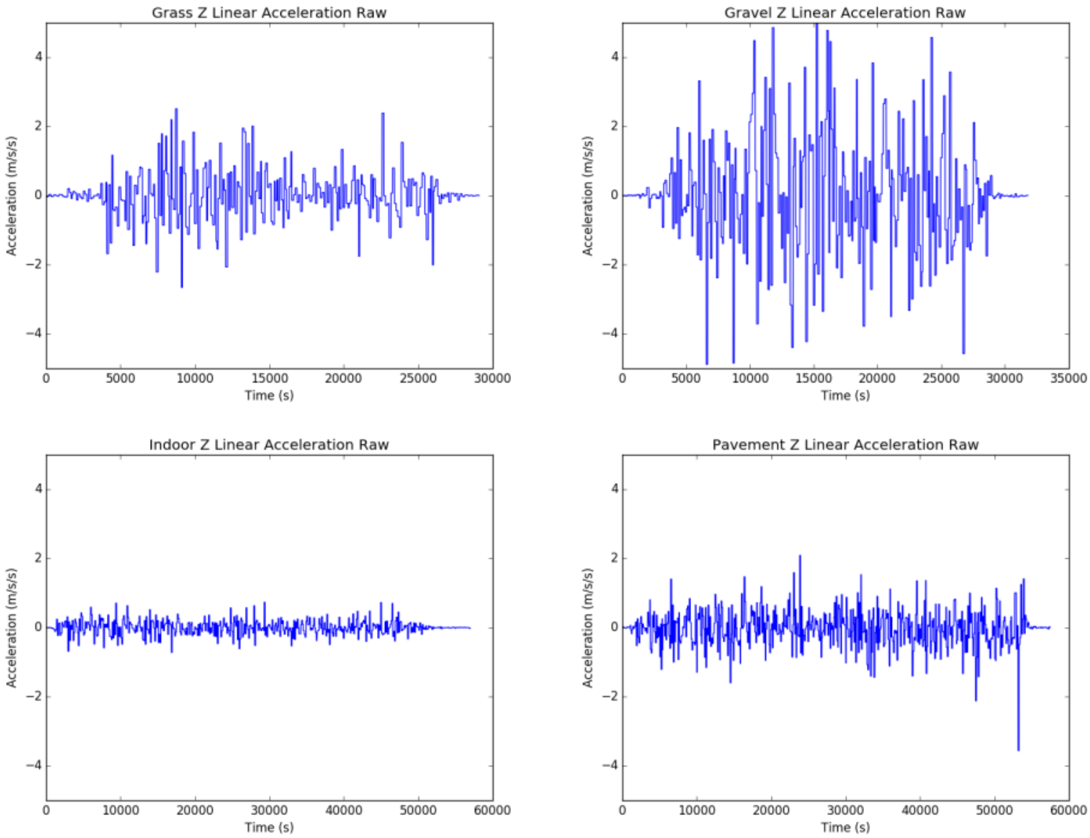

Figure 3: Raw plots of z-axis linear-acceleration recorded while traversing various urbanscapes (z is the direction perpendicular to the surface being traversed).

The wheelchair, as it traverses rough terrain, generally records acceleration impulses of great amplitude and frequency as shown in Figure 3. Peak variance aims to characterize the magnitude variations of the positive and negative impulse peaks, on the assumption that terrains such as gravel have greater magnitude variance than those such as indoor concrete.

The raw data is subdivided into sample subsets (generally of 50 data samples). The peaks and valleys of each data subset are identified, where a peak is simply defined as :

$$0 < \frac{X_k \ - \ X_{k-1}}{dt}, \ \frac{X_{k+1} \ - \ X_k}{dt} \ < \ 0$$

Thus a valley is defined as:

$$0 < \frac{X_{k+1} \ - \ X_k}{dt}, \ \frac{X_{k+1} \ - \ X_k}{dt} \ < \ 0$$

The variances of the peaks and valleys of each data subset are calculated and plotted.
In the case of the FFT analysis, we initially apply a fourth-order Butterworth filter and then perform an FFT (Section 6.1).

2.2.2 Optical Classifier

Other literature (5) suggests that optical data taken via camera is effective at differentiating between grass and dirt through common machine vision algorithms, which are discussed further in Section 2.2.2. Thus camera analysis can make up for some of the inherent limitations of IMU classification in differentiating between these terrains, though processing optical data can be more computationally expensive.

Figure 4: Some examples of common urbanscapes encountered by wheelchair users: (1) grass (2) asphalt (3) tiled pavement (4) gravel (5) indoor concrete

Our machine vision algorithms utilize one of two data pipelines: (1) a simple Hue-Saturation-Value (HSV) filter; (2) a gray-scale conversion, a gradient analysis of the gradient values in both the x and y axes; and a principal component analysis (PCA) (see Figure 5).

The HSV filter simply converts the standard Red-Green-Blue description of an image into a more intuitive model: the hue determines the actual color of a pixel on the color wheel; the saturation, the intensity of that particular hue; and the value, the amount of light that might shine on an object of that color. This enables more robustness of the optical analysis to varying conditions such as the effect of sunlight on white balance.

The gradient/PCA enables one to analyze the directional components of an image. By converting an image to grayscale (i.e. describing it on a linear 0 to 255 binary bit value), one can create an

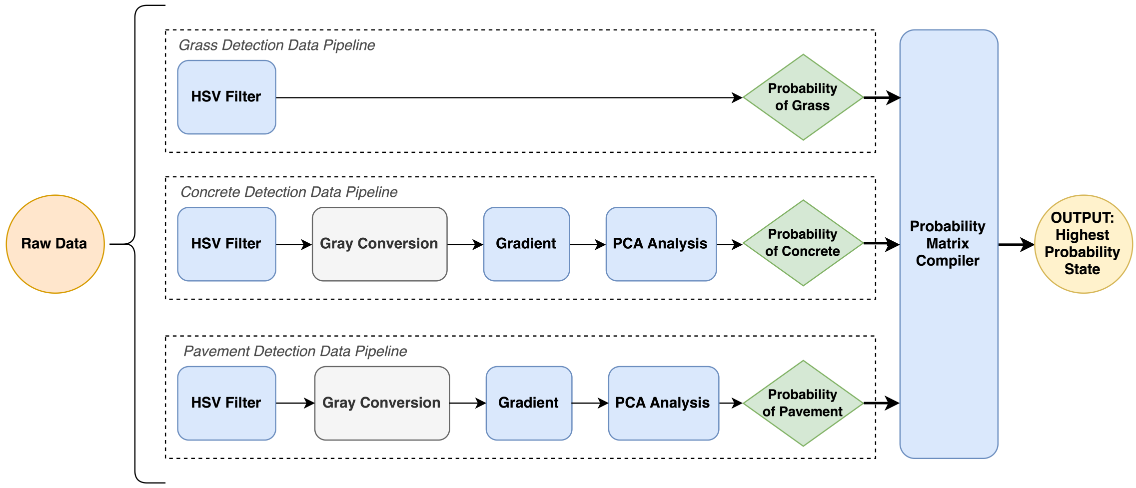

Figure 5: Optical terrain classification flowchart

$(N-1)\times(M-1)$ gradient matrix, where N and M are the pixel resolutions of the image horizontally and vertically. One can then calculate the pixel differentials in the two dimensions and perform PCA.

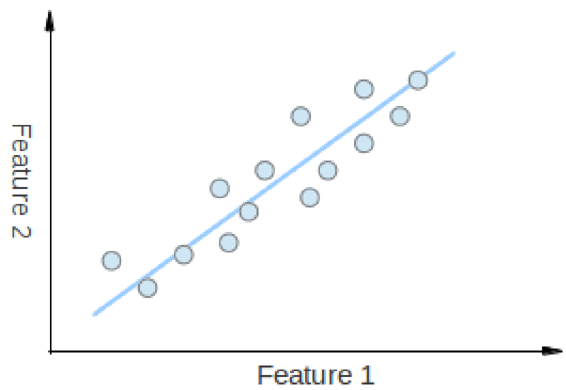

Figure 6: An example PCA (principal component analysis). This is one optical analysis method useful for determining terrain type, since different terrains often exhibit unique visual patterns.

In a data set, there is a chance that the feature dimensions (the y and z-axis linear accelerations, the $x-y$ image gradients, etc.) are correlated. PCA performs orthogonal transformations to discriminate the data set into linearly uncorrelated dimensions called principal components. Physically, what this looks like is that the coordinate axes are rotated to a frame such that axes lie where the dataset has the greatest variance. In Figure 6, one observes that the data points vary more along the correlated blue line than along the Feature 1 or 2 axes.

To calculate the principal components for a matrix M of p dimensions, one calculates the eigenvalue decomposition of a data covariance. To calculate this covariance, one first calculates the empirical mean of each row (or column) of the matrix and then subtract it from each data point.

$$mean \;=\; u[j] \;=\; \frac{1}{n}\sum_{i=1}^{n} X[i,j]$$

Where n is the number of groups of a single observation. U is the mean vector multiplied by a ones vector of dimension n to create an $n \times n$ matrix.

$$B \;=\; X - U$$

Then one can calculate the covariance matrix by multiplying the B matrix with its conjugate:

$$C \;=\; \frac{1}{n-1} B^* \;*\; B$$

From the deviations, one can calculate the covariance matrix. One then finds the eigenvectors and eigenvalues of the covariance matrix, which in turn describe the principal components of the data set.

2.2.3 Audio Classifier

A final additional parameter for classification, which was suggested by the Engineering Project Lab, is audio data. Different terrains can make significantly different sounds as they are travelled over – gravel for example is significantly louder than pavement and grass. Audio analysis can thus be performed using similar techniques as the IMU as another confirmation of terrain classification.

2.3 Data Acquisition Module using Smartphone App

2.3.1 Design Approach

One major design pivot we made early on in the project was to use the Android platform as an integrated DAQ and terrain classification system. Initially, we planned to separately acquire sensors and integrate them on a microcomputer platform, such as a Raspberry Pi. Smartphones contain powerful inertial sensors, cameras, and microphones in a highly condensed and accessible package. They opened the opportunities to eventually serving as development platform for real-time, live classification. This significantly changed the scope of this data acquisition portion of our project from a problem of sensor integration into a more software-orientated undertaking.

To access all of these sensors in an integrated fashion, we had to design and build a smartphone app capable of concurrently polling terrain sensors at a consistently high frequency.

After doing background research on the Android environment, we decided it was easiest to develop a separate Activity (Android's word for a user interface with a specific function) for each sensor, then test and integrate them into a single Activity. Each group member was given one sensor to focus on, so that we could eventually work together to combine them. Overall, we chose our design path to best fulfill the sponsors need for an accessible yet powerful platform for large-scale data collection.

2.3.2 Testing & Validation

Upon completion of our DAQ (data acquisition) Android app we collected data our three forms of data by rigidly mounting two smartphones to the rigid undercarriage beams of the wheelchair (see Figure 7) and having one of us traverse a specific terrain while the app was running. We attempted to best recreate the conditions of a typical wheelchair user by self-propelling on each surface.

Figure 7: Terrain data acquisition setup with two smartphones rigidly mounted to the wheelchair base (using a purchased phone mount). Large amounts of data was collected with this method, and various mounting locations on the wheelchair were tested.

The data was polled at maximum rates in order to collect as much data as possible (refer to Section 2.3.3). We found some limitations in the use our DAQ app due to its low maximum polling frequency, which we were unable to increase due to our lack of familiarity with the Android environment. Thus we ended up using externally available apps (1) for a large amount of data collection.

We varied parameters of smartphone data collection for testing purposes. Firstly, we tried different mounting locations on the wheelchair, such as directly attached to the seat, attached to the wheels, and mounted to the wheelchair undercarriage. We found the best location to be a rigid location somewhere close to the ground on the wheelchair (this can vary depending on the wheelchair used). We also discovered that commonly available smartphone mounts designed for bikes were a great tool for wheelchair attachment (see Figure 7). Finally, we tried attaching two smartphones on either side of the wheelchair, which was found to have little effect on data recordings.

As a method of verification, we compared the data collected by our app to that taken from an sensor app available on the Google Play Store (1). As discussed, this external app was able to reach higher sampling rates than ours, but it cannot be used to record all the sensors we need concurrently. Data collected from each app is compared in Figure 8. Note that the z axis is the direction perpendicular to the surface of the terrain being traversed, in this case pavement. Results are quite similar even though our app records at a slower sampling rate, this is reasonable based on the low frequencies appearing in the transformed signals (refer to Appendix A).

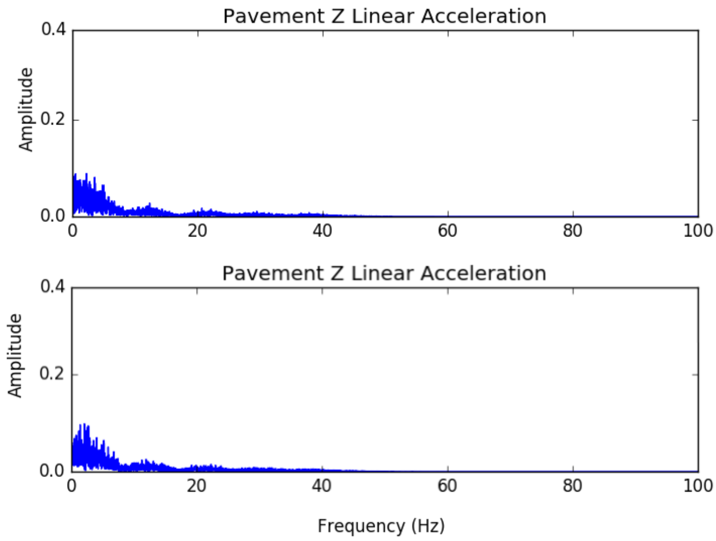

Figure 8: Comparing Fourier-transformed z-axis linear acceleration data recorded by our data acquisition app (a) to data simultaneously recorded by a similar available app (b) (see (1)).

2.3.3 Results

Through developing an Android app on HTC One M9 smartphones, we were able to successfully gain access to three sensors suitable for terrain classification. We created straightforward interfaces for collecting data from each sensor, and provided a method for concurrently acquiring data in a single integrated Activity. Concurrently running data collection on each sensor, we achieved sampling rates of 1 Hz and 4 Hz for the camera and IMU respectively. Audio recording was found to have little impact on the reliability of other sensors at a variety of sampling rates. The optimal rate was found to be 16 kHz. A user is able to turn off an polling of each sensor.

As one can see, the sampling rates of our data acquisition app is quite low, which limits the range of frequencies detectable in IMU and audio analysis (see Section 6.1). Collecting data from three sensors at the same time poses limitations on the processing speed of data acquisition. Our app has difficulties outperforming other currently available smartphone apps at single-sensor data collection, specifically in terms of maximum sampling frequency as well as performance on a wide array of phones. Our limited experience with the nuance of Android development may have hindered our ability to develop the optimal app for data collection.

Figure 9: Data acquisition app running on a HTC One M9 smartphone for collecting three sources of terrain data concurrently.

2.4 Terrain Classification

2.4.1 Design Approach & Results

In order to lend modularity and scalability to our system, each terrain classifier is its own encapsulated class and is free to subscribe to its own set of sensor values. This design assumes that a "Data Collection Interface" will be responsible for gathering all the available sensor values and broadcasting them to the terrain classifiers. See Section 4.3 for how to achieve this. We developed and tested the individual classifier states for the Grass, Gravel, Pavement, and Indoor Concrete classifiers. Below is a high-level description of each sensor:

1. **Grass**: Receives optical data. Convert the RGB image to HSV and multiply the saturation by a SATURATION_FACTOR = 1.4 to accentuate the intensity of the color. Apply a binary mask to the image, blacking out pixels that do not fall within the range of [GREEN_HSV_MIN = {42, 160, 160}, GREEN_HSV_MAX = {106, 255, 255}]. Identify regions of notable size that passed the filter and if the size is large enough, then grass has been identified.

2. **Gravel**: Receives IMU and optical data. For the IMU, analyze only the y linear accelerations. Assuming that the sampling rate is 200Hz, take the first 128 data points, apply a fourth ordered Butterworth low-pass filter, with the cutoff frequency at 20Hz, and perform the FFT. Should the mean-amplitude of the frequency components exceed 0.5 m/s/s then there is a probability that it is gravel. Analyze the optical component to check for any green hues so that the classifier has not confused it for grass, which can be equally strenuous to traverse. See Figures 13 and 15 for example graphs of the y and z linear accelerations.

3. **Concrete**: Receives IMU. For the IMU, analyze the mean amplitude of the roll acceleration, or the acceleration that best measures level height between the two wheels. Perform the same FFT pipeline as in Gravel, only this time ensure that the mean amplitude does not exceed 0.01 rad/s/s. Corroborate this finding with the optical data by measuring the smoothness of the image i.e. based on Figure 4, one expects concrete and other indoor surfaces to have a smooth shader texture. Exploit this characteristic by converting the sampled image to grayscale, calculating the pixel gradient in the x-y axes, and then using a Canny edge detector to ensure that there are no distinct edge patterns as would be found for gravel or pavement.

4. **Tiled Pavement**: Receives IMU optical data. Utilize the same FFT pipeline as the Concrete detection, but set the roll mean amplitude threshold to 0.02 rad/s/s since tiled pavement can introduce regular but significant bumps. Corroborate the IMU data with an optical classifier. Taking the sampled image, convert the sampled image to grayscale, calculate the x-y gradients, and detect the edges of the image to discern the shapes. If several objects of notable size can be found, then it's likely tiled pavement.

2.4.2 Testing and Validation

Sufficient data was gathered to train and test the classifiers individually, but due to the constraints of time, a system to run the classifiers concurrently and develop a confusion matrix has not been developed at this time.

Certain assumptions are made during the collection and analysis of classification data. Though we have accounted for terrain variations, decoupling the physical variations among wheelchair users from the data remains unexplored territory. Whether it be the number of pushes a user can exert in a minute or the individual strength of each wheel push, the accelerations recorded via the IMU will differ when it comes to difficult terrains. In our analysis, we assume that the contributions of the user-inputted torque and weight are negligible.

A testing strategy for each of the four terrains is discussed below:

Figure 10: Example photos of grass. Displayed on the top row are photos of grass at a range of colors; on the bottom, grass with other objects cluttered in the frame.

1. **Grass**: In order to rigorously test the HSV filter and contour identification, we took full frame grass photos at saturation varying from lush to pale green; hues varying from yellow-green to blue-green; and brightness varying from grass under direct sunlight to being shaded. We also took care to take ideal photos of grass thickets and more noisy ones with added objects such as roads and buildings.

2. **Gravel**: Since there is great variation in gravel sizes, we took pains to locate gravel of differing granularity. Gravel rock sizes varied from half a centimeter to an average inch in diameter. We attempted to eliminate variables by having the user output a constant torque with each wheel push.

3. **Indoor Concrete**: Indoor surfaces generally proved the easiest classify, as the highly-controlled

level surface made its signature in IMU recordings very distinct. However, there are significant variations in indoor surfaces, as aesthetic choices can differ. Therefore, greater energy was devoted to gathering accelerometer rather than optical data in this case.

4. **Tiled Pavement**: Tiled pavement proved the most difficult to classify. The shapes of the individual tiles vary significantly; outdoor pavement is sloped at noticeable angles, whether as a consequence of the landscape or to ameliorate drainage. This has made a one-size-fits-all solution difficult to find.

Each terrain classifier is expected to output a probability vector estimating the likelihood that based on its subscribed set of sensor data, the wheelchair user is in fact on that terrain. We adopted a very simplistic method for calculating this final probability. Using a testing dataset, we compiled the number of true-positives, false-positives, true-negatives, and false-negatives, and then divided each of those counts with the total number of trials to derive a probability. However, this idea naively relies on the law of large numbers and due to constraints of time, we lacked the breadth of data to produce a reliable number.

3 Conclusions

3.1 Android Development

As described in Section 2.3.3, we developed a data acquisition (DAQ) app capable of concurrently polling three sources of terrain data: camera, microphone, and linear/gyroscopic accelerometer. Such a data collection system is powerful when collecting large amounts of data on the parameters we believe to be best suited to terrain classification (see Section 2.2.1). This app could eventually be expanded to perform real-time classification on a power-assisted wheelchair, thus it could communicate with the wheelchair's motors to allow it to perform adaptive motorization control in different environments, as desired by the project sponsor.

However, Android imposes limitations on the potential of a real time DAQ/classification device. Background routines performed by the Android OS, such as network checks, can make data collection non-deterministic, i.e. a consistent sampling rate cannot be guaranteed. This limits the frequency range that can be recorded by our DAQ app, specifically in IMU data.

3.2 Data Collection Methods

We tested various methods of data collection with our smartphones. The best method we found was attaching the smartphones onto a rigid beam of the wheelchair with purchased mounting parts. This method minimized inertial movement damping created from more flexible materials, such as the seat. Placing the phone on different sides of the wheelchair had no discernible effect.

3.3 Overall Terrain Classification Performance

We explored multiple parameters for differentiating terrains, namely: audio data, optical data, and inertial/gyroscopic acceleration. Firstly, we identified y-axis and z-axis linear acceleration to be a good differentiator of terrain type. As the y-axis corresponds to the forward direction of the wheelchair, y linear acceleration naturally records the torque that the user exerts on the wheels. The trend is that generally, the more difficult the terrain is to traverse, the greater the recorded y linear accelerations. As for the z linear accelerations, the IMU best records the bumps of a surface along this axis. These observations corroborate Ojeda et al.'s paper(4), extrapolated to a larger wheelchair application rather their small rover robot.

In the case of optical classification, we were met with more varying results. A simple HSV filter proved robust in classifying grass a fair share of the time but when confronted with more challenging scenarios such as white balance in shade or high-exposure. Earlier explorations in analyzing the texture of the "grass" (i.e. the number of definable edges in a gradient filter) have proven fruitful in corroborating the HSV filter, though it proved computationally expensive. The same data pipeline of texture segmentation have proven equally fruitful in gravel and pavement, as gravel evinces the same chaotic image quality as grass (minus the green) and pavement has regular patterns of tiles.

4 Recommendations

4.1 Data Collection System

Although we found it difficult to integrate classification into the Android environment, for data collection purposes the Android DAQ was able to provide large amounts of reliable and easy-to-work-with data. Therefore, we would recommend using the Android DAQ to collect optical, audio and IMU data for the purpose of tuning classification algorithms. For collecting IMU data specifically, we recommend using the app "AndroSensor" that can be downloaded for free in the Google Play store. (1).

4.2 Suitable Classification Parameters

We recommend continuing to use IMU and optical analysis as a method for classification. Audio analysis was not fully researched and is suggested as an exploration point for future teams.

4.3 Next Steps for Classification Algorithms

The overarching architecture for classification is organized according to modular terrain classifier states. In this system, an initial "Data Collection Interface" (DCI) compiles the outputs of the various sensors and broadcasts the new sensor readings in the form of globally available, read-only variables. Each terrain classifier is free to subscribe to what it deems are the relevant set of sensor values: grass may require only optical data, but gravel may require a corroboration between IMU and optical data. This design places the burden of coordinating classifiers in a single overarching DCI and instead frees future developers to instantiate new terrain classifiers.

The "State Selection Interface" (SSI) requires the classifier states, when polled, to output a float value that describes the probability that they are currently on their described terrain. It compiles the outputted values from each of the classifiers into a probability vector. The overarching motor control system is expected to use this probability vector to help adapt its torque output.

There are some notable expectations and limitations to this design. Because each classifier can take a variable amount of time to complete (and the whole system itself run on a non-deterministic operating system), the SSI cannot guarantee an probability vector at a deterministic interval. The adoption of a real-time OS may help to rectify this, though it does add a significant layer of complexity.

4.4 Real-Time Terrain Classification System

Other capstone teams working with CARIS have developed an independent sensor platform for intention-detection that could be adapted to perform terrain classification. Thus the sponsor will have to choose which environment they will continue developing real-time classification on.

Going forward, we recommend the sponsor use their independently developed intention-detection system as the platform for development of a real time classification device. Since the limitations of Android have given us issues at this stage of the project, we found that using smartphones as the development platform for a combined data acquisition and classification device will be difficult.

5 Deliverables

1. **Final Classification Algorithms**: As discussed in Section 3.3, the algorithms differentiate four unique terrains with varying degrees of certainty. This is our primary deliverable to the sponsor as although the DAQ app is operational, the performance does not exceed other readily available options

2. **DAQ Android Application**: We will be providing our DAQ app as an alternative option for acquiring data. Documentation and source code are readily available on our GitHub repository (6).

3. **Terrain Data Set**: We accumulated masses of raw terrain data which can be used to further improve the classification algorithms or any other uses the sponsor may need in the future

4. **Terrain Classification Pipeline Design** Data pipeline outlined in Figure 2. Completed individual state verification, future work to integrate them.

6 Appendices

6.1 Appendix A: Fourier Transforms and Sampling

One of the best tools for signal analysis for any type of sensor is the Fourier Transform. Performing a Fourier Transform on a time-domain signal converts it into its frequency components, which can differentiate certain features of a signal obtained from terrain measurement of a terrain parameter.

Since digital sensors can only sample a terrain data signal at a discrete rate, the sampling of the signal must occur at a higher frequency than the frequency being measured in order to it to appear in the Fourier Transform of the signal. **Nyquist-Shannon Sampling Theorem** states that when converting a continuous signal to a discrete signal via sampling, the following condition must be met:

$$f_s > 2f_{lim} \tag{1}$$

Here f_s is the frequency of sampling and f_{lim} is the band limit frequency i.e. the highest frequency to be sampled in the continuous signal. This theorem is critical when designing sensor systems as it enforces a processing speed that can limit data-collection bandwidth, which can influence the reliability of terrain-recognition algorithms used by the wheelchair.

6.2 Appendix B: Fast Fourier Transforms of IMU Data

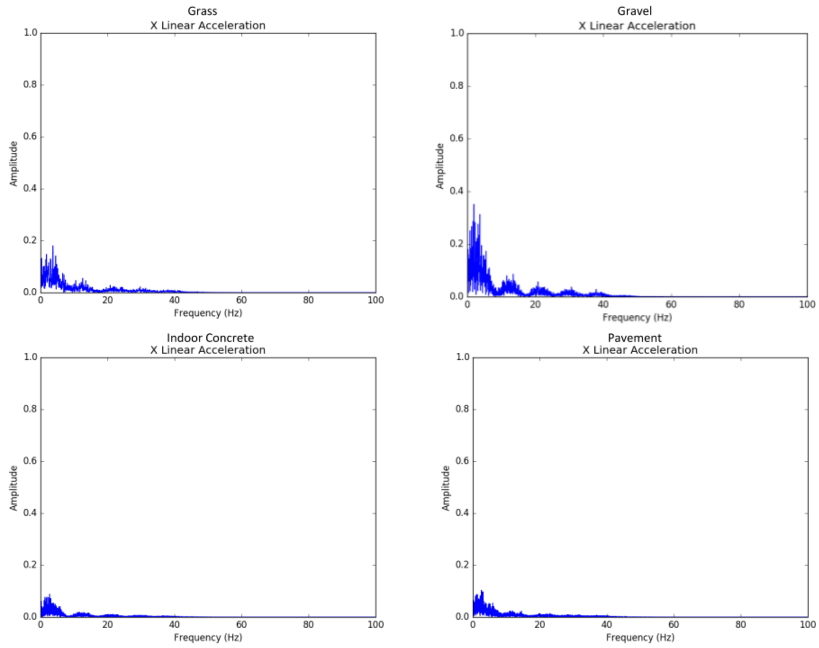

Figure 11: Fourier transforms of x-axis linear acceleration recorded while traversing common urban terrains, namely: (a) grass (b) gravel (c) indoor surface (polished concrete) (d) tiled pavement.

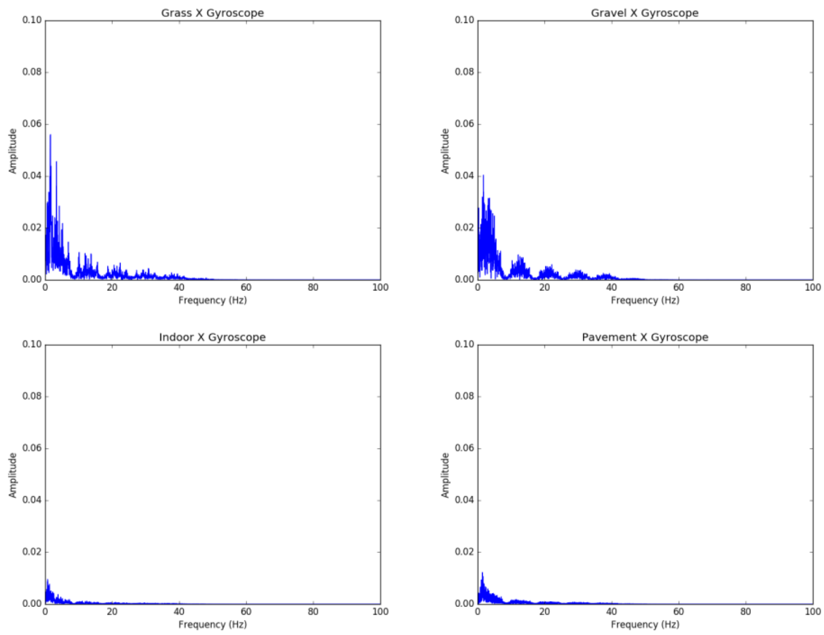

Figure 12: Fourier transforms of pitch acceleration recorded while traversing common urban terrains, namely: (a) grass (b) gravel (c) indoor surface (polished concrete) (d) tiled pavement.

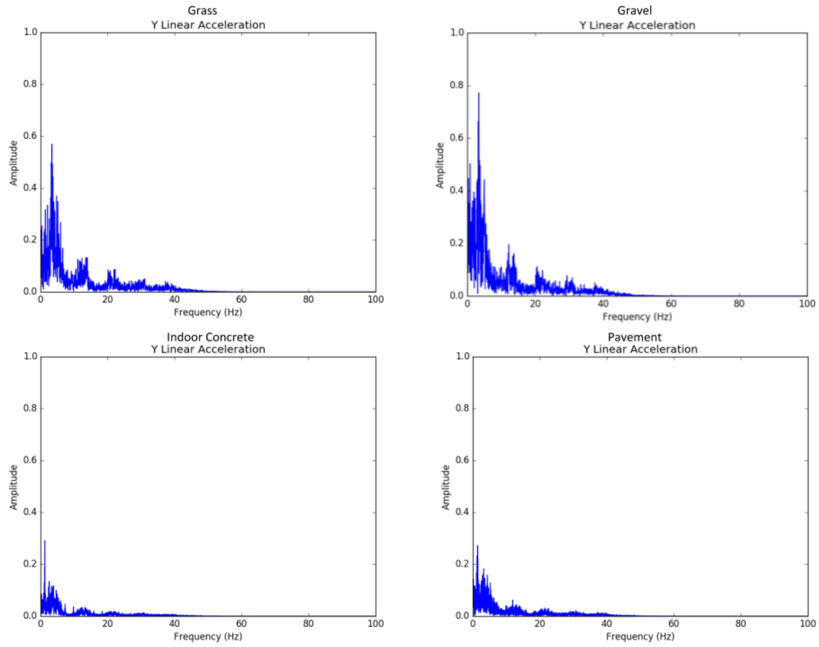

Figure 13: Fourier transforms of y-axis linear acceleration recorded while traversing common urban terrains, namely: (a) grass (b) gravel (c) indoor surface (polished concrete) (d) tiled pavement.

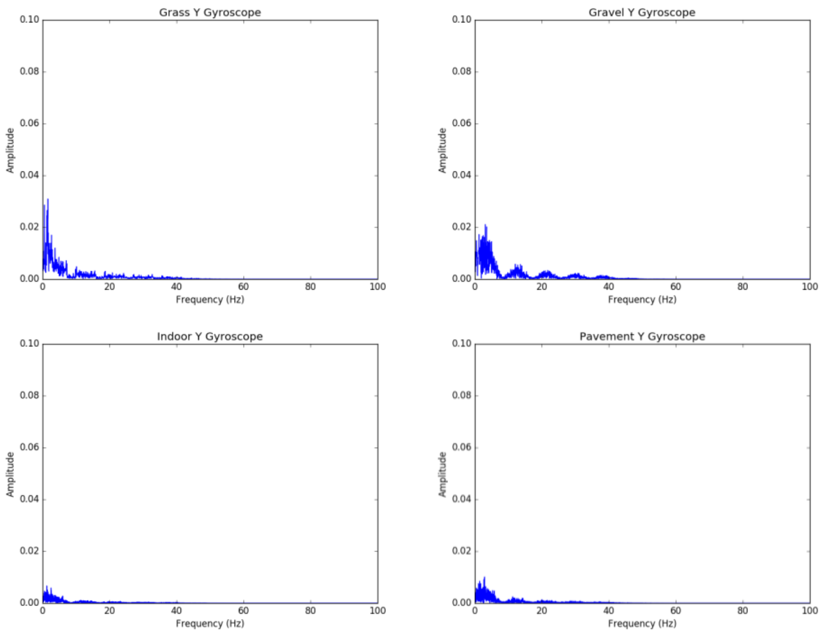

Figure 14: Fourier transforms of the roll acceleration recorded while traversing common urban terrains, namely: (a) grass (b) gravel (c) indoor surface (polished concrete) (d) tiled pavement.

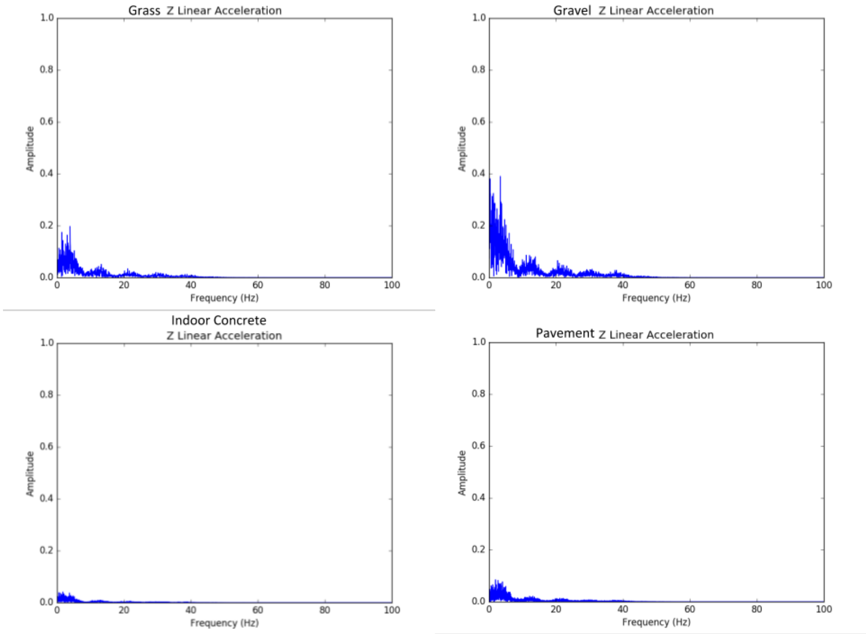

Figure 15: Fourier transforms of z-axis linear acceleration recorded while traversing common urban terrains, namely: (a) grass (b) gravel (c) indoor surface (polished concrete) (d) tiled pavement.

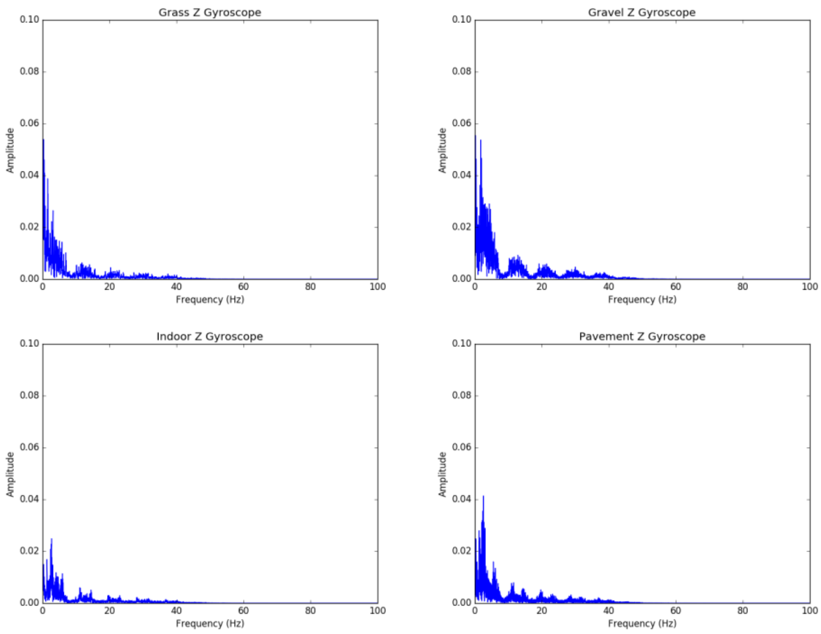

Figure 16: Fourier transforms of the yaw acceleration recorded while traversing common urban terrains, namely: (a) grass (b) gravel (c) indoor surface (polished concrete) (d) tiled pavement.

References

[1] "Androsensor." [Online]. Available: https://play.google.com/store/apps/details?id=com.fivasim.androsensor&hl=en_CA

[2] C. Labs, "Caris labs website." [Online]. Available: www.caris.mech.ubc.ca

[3] C. River Medical Products, "What is a power assist wheelchair?" [Online]. Available: www.clintonrivermedical.com/blog/what-is-a-power-assist-wheelchair

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

[5] G. A. Carpenter, S. Gopal, S. Macomber, S. Martens, C. E. Woodcock, and J. Franklin, "A Neural Network Method for Efficient Vegetation Mapping," *Remote Sensing of Environment*, vol. 70, no. 3, pp. 326–338, 1999.

[6] "Project Github Repository." [Online]. Available: https://github.com/AustinKhorram/CARIS-WHEELCHAIR