Predicting terrain traversability from visual and accelerometric feature correlation

Ravi Kiran Sarvadevabhatla
Department of Computer Science
University of Washington
Seattle, WA 98105
kiran@cs.washington.edu

Abstract

One of the fundamental problems of autonomous navigation in real-world outdoor environments is that of predicting *terrain traversability*. Typical approaches to this problem involve the use of vision-based and terrain-contact based features. The vision-based approaches have the advantage of long-range sensing, however they are prone to changes in lighting conditions and offer at best an indirect measure of traversability. The contact-based approaches are more direct, however they can only sense within the physical footprint of the robot. We present an approach that predicts terrain traversability by correlating visual appearance of terrains with associated contact-based properties as the robot traverses the terrain. Specifically, we use an approach wherein the predictions from a contact-only based classifier are used to re-train a visual feature-based classifier. Experimental results on hitherto unobserved terrains show that this approach improves the performance of the visual feature-based classifier while providing long-range predictions.

1 Introduction

Autonomous navigation in unstructured outdoor environments is a fundamental and challenging problem in Robotics. In this problem, a robotic vehicle senses the world and attempts to identify surroundings which provide the maximum affordance [1] in reaching the goal. A key aspect of this problem is the ability to predict the *traversability* of terrains encountered in various environments and act accordingly. Predicting terrain traversability involves identification of regions in the world which are most suitable for onward journey of the robot and forms a crucial component of path-planning approaches in applications [15]. In order to identify such regions, we need to identify obstacles (so that they can be routed around) and desirable regions in obstacle-free areas (so that they can be preferentially used). In this report, we focus on the latter and present a robotic platform which predicts terrain traversability from sensor data.

Typical approaches to this problem involve the use of vision-based [3, 4, 5, 6] and terrain-contact based sensor [11, 12, 13, 14] data. The former use cameras (monocular and stereo) fitted on the robot to obtain images of the terrain. These images are then processed and labeled with desirability values. In order to use this prediction information, the camera(s)

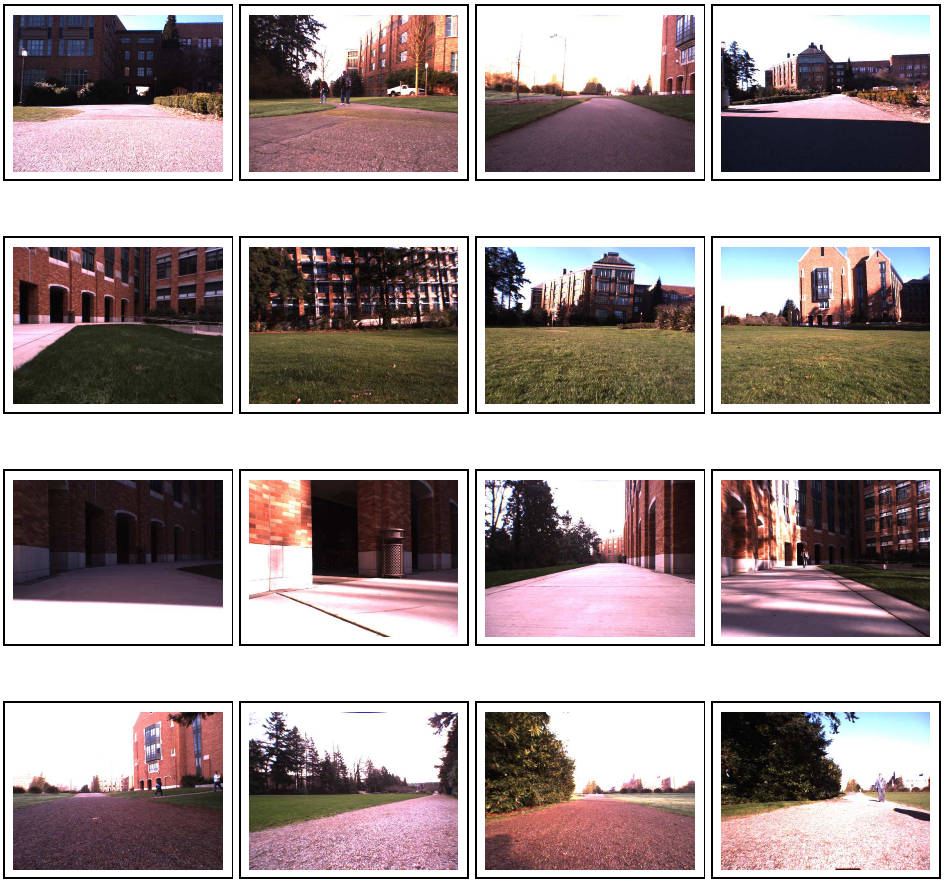


Figure 1: Terrains encountered by the robot. Each row represents a terrain(top to bottom - `road`,`grass`,`concrete`,`gravel`. Notice the effect of lighting changes on terrain appearance.

must be calibrated with the ground-plane reference of the robot. Typically, this reference plane is assumed to be the one that passes through the base of each wheel of the robot. Depending on the ruggedness of the terrain, this assumption may not be correct. Also, the lighting conditions in outdoor environments contain a large dynamic range which affects appearance – a patch of grass in shade can very well look like a patch of dark soil. Therefore, it is very important to use color-constancy based features for the prediction to be invariant to lighting. Another important aspect is the change of appearance with distance. Because of perspective imaging, two terrain patches of same size can occupy two very different sized areas in the image depending on their distance from the camera. This affects their appearance and therefore a good prediction system must take this distance from camera into account. Many of the approaches deal with the prediction problem by treating it as an instance of image classification. However, as [1] point out, the mapping from a classification label (E.g.`tree`,`grass`,`rock`) to the underlying traversability they afford can be quite non-uniform and hard to define. A better idea would be to model the traversability explicitly while using prior knowledge provided by the classification-label approaches.

Visual features offer at best an indirect *estimate* of the traversability and even the best approaches are prone to changes in lighting conditions. An alternative is to use contact-based sensors such as accelerometers, gyroscopes and IMUs (Inertial Measurement Units). These measure properties such as acceleration, pitch, roll etc. They are quite sensitive to even minor variations in the terrain and therefore afford greater prediction accuracy. Typically, features are extracted from the data and used to characterize terrains, typically using terrain classes. However, the data from these sensors can be quite noisy which can impede good classification. More significantly, these features cannot be used for prediction as the sensing range of the contact-based approaches is limited to the extents of the robot (unlike cameras which can provide long range data measurements).

The advantages and the limitations of vision and contact-based approaches therefore suggest a *fusion* approach wherein the robot uses synchronized sensor data from both (and in general, multiple [16]) modalities. In this approach, the visual features can be correlated with accelerometric features and used to build a model which predicts traversability more robustly than either of the two approaches. We present one such approach and demonstrate its effectiveness using experimental results on a robotic platform equipped with visual and accelerometric sensors.

The rest of the report is structured as follows: We explore the design space of outdoor traversability prediction in Section 2. In Section 3, we describe our terrain traversability system formally while specifying the associated implications and assumptions. In Section 4, we describe the robotic system built for traversability prediction and various experiments to assess the prediction performance. We conclude with a discussion on alternative approaches and avenues for future work in Section 5.

2 Related Work

The approaches for predict terrain traversability fall broadly into three categories. The first category is based on *visual-features*. In these approaches, vision sensors collect image data which is processed for delineating traversable regions in images. One class of these approaches views the problem as one of obstacle detection and usually consider all (detected) obstacle-free to be equivalent. These approaches use visual appearance features from images and $3 - D$ characterization of the terrains (shape,extents) to model obstacles (or their absence). Early work in this area used apriori characterization of the environments [5, 10] but these work well when the terrain is constrained and the apriori characterizations can be used with relatively little modification. Some of the approaches augment the monocular image-based models [8] with terrain analysis from stereo data. This approach is quite popular, especially for off-road traversability prediction and planetary exploration [10]. Another set of approaches use the notion of *learning* the traversability using neural-networks [7], reinforcement-learning [9] and unsupervised-learning [1]. The work in [1] is interesting as it imposes no restrictions on the terrain model. Instead, the problem is posed as one of *affordance*. The robot attempts to characterize this affordance by the consequences of its traversal in the world and modifies the affordance appropriately. Another class of visual-feature based approaches focus on the differences between obstacle-free regions. The approach presented in the report squarely belongs to this category. Most of these approaches assume that the obstacles have been filtered out or are insignificant. The defining characteristic of these approaches is that they seek to impose a ranking among obstacle-free regions. This ranking can be propagated to higher-level functions of the robot for path planning [17]. This approach eliminates the need for modeling peculiarities of the training data in learning approaches and provides a more flexible characterization that goes beyond a discretized , class-based description of traversable regions.

The second category of approaches aim to characterize the terrain by its *contact-based* properties such as acceleration, jitter, pitch and roll. These properties are measured using a variety of sensors such as gyroscopes, IMUs, MSBs [11, 12, 13, 14]. The mapping to traversability is typically done using the learning approaches described above. While these methods provide the most direct method of characterizing terrains, the sensor data is often noisy and tends to make learning difficult.

The third category contains methods which employ both visual and contact-based methods in tandem for prediction. [2] use a LIDAR and stereo camera to detect obstacle regions. Another approach [17] propagates a load-bearing surface model into the world ahead using long-range stereo data and a surface deformation model. The approach in [18] estimates the height of vegetative surface and the underlying terrain model using multiple Markov random field models. This work is probably the closest in spirit to the approach presented in this report.

3 Terrain Traversability Prediction System

Our strategy is as follows: Training data is collected by allowing the robot to traverse different kinds of environments $\mathbf{E}$. One such environment is shown in Figure 2. The visual features as viewed from a particular position $\mathcal{P}^{(t)}$ are associated with accelerometric features of areas traversed subsequently $\mathcal{P}'^{(t')}$, producing sensor feature pairs. These feature vectors are shown as vertical striped bars in Figure 2. The sensor feature pairs are tagged with traversability labels. The tagged feature pairs are used to create a *traversability mapping* $\mathcal{F}$. In Figure 2, we assume that this is already available to us. Using this mapping, the robot *learns* to associate visual features (as viewed from a particular location in a possibly new environment) to predict the associated accelerometric features and concurrently, the traversability labeling ($\mathbf{L}_{\mathcal{P}^{(t)}}$ in Figure 2).

A robot capable of performing this task of predicting terrain traversability imposes certain fundamental requirements:

- The robot's cameras need to be calibrated – extrinsically and intrinsically – so that the association between visual and accelerometric features can be captured properly.
- The odometric, visual and accelerometric sensor streams must be synchronized appropriately to enable the association mentioned above.
- The robot's dimensions need to be known in order to determine its footprint in camera images.
- The visual and accelerometric features must be chosen carefully so as to sustain a good prediction capability in the system.

To make the problem tractable, we make certain assumptions, the primary among which are as follows:

- No negative obstacles: We assume that there are no obstacles which can potentially damage the robot such as ditches, tall obstructions, cliffs etc. We distinguish negative obstacles from other obstacles in that the latter do not pose a threat to the continuing functionality of the robot.
- Flat ground-plane : As mentioned before, we define the area of operation for the robot to be a flat $2-D$ surface. In the absence of negative obstacles, this is a reasonable assumption to make. In addition, it enables us to easily capture the association between visual features and the corresponding accelerometric features, thanks to the calibration of camera – extrinsic and intrinsic.

- The traversability labeling $\mathbf{L}_{\mathcal{P}^{(t)}}$ is restricted to represent what we define as *terrain classes*: `grass,gravel,concrete,road` (see Figure 1). Therefore, $\mathbf{L}_{\mathcal{P}^{(t)}} \in \{1, 2, 3, 4\}$.
- The accelerometric and visual properties are a function of the discretization parameters(step-size in each direction). Also, these properties are considered at the approximate locations of the wheels present on the front axle of the robot. We assume that these properties do not change within the physical extents occupied by the robot – an assumption which holds fairly well except at the inter-terrain-class boundaries and highly undulating terrains.
- The accelerometric features for a fixed pair of viewing and viewed positions e.g. $(\mathcal{P}^{(t)}, \mathcal{P}'^{(t')})$ above are deemed *repeatable*, i.e multiple traversals of the same position produce the same accelerometric readings.

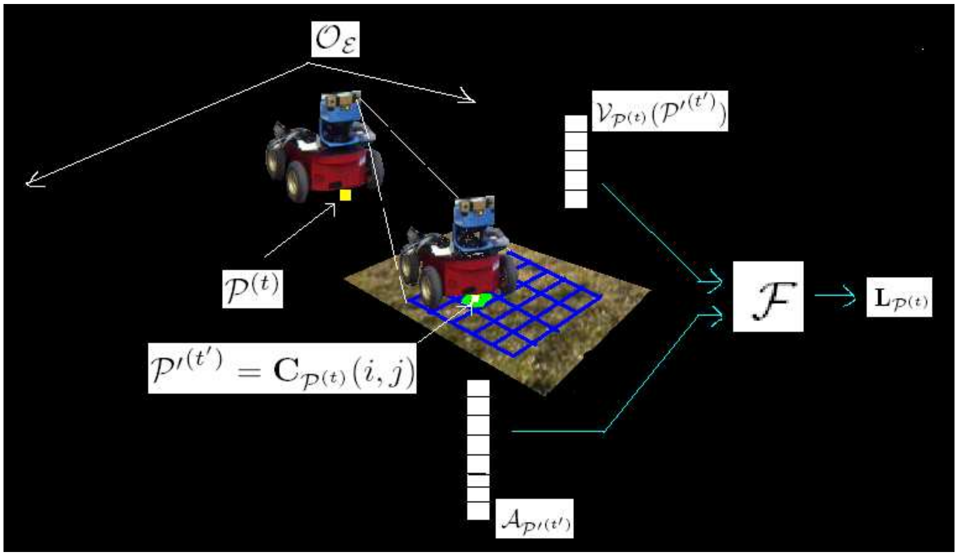

$\mathcal{O}_\mathcal{E}$
$\mathcal{V}_{\mathcal{P}^{(t)}}(\mathcal{P}'^{(t')})$
$\mathcal{P}^{(t)}$
$\mathcal{F}$
$\mathbf{L}_{\mathcal{P}^{(t)}}$
$\mathcal{P}'^{(t')} = \mathbf{C}_{\mathcal{P}^{(t)}}(i, j)$
$\mathcal{A}_{\mathcal{P}'^{(t')}}$

Figure 2: An illustration of the terrain traversability system. The robot is in an environment $\mathcal{E}$ at a position marked $\mathcal{P}^{(t)}$. Refer to Section 3 for notation. The visual features are correlated via $\mathcal{F}$ with the accelerometric features (both shown as vertical bars) to arrive at a labeling for the patch shaded in green.

We now describe the problem formally:

A typical environment $\mathcal{E} \in \mathbf{E}$ (See Figure 2) traversed by a robot is usually represented by a $3 - D$ coordinate system. For the purposes of the problem, however, we consider a $2 - D$ projection of this coordinate system, which is reasonable if we assume that the robot does not "fly off the projection plane". Let $\mathcal{O}_\mathcal{E}$ be the origin of this latter coordinate system. Let $\mathcal{P}^{(t)}$ (yellow square in Figure 2) be the position of the robot w.r.t $\mathcal{O}_\mathcal{E}$ at time t[1]. The robot has a stereo camera attached to it. Assume a $2 - D$ discretization $\mathbf{C}_{\mathcal{P}^{(t)}}$ corresponding to the camera's left eye field of view, i.e each grid cell $\mathbf{C}_{\mathcal{P}^{(t)}}(i, j)$ (filled in as a green patch in Figure 2) corresponds to a location referenced as $\mathcal{P}'$ w.r.t $\mathcal{O}_\mathcal{E}$. Each grid cell is associated with a set of visual properties $\mathcal{V}_{\mathcal{P}^{(t)}}(i, j)$. Note that these properties are defined relative to

[1] We ignore the physical extents of the robot for the time being and reduce its location to a point in $\mathcal{E}$

robot's current position. As the robot moves forward, it traverses positions viewed in the camera's field of view from a previous location. Specifically, let ${\mathcal{P}'}^{(t')}$ be the position of the robot at time t' $(t' > t)$ such that ${\mathcal{P}'}^{(t')} = \mathbf{C}_{\mathcal{P}^{(t)}}(i, j)$, i.e it is visible from an earlier location $\mathcal{P}^{(t)}$. Let $\mathcal{A}_{\mathcal{P}'^{(t')}}$ be the accelerometric properties measured at ${\mathcal{P}'}^{(t')}$. Define a *traversability labeling* $\mathbf{L}_{\mathcal{P}^{(t)}}$ over $\mathbf{C}_{\mathcal{P}^{(t)}}$ where $\mathcal{L}_{\mathcal{P}^{(t)}}(i, j) \in \mathbb{R}$. Define a *traversability mapping* $\mathcal{F}$ such that :

$$\mathcal{F} : \{\mathcal{V}_{\mathcal{P}^{(t)}}({\mathcal{P}'}^{(t')}), \mathcal{A}_{\mathcal{P}'^{(t')}}\} \rightarrow \mathbf{L}_{\mathcal{P}^{(t)}} \tag{1}$$

Assume that the robot is now in a hitherto unseen environment $\mathcal{E}'$ at position ${\mathcal{P}''}^{(t'')}$ at time t''. Define the coordinate systems and discretization grid as before. Assume a perfect[2] traversability labeling ${\mathbf{L}^{\mathbf{perf}}}_{\mathcal{P}''^{(t'')}}$ exists. The problem of *predicting terrain traversability* is to estimate $\mathcal{A}_{\mathcal{P}'''^{(t''')}}$ for all locations ${\mathcal{P}'''}^{(t''')}$ as seen from ${\mathcal{P}''}^{(t'')}$ and use $\mathcal{F}$ to find a labeling $\mathbf{L}_{\mathcal{P}''^{(t'')}}$ such that :

$$g(\mathbf{L}_{\mathcal{P}''^{(t'')}}, {\mathbf{L}^{\mathbf{perf}}}_{\mathcal{P}''^{(t'')}}) = 0 \tag{2}$$

where $g(\mathbb{R}^2, \mathbb{R}^2) \rightarrow \mathbb{R}$ is an error function that measures the discrepancy between the ideal labeling and one that is produced using the mapping function.

Special cases of the above model arise when vision and accelerometric features are used by themselves. Let us assume that only vision features are available. Equation 1 becomes :

$$\mathcal{F}^V : \{\mathcal{V}_{\mathcal{P}^{(t)}}({\mathcal{P}'}^{(t')})\} \rightarrow \mathbf{L}_{\mathcal{P}^{(t)}} \tag{3}$$

In this case, the problem of *predicting terrain traversability* does not involve any estimation of $\mathcal{A}_{\mathcal{P}'''^{(t''')}}$. Instead, for all locations ${\mathcal{P}'''}^{(t''')}$ as seen from ${\mathcal{P}''}^{(t'')}$, we use Equation 3 to find a labeling $\mathbf{L}_{\mathcal{P}''^{(t'')}}$ with the same property as given in Equation 2.

Similarly, when only accelerometric features are available, the traversability mapping function becomes :

$$\mathcal{F}^A : \{\mathcal{A}_{\mathcal{P}^{(t)}}({\mathcal{P}'}^{(t')})\} \rightarrow \mathbf{L}_{\mathcal{P}^{(t)}} \tag{4}$$

An alternative definition of the traversability mapping is :

$$\mathcal{F}^A : \{\mathcal{A}_{\mathcal{P}^{(t)}}(\mathcal{P}^{(t)})\} \rightarrow \mathbf{L}_{\mathcal{P}^{(t)}} \tag{5}$$

This is reasonable since we can measure the accelerometric properties at a particular position $\mathcal{P}^{(t)}$ (i.e directly beneath the robot) whereas the same cannot be done for visual features. However, for consistent evaluation, we use the mapping from Equation 4.

We model the learning of traversability mapping functions (e.g. Equation 1) as a multi-class classification problem, i.e. $\mathcal{F}$ from Equations 2, 3, 4 is considered to be a function learnt by the classifier. For our experiments, we use the Adaboost classifier [28]. Adaboost is a popular classifier for solving two-class classification problems and provides robust class predictions sans over-fitting. Suppose the training data-set has N labeled samples $\mathcal{D} = \{(x_1, t_1), \ldots (x_N, t_N)\}$ where t_j is the label of feature vector $x_j, j = 1, \ldots N$ and $x_j \in \{1, 2, \ldots K\}$. A simple extension of Adaboost for problems with K classes can be

[2] According to a reasonable metric of perfection.

obtained by training K two-class classifiers $C_1, C_2 \dots C_K$. We split the training data into two disjoint sets S_1 and S_2[3] ($S_1 \cup S_2 = D$) where S_1 contains all the samples belonging to terrain class i [4]and S_2 contains the rest of the data-set. These two sets are considered as two classes for training C_i, i.e for $j = 1, 2 \dots N$:

$$t_j^{new} = \begin{cases} -1 \text{ if } t_j \notin C_i \\ +1 \text{ otherwise} \end{cases} \tag{6}$$

Let θ_i be the parameters of the trained C_i. Given a test feature vector x, it produces a real-valued confidence $l_{C_i}^x$ where $-1 \leq l_{C_i}^x \leq 1$. x is labeled with the class t^x such that :

$$t^x = \arg\max_i l_{C_i}^x \tag{7}$$

where $i = 1, 2 \dots K$.

We can also obtain a probability distribution over multiple classes by the *softmax* transformation.

$$P(\omega_i|x) \propto exp\left(-kl(x;\theta_i)\right) \tag{8}$$

where k is a parameter that controls the shape of the distribution and $l(x;\theta_i) = l_{C_i}^x$.

The predicted class is determined in the same manner as Equation 7. Note also that we can consider $\theta = \{\theta_i, i = 1 \dots K\}$ as a parameterization of mapping functions $\mathcal{F}$ in Equations 2, 3, 4.

Let $P^{vis}(\omega_i|x)$, $P^{acc}(\omega_i|x)$ be the predictions for using only visual and accelerometric features respectively. A simple way of combining these predictions is :

$$P^{combined}(\omega_i|x) = P^{vis}(\omega_i|x) P^{acc}(\omega_i|x) \tag{9}$$

where ω_i takes the values of different terrain-classes.

Equation 9 implicitly assumes that all terrain classes contribute equally to the result. Also, it does not take into account the differing performance rates with visual and accelerometric features. Therefore, we derive a sensor fusion method that takes these factors into account. Note that Equation 9 provides a terrain-class conditional likelihood whose value forms part of the prediction process. To incorporate prior knowledge about the performance on different terrain classes itself, we model their posterior distribution.

$$P(\omega_i^{true}|x) = \sum_{m=vis,acc} \sum_j P(\omega_i^{true}|\omega_j^{est}, m, x) P(\omega_j^{est}|m, x) P(m|x) \tag{10}$$

where $P(\omega_i^{true}|x)$ is the true posterior distribution for x, $P(\omega_i^{true}|\omega_j^{est}, m, x)$ encodes performance of the classifier for a fixed m and $P(m|x)$ denotes an importance weighing factor per sensor modality. For simplicity, we set $P(m|x) = c, 0 \leq c \leq 1$ where c is a fixed constant to indicate the fact that we give equal importance to all the sensors. In our case $c = \frac{1}{2}$. We have :

$$P(\omega_i^{true}|x) \propto P(\omega_i^{true}|m = vis, x) + P(\omega_i^{true}|m = acc, x) \tag{11}$$

[3] Samples belonging to these sets are labeled -1 and $+1$ respectively

[4] Class i will henceforth be referred to interchangeably as ω_i.

Table 1: Features extracted per channel

Feature-code	Description
v1	Expected value of feature
v2	Expected standard deviation
v3	Entropy
v4	Bin centers from histogram (5 bins)

$$P(\omega_i^{true}|x) \propto \underbrace{\sum_j P(\omega_i^{true}|\omega_j^{est}, m = vis, x)}_{A} \underbrace{P(\omega_j^{est}|m = vis, x)}_{B} + \sum_j P(\omega_i^{true}|\omega_j^{est}, m = acc, x) P(\omega_j^{est}|m = acc, x) \qquad (12)$$

The term denoted by A in Equation 12 can be written as :

$$P(\omega_i^{true}|\omega_j^{est}, m = vis, x) = \frac{P(\omega_i^{true}, \omega_j^{est}|x)}{\sum_k P(\omega_k^{true}, \omega_j^{est}|x)} \qquad (13)$$

The ω_j^{est} is introduced so as to factor the effect of mislabelings into the prediction mechanism. The estimates seem coupled in the sense estimating $P(\omega_i^{true}|x)$ requires estimation of $P(\omega_k^{true}|x)\ \forall k \in \{1, \ldots K\}$. However, we can obtain an estimate for this from the confusion matrix corresponding to visual features. (E.g. Table 4). Let this confusion matrix be $\mathcal{M} = [m_{pq}], 1 \leq p, q \leq K$. We note that $m_{pq} = P(\omega_p^{true}, \omega_q^{est}|x)$. Therefore,

$$P(\omega_i^{true}|\omega_j^{est}, m = vis, x) = \left[\frac{m_{ij}}{\sum_k m_{kj}}\right] \qquad (14)$$

The term B in Equation 12 is the normal prediction (c.f. Equation 8). Therefore, the fusion strategy in Equation 12 helps bias the normal prediction with prior knowledge. We perform a similar computation when $m = acc$ for accelerometric features and combine them as shown in Equation 12.

In the next section, we describe different experiments that examine the performance of the prediction system with various combinations of sensor data, data fusion strategies and their impact on overall prediction.

4 Experiments

In order to evaluate the performance of the mapping functions and the predictive capabilities they provide, we used a robotic platform for conducting experiments (see Figure 3). The base robot is a Pioneer-2 AT ActivMedia [21] outfitted with SICK laser range finder and a wheel-tick odometer which runs at 10 Hz. A Bumblebee color stereo camera functions as the vision sensor [22] and provides images at around 2 fps [6]. Accelerometry is read from a Multi-Sensor Board (MSB) developed at Intel Research, Seattle [23],

[5] For consistency, the means were ranked by the number of samples closest to each mean

[6] The low fps is related to software issues. In isolation, the camera provides a 15 fps frame-rate

Table 2: Vision-based features

Feature	Number of features
$v1 - v4$ features - R,G,B channels	24
k-means on RGB values [5] ($k = 3$)	9
k-means on color constancy features - $\frac{R}{B}$, $\frac{G}{B}$	18
$v1 - v4$ features - Hue,Saturation, Intensity channels	24
k-means on HSV values($k = 3$)	9
DOOG texture features [20]	30

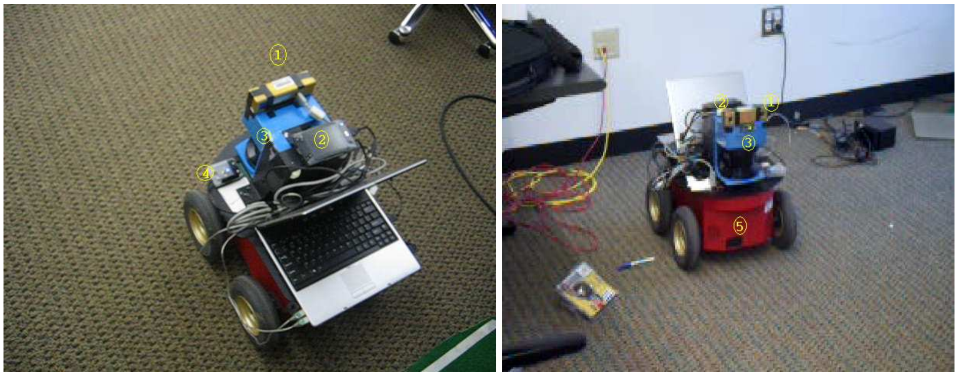


Figure 3: The robot platform built for data-collection and experiments. 1 = Bumblebee stereo camera, 2 = HP iPAQ PDA, 3 = SICK laser range-finder, 4 = MSB, 5 = Pioneer 2 AT robot

which functions as the accelerometry sensor. This board provides information on many sensor channels including barometric pressure, light frequencies, sound and acceleration. However, we use just the accelerometric readings along the canonical X, Y, Z axes for our experiments. The robot was tele-operated using a Logitech Wireless Rumblepad joystick [24]. A LinuxCertified LC-2000 laptop [25] runs the associated software for the robot. The different sensors and devices attached to the robot (odometer, laser range finder, joystick) are connected to the laptop via a USB interface. Because of the limitation on the number of USB devices supported on the machine, the MSB is run via software on a HP iPAQ PDA [26]. The software controls for the robot are provided via the CARMEN [27] program. The data collected by CARMEN includes information from odometry, laser range-finder,stereo image pair from the camera and the associated disparity image – all of which are time-stamped. These time-stamps are used to synchronize with the MSB data collected via the iPAQ.

Using this setup, we collected sensor data on 26 runs covering 4 different kinds of terrains – `road, grass, concrete, gravel` , under widely varying environmental conditions (see Figure 1). Each run consisted of collecting sensor data from the robot platform as the robot moved in the environment. The odometer provided location information relative to the starting position. The stereo camera simultaneously captured color images and these were associated with the odometric position. The accelerometric data collected from the iPAQ was synchronized with odometry and camera images using time-stamp information. Subsequently, feature extraction was performed on the visual data and accelerometric data.

In our experiments, we made the assumption that the ground plane is located a distance equal to the (known) height of the robot passing through its wheels. Using the camera calibration information, we overlay a virtual discretized grid on the ground plane (blue grid in Figure 2). The origin of the grid is adjusted to lie within the visual range of the camera on the robot. The discretized grid is then projected onto the camera images. Visual features are extracted from all the grid cells (patches). The relative odometry of the robot is also projected onto the image so that we can identify the grid occupied by the robot (filled green patch in Figure 2)[7].

We extract two kinds of visual features from each patch. The first kind operates on channels of image representations, in our case from RGB and HSV space. For each channel, we obtain a histogram whose bins correspond to the possible values of the channel. We used 10 bins in the range $0-1$. The histogram is normalized to obtain a probability distribution. We then compute the following features on the channel values for each patch: (i) expected value of channel (ii) expected standard deviation (iii) entropy (iv) 5-bin histogram centres – A 5 bin histogram of the channel values in the patch is obtained and the corresponding bin centres are used . We refer to these features as $v1 - v4$ features (see Table 1). The second kind of features are extracted over multiple channels and typically represent global properties of the patch. We also perform k-means clustering on RGB, HSV features and normalized color space – $\frac{R}{B}, \frac{G}{B}$ (see Table 2). The latter features are quite popular in color-constancy studies [19].

We extract accelerometric features from a patch which contain the projected footprint of the robot and associate them with the visual features from that patch. The accelerometric features are based on Fourier-frequency analysis of the sensor data from the MSB.

To summarize, we consider each patch containing the footprint of the robot, extract visual and associated accelerometric features as described above and finally label the features with the appropriate terrain class viz. `road, grass, concrete, gravel`.

4.1 Evaluating predictions on traversed paths

As a baseline set of experiments, we evaluated the performance only on terrain patches traversed by the robot. For these experiments, the labeled data was split randomly and 70% was used for training while the rest was used for evaluating the prediction performance.

4.1.1 Uniform lighting conditions

Table 3: Confusion Matrices for terrains - Uniform Lighting conditions (RD - `road`, GR - `grass`, CO - `concrete`, GV - `gravel`).

		RD	GR	CO	GV
Visual	RD	99.9401	0.0599	0	0
	GR	0	99.3674	0	0.6326
	CO	0	0	98.5877	1.4123
	GV	0.0155	0	0.1864	99.7980
Accelerometric		RD	GR	CO	GV
	RD	100.0	0	0	0
	GR	0	99.0438	4.0972	3.2873
	CO	25.6610	0.3629	73.5614	0.4147
	GV	1.4043	0.1404	0	98.4553

Uniform lighting conditions refer to the fact that the data was collected in the absence of

[7] Strictly speaking, it should be grids occupied by the front wheels of the robot. We consider the single patch corresponding to left front wheel in the report without loss of generality.

bright sunlight. Table 3 shows the confusion matrices when using visual features and accelerometric features. The high accuracies reflect the suitability of visual features, at least as a baseline set for prediction (see also Figure 4). The accelerometric labeling is also quite accurate. However, some of the terrains patches have incorrect predictions. This happens when the terrains share contact-based properties, e.g. a well-paved road and concrete surface can 'feel' quite similar to the robot (row 3 of Table 3).

4.1.2 Nonuniform lighting conditions

Table 4: Confusion Matrices for terrains - Nonuniform Lighting conditions (RD - road, GR - grass, CO - concrete, GV - gravel).

		RD	GR	CO	GV
Visual	RD	85.521100	0.059700	2.128100	12.291200
	GR	0.054100	96.661900	0.000000	3.284000
	CO	3.673600	0.000000	76.362100	19.964200
	GV	1.465100	4.213500	0.463000	93.858400
Accelerometric		RD	GR	CO	GV
	RD	91.139200	0.642600	6.504400	1.713700
	GR	1.064000	93.868300	0.577100	4.490500
	CO	2.416700	0.430600	96.583300	0.569400
	GV	0.634500	4.012500	0.111600	95.241500
Combined		RD	GR	CO	GV
	RD	91.695600	0.077100	5.086700	3.140700
	GR	0.000000	99.095200	0.000000	0.904800
	CO	0.859100	0.000000	98.739100	0.401800
	GV	0.135500	4.384500	0.023600	95.456400

Next, we relax the above assumption and evaluate the performance without any lighting-condition based constraint on the data. We also examine the effect of combining the predictions of visual and accelerometric classifiers. Table 4 shows the effect of large dynamic range in lighting typical of outdoor scenes (c.f. Table 3). Note however that the accelerometry performance remains unchanged. This invariance to lighting conditions makes contact-based sensors attractive from the viewpoint of prediction (Figure 5). Even when they are combined in a simple fashion such as that given by Equation 9, the prediction results are quite superior (see Table 4) to visual features or accelerometric features alone, highlighting the advantages of a correlation-based approach.

4.1.3 Evaluating sensor fusion strategies

Table 5: Confusion Matrices for terrains - Combining visual and accelerometric classifiers and using prior information (RD - road, GR - grass, CO - concrete, GV - gravel).

		RD	GR	CO	GV
Combined (with prior info)	RD	93.237000	0.115600	4.238900	2.408500
	GR	0.036200	99.475200	0.000000	0.488600
	CO	0.872900	0.000000	99.043900	0.083100
	GV	0.277000	4.761600	0.064800	94.896600

We use the combination strategy presented in Equation 12. The results are presented in Table 5. While the improvement is marginal, this could be due to the already high

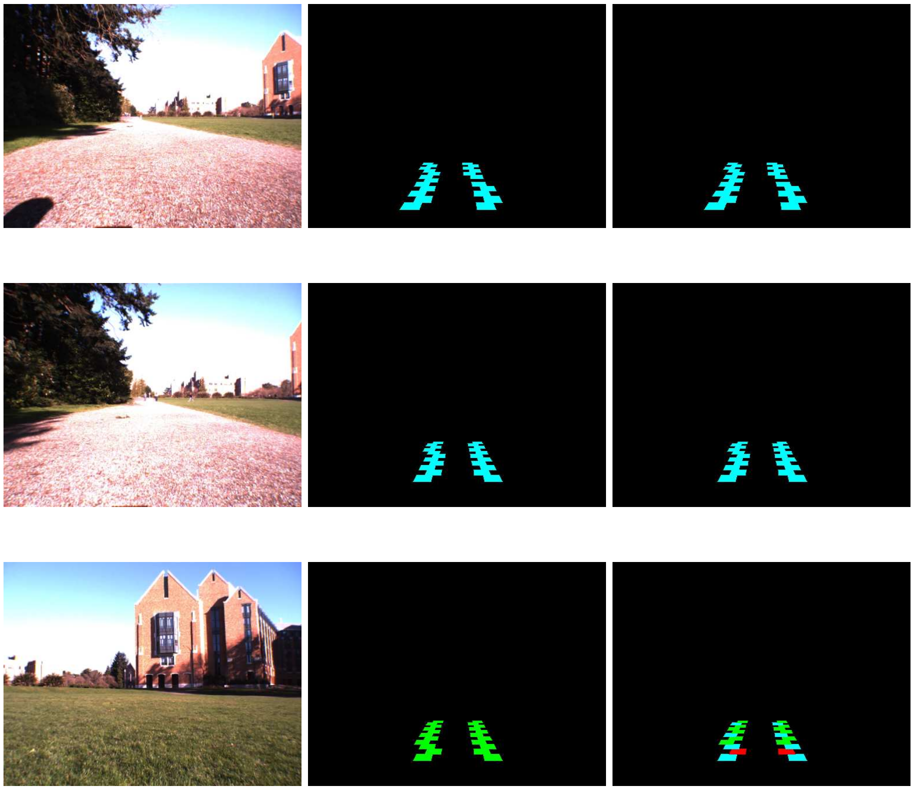


Figure 4: Images from the robot during a test run and the corresponding visual (column 2) and accelerometric (column 3) predictions *on the path taken later by the robot*. Terrain is indicated by color (green=`grass`, cyan=`gravel`, red =`road`). Notice that both visual and accelerometric features work well in this case.

prediction accuracies seen in Table 4. The strategy's effect might be more visible when a large number of terrain classes are present – a possibility we do not address in this report.

It is worthwhile to mention that in the above set of experiments, we have considered only those patches on the discretized grid which contain the physical footprint of the robot for a given position $\mathcal{P}^{(t)}$, i.e we consider a subset of $\mathbf{C}_{\mathcal{P}^{(t)}}$ for evaluation purposes. In the next set of experiments, we relax this assumption and consider the complete discretization grid $\mathbf{C}_{\mathcal{P}^{(t)}}$ for each odometric position of the robot.

4.2 Evaluating predictions on untraversed patches

The previous set of experiments were primarily concerned with determining the suitability of visual and accelerometric features wherein the evaluation was done over labeled patches traversed by the robot. We now consider the more realistic scenario of a hitherto unseen environment (i.e. not traversed by the robot) and examine the prediction performance for the same.

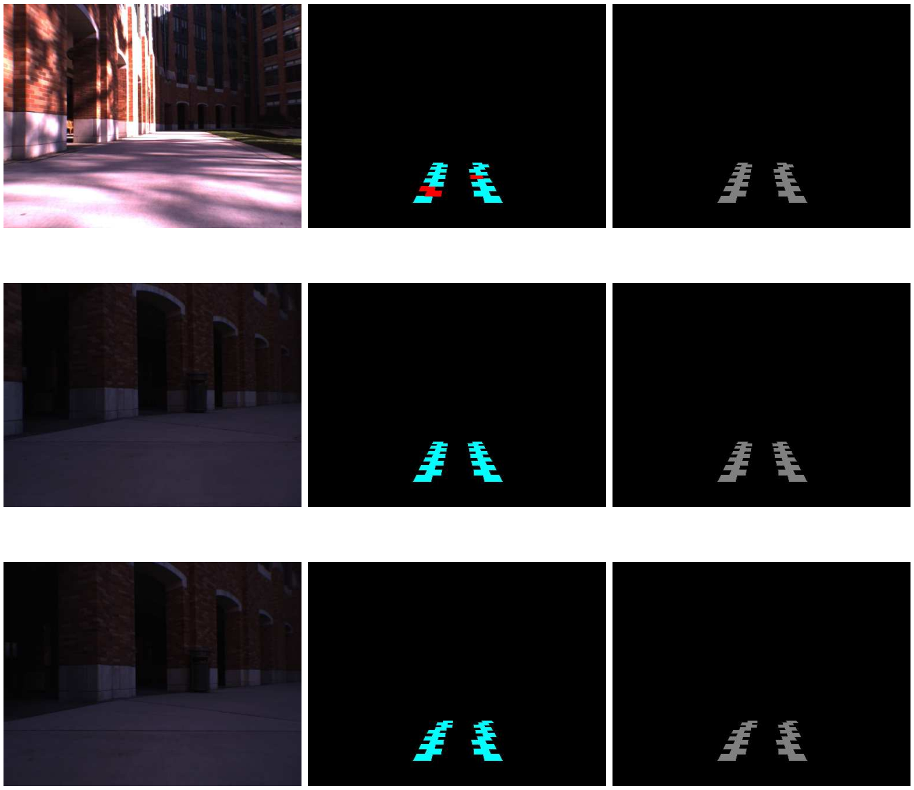


Figure 5: Images from the robot during a test run and the corresponding visual (column 2) and accelerometric (column 3) predictions *on the path taken later by the robot*. Terrain is indicated by color (green=grass, cyan=gravel, red =road, gray =concrete). Notice that many of the visual predictions are incorrect because of the large dynamic lighting range. This does not affect accelerometry which gives perfect prediction.

4.2.1 Determining desirable regions of traversability from visual features

Instead of a naive extension of the previous section's approach, i.e visual-feature based predcition for each patch in the grid, we consider an alternate approach. Let us consider a situation where the *manner* of collecting training data is relevant. For instance, the training data could consist of runs which lead to a goal. If the straight line path between starting position and the goal is desirable at all points along itself, the robot would have been guided down that route. However, the path to the goal might be circuitous because of environmental constraints and therefore the path need not be straight. At each point in the path, we can determine the straight line direction to the goal and the current heading of the robot. These lines can be projected onto the images taken from the current position. Image patches can be taken w.r.t these lines to identify 'desirable' and 'non-desirable' patches. Patches collected in this fashion can then be used to train classifiers to identify 'desirable' regions.

Let us consider a situation where we have images and $p = 3$ patches in every image identified as 'desirable' by the method mentioned above. By extracting suitable features, we can determine other such regions in the image and 'grow' the desirable patches into

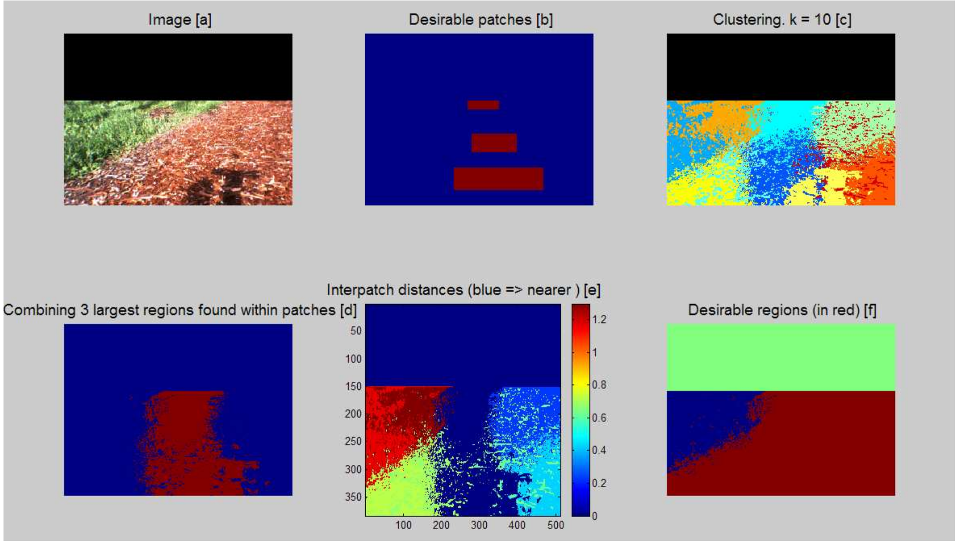


Figure 6: Procedure for extending pre-determined desirable regions to rest of image.

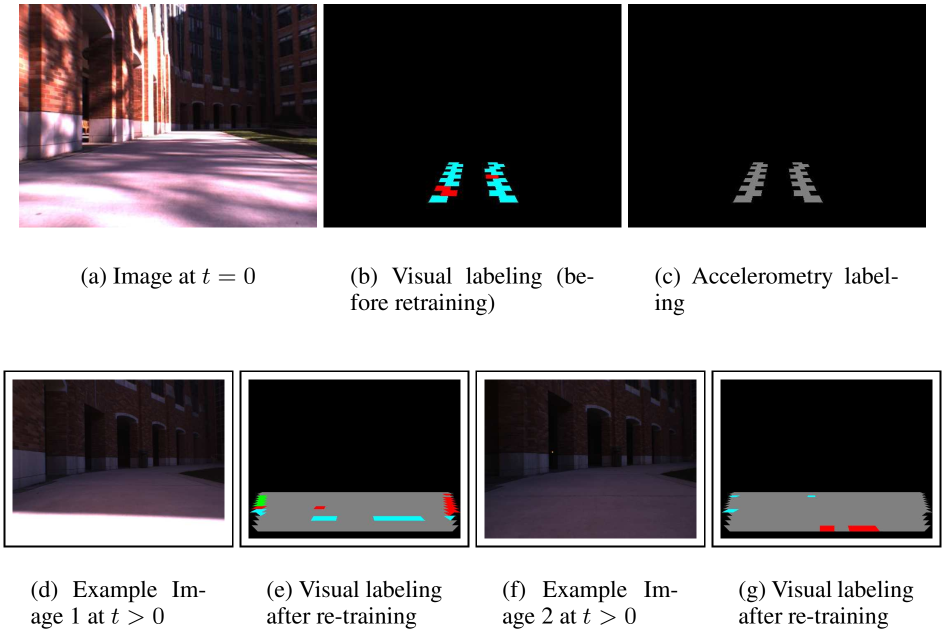
regions which can be labeled 'desirable'. We use the following procedure:

1. Typically we are interested in desirable regions similar in appearance to the 3 designated patches (Figure 6(b)). The appearance properties of regions further away from the robot can be different from those nearer to it and therefore, we crop the top portion of the image corresponding to regions closer to horizon w.r.t robot's current position. (Figure 6(a))
2. Extract the following features from each pixel of the image: Red channel value, Green channel value - g, Blue channel value - b, Hue channel value, Color channel ratio : $\frac{g}{b}$, pixel's row, pixel's column[8].
3. Perform k-means clustering on the feature vectors. We use $k = 10$. (Figure 6(c))
4. Identify clusters which intersect with the 3 `desirable` patches and mark them desirable. (Figure 6(d))
5. Extract features as in (2) above from all the clusters and compute the mean feature vector.
6. Compute the Euclidean distance between the feature vector from the desirable clusters and the remaining clusters. (Figure 6(e))
7. Choose a threshold and merge the clusters closest in distance to the `desirable` clusters to obtain all the desirable regions in the image. (Figure 6(f)).

While this method works well in general, it occasionally fails when image regions are saturated or the desirable patches have a large dynamic range in appearance. It must be mentioned that advanced color-constancy methods exist which can handle some of these issues but even these typically do not compensate the full range of lighting conditions such as those seen in Figures 1.

[8] Including the pixel's row and column as features enforces spatial consistency on desirable regions.

4.2.2 Using accelerometric predictions to improve visual classifier predictions



(a) Image at $t = 0$

(b) Visual labeling (before retraining)

(c) Accelerometry labeling

(d) Example Image 1 at $t > 0$

(e) Visual labeling after re-training

(f) Example Image 2 at $t > 0$

(g) Visual labeling after re-training

Figure 7: The accelerometric predictions in (c) are used to re-train the visual classifier. The results after re-training are shown in the second row (red=`road`, green = `grass`, cyan = `gravel`, gray = `concrete`.

As the previous set of experiments show, the visual feature-based predictor is sensitive to lighting conditions while this is not the case with accelerometry. Therefore, we can use the predictions from the accelerometry-based predictor and use them to revise the decisions of the visual feature-based predictor, thereby improving the performance of the latter. To verify this, we performed an experiment (see Figure 7) wherein the robot initially moves some distance. The vision-based classifier initially labels the patches traversed by the robot. In an environment with mixed lighting conditions such as those in the figure, it does a poor job (Figure 7(b)). In contrast, the accelerometry-based predictor is quite accurate in recognizing the terrain (Figure 7(c)). Therefore, we use these predictions to ”re-label” the traversed patches. To perform this re-labeling, we need to first re-train the visual-feature classifier with the patches it had previously misclassified, but now using the (accurate) labels from accelerometry predictions. Let this data be $\mathcal{D}^{re-train} = \{(x'_1, t'_1), \dots (x'_N, t'_N)\}$. Retraining follows the same procedure as in Section 4.1, except that existing decision stumps (which form part of the parametric representation in Adaboost) are modified so as to reduce the overall empirical training error [29]. However, this method of on-line learning requires a on-line weak learner for Adaboost. We use a decision stump for training but this is done offline. Therefore, we re-train the visual classifier with the old training data $\mathcal{D}^{old}$ augmented with $\mathcal{D}^{re-train}$. We first define a distribution over the entire data :

$$w_x = \begin{cases} \alpha \text{ if } x \in \mathcal{D}^{re-train} \\ \frac{\alpha}{100} \text{ if } x \in \mathcal{D}^{old} \end{cases} \quad (15)$$

where w_x is the weight or *importance* of training sample x. The above distribution reflects the intuition that the recently re-labeled data $\mathcal{D}^{re-train}$ should be modeled with greater certainty. As the robot proceeds ahead, the parameters of re-trained visual classifier show improved prediction (see second row in Figure 7). In general, this approach works very well in spite of the sensor-specific limitations of visual and accelerometric classifiers.

This success of this approach depends to a large extent on the accuracy of the accelerometric classifier. In some cases, the classifier exhibits mislabeling which can "bleed" into visual labeling via re-training. This is often due to the similar contact-based properties two surfaces exhibit. In the next section, we discuss ideas for future work that address these issues.

5 Conclusions

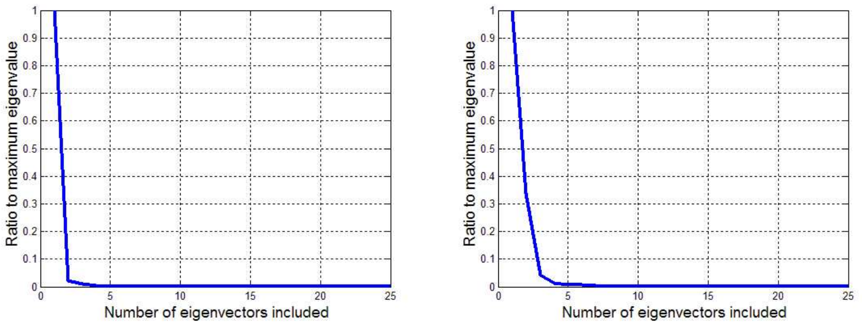
In this report, we have presented a system which learns the correlation between accelerometric and associated visual properties of terrains for predicting traversability. As the experiments, particularly the last set demonstrate, better predictions can be obtained by combining the best approaches from vision-only and accelerometry-only approaches in an intelligent yet straightforward fashion. The current work has plenty of scope for improvement :

- Our current model implicitly assumes that adjacent terrain patches are spatiotemporally independent. A better and more consistent prediction can be obtained with smoothing.
- In our work, we have implicitly assumed absence of significant obstacles. In reality, this might not be the case and therefore, stereo and laser based features could be added to the existing set of vision-based features. Extending the model to handle obstacles can be done by associating these features with the lowest measure of traversability.
- The MSB which provides accelerometric data could be augmented with other contact-based sensors such as gyroscopes and IMUs to enhance the correlation.
- The image data can serve as a corpus for exploring color-constancy features which can be usefully added to current set of features.
- In order to re-train the classifier(Adaboost), the current model reuses the *entire* training data along with new training input. Instead of this naive method, it would be better to use an online-learning approach which can modify the existing set of learnt parameters with incoming data for prediction [29].
- A continuous measure of traversability may be more appropriate in this scenario as opposed to discrete class-based approaches. One possible approach could be to set up a regression model between the vision-based and accelerometric features so that given the former, the latter can be predicted for a novel terrain patch. Although the feature dimensions involved are quite large (112 and 267 respectively), dimensionality reduction techniques could be applied. Figure 8 shows plots of the associated eigenvalues.

  Another alternative could be to perform regression and dimensionality reduction simultaneously using methods such as partial least squares regression [30]. Given the complex nature of terrains and associated accelerometric information, the regression model may perform poorly. In this case, the features can be clustered wherein each cluster is associated with notions of *ease* of traversability. To verify this, we performed a preliminary experiment in clustering the dimensionality-reduced accelerometric features. After clustering, each feature was associated with the closest cluster. Figure 9 shows how the samples of a given terrain (x-

axis) are distributed across clusters (y-axis). The clusters tend to maximally model a single terrain class while spanning all the terrain classes.

- As can be seen in Figure 1, the appearance of a terrain patch changes as a function of its distance of the robot. Therefore, instead of burdening a single classifier to learn appearance over a range of distances, we could have multiple classifiers, each of which learns appearance for a specific distance or a much smaller range of distances.
- Occasionally, the visual-feature classifier may perform better than the accelerometric classifier for a given area. For instance, if the ground underlying a grass patch is very uneven, the accelerometric classifier tends to classify it as `gravel`. Therefore, it would be useful to have a *meta-predictor* which adaptively enables and disables re-training of visual-feature classifier.



(a) #Original dimensions = 112

(b) #Original dimensions = 267

Figure 8: PCA eigen-value plot for visual(left) and accelerometric(right) features. Notice that the so-called knee of the plot is achieved at a relatively small dimension.

6 Acknowledgments

I would like to thank Dieter and Steve for being great advisors and critics. It has been a great entry into grad-level research with them. Dirk Haehnel (Intel Research, Seattle) was instrumental in the set-up of the robot platform and provided much needed support in using the CARMEN software. David Duhon (U.Mass, Amherst) was involved in preliminary experimentation and building the robotic platform for data-collection. Jiwon Kim helped out with stereo camera accessories. Adding the stereo camera module to the data-collection system would have been a difficult task if not for the generous advice and help from the support team at Point Grey Research, manufacturers of the Bumblebee stereo camera. I would also like to thank Alvin Raj, Brian Ferris for providing a glitch-free version of iPAQ-MSB software and Pravin Bhat for outdoor videos of the robot. Finally, I would like to acknowledge the useful discussions with team SRI – Bob Bolles, Moti Agrawal and Kurt Konolige.

References

[1] “Traversability classification using unsupervised on-line visual learning for outdoor navigation”, Dongshin Kim, Jie Sun, Sang Min Oh, James M. Rehg, Aaron F. Bobick,

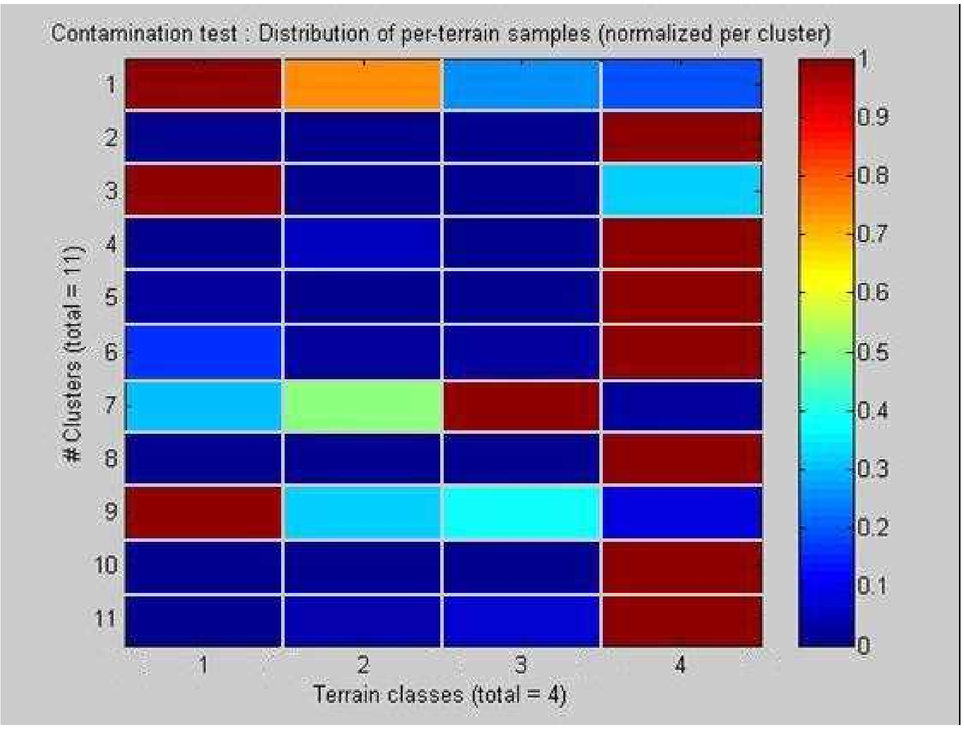


Figure 9: Clustering of accelerometric features (dimensionality-reduced). The color indicates the degree of association (red implies most and blue implies least)

IEEE 2006 International Conference on Robotics and Automation (ICRA-2006), Orlando, Florida.

[2] "Obstacle Detection and Terrain Classification for Autonomous Off-Road Navigation", R. Manduchi, A. Castano, A. Talukder, L. Matthies, *Autnomous Robots 18, 81-102, 2005*.

[3] "Autonomous navigation in ill-structured outdoor environment",J. Fernandez , A. Casals , *Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on , vol.1, no.pp.395-400 vol.1, 7-11 Sep 1997*.

[4] "Vision for Mobile Robot Navigation : A Survey", G. N. DeSouza, A. C. Kak, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI-2002), Vol. 24, No. 2, Feb 2002*.

[5] "Road detection for robot navigation ", G. A. Cervantes, M. Devy, R.M. Cid, *Proc 3rd International Symposium on Robotics and Automation, ISRA 2002*, Toluca, Mexico.

[6] "A View-Based Outdoor Navigation Using Object Recognition Robust to Changes ofWeather and Seasons", Hiroaki Katsura, Jun Miura, Michael Hild, and Yoshiaki Shirai, *Proc. 2003 IEEE-RSJ Int. Conf. on Intelligent Robots and Systems, pp.2974-2979, 2003*, Las Vegas.

[7] "Evolution of an artificial neural network based autonomous land vehicle controller", S. Baluja, *IEEE Transactions on Systems, Man and Cybernetics, Part B, Vol. 26, No. 3, June, 1996, pp. 450 - 463*.

[8] “Outdoor autonomous navigation using monocular vision”, Royer, E.; Bom, J.; Dhome, M.; Thuilot, B.; Lhuillier, M.; Marmoiton, F., *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol., no.pp. 1253- 1258, 2-6 Aug. 2005.*

[9] “High Speed Obstacle Avoidance using Monocular Vision and Reinforcement Learning”, J Michels, A Saxena, AY Ng, *Proceedings of the 22nd international conference on Machine learning(ICML-2005), pp. 593-600*, Bonn, Germany.

[10] “Stereo-based tree traversability analysis for autonomous off-road navigation”, A. Huertas, L. Matthies, and A. Rankin, *In IEEE Work-shop on Applications of Computer Vision, 2005.*

[11] “Vibration-based Terrain Analysis for Mobile Robots”, Brooks, C.; Iagnemma, K.; Dubowsky, S.,*Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol., no.pp. 3415- 3420, 2-6 Aug. 2005*

[12] “Learning from accelerometer data on a legged robot.”, D. Vail, M. Veloso ,*In Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal, 2004.*

[13] “ Visual and Tactile-Based Terrain Analysis Using a Cylindrical Mobile Robot”, G. Reina, M. M. Foglia, A. Milella, A. Gentile, *Journal of Dynamic Systems, Measurement, and Control – March 2006 – Volume 128, Issue 1, pp. 165-170.*

[14] “Vibration-based Terrain Classification for Planetary Rovers”, Brooks C. , Iagnemma K , *IEEE Transactions on Robotics, Vol. 21, No. 6, pp. 1185-1191, December 2005.*

[15] “ Recent progress in local and global traversability for planetary rovers”, Singh S, Simmons R, Smith T. , Stentz A., Verma V., Yahja A., Schwehr K, *Proceedings. ICRA ’00. IEEE International Conference on Robotics and Automation, 2000.*

[16] “Classifier Fusion for Outdoor Obstacle Detection”, Cristian S. Dima, N. Vandapel, M. Hebert, *Proocedings. ICRA 2004. IEEE International COnference on Robotics and Automation, 2004.*

[17] “Path Planning with Hallucinated Worlds”, B. Nabbe, S. Kumar, M. Hebert, *Proceedings: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, October, 2004.*

[18] “Interacting Markov Random Fields for Simultaneous Terrain Modeling and Obstacle Detection”, *Proceedings of Robotics: Science and Systems, 2005*, Cambridge MA, USA.

[19] “Color Constancy for Landmark Detection in Outdoor Environments”, Eduardo Todt, Carme Torras, *Proceedings of Eurobot (2001).*

[20] “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics”, Martin. D., Fowlkes. C., Tal. D., and Malik. J , *In Int. Conf. on Computer Vision, vol. 2, 416-423.*

[21] ActivMedia Robotics : `http://www.activmedia.com`

[22] Point Grey Research, Vancouver, Canada : `http://www.ptgrey.com`

[23] Intel Research, Seattle : `http://www.intel-research.net/seattle/`

[24] Logitech International : `http://www.logitech.com`

[25] Linux Certified Computers : `http://linuxcertified.com`

[26] HP iPAQ : `http://www.hp.com/country/us/en/prodserv/handheld.html`

[27] The CMU CARMEN repository : `http://www.cs.cmu.edu/∼carmen`

[28] “A decision-theoretic generalization of on-line learning and an application to boosting”, Y.Freund, R.E.Schapire, *European Conference on Computational Learning Theory, 1995*, .

[29] “Online Bagging and Boosting”, Nikunj C. Oza and Stuart Russell, *Eighth International Workshop on Artificial Intelligence and Statistics(AISTATS). January 4-7, 2001.*

[30] “Partial Least Squares (PLS) Regression”, Herve Abdi, `http://www.utdallas.edu/~herve/Abdi-PLS-pretty.pdf`