2012 IEEE International Conference on Robotics and Automation
RiverCentre, Saint Paul, Minnesota, USA
May 14-18, 2012

Indoor Robotic Terrain Classification Via Angular Velocity Based Hierarchical Classifier Selection

David Tick, Tauhidur Rahman, Carlos Busso, and Nicholas Gans

Abstract— **This paper proposes a novel approach to terrain classification by wheeled mobile robots, which utilizes vibration data. In our proposed approach, a mobile robot has the ability to categorize terrain types simply by driving over them. Classification of terrain is based on measurements obtained from an inertial measurement unit strapped directly to the robot's chassis. In contrast to the previous approaches, we use acceleration and angular velocity measurements in all cardinal directions to extract over 800 features. Sequential Forward Floating Feature Selection is used to narrow down this large group of features to a set of 15 to 20 that are the most useful. The reduced set of features is used by a Linear Bayes Normal Classifier to classify terrain. Furthermore, different feature sets are generated for different velocity conditions, and the classifier switches based on the current robot velocity. Experimental results are presented that show the strong performance of the proposed system, including 90% accuracy over 20 continuous minutes of driving across different terrains.**

I. INTRODUCTION

In the last decade, the topic of robotic terrain classification (RTC) has become an increasingly active field of research. Identifying terrain can allow a mobile robot to avoid dangerous regions or choose among control options. We propose that identifying terrain can aid in localization and mapping. If the currently identified terrain is known to exist only in a certain location, the probable location of the robot can be narrowed down. It may also be possible to adjust the location estimator based on characteristics of the identified terrain.

Approaches to solve the RTC problem include physics-based probabilistic models [1], principle component analysis [2], Linear Discriminant Analysis [3], probabilistic neural networks [4]–[7], support vector machines [8], [9], and K Nearest Neighbor [10]. These works have yielded some impressive results with respect to accuracy, but the comparisons of the approaches have been inconclusive [10], [11].

Many of the previous approaches use vibration data from an inertial measurement unit (IMU), but typically use two or three signals. In contrast to the previous approaches, we use all six acceleration and angular velocity measurements. These measurements are used to extract over 800 features. Sequential Forward Floating Feature Selection is used to narrow down this large group of features to a set of 15 to 20 that are the most useful for classifying terrain. The reduced set of features is used by a Linear Bayes Normal Classifier to classify terrain. In addition, it is observed that correct classification is dependent on the angular velocity of the robot. Therefore, we train the system using data collected under three turning conditions to choose a classifier based on the current angular velocity.

This work was not supported by any organization D. Tick is with the Department of Computer Engineering; T. Rahman, C. Busso, and N. Gans are with the Department of Electrical Engineering, University of Texas at Dallas, Dallas, TX, USA dqt081000@utdallas.edu, txr100020@utdallas.edu, busso@utdallas.edu, ngans@utdallas.edu

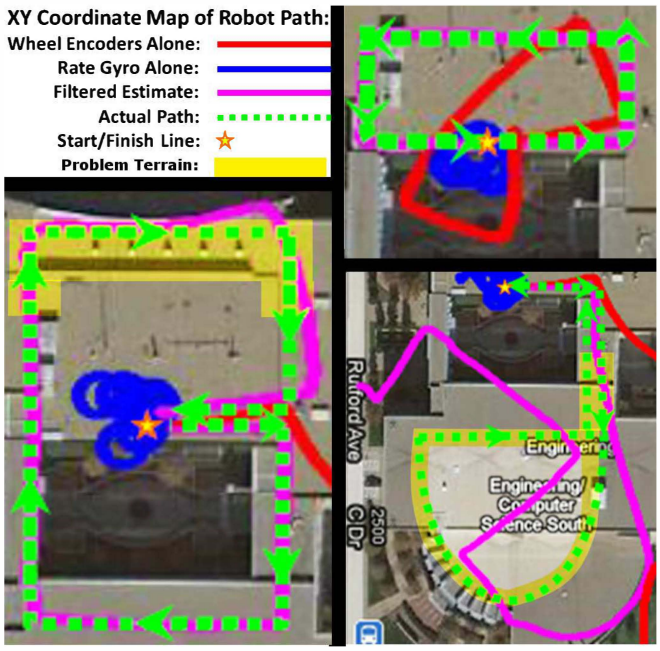

Fig. 1: Top Left: Legend. Top Right: Trial of system (Approx. 250 meters long) where the robot drove only on smooth tiled linoleum (2nd floor of the ECSC). Bottom Left: Trial of system (Approx. 500 meters long) where the robot drove mostly on smooth tiled linoleum, then drove over a short stretch of problematic terrain. Bottom Right: Trial of system (Approx. 750 meters long) where the robot drove on problematic terrains for a long period of time.

To our knowledge, these techniques have not been used for RTC, and this represents a rigorous approach to the problem. We conduct a long-duration experiment of nearly 20 minutes that covers five terrain types at various linear and angular velocities. Despite these challenging experimental conditions, identification accuracy using our approach is over 89%, including distinguishing very similar terrains.

Most of the previous literature is interested in improving control of mobile robots by identifying terrain, which can affect the slippage of wheels, orientation of the robot, etc. The research presented here is motivated by a different problem, localization. In the course of developing a novel sensor-fused odometry system [12]–[14], it was discovered that our system did not produce an accurate localization estimate when operating on certain types of problematic terrain (see Fig. 1 for details). The problematic terrains are

terrazzo and two types of ceramic tiles. These terrains are significantly more slippery and/or bumpier than the very smooth and tractionable tiled linoleum. Because of these terrain properties the localization estimate drifts whenever the robot runs over them. Extended duration on these problematic terrains causes dramatic errors. The images in Fig. 1 are representations of the localization estimates produced by the localization system presented in [13] that uses only IMU and wheel odometry fused via an Extended Kalman Filter [15], [16]. The problematic terrains clearly cause a drift of the system, which works well on linoleum.

If the robot can accurately detect when it is on a problematic terrain, then ideally, it can compensate for the induced drift. Furthermore, if it can classify terrains by type, then a separate compensation can be tailored for each terrain. Classification of the terrains may also allow the robot to detect areas of a region which exhibit unique anomalies of sufficiently small area to be useful as recognizable landmarks in a terrain based SLAM.

Section II presents an expanded literature review, discussion of the robot model, and information about the employed classification and feature extraction and selection algorithms. Section III discusses our novel application of classifier training techniques. Section IV describes a large scale experiment conducted in a chaotic and realistic deployment environment, which demonstrates the high degree of robustness, utility, and accuracy of our RTC methods. Results from this experiment are presented and discussed. Section V presents conclusions and potential avenues for future work.

II. BACKGROUND

A. Related Work

While RTC has received some notable attention, it remains very much an open problem. Iagnemma et al. presented a probabilistic model that could be used for controlling a robot based on what type of terrain it is on [1]. Iagnemma and his coauthors presented a number of related works. Of particular note is [2], which established the utility of vibration, visual, and tactile information in solving the RTC problem.

Brooks et al. propose an RTC technique based on vibrations measured by an accelerometer rigidly attached to the chassis of a rover [3]. The authors use Linear Discriminant Analysis to train a classifier offline, then use the resulting discriminant to classify or label terrain online. Primary Component Analysis is used to reduce the feature set of the training data. Sadhukhan, Moore, Collins, DuPont, Coyle, and Roberts collaborated in a series of projects that used Probabilistic Neural Networks (PNN) as a classification technique [4]–[7]. The focus of their work is adaptive switching control for a mobile robot operating outdoors. Weiss et al. proposed a Support Vector Machine using Power Spectral Density Analysis with a Support Vector based representation of the data [8], [9]. Weiss et al. compared the performance of their SVM based techniques with the PNN method forwarded by Moore and Collins [10]. They also investigate the performance of a K Nearest Neighbor instance based classifier in [10], as well as probabilistic methods and decision trees. Giguere and Dudek compared the performance of both supervised (Artificial Neural Network) and unsupervised (Gaussian Mixture Models, Hidden Markov Models, K-means, and Cost Function Analysis) learning methods [11].

Previous work toward solving the RTC problem has yielded impressive results with respect to accuracy. However, there appears to be no long-duration experiments in realistic operating scenarios. In the work outlined above, test experiments are generally rigidly controlled scenarios where a robot is set to a constant velocity on a straight line over one terrain type at a time. Brooks et al. do tests where a rover is driven over three different outdoor terrains in one traverse, but velocity and orientation are kept constant during each test [3]. Weiss et al. collect data at multiple speeds and while turning in [8]. However the data is collected by manually dragging a cart rather than a robot, also their algorithm is tested on held out training data.

In our view, the work in Giguere and Dudek [11] presents the most practical demonstration of RTC. A two-wheeled differential drive robot, equipped with a tactile probe, is placed on a small carpet surrounded by a hard floor. The robot runs a control algorithm that detects when the robot has left the carpeted area and then returns to it. It should be noted that the actual classifier performance was tested and evaluated offline using a mixture of held out training data and synthetically generated data.

B. Robot Motion and Inertial Measurement Unit

One common approach to the problem of mobile robot localization is to utilize a strap-down Inertial Measurement Unit (IMU) [17]–[19]. A strap-down IMU consists of a set of linear accelerometers and rate gyroscopes attached rigidly to a robot that measure inertial forces induced by motion of the robot. IMUs make measurements in a moving body frame, here called $\mathscr{F}_b$. At each sampling time, the IMU measures linear acceleration and angular velocities. These measurements are then rotated and integrated over time to obtain localization with respect to a static world frame, here called $\mathscr{F}_w$. In this work, we take $\mathscr{F}_b$ and $\mathscr{F}_w$ to have the same origin and aligned axes at initial time t_0. The z-axis of both frames is oriented up from the ground plane for all time. These frames are illustrated in Fig. 2.

The body reference frame, $\mathscr{F}_b$, is attached to the robot's center of rotation, with the x-axis aligned along the robot's heading, the y-axis oriented toward the robot's left side along the left wheel axle, and the z-axis oriented upwards. An IMU is rigidly attached to the robot, with its axes and origin aligned to match $\mathscr{F}_b$ as shown in Fig. 2.

The robot follows the kinematic unicycle model [20]. Thus, the robot moves in a plane spanning the x and y axes of $\mathscr{F}_w$ and $\mathscr{F}_b$. The robot has two degrees of freedom and nonholonomic velocity constraints. It can rotate about the z-axis of $\mathscr{F}_b$ with an angular velocity $\omega_z \in \mathbb{R}$ and translate along the x-axis of $\mathscr{F}_b$ with a linear velocity $v_x \in \mathbb{R}$.

The location of the robot at time t in $\mathscr{F}_w$ is the origin of $\mathscr{F}_b(t)$, given by $[x, y, 0]^T$. The orientation of the robot is

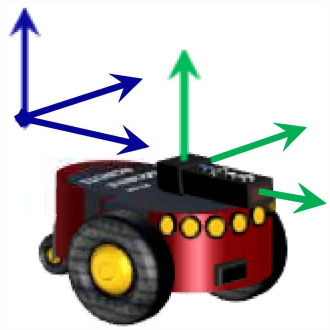

Fig. 2: Robot Body Frame $\mathscr{F}_b$ and Static Inertial World Frame $\mathscr{F}_w$.

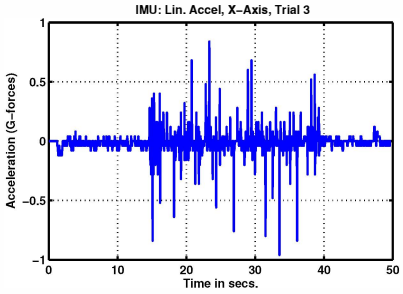

Fig. 3: Raw IMU signal: Linear Acceleration along X axis during Trial 3.

θ, and is measured by the angle between the x-axes of $\mathscr{F}_w$ and $\mathscr{F}_b$. The instantaneous linear and angular velocities of the body frame $\mathscr{F}_b$, as measured in the body frame $\mathscr{F}_b$ at time t, are described as a pair of vectors (v_b, ω_b), where $v_b \in \mathbb{R}^3$ and $\omega_b \in \mathbb{R}^3$. Note that while $v_b \in \mathbb{R}^3$ and $\omega_b \in \mathbb{R}^3$, the commanded motion of the robot itself is described by the pair of scalars $(v_x \in \mathbb{R}, \omega_z \in \mathbb{R})$. That is, the robot can only actuate its motion along two axes, however it can measure movement along all six axes. Due to friction, slippage, vibration, etc, all six velocity terms may be non-zero.

Fig. 3 shows an example of a time domain signal from the IMU. This time domain signal illustrates a simple experiment in which the robot was driven first over smooth tiled linoleum, then over raised ceramic tiles, then back over smooth tiled linoleum again. The large vibrations in the signal during the time period stretching from 15 seconds to approximately 40 seconds can be clearly seen. This is the same time period that the robot was on the ceramic tiles.

We can also look at the frequency domain of the same IMU signal as a function of time in order to find features as well. This is calculated, using a spectrogram, shown in Fig. 4. The more power in the signal at a given time, the larger the inertial disturbance being measured along that inertial axis. Assuming that the robot's linear velocity and orientation remain constant as it traverses that terrain, then Figs. 3 and 4 show that a strong correlation exists between the specific terrain deformations of a region and the changes that clearly occur in both the time and frequency domain of the IMU's output signal.

C. Linear Bayes Normal Classifier

Bayesian classifiers look for patterns that can be predicted based on previous knowledge of similar datasets. They separate instances of a class using a priori probabil-

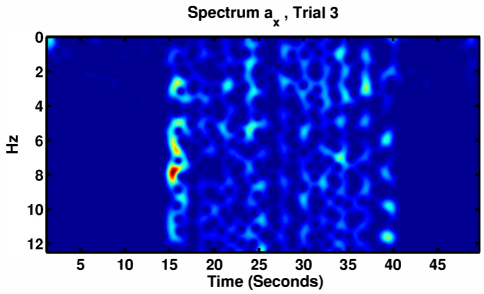

Fig. 4: Spectrogram of raw IMU signal: Linear Acceleration along X axis during Trial 3.

ities in the form a conditional probability density function (CPDF). Bayesian classifiers estimate the boundary between the classes so that the Bayes error is minimized. This error metric is considered to be optimal, assuming that the probability distributions are representative of the features. By minimizing the Bayes error, Bayesian classifiers ensure the maximum class separability. Let θ_t represent the class, and X represent the observations, then Bayes's Rule says

$$P(\theta_t|X) = \frac{p(X|\theta_t)P(\theta_t)}{p(X)} \tag{1}$$

where $P(\theta_t|X)$ is the conditional probability distribution, $p(X|\theta_t)$ is the likelihood, $P(\theta_t)$ is the priori distribution of the classes, and $p(X)$ is the distribution of the observation. Classification is done by selecting a class that maximizes the a posteriori probability $P(\theta_t|X)$ according to the Maximization of A Posteriori (MAP) decision rule

$$p(X|\theta_t)P(\theta_t) = \max_j \left\{ p(X|\theta_j)P(\theta_j) \right\}. \tag{2}$$

The CPDF for each class can be estimated either parametrically or non-parametrically. An accurate nonparametric density is hard to estimate, especially when the number of features is high. One of the most popular parametric probability density functions is the normal distribution, which only requires an estimate of the mean M_t and covariance matrix Σ_t for the class θ_t

$$\begin{aligned} p(X|\theta_t) \quad = \quad & \frac{1}{(2\pi)^{N/2}|\Sigma_t|^{1/2}} \\ & \times \exp\left\{ -\frac{1}{2}(X-M_t)^T \Sigma_t^{-1}(X-M_t) \right\}. \end{aligned} \tag{3}$$

D. Sequential Forward Floating Feature Selection

Feature selection avoids problems related to high dimensionality, over-fitting, and lack of training data. By lowering the number of features, we also lower the amount of required computation. This is essential for real-time pattern classification algorithms. Feature selection involves two aspects. First, small subsets of the original set of extracted features are evaluated with a criterion function designed to measure the potential utility of the subset in question. Since an exhaustive search through large feature spaces is computationally intractable, an optimal subset of features must be selected using a search algorithm.

A good feature selection algorithm must balance the optimality of its result against the computational efficiency of its

Fig. 5: Five types of terrain found in the ECSC at UTD. The left column has pictures taken from a distance, while the right column shows close up views of each type of terrain.

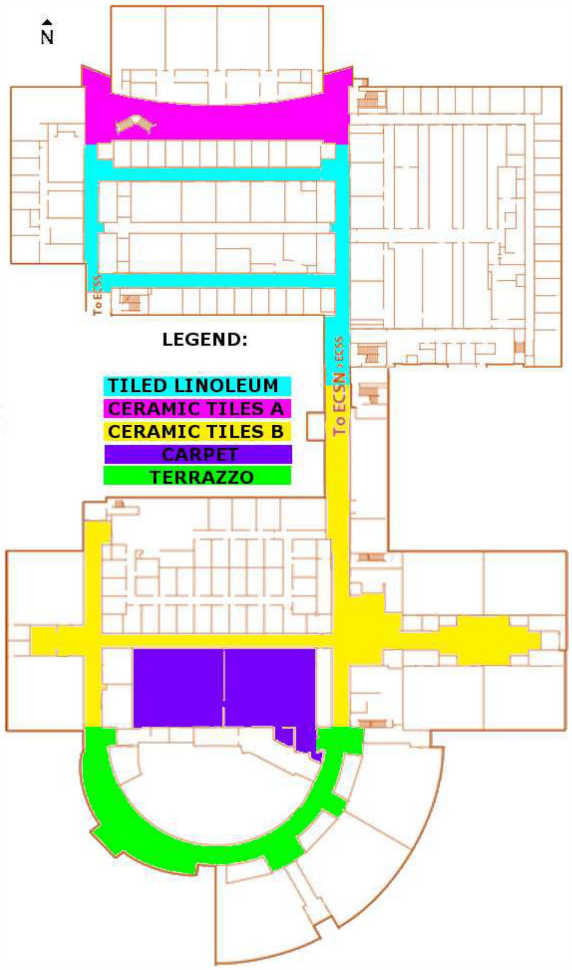

Fig. 6: A map of the ECSC at UTD color coded by terrain. The areas in color constitute the area where testing was conducted. Halley was then driven around the color coded areas in a circuit and classified the terrain. This experiment is meant to replicate a realistic type of path that a mobile robot might take in an actual application scenario.

search. This motivated researchers to propose many feature selection algorithms including the sequential background selection (SBS) method [21] and its counterpart known as sequential forward selection (SFS) [22]. Although these search algorithms are suboptimal, these algorithms are very computationally efficient when compared to an exhaustive search. The Sequential Forward Floating Selection method was introduced by Pudil et al. to deal with the nesting problem [23]. According to this algorithm, the selected subset starts from an empty set. In each step, it creates a new subset by adding a feature as in the SFS. Then it searches for features to be eliminated from the subset as in the SBS method until performance no longer increases.

III. APPROACH

The goal of this paper is to teach a robot to recognize five types of terrain shown in Fig. 5 (these are the different terrain types present in our building). Ideally, the robot will drive over any of these five terrain types for three seconds (or approximately 1.8 meters at maximum speed), and then be able to determine which of the five terrains it is currently on. The robot would then report a label every one second (or approximately 600mm at top speed) after that. In this work, only indoor terrain is considered. To allow long-term experiments along complicated paths, we guarantee a high degree of accuracy in realistic environments and situations. *Our design goals are 90% correct classification, while the robot is able to operate at any velocity within its range, operate a non-constant velocity, and be able to make turns in order to reach destinations and avoid obstacles.*

There are many design aspects to consider in developing such a system. The intention here is to create a modular system that can be used in a modular fashion with other mobile robots of various configurations. The goal is not to design a robot specifically for terrain classification, but rather to design a terrain classification system specifically for mobile robots. Therefore we eschew solutions that are dependent on the structure of the vehicle.

A strap-down IMU is affixed to a two-wheeled mobile robot as described in Sec. II. The robot is a Mobile Robots Pioneer3-DXSH named Halley. A low-cost (less than $50) IMU is used, specifically a Nintendo Wiimote, which utilizes MEMS. Since the robot ideally only rotates and translates on its z and x axes (i.e. as measured from F_b), respectively,

any inertial disturbances/vibrations detected on other axes is partially due to properties of the surface that the robot is traveling on. If the robot moves onto a different surface, then the vibrations will change. In this study, we exploit this observation to solve the RTC problem. If characteristics such as friction, texture, the size of tiles, width, depth, or absence of grout lines, or any other physical characteristic of one region is different than another region, the robot will ideally be able to detect this, and classify that region's terrain.

The robot's maximum linear velocity during training and testing is approximately 600mm/sec plus vibrations. The current version of the open source Wiimote software API [24] that we employ allows us to take data from the Wiimote's inertial sensors. A sampling rate of 25 Hz is selected through experimentation. The robot's software is implemented in C++ and run on a Dell M6400 mobile workstation that is mounted on top of Halley. The PC has an Intel Core 2 Duo CPU T9600 clocked at 2.80 GHz.

A. Feature Extraction

Six low-level descriptors of linear acceleration and angular velocity in the time domain are generated by the IMU, as measured along and about its six inertial axes. The six IMU signals are windowed in three second segments with a two second shift. Altogether we compute 864 different statistics from the IMU's six signals and corresponding derivatives, which are expected to capture discriminatory features inherent to the vibration signal data being used to solve the RTC problem. Table I also shows the percentage of the 864 statistics that are comprised by each IMU signal (e.g. the accelerometer signal along the robot x direction was used in 19.12% for the statistics). For detailed information about the statistics computed from the signals, readers are referred to Eyben et. al [25].

B. Feature Selection

Sequential forward floating feature selection is used to select the best features from the original feature set for solving the RTC problem. The original feature set includes 864 features extracted from the IMU's output. Table I describes the set of statistics extracted from the six IMU signals (denoted acclx, accly, acclz, gyrox, gyroy, gyroz for the accelerometer or gyro measurement on a particular axis) over these three second segments. These statistics are used as features. The types of functions utilized to generate these statistics include different measure of mean (e.g. arithmetic mean(amean)), moments (e.g. skewness), extremes (e.g. minimum), frequency spectrum (e.g. linear predictive coefficients(lpc)), timing and duration (e.g. risetime), regression (e.g. coefficients of linear and quadratic regression), zero crossing rate, quartile ranges etc.

During the training phase, sequential forward floating feature selection is run three times for building three separate models corresponding to three operational modes:

- STRAIGHT: The robot is driven in a straight line at maximum velocity.
- CLOCKWISE (CW): The robot is driven in many continuous clockwise ellipses at variable linear and angular velocities.
- COUNTER-CLOCKWISE (CCW): The robot is driven in many continuous counter clockwise ellipses at variable linear and angular velocities.

TABLE I: Features selected for each low level signals by SFFS.

Low level Descriptors	Functionals
acclx (19.12%)	de-skewness, de-amean, de-amean, de-qmean, de-nzamean, amean, absmean, kurtosis, max, iqr1-2, nzgmean, de-lpgain, left-ctime
accly (13.24%)	de-pctlrange0-1, amean, de-qmean, qmean, lpc2, variance, qregerrA, de-lpc1, de-qregc2
acclz (20.59%)	quartile1, maxameandist, peakMean, duration, risetime, kurtosis, variance, de-stddev, kurtosis, percentile99.0, linregerrA, de-iqr1-3, de-dct6, pctlrange0-1
gyrox (20.59%)	de-iqr1-2, qregc2, zcr, nnz, meanPeakDist, peakDistStddev, duration, qregerrA, quartile1, pctlrange0-1, de-lpc1, minPos, de-stddev, de-absmean
gyroy (13.24%)	de-pctlrange0-1, minameandist, dct5, qregerrA, qmean, de-absmean, nzgmean, lpc2, min
gyroz (13.24%)	quartile1,mcr, de-centroid, nzgmean, de-iqr2-3, lpc4, lpc3, de-falltime, de-downleveltime90

C. Linear Bayes Normal Classifier

A Linear Bayes Normal Classifier is used for the RTC task. The IMU captures inertial data for each of the five different terrains shown in Fig. 5. During training, the robot is driven over each surface by a human operator in multiple independent sessions. In total a minimum of 30 mins x 3 modes x 5 terrains = 450 mins = 7.5 hours of data was collected for training the classifier.

The data is then divided into three separate partitions called training, development and testing with a ratio of $[4:1:1]$, respectively. The data is divided in such a way that the data from one session is included in only one partition. Next, three different classifiers are built on the training partition for STRAIGHT, CW and CCW, respectively. The development partition is used for feature selection as described in the previous section. The training and testing partitions are randomly down-sampled to compensate for any unbalance that may exist with respect to the class sizes amongst the partitions. Table II shows the performance of each classifier when it is tested on different modes.

D. Angular Velocity Based Hierarchical Classifier Selection

Table II shows that the accuracy of the classifiers is above 90% when the training and testing partitions are modally matched (models A, B and C). However, the performance significantly decreases when the training and testing partitions are modally mismatched (models D, E, F, G). This is a clear indication that the features selected for classification are very much mode dependent.

Our first approach toward solving this issue was to merge the corresponding training, development and testing partitions for all three modes (STRAIGHT, CCW or CW). Then, we used the joint development set to select the best

TABLE II: Training Results for Mode-based Classifier Models

Model	Train	Test	Accuracy %
A	STRAIGHT	STRAIGHT	90.69
B	CCW	CCW	93.16
C	CW	CW	92.54
D	STRAIGHT	CCW	57.45
E	STRAIGHT	CW	49.48
F	CCW	CW	31.22
G	CW	CCW	56.59
H	ALL	STRAIGHT	65.37
I	ALL	CCW	86.44
J	ALL	CW	67.65
K	ALL	ALL	74.37
L	ALL+Ang.vel	ALL+Ang.vel	80.65
M	Hierarchical	ALL	91.67

20 features. These features are used to train the a classifier, referred to as ALL, that aims to capture the intrinsic variability introduce by the three modes. Table II shows that the mode independent classifier ALL outperforms the other single mode classifiers when mismatched against CW (compare models E, F and J) and CCW (compare models D, G and I) test data. However, when the training and testing partitions for ALL are modally matched, the accuracy of the classifier dramatically decreases (compare models A, B and C vs. K).

The next attempt to correct this condition, was the creation of model L. During both training and testing, the robot is estimating its current position, orientation and velocity using an Extended Kalman Filter [13], [15], [16]. Model L was made by including the robot's angular velocity at the feature level. We estimate the same functionals described in section II-D producing 1008 features (144 functionals for the 6 IMU signals plus angular velocity). This set was also reduced to 20 features using SFFS. Table II shows that the improvement of model L over ALL in a modally matched test (compare models L and K) was significant but did not yet meet our goal of 90% accuracy.

The final, and ultimately successful, implementation is a hierarchical classifier. This classifier is motivated by the high performance achieved when the training and testing sets are drawn from the same mode (models A, B, and C). The proposed approach consists in (1) identifying the mode of the robot (STRAIGHT, CCW or CW) and (2) selecting the corresponding classifier that matches that mode. With this scheme, we avoid mismatches between the training and testing conditions. To identify the correct mode, we use the mean of the estimated angular velocity of the robot over the three-second window. If the mean is less than negative ten degrees per second, then we test that instance using the CW classifier. If the mean is more than ten degrees per second, then we test that instance using the CCW classifier. Otherwise, the STRAIGHT classifier is selected to test that instance. Table I shows that the mode independent hierarchical classifier yields accuracy of 91.67% (model M). It performs significantly better than the three separate mode dependent classifiers when they are tested in mismatched modes.

TABLE III: Accuracy Results for Test Experiment

Terrain	TLin	CTA	CTB	Cpt	Trz	ALL
Accuracy %	93.03	82.93	89.21	88.29	89.08	89.65
Sample %	28.47	7.08	20.79	19.15	24.50	100.0

TABLE IV: Confusion Matrix for Test Experiment

	TLin	CTA	CTB	Cpt	Trz
Tiled Linoleum	307	2	2	10	9
Ceramic Tiles A	1	68	13	0	0
Ceramic Tiles B	1	24	215	0	1
Carpet	11	4	1	196	10
Terrazzo	16	2	2	11	253

IV. Validation Under Realistic Conditions

Fig. 6. Shows a map color coded by terrain of the area where testing data was collected. In test experiments, the robot is driven around in a circuit such that all areas colored in the map are traversed in a single test. These tests are conducted in the middle of the work day when there are many people walking around the Engineering and Computer Science Complex at the University of Texas at Dallas. Halley is driven by a human operator using a joystick. The human operator can take any path that satisfies the following conditions:

1) All color coded areas of the map must be visited at least once.
2) The experiment must last at least 15 minutes.
3) The robot may stop *only* to avoid obstacles or pedestrians.
4) The robot must operate in all three modes on each terrain.

The robot traverses crowded hallways where it must travel with heavy, erratic pedestrian traffic. The human operator must drive Halley at varying speeds and also maneuver very tightly around several construction/maintenance areas, as well as through an open computer lab full of students. The data is collected as described above and tested offline. The experiment presented here took 1159 seconds (19 mins. 19 secs.) to complete, and offline classification took approximately 0.1737 sec/frame * 1159 frames of data = 201.3183 secs (3 mins. 21.3183 secs.) total processing time, where one 'frame' means a single one second window of test data from which a single label is generated.

Results from this experiment are shown in Table III. These results show that our classification system performs with an overall accuracy of 89.65% for all samples. Four out of five terrains are classified with an accuracy of over 89%. The one exception being Ceramic Tiles A (CTA), which is still classified with an accuracy of above 82%. The second row of Table III, tells what percentage of all samples each terrain represents. CTA represents only 7.08% of the samples taken, while each of the other terrains have approximately three to four times more samples than CTA. This relatively lower sample size may indicate lower confidence in this accuracy measurement. Additional experiments will be carried out.

Table IV shows the confusion matrix for this experiment where the rows represent the classifier results and the columns represent the ground truth. Note that out of the 14 incorrectly labeled samples of CTA, 13 were labeled as Ceramic Tiles B (CTB). Also note that out of the 26 incorrectly labeled samples of CTB, 24 were labeled as

TABLE V: Confusion Matrix with Merged CT Class

	TLin	CT	Cpt	Trz
Tiled Linoleum	307	4	10	9
Ceramic Tiles	2	320	0	1
Carpet	11	5	196	10
Terrazzo	16	4	11	253

TABLE VI: Accuracy Results with Merged CT Class

Terrain	TLin	CT	Cpt	Trz	ALL
Accuracy %	93.03	99.07	88.29	89.08	92.84
Sample %	28.47	27.87	19.15	24.50	100.0

CTA. CTB and CTA are very similar surfaces with equally similar tactile properties. The fact that the classifier is able to distinguish them so well speaks to the effectiveness of our proposed RTC method. Additionally, combining CTA and CTB into a single class has a dramatic effect on the classification accuracy. To illustrate this, Table V presents a confusion matrix that considers CTA and CTB to be a single terrain class Ceramic Tiles (CT). Merging CTA and CTB has a dramatic positive effect on the CT class accuracy and also boosts the total accuracy of the proposed classifier by more than 3% for this experiment as shown in Table VI.

V. CONCLUSIONS AND FUTURE WORK

Solving the RTC problem is an important step toward creating truly autonomous wheeled mobile robots. By teaching a robot to recognize different types of terrain, one can better ensure the safety and proper operation of a robot as well as enhance its functional capabilities in many different ways.

This paper presented a classifier that recognizes physical characteristics of a given terrain, manifested as vibrations sensed by an IMU. This data is combined with a thresholded state descriptor *before* extracting many statistical features in both time and frequency domains. The set of these statistical features is reduced using sequential forward floating feature selection. The results show that the classifiers are sensitive to the robot's current state of the operation (moving straight versus turning left or right). This limitation is overcome by a multi-class, hierarchical classifier. This approach is validated with long-duration experiments which include realistic operational conditions that a robot performing RTC indoors might encounter. The success rate is over 89% for five terrain types and over 92% when two similar terrains are classified together.

Future areas of work include implementing an on-line version of the classifier that can be used during deployment. This will allow us to investigate improved localization methods by allowing them to compensate for specific errors that occur as a result of operation on specific terrains. It may also be feasible to detect small yet distinctive irregularities that may occur in large regions of otherwise uniform terrain. If these surface anomalies are of sufficient uniqueness and of small enough size, then it stands to reason that they can serve as landmarks in a Terrain-based SLAM (TeSLAM).

REFERENCES

[1] K. Iagnemma, F. Genot, and S. Dubowsky, "Rapid physics-based rough-terrain rover planning with sensor and control uncertainty," in *Proc. IEEE Int Robotics and Automation Conf*, vol. 3, 1999, pp. 2286–2291.

[2] K. Iagnemma, C. Brooks, and S. Dubowsky, "Visual, tactile, and vibration-based terrain analysis for planetary rovers," in *Proc. IEEE Aerospace Conf*, vol. 2, 2004, pp. 841–848.

[3] C. A. Brooks and K. Iagnemma, "Vibration-based terrain classification for planetary exploration rovers," *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1185–1191, 2005.

[4] D. Sadhukhan, C. Moore, and E. Collins, "Terrain estimation using internal sensors," in *Proc. of the 10th International Conference on Robotics and Applications*, Honolulu, HI, August 2004, pp. 447–800.

[5] E. M. DuPont, R. G. Roberts, and C. A. Moore, "Speed independent terrain classification," in *Proc. Proceeding of the Thirty-Eighth Southeastern Symp. System Theory SSST '06*, 2006, pp. 240–244.

[6] E. DuPont, C. Moore, E. Collins, and E. Coyle, "Frequency response method for terrain classification inautonomousground vehicles," *Autonomous Robots*, vol. 24, pp. 337–347, 2008.

[7] E. Coyle, E. G. Collins, and R. G. Roberts, "Speed independent terrain classification using singular value decomposition interpolation," in *Proc. IEEE Int Robotics and Autom Conf*, 2011, pp. 4014–4019.

[8] C. Weiss, H. Frohlich, and A. Zell, "Vibration-based terrain classification using support vector machines," in *Proc. IEEE/RSJ Int Intelligent Robots and Systems Conf*, 2006, pp. 4429–4434.

[9] C. Weiss, M. Stark, and A. Zell, "SVMs for vibration-based terrain classification," in *Autonome Mobile Systeme 2007*, ser. Informatik aktuell, K. Berns, T. Luksch, and W. Brauer, Eds. Springer Berlin Heidelberg, 2007, pp. 1–7.

[10] C. Weiss, N. Fechner, M. Stark, and A. Zell, "Comparison of different approaches to vibration-based terrain classification," 2008, tech. Rep.

[11] P. Giguere and G. Dudek, "A simple tactile probe for surface identification by mobile robots," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 534–544, 2011.

[12] D. Tick, J. Shen, and N. Gans, "Fusion of discrete and continuous epipolar geometry for visual odometry and localization," in *Proc. IEEE International Workshop on Robotic and Sensors Environments*, Arizona State University, Phoenix, AZ, Oct. 2010, pp. pp. 134–139.

[13] J. Shen, D. Tick, and N. Gans, "Localization through fusion of discrete and continuous epipolar geometry with wheel and IMU odometry," in *Proc. 2011 American Control Conference*, O'Farrell Street, San Francisco, CA, USA, June-July 2011, pp. pp. 1292–1298.

[14] D. Tick, J. Shen, Y. Zhang, and N. Gans, "Chained fusion of discrete and continuous epipolar geometry with odometry for long-term localization of mobile robots," in *IEEE Multi-conference on Systems and Control*. Sheraton Denver Downtown Hotel, Denver, CO, USA.: IEEE, Sept. 2011, p. pp. 668 674.

[15] R. E. Kalman, "A new appproach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[16] R. G. Brown, *Introduction to Random Signal Analysis and Kalman Filtering*. John Wiley & Sons, 1983.

[17] B. Barshan and H. F. Durrant-Whyte, "Inertial navigation systems for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 11, no. 3, pp. 328–342, 1995.

[18] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology 2nd Ed.* Institution of Engineering and Technology, 2004.

[19] G. Dudek and M. Jenkin, "Inertial sensors, GPS, and odometry," in *Springer Handbook of Robotics*, 2008, pp. 477–490.

[20] J. Laumond and J. Risler, "Nonholonomic systems: controllability and complexity," *Theoretical Computer Science*, vol. 157, pp. 101–114,, 1996.

[21] T. Marill and D. Green, "On the effectiveness of receptors in recognition systems," *IEEE Trans. Inf. Theory*, vol. 9, no. 1, pp. 11–17, 1963.

[22] A. W. Whitney, "A direct method of nonparametric measurement selection," *IEEE Trans. Comput.*, no. 9, pp. 1100–1103, 1971.

[23] P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119 – 1125, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0167865594901279

[24] gl.tter. (2007-2010) Wiiyourself! native c++ wiimote library v1.15 rc3. gl.tter.org. [Online]. Available: http://wiiyourself.gl.tter.org

[25] F. Eyben, M. Wöllmer, and B. Schuller, "OpenSMILE: the Munich versatile and fast open-source audio feature extractor," in *ACM International conference on Multimedia (MM 2010)*, Firenze, Italy, October 2010, pp. 1459–1462.