

Surface Classification for Sensor Deployment from UAV Landings

David Anthony¹, Elizabeth Basha², Jared Ostdiek³, John-Paul Ore¹, and Carrick Detweiler¹

Abstract—Using Unmanned Aerial Vehicles (UAVs) to deploy sensor networks promises an autonomous and useful method of installation in remote or hard to access locations. Some sensors, such as soil moisture sensors, must be physically installed in soft soil, yet UAVs cannot easily determine soil softness with remote sensors. In this paper, we use data from an onboard accelerometer measured during UAV landings to determine the softness of the ground. We collect and analyze over 200 data sets gathered from 8 different materials: foam, carpet, wood, tile, grass, dirt, concrete, and woodchips. Based on this analysis, we examine a number of features from the accelerometer and four classification algorithms: LDA, QDA, SVM, and binary decision trees. The decision tree performs well and is simple to implement onboard the UAV. We implement this in our UAV control system and perform experiments to verify that the UAV can accurately classify the softness of the surface with 90% accuracy. This lays the groundwork for our future work on developing a UAV capable of installing sensors in soft soil.

I. INTRODUCTION

Deploying sensor nodes to create a monitoring system outdoors is time consuming, especially when sensors must be placed in remote or hard to access locations. Unmanned Aerial Vehicles (UAV) can automate deployment to enable faster deployment times as well as provide mechanisms for easier system repair when deployed sensors fail [1]. Yet UAVs face challenges in knowing where they can safely deploy sensors and how to install the sensors. Cameras and remote sensors can be used to identify target installation locations; however, every gram of cameras and sensors replaces mass that could be devoted to carrying more or larger sensors to distant locations. Further, UAVs cannot easily measure some physical parameters remotely, such as how yielding the soil is for installing a moisture sensor.

In this paper, we develop a surface classification methodology and algorithms that a quadcopter UAV can use to identify whether a surface is hard or soft to enable proper sensor installation. Our approach relies exclusively on acceleration data acquired during landing using a sensor available on



Fig. 1. UAV with sensor deployment mechanism.

nearly all UAVs. Figure 1 shows our UAV equipped with our sensor deployment mechanism that both deploys and anchors the sensor by inserting a “spike” into soft ground. The spike can also be replaced with a soil moisture, conductivity, temperature, or other contact-based sensors that must be inserted into the ground.

To accurately classify soft and hard surfaces, we develop a set of classification features extracted during indoor and outdoor UAV landing experiments. Using these features, we evaluate four different classification algorithms: (1) linear discriminant analysis (LDA), (2) quadratic discriminant analysis (QDA), (3) a support vector machine (SVM) classifier, and (4) a decision tree classifier. We find that a decision tree classifier based on the accelerometer’s maximum upward acceleration and frequency with maximum amplitude performs as well or better than the other approaches and is also computationally simple to implement in real time on our UAV. We evaluate our approach using indoor experiments and verify its performance in classifying hard versus soft landing surfaces. Our indoor experiments yield 90% accuracy in classifying with all hard surfaces correctly classified and some soft surfaces misclassified; we prefer that the UAV misclassify soft where it simply avoids a deployment site compared to hard where it could damage the system by incorrectly deploying.

We organize the paper as follows. Section II outlines prior work in surface classification. Section III describes the UAV system used for classification. Section IV explores features of indoor and outdoor data sets. Given these features, Section V develops, analyzes, and compares our four classification algorithms. Section VI describes the experiments. We conclude and discuss future work in Section VII.

We are grateful to National Science Foundation (NSF) RI (IIS-1116221), NSF CNS (CSR-1217400 and CSR-1217428), and USDA #2013-67021-20947 which partially supported this work. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

¹C. Detweiler, D. Anthony, and J-P Ore are with the Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588, USA `carrick, danthony, jore` at `cse.unl.edu`

²E. Basha is with the Department of Electrical and Computer Engineering, University of the Pacific, Stockton, CA 95211, USA `ebasha` at `pacific.edu`

³J. Ostdiek was with the Department of Biological Systems Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588, USA during this work and is now with Stanford University. `jaredostdiek` at `gmail.com`

II. RELATED WORK

There is a wide range of work related to surface classification, mainly in two categories: (1) those that actively touch to characterize the surface, and (2) those that sense remotely.

Within active touch methods, whiskers, accelerometers, and other tactile sensors have classified surface types or objects by identifying shapes or textures. Dallaire *et al.* use a triple-axis accelerometer as a contact-based tactile sensor to accurately classify a large set of surface types like aluminum, linoleum, and carpet, using an SVM classifier [2]. They achieve a surface classification rate of 96.7% using 1 second of data and a variety of features including the variance up to the fifth standard moment. Romano and Kuchenbecker use a PR2 robot with a custom acceleration and force sensor to classify 15 different surfaces with a recognition rate of 80% using an SVM classifier [3]. Fishel *et al.* develop a Bayesian exploration approach that obtains 95.4% performance in discriminating between 117 textures using a finger-like tactile sensor [4]. These types of approaches require rubbing the sensor over the surface at a variety of precise distances and angles. In contrast, our method uses one accelerometer during one landing with a range of distances and orientations. The drawback of using fewer constraints is lower overall classification success. Fox *et al.* also relax the requirements on a precise orientation and distance while using a whisker to discriminate between two surfaces with 65% accuracy after two passes [5]. This shows the difficulty of surface classification by contact in unconstrained setups.

Sensors have also helped ground robots classify the surfaces over which they drive. Roy *et al.* tap a microphone on different floor materials and examine the normalized cross correlation between the frequency response of the tap and a library of known surfaces. This method obtains a 98% classification performance of six surface types [6]. However, the experiment occurred in a controlled lab environment by tapping the same object in the same spot, so it is unclear how well this approach would generalize. Brooks *et al.* use the on-board accelerometer on a robot with LDA to classify the terrain over which it is driving [7]. They divide the signal into short segments and analyze the power spectral densities and the signals in the Fourier domain to create a voting system for surface type. All of these approaches use long sequences of data to accurately classify the surfaces, whereas we use a quick burst of information from a single landing.

Non-contact sensors can also classify terrain, with the most popular types of sensors being cameras and LIDAR sensors. Larson *et al.* use a single camera and measure the “gait bounce” to determine characteristics of the terrain for legged robots [8]. Vandapel *et al.* use LIDAR to classify safe and unsafe driving areas [9]. Manduchi *et al.* use LIDAR and a color stereo camera pair to classify terrain for obstacle avoidance in off-road settings where a ground plane cannot be easily computed [10]. While these techniques are appealing and applicable in many settings, terrain analysis from cameras is subject to errors as lighting conditions change as well as seasonal changes. For the types of small

UAVs we are using, carrying a LIDAR and the associated processing systems would consume nearly all the payload.

Finally, UAVs have been used for terrain mapping and characterization. Templeton *et al.* use a vision system on a helicopter to compute safe landing areas free of obstacles [11]. Thrun *et al.* use a laser-scanner mounted on a helicopter to reconstruct the 3D map of the environment [12]. Ahmadi *et al.* process aerial imagery to classify terrain into various categories such as water, grass, trees, buildings, and roads by using both color and texture features [13]. Scherer *et al.* use a camera and laser scanner to classify safe areas to fly along a river [14]. Our work differs from these in that we examine the surface with one landing using simple, light, cost-effective sensors and therefore do not require heavy cameras or laser scanners to classify the surface. In the future, as cameras and processors get smaller, we plan to merge these remote sensing techniques ours to identify candidate locations for deployment, so that automated deployment of sensor networks is possible [15].

III. SYSTEM OVERVIEW

In this section, we provide an overview of the system hardware and software, including the UAV, accelerometer hardware, altitude estimation system, and the ground station.

A. UAV

The experiments in this paper utilize an Ascending Technologies Hummingbird quadrotor UAV, shown in Figure 1. The Hummingbird is capable of autonomous flight, utilizing onboard processors and sensors. The UAV’s sensors include accelerometers, gyroscopes, GPS, and a barometric pressure sensor. The UAV also has a 802.15.4 ZigBee radio so that an operator can remotely monitor and control the UAV.

B. Sensor Deployment and Data Collection

The UAV carries a 58g sensor that can be deployed with the sensor deployment mechanism shown in Figure 2. The sensor is held in place by an electromagnet that can release the sensor once the system finds a suitable location. A “spike” can be rotated into place to anchor the sensor to the ground after deployment. Alternatively, the spike can be replaced with an insertable sensor, such as a soil moisture probe. We leave a detailed discussion and analysis of the deployment mechanism to future work.

We augmented the UAV with a 3-axis accelerometer mounted underneath the center of the UAV, and a second ZigBee radio to collect the acceleration data we use for the surface classification. Ultimately, the classification approach could be implemented using the UAV’s IMU accelerometer and onboard microcontroller, but the additional accelerometer allowed us to transmit readings to an offboard computer to enable computation of complex features and classifiers while not interfering with the command radio link. The accelerometer is an Analog Devices ADXL335 that has a minimum $\pm 3g$ measurement range. An ATmega 1284p microcontroller samples the accelerometer and transmits the data to an offboard computer via the second ZigBee radio

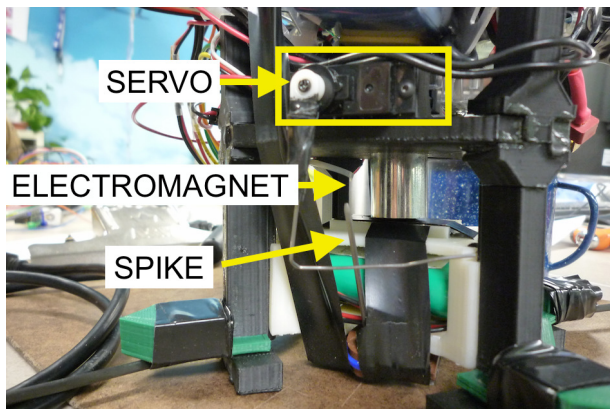


Fig. 2. Sensor deployment mechanism.

link. As Section V discusses, we achieve good results with features and classifiers that we can easily implement on this microcontroller. The radio bandwidth constrains the sensor sampling rate to 167Hz; this captures the features of interest.

C. Altitude Estimation

Accurately classifying the landing surface requires precise control of the UAV at low altitudes ($< 0.5\text{m}$). The fused height estimate provided by GPS, barometric altimeter, and IMU is not stable, and can rapidly drift over several meters in an outdoor environment. This altitude estimate is not accurate enough for this experiment. Therefore, we augment the system with a Sharp GP2Y0A21YK infrared (IR) rangefinder. This sensor has a measurement range of 10-80cm. It attaches to the arm of the UAV, as seen in the lower left of Figure 1.

The same microcontroller used for the accelerometer samples the IR rangefinder measurements and transmits the data. We use attitude information from the UAV's IMU to correct the range readings for the UAV's roll, pitch, and yaw. We then filter the IR range information to obtain good height estimates when operating less than 0.5m from the ground. The sensor does not work well in direct sunlight, but works outdoors in shade and provides sufficiently accurate height estimates for control of our UAV. In the future, we will integrate short-range ultrasonic range finders to enable outdoor operation in direct sunlight.

IV. FEATURE ANALYSIS AND CLASSIFICATION

We use the system to collect a base set of data on a range of surfaces. With this data, we analyze the different surfaces to determine which features seem useful for classification. We prefer low complexity, easy to compute features that are possible to implement on a UAV with limited processing power. These features then lead us to a set of classification algorithms we can use to effectively determine surface type. In this section, we first describe our data collection process, then our feature analysis before examining classification algorithms in Section V.

A. Indoor Data Collection

Our initial experiments focus on indoor surfaces of various hardnesses as seen in Figure 3. We explore four different surfaces: (1) hard and rigid wood platform, (2) tightly woven

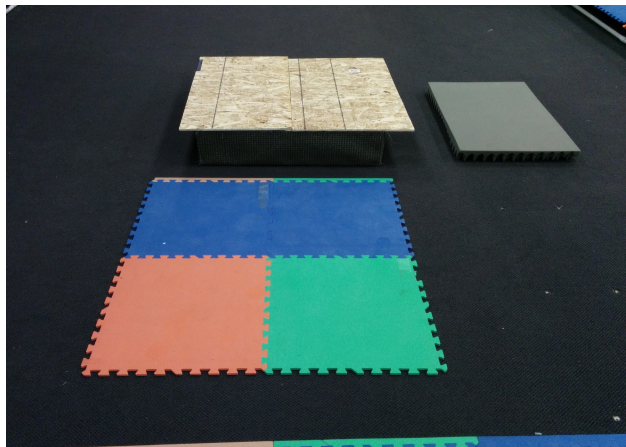


Fig. 3. Test surfaces. Clockwise from top left: Wood, foam, carpet, tile.

carpet over a concrete floor, (3) 2 cm rubber tile, and (4) soft and flexible foam pad. This set will help us understand how precisely the accelerometer data can differentiate between a range of hardnesses.

To collect the data, the UAV hovers approximately 20cm above each surface, and begins recording z-axis acceleration data. The motors are then commanded to turn off, which causes the UAV to descend. After the UAV reports that its motors are off, the system continues logging data for four seconds. The system then analyzes the recorded data to find the maximum acceleration reading, which corresponds to the initial impact on the surface. From this point, we use 400 samples (approximately 2.5s) to compute our different statistical features. We collected 80 different indoor samples, 20 from each of the different indoor surfaces.

B. Outdoor Data Collection

We also collected data from outdoor surfaces of various hardnesses. The outdoor surfaces used are grass, mulch woodchips, bare dirt, and concrete. We consider the grass and woodchips as soft, and the dirt and cement as hard since during our tests the dirt was dry and highly compacted.

An IR rangefinder controls the UAV's altitude while the pilot controls the (x, y) position. This combination ensures the UAV lands in the desired test location. Other than the flight method, we use the same data collection procedure as the indoor experiments, although we discarded any landings where the UAV tipped over on uneven terrain. Overall, we collected 71 landing samples on cement, 54 on woodchips, 56 on dirt, and 58 on grass.

C. Feature Analysis

We now analyze our experimental data to determine what, if any, features exist to differentiate surfaces and utilize with our classification algorithms.

We first explore the results of single experiments on the surfaces to ensure differences exist in the data. Figure 4 shows one second of characteristic z-axis accelerometer readings for the outdoor surfaces. Each surface has a distinctive response, which hints at the ability of the system to automatically classify which surface it has landed on. For example, the UAV makes a series of small bounces when it

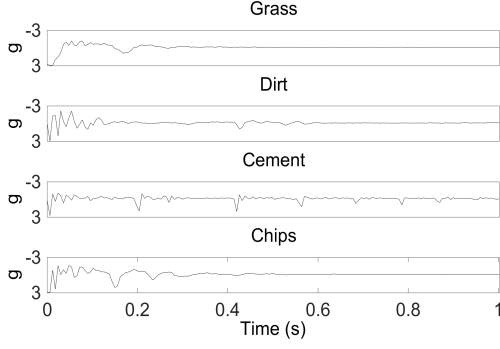


Fig. 4. Acceleration data from landings on grass, dirt, cement, and woodchips

lands on the hard concrete surface, which is reflected in the decaying series of spikes in the concrete data. On the other hand, grassy surfaces cushion the UAV's landing.

We develop a number of features that may characterize each trial, d . Specifically, the features we explore are:

- **Maximum Value:** $\max(d)$
- **Minimum Value:** $\min(d)$
- **Peak Frequency:** The frequency $f, f > 0$ with the largest amplitude.
- **Variance of Signal:** $E[(x - \mu)^2]$
- **2nd Maximum:** The second largest acceleration measurement.
- **2nd Minimum:** The second smallest acceleration measurement.
- **Ratio:** The ratio of the max to the 2nd max.
- **Skewness:** $\frac{E[X - \mu]^3}{\sigma^3}$
- **Kurtosis:** $\frac{E[X - \mu]^4}{\sigma^4}$
- **Mean:** $E[X]$
- **Settling Time:** After subtracting the mean of the final 20 samples, find the time between the largest spike and the spike that has 50% of this value.

Figure 5 shows some of these statistical measures computed for each trial, grouped by the surface type. This provides further evidence that surfaces do demonstrate differentiation based on these features that could lead to a viable classifier. The maximum value shows that the harder surfaces, such as cement and dirt, consistently have higher values than surfaces such as grass, because the UAV rebounds much harder on these surfaces when landing. Figure 5 also confirms our intuition developed from Figure 4. The plot of the peak frequency shows that the harder surfaces, such as concrete, dirt, and tile, have higher peak frequencies than the softer surfaces, such as grass and foam.

Finally, we perform a principle component analysis (PCA) on the data to better analyze our indoor data features. This analysis fails to identify descriptive features, as the eigenvectors with the largest eigenvalues have multiple components

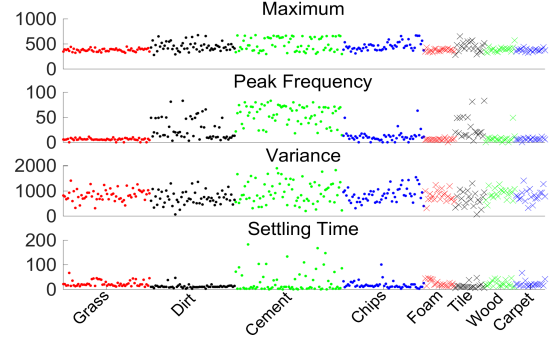


Fig. 5. Features for different trials and surfaces

with approximately the same values. Two factors cause the lack of distinguishing features in the PCA. First, not all features in the potential set are independent. For example, the second maximum is related to the first maximum, and the surface the UAV landed on. Therefore, it often explains a similar amount of variance as the first maximum. Second, the data has noise, which distorts the potential features and weakens their predictive powers. Thus, we manually use different feature subsets to identify the most accurate predictors when developing classifiers.

V. INDOOR AND OUTDOOR CLASSIFIER CREATION

Using the features developed in the prior section, we explore four different classification algorithms on the indoor and outdoor datasets: (1) Linear Discriminant Analysis (LDA), (2) Quadratic Discriminant Analysis (QDA), (3) Support Vector Machine (SVM) Classifier and (4) Decision Tree Classifier. For the indoor materials, our goal is to differentiate between the soft foam and the other three surfaces, which are significantly harder. Outdoors, we consider the grass and woodchips as soft, and the dirt and cement as hard.

All classification procedures use 10-fold cross validation for evaluation based on the data collected indoors and outdoors as described in Section IV-A. Experimentation with various subsets of features reveals that the maximum acceleration and the peak frequency create the most accurate classifiers. We find none of the approaches do particularly well for discriminating between all of the surfaces (see the confusion matrices in Tables I and II), most likely due to the similarity of the harder surfaces. However, all approaches do well discriminating between hard and soft surfaces.

1) *Linear Discriminant Analysis:* We start with LDA to classify based on a linear combination of the features. This linear combination separates the data into two or more classification categories, relying on the assumptions that the feature data are normally distributed with the same covariance matrices. We used Matlab R2014a's *fitcdiscr* function to perform this classification [16].

Table I shows the performance of of LDA when trying to exactly classify the four indoor surfaces. This classifier has an accuracy of 63.75%. It correctly classifies all of the soft foam landings as foam, but does not do as well in

True Category	Classified Category															
	LDA				QDA				SVM				Decision Tree			
	Carpet	Foam	Tile	Wood	Carpet	Foam	Tile	Wood	Carpet	Foam	Tile	Wood	Carpet	Foam	Tile	Wood
Carpet	0	2	10	8	0	0	12	8	15	0	2	3	3	0	11	6
Foam	0	20	0	0	0	19	1	0	4	13	2	1	1	19	0	0
Tile	0	2	17	1	0	1	18	1	15	1	2	2	10	1	8	1
Wood	1	1	4	14	5	0	5	10	16	0	1	3	5	0	3	12

TABLE I
INDOOR CLASSIFICATION RESULTS, ALL SURFACES.

True Category	Classified Category															
	LDA				QDA				SVM				Decision Tree			
	Cement	Woodchips	Dirt	Grass	Cement	Woodchips	Dirt	Grass	Cement	Woodchips	Dirt	Grass	Cement	Woodchips	Dirt	Grass
Cement	54	2	13	2	52	2	17	0	70	0	1	0	58	2	11	0
Woodchips	2	27	2	23	2	33	4	15	46	2	2	4	3	27	11	13
Dirt	16	14	9	17	16	19	10	11	46	2	4	4	17	15	17	7
Grass	0	5	0	53	0	8	0	50	28	6	5	19	0	6	5	47

TABLE II
OUTDOOR CLASSIFICATION RESULTS, ALL SURFACES.

True Category	Classified Category							
	LDA		QDA		SVM		Decision Tree	
	Hard	Soft	Hard	Soft	Hard	Soft	Hard	Soft
Hard	55	5	59	1	60	0	59	1
Soft	0	20	1	19	9	11	1	19

TABLE III
INDOOR CLASSIFICATION RESULTS, HARD VERSUS SOFT SURFACES.

True Category	Classified Category							
	LDA		QDA		SVM		Decision Tree	
	Hard	Soft	Hard	Soft	Hard	Soft	Hard	Soft
Hard	84	43	94	33	119	8	101	26
Soft	3	109	8	104	78	34	21	91

TABLE IV
OUTDOOR CLASSIFICATION RESULTS, HARD VERSUS SOFT SURFACES.

discriminating between the hard surfaces. Table III shows the LDA classification results when only differentiating between hard and soft indoor surfaces. The system is able to correctly the soft landings, but misclassifies five hard surfaces as soft. This is undesirable, as it could lead to deploying a node in an undesirable location, and risks damaging the system. Overall, however, accuracy for LDA classification of indoor surfaces is high at 93.75%.

On the outdoor dataset, the system is able to achieve an 59.83% classification accuracy for all surfaces and 80.75% between soft and hard surfaces as shown in Tables II and IV, respectively. The classification rate is very poor for all surfaces; however, we are less concerned with this classification since we are primarily interested in identifying soft surfaces. For the hard and soft, the confusion matrix reveals that the system has a tendency to classify hard surfaces as soft surfaces, which may lead to UAV damage. However, it rarely misclassifies the soft surfaces.

2) *Quadratic Discriminant Analysis*: To see if we can improve on the LDA results, we next try a quadratic discriminant analysis (QDA) classification technique on the data, which should capture some non-linear relationships between the features. This approach is similar to LDA; it too assumes that the feature data are normally distributed, but it does not assume that the covariances are identical. Relaxing this assumption results in quadratic separation of the classes. Again, we start by testing the performance of QDA in fully classifying the surface types; to do so, we use the same *fitcdiscr* function from Matlab. As with the LDA analysis, changing the features used in the classifier does not yield significant classification improvements.

Table I contains the confusion matrix of the 10-fold cross-validated QDA classifier, when ran on the entire indoor

data set. This classifier only achieves a correct classification accuracy of 58.75%, with similar misclassifications to the LDA classifier. Table III shows the performance of QDA when classifying the indoor data into hard and soft categories with the resultant 97.5% accuracy. QDA incorrectly classifies only one of the hard and soft surfaces, and is thus very safe to operate.

Outdoors, Table II shows that QDA performs marginally better than the LDA, with a 60.67% accuracy, for all surfaces. The improvement is most noticeable when landing on the woodchips. For the hard and soft classification, Table IV shows that the QDA achieves an accuracy of 82.85%. This is slightly better than LDA, but still suffers from a significant number of misclassifications of hard as soft.

3) *SVM Classifier*: We next explore the SVM approach. The SVM is built using Weka 3.6.11 [17] and LIBSVM 3.18 [18] with the default options and 10-fold cross validation. A SVM divides the data with hyperplanes that maximize the distance of the data points from the hyperplane; these separations create the different classes.

The SVM performs far worse on the indoor dataset with all surfaces than the other approaches, as seen in the confusion matrix in Table I. It correctly classified only 41.25% of the indoor trials, which is far below the other approaches. The SVM fails to accurately detect even the foam surfaces, which the other classification techniques successfully classified. Using feature selection methods such as sequential feature searches or ranking the PCA components does not yield improvements on the performance. The SVM does show similar improvements to classification accuracy when limited to classifying the surfaces as hard or soft, as shown in Table III, achieving 88.75%. The SVM does have a tendency to misclassify soft surfaces as hard, unlike the other classifiers.

This could be a result of having more hard surfaces in the training set than soft surfaces.

Outdoors, the SVM performs significantly worse and achieves an accuracy of 39.75% for all surfaces and 64.02% for soft versus hard. Table IV shows that the SVM has a strong tendency to misclassify soft surfaces as hard, as in the indoor setting. Unlike the indoor settings, the two populations have similar size. While this conservative tendency makes it less likely that the UAV will damage itself by deploying a node to a hard surface, it may lead to unacceptably long deployment times, as the system will ignore many favorable locations.

4) *Decision Tree Classifier*: Our final classifier approach uses a decision tree. Decision trees create a set of nodes that “test” for certain criteria; the path between these nodes thus provides the classification rules and the final leaf node is the classification. We used the Matlab *fitctree* function to build the decision tree classifier. We train the decision tree with the same peak frequency and maximum amplitude features as the other classifiers. The decision tree uses the node error as the splitting criterion and the classification is pruned in order to create the optimal sequence of pruned subtrees. Figure 6 reveals that the best decision tree only has two levels and uses the two features; this leads to an efficient implementation that we could easily implement on microcontrollers onboard a UAV.

Tables I and III shows the decision tree indoor performance of 62.5% and 97.5% for all surfaces and soft versus hard, respectively. For the outdoor datasets, Tables II and IV show that the decision tree achieves 62.34% and 80.33% for all surfaces and soft versus hard, respectively. In both indoor and outdoor settings, the decision tree has similar performance to the LDA and QDA classifiers. The most noticeable difference is in the outdoor setting when classifying between two different surfaces. The decision tree is not biased to choosing one surface type over the other, unlike the LDA and QDA, which frequently misclassify hard surfaces as soft. This will force the UAV to operate in a safer manner, and make it less likely to damage itself. Also, it is not overly conservative like the SVM, and less likely to miss deployment opportunities.

Overall, we see improved results when focusing on classifying hard versus soft. As this solves our overall deployment problem, we concentrate on the two classifier system, where all classifiers performed well. Because of the ease of implementation, we implement the decision tree approach on our platform.

VI. EXPERIMENTS

We performed experiments indoors to validate our approach. In this experiment, the UAV travels to each of the four indoor surfaces. The UAV first lands on the surface and runs the decision tree classifier shown in Figure 6, which was generated using the previously collected data, as described in Section V. If the surface is soft, it will then re-launch itself, and perform a simulated sensor node deployment. After this is complete, or if the surface is hard, it will proceed to the

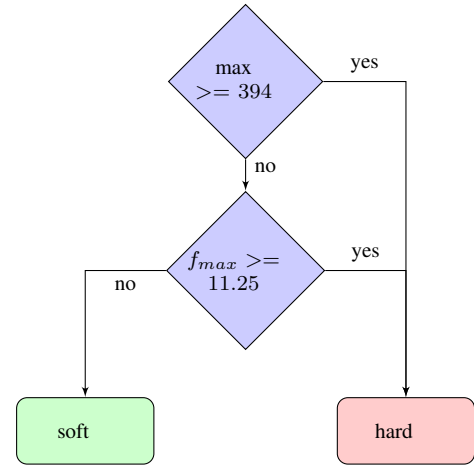


Fig. 6. Indoor decision tree

next surface and repeat the experiment. We repeated this process ten times.

Table V summarizes the results of the experiment. The ‘H’ entries in the table indicate that the system classified a surface as hard, and an ‘S’ entry indicates the surface was classified as soft. The overall classification rate is 90% accurate. All of the hard surfaces were correctly classified, although it did struggle with the soft surface, incorrectly classifying the soft foam as hard four out of ten times.

Experiment	Surface			
	Tile	Foam	Wood	Carpet
1	H	S	H	H
2	H	H	H	H
3	H	S	H	H
4	H	H	H	H
5	H	S	H	H
6	H	S	H	H
7	H	H	H	H
8	H	H	H	H
9	H	S	H	H
10	H	S	H	H

TABLE V
INDOOR VERIFICATION RESULTS

This incorrect classification of the soft foam may be due to the UAV dropping from a slightly different height, or impacting the foam at a different location than it was before. However, we prefer that the system is conservative in classifying a surface as soft, since attempting to install a sensor in an incorrectly classified hard surface could damage the sensor or UAV.

VII. CONCLUSION

In this paper, we examine using a UAV to classify surfaces based on accelerometer data. We collect indoor and outdoor datasets, analyze their features, and develop classifiers based on those features. The decision tree classifier performs well in discriminating between soft and hard surfaces as well as being easy to implement onboard a UAV. We implement this classifier on our UAV control system and develop

classifiers for both indoor and outdoor surfaces. For our indoor experiments, the classification accuracy between hard and soft surfaces is 90%. All of the classification errors are in classification of a soft surface as hard, which is a safe error as it would not cause an erroneous sensor deployment.

In future work, we will investigate improvements to our features and classification approaches by incorporating other onboard sensors such as the gyroscope, or accelerometers with larger dynamic ranges and bandwidths. Additional classification features will be investigated, in particular, those related to the frequency response of the landing. In addition, we plan to look at classifying surfaces based on a range of softness and retesting (performing another landing) on surfaces that are borderline soft to ensure that we only deploy sensors in soft soil. Furthermore, we will test different misclassification cost parameters to optimize the behavior of the system. Finally, we will add the ability to allow the UAV to deploy sensors that must be inserted into the soil, such as soil moisture sensors. This work will ultimately result in easier deployment of sensors in remote or hard to access locations.

REFERENCES

- [1] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme, "Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4, 2004, pp. 3602–3608.
- [2] P. Dallaire, D. Emond, P. Giguere, and B. Chaib-Draa, "Artificial tactile perception for surface identification using a triple axis accelerometer probe," in *2011 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, Sep. 2011, pp. 101–106.
- [3] J. Romano and K. Kuchenbecker, "Methods for robotic tool-mediated haptic surface recognition," in *IEEE Haptics Symposium (HAPTICS)*, Feb. 2014, pp. 49–56.
- [4] J. A. Fishel and G. E. Loeb, "Bayesian exploration for intelligent identification of textures," *Frontiers in Neurorobotics*, vol. 6, Jun. 2012.
- [5] C. W. Fox, B. Mitchinson, M. J. Pearson, A. G. Pipe, and T. J. Prescott, "Contact type dependency of texture classification in a whiskered mobile robot," *Autonomous Robots*, vol. 26, no. 4, pp. 223–239, May 2009.
- [6] N. Roy, G. Dudek, and P. Freedman, "Surface sensing and classification for efficient mobile robot navigation," in *1996 IEEE International Conference on Robotics and Automation, 1996. Proceedings*, vol. 2, Apr. 1996, pp. 1224–1228 vol.2.
- [7] C. Brooks and K. Iagnemma, "Vibration-based terrain classification for planetary exploration rovers," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1185 – 1191, Dec. 2005.
- [8] A. C. Larson, G. K. Demir, and R. M. Voyles, "Terrain classification using weakly-structured Vehicle/Terrain interaction," *Autonomous Robots*, vol. 19, no. 1, pp. 41–52, Jul. 2005.
- [9] N. Vandapel, D. Huber, A. Kapuria, and M. Hebert, "Natural terrain classification using 3-d ladar data," in *2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04*, vol. 5, May 2004, pp. 5117 – 5122.
- [10] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," *Autonomous Robots*, vol. 18, no. 1, pp. 81–102, Jan. 2005.
- [11] T. Templeton, D. Shim, C. Geyer, and S. Sastry, "Autonomous vision-based landing and terrain mapping using an MPC-controlled unmanned rotorcraft," in *2007 IEEE International Conference on Robotics and Automation*, Apr. 2007, pp. 1349 –1356.
- [12] S. Thrun, M. Diel, and D. Hähnel, "Scan alignment and 3-d surface modeling with a helicopter platform," in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, S. Yuta, H. Asama, E. Prassler, T. Tsubouchi, and S. Thrun, Eds. Springer Berlin / Heidelberg, 2006, vol. 24, pp. 287–297.
- [13] P. Ahmadi and S. Sadri, "A two-step approach for surface type classification of aerial images," in *2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN)*, May 2011, pp. 294 –298.
- [14] S. Scherer, J. Rehder, S. Achar, H. Cover, A. Chambers, S. Nuske, and S. Singh, "River mapping from a flying robot: state estimation, river detection, and obstacle mapping," *Autonomous Robots*, vol. 33, no. 1-2, pp. 189–214, Aug. 2012.
- [15] C. Tricaud and Y. Chen, *Optimal Mobile Sensing and Actuation Policies in Cyber-physical Systems*. Springer, 2012.
- [16] MATLAB, 2014a. Natick, Massachusetts: The MathWorks Inc., 2014.
- [17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explorations*, vol. 11, no. 1, 2009.
- [18] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.