Unsupervised Visual Terrain Classification from Proprioception

Michael Turmon, Max Bajracharya, Benyang Tang, Larry H. Matthies
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
email: {turmon, maxb, bytang, matthies}@jpl.nasa.gov

***Abstract*— We describe a novel system that learns the visual appearance of terrain classes from vehicle proprioceptive sensors, and requires no operator supervision. This enables an unmanned ground vehicle to learn continuously throughout its life. In particular, the vehicle can exploit large amounts of logged data to learn visual classes that are invariant to illumination and environmental conditions, because they include all observed variations. The approach uses an unsupervised clustering of high-rate (150 Hz) inertial measurement unit (IMU) data to identify classes of terrain with different roughness characteristics. We compare two clustering algorithms in learning terrain classifications: finite mixture models (FMMs) and hidden Markov models (HMMs). Because the HMM uses temporal continuity to constrain its clustering, it is able to attain estimated error rates of about 8% in our tests, half the error rate of a similar FMM. After learning the terrain classes, we use stereo information to back-project the vehicle's classified path into imagery, from which a multiclass support vector machine (SVM) visual classifier is learned. An uncertainty measure produced by the HMM allows screening of non-representative imagery from the SVM training set. After training, visual terrain classification can be performed online, in real time, by an SVM aboard the vehicle.**

I. Introduction

The ability of unmanned ground vehicles (UGVs) to predict the physical terrain properties of their environments is critical for high performance autonomous navigation. Traditional systems have relied heavily on terrain geometry to infer traffica-bility of terrain, but most of these approaches are not sufficient to determine soil and vegetation properties, such as sinkage, slip, or compressibility. These properties are necessary for large UGVs to adapt their planned path, traction control, and speed for safety and passenger comfort. They are even more critical for smaller UGVs to ensure stability and exploit the full capabilities of the vehicle (such as accelerating into a patch of mud that would immobilize the vehicle at a slower speed, or dynamically overcoming vegetation).

Historically, autonomous navigation of UGVs has been limited to obstacle avoidance guided by geometric analysis, either specified with static rules or learned from labeled training data. This leads to brittle behavior in new environments. More recently, systems have exploited self-supervised learning, where vehicle sensor data is used to update trafficability of terrain classes. However, these approaches use a supervised classifier learned from labeled sensor data, so they require training data from an operator, and they do not generalize well to new terrains. To avoid these problems, here we develop an *unsupervised* method for learning the visual appearance of terrain from proprioceptive data. This removes the need for manual labeling and enables the system to exploit large amounts of sensory data to learn throughout its lifetime.

We are interested in characterizing mostly flat load-bearing surfaces such as trails, fields, and roads. We therefore learn the association of a terrain's surface elevation profile [27] to its appearance, which provides an indication of the terrain type while being more distinguishable than slip or soil properties on these flat surfaces. We use the notion of automatic supervision, originally proposed for learning terrain slip characteristics [1], to learn the appearance of terrains with differing surface elevation profiles.

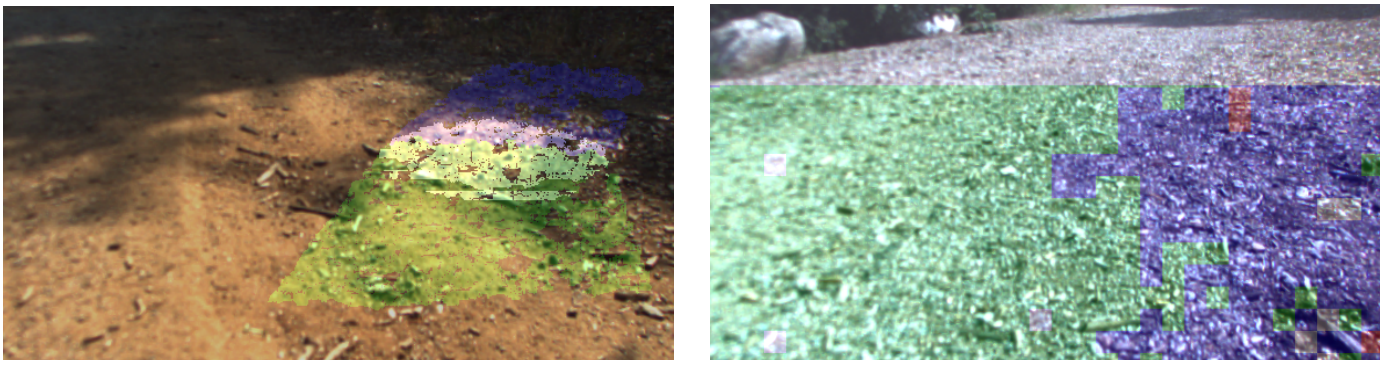

Fig. 1. An underfoot proprioceptive classification is projected back into imagery (left panel), yielding patches of dirt (green overlay) and mulch (blue). These and other patches train a visual classifier to label new scenes (right panel), such as this dirt path (green overlay) flanked by mulch (blue).

As the vehicle drives, terrain roughness measurements from an inertial measurement unit (IMU) are collected and added to a database, which is then segmented into terrain classes. The path classified by the IMU is projected into camera images acquired while driving (Figure 1, left), from which the visual appearance of the terrain classes is learned. The visual classifier is then used onboard the vehicle to predict the terrain class (Figure 1, right).

Our main contribution is to develop an unsupervised proprioceptive classifier that can successfully bootstrap a visual classifier. We address the key challenge of an unsupervised classifier introducing contamination during clustering by formulating mechanisms to make the unsupervised classifier robust to outliers, introduction of an outlier class, and computation of a classification confidence. Used to supervise an appearance-based terrain classifier, this allows a vehicle to learn a lighting-invariant visual terrain classifier by observing terrain under many different environmental conditions. We demonstrate the end-to-end system's effectiveness onboard a man-portable vehicle driving over complex natural terrain.

A. Related Work

Terrain classification systems for autonomous robot navigation have conventionally used rule-based approaches [14, 16] or offline, supervised learning [7, 17, 26] to segment imagery, laser range finder, or proprioceptive data. For example, [17] reports good vision-based classification with color distributions learned from examples taken over varying conditions, but points out that a large amount of training data covering all expected environmental conditions is required. Other work classifies terrain based on vibration signatures from vehicles on pre-defined terrains such as asphalt, gravel, and grass [25, 5]. Normally, data for supervised learning is labeled by a human operator, which is tedious and subjective. Recently, imitation learning that is robust to imperfect training input has been used to teach a UGV autonomous navigation [22], but the classifier remains static after being taught.

Self-supervised learning for navigation has been used to make terrain classifiers more adaptive, as well as extend the sensing range of vehicles [12]. Long-range visual classifiers have been trained based on safe regions identified by a scanning laser range finder [6] or stereo geometry [3, 19, 10, 13]. Similarly, traversability in overhead imagery has been learned from local images [23]. Also, short-range visual classifiers have been learned from wheel slip [2] and vibration [4] for planetary rovers. While the features and methods of learning vary for these approaches, each still requires a source of supervision, normally from another terrain classifier trained offline with human supervision.

Unsupervised learning has commonly been used for segmenting image, proprioceptive, or other sensor data. For proprioceptive terrain classification, [8] uses IMU, motor current, and leg angle to detect various level terrain types. However, we are interested in autonomously learning the visual appearance of terrains associated with different physical properties, and, following [1], adopt the notion of automatic supervision from proprioceptive sensors, in which proprioceptive sensor data is used to guide the learning of terrain appearance. The vehicle slip and terrain slope are used to learn vehicle slippage associated with visual terrain classes. Because vehicle slip is not sufficient to segment terrain types in all cases (e.g., flat terrain), the visual and proprioceptive features are treated jointly in [1].

B. Notation

Our approach tries to extrapolate traversability information learned from underfoot sensors, like accelerometers, gyroscopes, and wheel encoders, to sites farther away from the robot, where only visual features plus their locations (derived from stereo imaging) are known. We briefly establish notations for these concepts. The underfoot sensor readings are denoted u or $u(t)$. The visual features obtained from RGB imagery are denoted v.

We are concerned with two estimates of surface type T, one obtained from underfoot information (T_u), and the other from visual appearance (T_v). In some experiments, we may have a human-labeled ground truth surface type (T_0). (We drop the subscripts when context makes the source clear.) In this work, T takes on integer values in $\{1, \ldots K\}$, with 0 indicating an unknown-terrain class. In our approach, we use T_u to produce an estimated surface type. This estimate is then used as an oracle for T by the visual classifier, which produces its own estimate T_v. We may view the three types T_0, T_u, and T_v as successively more noisy estimates of surface type.

II. Unsupervised Proprioceptive Terrain Classification

The underfoot information we have is the 3D vehicle-centered accelerations and angular rates, plus the speed along the line-of-travel. In our experimental setup, the accelerations and rates are sampled with different sensors, but fast enough so we may assume they are coincident, grouping all seven real numbers into a time series of vectors $\tilde{u}(i) = \tilde{u}(t_i)$.

To derive informative features, it is helpful to operate on a sliding window of width W samples. In this view, the proprioceptive training data consists of a time series of matrix-valued *snapshots*

$$\mathcal{T}_{\mathrm{u}} = \{u(i) : 1 \leq i \leq N_{\mathrm{u}}\}, \quad \text{where} \tag{1}$$

$$u(i) = \left[\tilde{u}(i - W + 1), \ldots, \tilde{u}(i - 1),\, \tilde{u}(i)\right] \tag{2}$$

Typically $W = 128$, and each training sample covers about one second of data. As outlined above, we wish to assign each terrain snapshot $u(i)$ to a learned discrete class, which we denote as $T_u(i)$.

A. Proprioceptive Feature Selection

For definiteness, it is helpful to discuss feature selection first. We experimented with several feature types, including spectrogram-based features and features based on autoregressive modeling within each $u(i)$. In tests using an automobile-size vehicle with pneumatic tires and suspension, the spectrogram features showed resonances at a handful of characteristic frequencies, which broadened and shifted depending

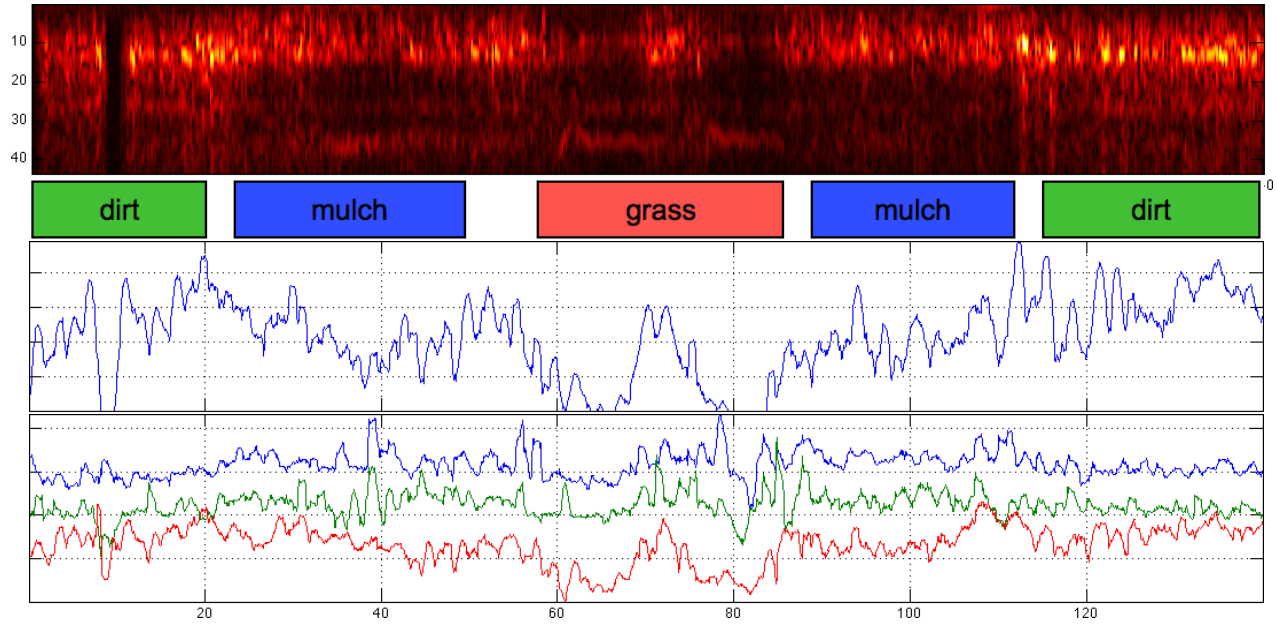

Fig. 2. Proprioceptive features observed on a typical run of 140 s, sampling at 150 Hz. All panels are time-aligned on the horizontal axis. Top image: spectrogram of z acceleration (snapshot $W = 64$, heated object color scale). Approximate ground truth is indicated below the spectrogram; no label means indeterminate or mixed class. Middle plot: log RMS energy of z acceleration ($W = 128$ samples). Bottom plot: three log-ratio features used in clustering.

on terrain. However, our main test vehicle is a Packbot-class system which does not have a suspension, so the individual modes visible within the spectral features were noisy, and did not carry much independent information.

The top panel of Figure 2 shows a typical spectrogram computed from the z acceleration channel of a Packbot run. The middle plot shows the RMS energy of the same channel on a log scale: equivalent to just the sum of the spectrogram. The energy time series is contaminated by many transient features, but, in general, a rough correspondence with the three surface types is present. So we settled on time-domain features.

Within this class, we experimented with features derived from the hypothesis that the proprioceptive snapshot may be approximated by an order-p vector autoregressive (VAR) model [15] with unknown matrix parameters,

$$\tilde{u}(i) = A_1\tilde{u}(i-1) + \cdots + A_p\tilde{u}(i-p) + \nu_i \quad ,$$

where ν_i is white Gaussian noise. The relevant sufficient statistics to determine the feedback matrices A_τ, $1 \le \tau \le p$, are the the cross-covariance matrices

$$R(\tau) = \mathrm{Cov}\big(\tilde{u}(i), \tilde{u}(i+\tau)\big), \quad \tau = 0, \ldots, p-1 \quad , \qquad (3)$$

which may be estimated from the snapshot. In practice, a sparse VAR model would be used, which only relies on a subset of these covariances. We found, in fact, that neither the time lags nor the cross-coordinate information yield extra information for our data compared to simply $\mathrm{diag}\, R(0)$, which is the squared energy in each coordinate of the snapshot.

In summary, the features we found most useful were the RMS energy in each channel of $\tilde{u}(i)$, averaged over a window of W samples. The most critical confounding variable in our test setup is vehicle speed: increasing speed causes an increase in RMS energy. (Vehicle load did not vary in our tests.) An immediate way to remove velocity dependence is to use ratios of RMS energies. The three features shown in the bottom plot of Figure 2 are representative of those we used for clustering: log-ratio of roll velocity to cross-track (y) acceleration, log-ratio of pitch velocity to along-track (x) acceleration, and log-ratio of along-track acceleration to vehicle speed. The last feature is needed because the first two ratios are completely independent of the absolute scale of vibrations.

B. Clustering Methodology

Given these features, we investigated two approaches to unsupervised classification: standard clustering using finite Gaussian mixture models (FMMs), and time-linked clustering using hidden Markov models (HMMs). Both models assume conditionally Gaussian outputs,

$$u(i) \mid T(i) = k \sim N(\mu_k, \Sigma_k), \ 1 \le k \le K \quad , \qquad (4)$$

where the means μ_k are arbitrary d-dimensional vectors, and the covariances Σ_k are symmetric positive-definite $d \times d$ matrices. The FMM assumes that classes are independent across time, and identically distributed according to

$$P(T(i) = k) = \alpha_k, \ 1 \le k \le K \quad . \qquad (5)$$

The HMM allows the classes to be linked in time with

$$P(T(1) = k) = \alpha_k, \ 1 \le k \le K \qquad (6)$$

$$P(T(i) = k' \mid T(i-1) = k) = A_{k,k'}, \ 1 \le k, k' \le K \quad , \qquad (7)$$

where α is a probability vector and A is a $K \times K$ Markov transition matrix with nonnegative rows that sum to unity. As

A becomes more diagonally dominant, the temporal linkage increases. If each row of A equals the initial distribution α, the HMM reduces to a FMM with the corresponding α. So, the FMM class structure is captured by the weight vector α, while the HMM requires the transition matrix A and the initial state α. Both the FMM and HMM depend on the means and covariances (μ_k, Σ_k).

The free parameters of the FMM and the HMM are determined using the principle of maximum likelihood (ML)

$$\hat{\theta} = \arg\max_\theta \log P(\mathcal{T}_\mathrm{u} \mid \theta) \tag{8}$$

$$\theta = (\alpha,\, \{\mu_k,\, \Sigma_k\}_{k=1}^K) \quad \text{(FMM)} \tag{9}$$

$$\theta = \big(\alpha,\, A,\, \{\mu_k,\, \Sigma_k\}_{k=1}^K\big) \quad \text{(HMM)} \tag{10}$$

which is done in both cases via the well-known Expectation-Maximization (EM) algorithm. For FMM parameter estimation, we have used a relatively straightforward EM code (e.g., [18, sec. 3.2]). For the HMM, we have used a code which regularizes the parameter space so that similar means and near-singular covariances are avoided [9]. This HMM code also allows deterministic annealing to escape local minima, but we did not find that necessary in this application. Sample models are shown in the leftmost two panels of Figure 4.

C. Model Robustness

Models selected by maximum-likelihood can be highly sensitive to particular features of the training data. For robustness, it is important to moderate these influences by constraining the model. We describe two important steps: for the HMM, we constrain the transition matrix and the initial state, and for both models, we introduce a broad default class.

The HMM transition matrix A will reflect only the terrain transitions present in the training data. In particular, if the training data does not contain any high-probability transitions from terrain k to k', then $A(k,k') \approx 0$—even if there were many transitions from k' to k. Indeed, reversing the route will roughly correspond to replacing A by its transpose, an undesirable sensitivity to an arbitrary choice. Perhaps the only aspect of the estimated A that is intrinsic to the terrain is its diagonals, which are linked to the observed dwell time in a given state. (The dwell time of the HMM in state k is geometrically distributed with mean equal to $1/(1 - A_{k,k})$ epochs.) So, when the derived HMM equals $\hat{A}$, we have used a HMM with $A = aI + (1 - a/K)\mathbf{1}$, where $a = \mathrm{tr}(\hat{A})/K$. A refinement, unnecessary in our experiments, would be to estimate a constrained HMM with this parametric form within the EM iteration itself (similar to [21]).

The initial HMM state raises a similar issue. Our experiments used one long sequence to fit the model. From one observation, there is no statistical basis for estimating the initial state probability, although α is formally defined by (8). Accordingly, we discard the ML initial state vector and simply use a uniform distribution on the hidden states. In any event, because of the strong influence of the data on the state, the initial condition fades immediately and does not materially affect classifications.

When the selected FMM or HMM classifier is used on novel data, new terrain types or new noise sources will, if present, be forced into one of the K existing classes. To capture data that falls outside the existing model structure, we modify the ML classifier by inserting a new class, labeled 0, with a highly diffuse structure. For consistency with the existing modeling setup, we used a Gaussian with $(\mu_0,\, \Sigma_0)$ equal to that of the pooled data. For the FMM, we let the corresponding weight vector equal $[\,\epsilon \quad (1-\epsilon)\alpha\,]$ for a tiny problem-dependent ϵ. In the case of the HMM, we augment the Markov transition matrix:

$$\begin{bmatrix} 1 - K\epsilon & \epsilon & \cdots & \epsilon \\ \epsilon & & & \\ \vdots & & (1-\epsilon)A & \\ \epsilon & & & \end{bmatrix}$$

for a similar ϵ. The initial state vector is uniform over the enlarged set of $K+1$ classes.

D. Measuring Uncertainty

When training the visual classifier, it is useful to compute not just the most likely class assignment, but also the posterior probability of proprioceptive class membership, namely, $P(T(i) = k \mid \mathcal{T}_\mathrm{u})$. We can use this posterior as a screen on the patches admitted as training examples to the visual classifier: if $P(T(i) = k \mid \mathcal{T}_\mathrm{u}) < \tau$, then the sample is discarded, even though k is the most likely class.

Because time epochs are independent in the FMM model, the posterior probability of class membership reduces to

$$p\big(T(i) = k \mid u(i) = u\big) = \alpha_k N(u; \mu_k, \Sigma_k) \Big/ \sum_{j=1}^K \alpha_j N(u; \mu_j, \Sigma_j) \tag{11}$$

where $N(u; \mu, \Sigma)$ is the value of the corresponding normal density at u.

Computing this probability for the HMM is more complex, because past and future classifications influence each point. The method for finding the full sequence of probabilities is the forward-backward algorithm [20] which is an ingredient in the EM estimation procedure as well. This "smoother" runs with time and working state proportional to KN_u. Note that although the probability of the most likely class according to (11) is at least $1/K$, the HMM posterior probability of the most likely class at a given time can fall below this because of the temporal constraint. That is, the most likely class-sequence may contain individual class assigments that are unlikely, because they maintain smoothness of the sequence.

III. Self-Supervised Visual Terrain Classification

This system uses the proprioceptive class as training data to associate surface elevation profile with visual appearance. To relate visual appearance to the learned proprioceptive classification, stereo data from the vision system is used to back-project the class-tagged vehicle path into the image series. After the vehicle has driven a fixed distance from each image frame, the path is projected into the image. Visual

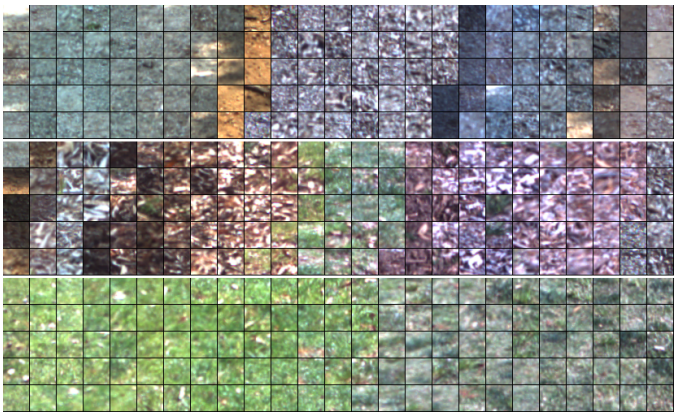

Fig. 3. Example patches used to train the visual classifier. Patches are 32×32 pixels, extracted from regions with a given proprioceptive class. The first five rows are dirt, the next five are mulch, and the last five are grass. Temporal position within the training set progresses left-to-right within each class.

odometry [11] is used to maintain the vehicle pose, which helps avoid artifacts from vehicle slip. Dense stereo processing is used to compute the range to the terrain for each image pixel. A vehicle-sized patch at each pose in the vehicle's path is then intersected with the three-dimensional terrain, yielding the set of pixels in the image over which the vehicle drove (see Figure 5). Each site in an image is thus associated with an interval of time during which the vehicle was present there. If the proprioceptive label changes during this time interval, the class is in doubt. We regard this as having the same significance as the unknown-terrain class.

The resulting appearance training data consists of pairs

$$\mathcal{T}_{\mathrm{v}} = \{(v(i), T(i)) : 1 \leq i \leq N_{\mathrm{v}}\} \tag{12}$$

where each $v(i)$ is an image patch, and each $T(i)$ is a proprioceptive terrain class determined by the learned proprioceptive model. In our tests, we discarded instances of the unknown-terrain class. The training data typically consists of a few thousand examples, drawn about equally from each class.

Figure 3 shows a representative sample of patches drawn from a three-class training set; each class is quite inhomogeneous. The intra-class variability is due to effects like lighting, which might be removed by preprocessing, but also to the inherent many-to-one map of visual appearance to proprioceptive feel (different-looking terrains may feel the same). The various dirt patches in Figure 3 are an example of intra-class lighting variations, while the grass patches are a mild example of different grass colors having the same feel.

Because much of this variation is irreducible, we employ a multi-class SVM [24] to associate visual appearance with a classification. We use the multi-dimensional (8×8×8) color histogram within 32×32 pixel patches as the SVM feature vector input. In similar problem domains [3, 12] we have experimented with direct and normalized RGB tuples, and Gabor textures, but found that the histogram features consistently had a performance edge. We used a One-Against-All (OAA) method to implement the multiclass SVM. At test time, some examples are not claimed by any of the K SVMs in the OAA ensemble; these are given a "decline to classify" label.

IV. Experimental Results

The system was tested on a small tracked vehicle driving at speeds from 0.5 to 2.0 m/s with a constant load in a variety of unprepared, natural terrains with slopes up to 15°. We used an iRobot Packbot chassis with a low-cost IMU and a 640×480 pixel, 0.8 m baseline, stereo color camera pair with a 90° field of view. IMU data was captured at 150 Hz and imagery at 15 Hz. Some runs included driving over only one terrain type, while others covered several types, including mixed terrains at spatial boundaries. All runs included driving over the terrain in various directions, with various sun angles, and in shadows. The terrain was completely unprepared, and manifests significant complexity: tufts of grass among the mulch, patches of dirt in the grass, and mulch that has spread onto the dirt and asphalt roads.

A. Proprioceptive Learning

To analyze the proprioceptive learning system in isolation, we gathered a data set which had reliable ground truth by performing runs in each of four surface types (asphalt, dirt, mulch, and grass) and concatenating these four logs. All surface types had examples across the full range of vehicle speeds. The ground truth labels, of course, are not supplied to either algorithm. The ground truth is not absolutely reliable; indeed, because the terrain was not prepared by us, ground truth is based only on spatial context and visual appearance.

We trained a four-component FMM and a four-component HMM on this dataset with results shown in Figure 4. For both the FMM and HMM, we used various initializations and runs to thousands of EM updates to assess local maxima in (8). We did not find prominent local maxima in the HMM optimization besides the well-known ones around singular covariances, which both codes are designed to avoid. The FMM did have one other prominent local maximum, which combined part of the dirt class into the mulch class. The likelihood of this maximum was lower than the solution shown in Figure 4, but existence of prominent local maxima in the FMM objective function seems to be an effect of removing the constraint implied by temporal linking.

The HMM outperformed the FMM in two other ways. First, the error rate of the HMM relative to ground truth was 8.10%, versus 18.0% for the FMM, computed over 6600 feature vectors. (This difference is far greater than could be reasonably accounted for by errors in the ground truth.) Second, the temporal pattern of errors for the FMM was spread throughout the data, while those for the HMM were in isolated clumps. The clumped pattern is more characteristic of a terrain that is mostly one class, but is occasionally interspersed with other roughness types.

We used both models to classify a separate test run. In this run, the robot drove freely across dirt, mulch, and grass terrains, so there are substantial mixed terrains. The log-energy feature vectors for this run are shown in Figure 2. The FMM

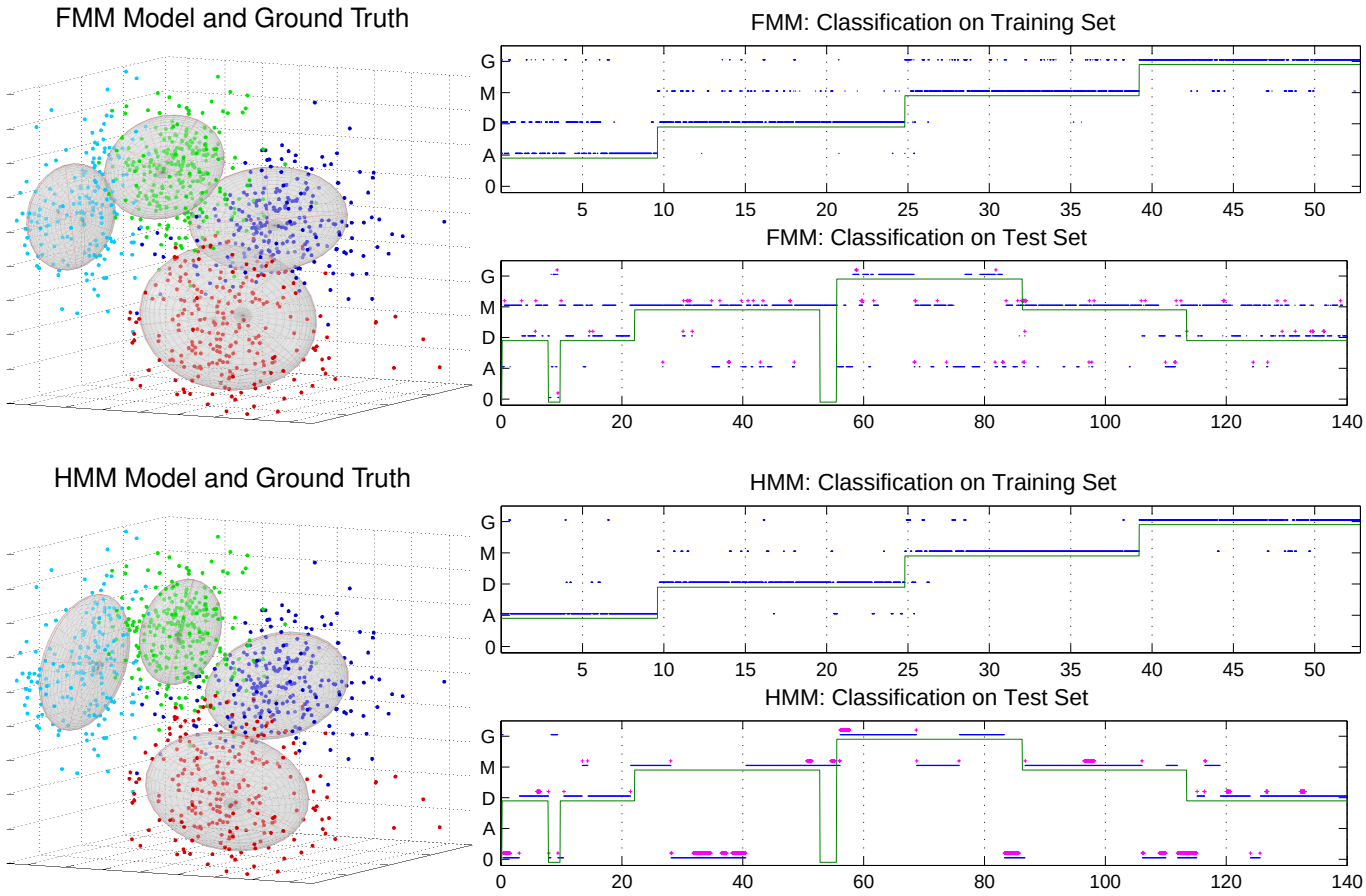

Fig. 4. Proprioceptive models and classifications. Perspective plots at left show extracted clusters, in a 3D feature space of log-energy features, for the FMM (top) and HMM (bottom). Ground truth labels for the training data are overlaid upon both sets of clusters (cyan: asphalt; green: dirt; blue: mulch; red: grass). To the right of each model plot, a pair of time series of classifications is shown. The top plot in each pair shows the training set, and the bottom plot shows a separate test set. The time axis is in seconds, and the discrete class labels are abbreviated on the ordinate, with $\emptyset$ for the unknown-terrain class. The approximate ground truth is green, and the learned classification is blue. In the test set classifications, the intervals of uncertainty in the FMM/HMM classification are in magenta.

labeling (Fig. 4) seems too noisy to use reliably for navigation without some postprocessing. The HMM, however, is able to recover the ground truth well enough to provide a basis for appearance-based planning, as we will see in the next section.

There are three notable features in the HMM time series. First, there is a long interval of apparent misclassification, around $t = 70$ s, where $T_0 =$ grass but $T_u =$ mulch. Comparing with the proprioceptive time series in Figure 2, we can see that this interval did not in fact have the same proprioceptive feel as the surrounding grass; this was also observed in a differently-oriented pass over the same area. From both the image log and an after-the-fact site inspection, this area is very close in appearance to its surroundings. This time interval illustrates both the approximate nature of ground truth, and the fact that visual class and proprioceptive class do not always coincide. We discuss the implications of this fact in the conclusions.

The other two observations concern uncertainty in proprioceptive labels. First, there are two long intervals where the HMM class is unknown, around $t = 30$ s and $t = 110$ s. Both occur when the robot is on a slope (the first interval is uphill, the second downhill). The training data did not include slopes, nor have we accounted for them by modeling, so there was no way for the classifier to determine the effect of slope on terrain feel.

The other instances of the unknown class fall around terrain changes. For instance, the middle panel of Figure 5 shows a typical grass-to-mulch transition (around $t = 85$ s). The apron of mixed grass and mulch with the white overlay corresponds to the unknown class. Other terrain changes (Fig. 1, left panel) show shorter spans of unknown terrain.

Finally, the uncertainty estimate produced by the forward-backward algorithm is shown as a magenta annotation in the HMM test set plot. (The threshold for plotting is $\tau = 0.7$.) One such uncertain interval, around $t = 95$ s, is back-projected into imagery in the last panel of Figure 5. The uncertain classification, shown as a striped overlay, seems due to the presence of large tufts of grass within the mulch.

B. Visual Learning

To test the end-to-end system, we extracted patches from labelings the HMM system made of the test data set just described; these labels are in the lower right plot of Fig-

Fig. 5. Back-projected proprioceptive classifications. Top: Classification is invariant to lighting. Middle: Transition from grass (red overlay) to unknown-terrain class. Bottom: Region labeled mulch (blue overlay) but partly with low confidence (striped area) due to intervening grass. Another back-projected classification is in the left panel of Fig. 1.

ure 4. A temporally-ordered sampling of this three-class proprioceptively-classified data set is in Figure 3. The strip of grassy patches within the mulch training data corresponds to the visual/proprioceptive disconnect observed around $t = 70$ s.

Our testing of the visual subsystem was directed at determining how well a visual classifier captures the wide variety of visual features having the same proprioceptive feel. We trained the SVM classifier on a set of 1347 dirt patches, 1888 mulch, and 576 grass, and tested on a set of a similar size and makeup. The test patches were drawn from different frames within the same run, alternating every five frames, or about 0.5 m

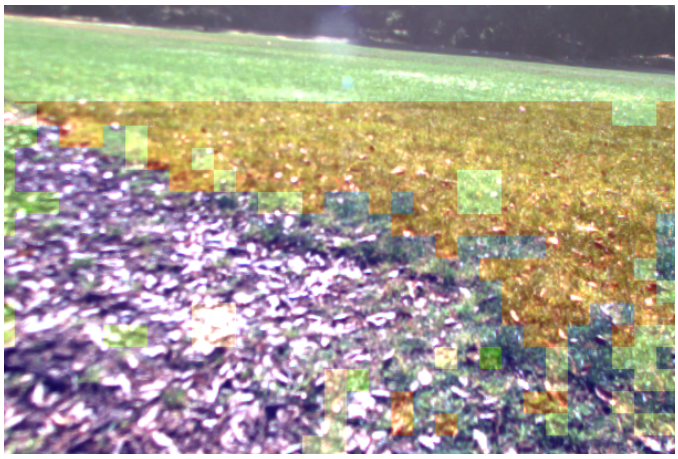

Fig. 6. Example image from visual classifier showing a mulch (blue overlay) and grass (red overlay) boundary. For another classified scene, see the right panel of Fig. 1.

of pose. We performed a grid search to determine the best SVM parameters on the training data, selecting a regularization coefficient of 50.0 and an RBF kernel with width 7.0.

Unrestricted driving within our unstructured test environment poses realistic learning challenges, but does not, as we have seen, permit reliable ground truth, so we cannot yet measure end-to-end error rates. However, we can approximate them as the sum of a proprioceptive error rate within more structured conditions, plus an approximation error between the visual classifier and the proprioceptive training signal.

The confusion matrix of underfoot classification T_u versus visual classification T_v is as follows, where both percentages and counts are given.

		T_v = **∅**		**D**		**M**		**G**	
		%	(#)	%	(#)	%	(#)	%	(#)
T_u	**D**	1.7	(23)	90.6	(1260)	7.5	(104)	0.3	(4)
	M	1.4	(26)	5.4	(101)	91.3	(1720)	1.9	(36)
	G	5.5	(31)	0.0	(0)	6.7	(38)	87.8	(497)

The overall error rate on the 97.9% of the instances which the SVM classified is 7.5%.

The error patches, where $T_u \neq T_v$, are in Figure 7. The dominant confusion in terms of impact on error rate is between dirt and mulch. One key factor in this confusion is the time interval around $t = 110$ s which contains mixed terrain: this is where roughly the last nine columns of $T_u =$ mulch, $T_v =$ dirt errors occur. The second main confusion is between mulch and grass, due to two factors. First, as noted, the presence of zones that look like grass but feel like mulch, and are so labeled in the test set. These chips cause the third type of error ($T_u =$ mulch, $T_v =$ grass). Second, the presence of these chips in the *training* set, which causes some ordinary grass chips to be labeled as mulch by the SVM.

Figure 6 shows a classified image in which the mulch/grass boundary is identified. (The foreground of the image is mulch, and it is almost completely covered by blue overlay.) Another classified image from the same run is in the right panel of

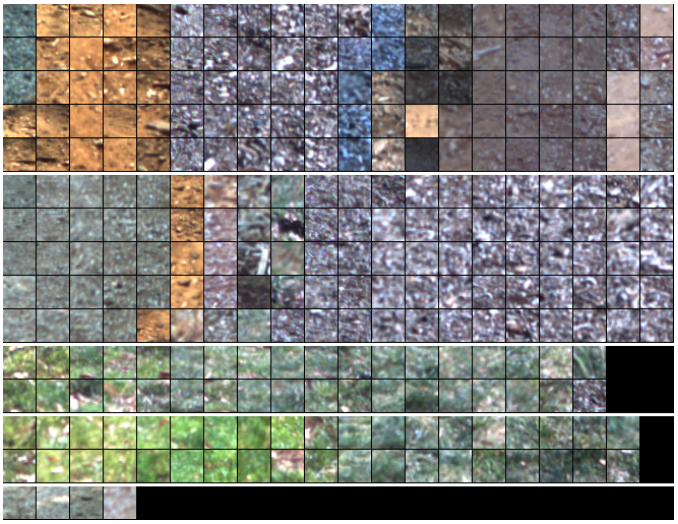

Fig. 7. Chips misclassified by the visual classifier. First five rows: $T_u =$ dirt, $T_v =$ mulch. Next five: $T_u =$ mulch, $T_v =$ dirt. Next two rows: $T_u =$ mulch, $T_v =$ grass. Next two: $T_u =$ grass, $T_v =$ mulch. Last: $T_u =$ dirt, $T_v =$ grass.

Figure 1. The dirt path is identified right up to its edge, and the mulch along the path border is correctly delineated. If used in practice, it would be advisable to take advantage spatial context in forming the visual classifications, for example with a spatial Markov random field, but for our purposes, spatial smoothing would suppress useful performance cues.

V. Conclusions

We have shown that hidden Markov model clustering can identify proprioceptive surface features using high-rate IMU data. Comparison with temporally-unlinked clustering by Gaussian mixtures shows that conventional independence assumptions result in more local maxima and significantly higher error rates on the data we examined. We developed techniques for adapting the learned HMM by adding a state to absorb previously unseen vibration signatures, and constraining its transition probability matrix for robust operation. We also developed an indication of label reliability which relies on the forward-backward algorithm to compute the posterior probability of each individual proprioceptive label.

Linking this novel proprioceptive classifier to a robust multiclass SVM results in a system that can learn the visual appearance of each driving surface, over time, with no operator intervention required. We demonstrated the end-to-end feasibility of the system, and derived an approximate composite error rate of 17% without using spatial smoothing. Training both of these classifiers is tractable; the relevant parts of the visual classifier can run on the robot.

These results suggest two directions for future work. First, although we developed effective means of normalizing features for vehicle speed, there will be other confounding variables in practice, such as vehicle load and surface slope. One method to limit the influence of these variables is to use the IMU signal in the presence of a vehicle model to derive the forces acting on the vehicle.

The second issue is that terrain with different proprioceptive properties may not be visually distinguishable. One mechanism to deal with this problem is to constrain the proprioceptive clustering using visual properties of underfoot regions, which could also be a tool to produce more robust clusterings. Another mechanism is to contract the observed visual classes to their core elements. This would have the effect of suppressing outlying instances which could be accounted for by membership in another class.

Acknowledgments

The research described here was carried out by the Jet Propulsion Laboratory, California Institute of Technology, with funding from the ARO. We thank Robert Granat for use of his algorithm and code for regularized fitting of hidden Markov models.

References

[1] A. Angelova, L. Matthies, D. Helmick, and P. Perona. Learning slip behavior using automatic mechanical supervision. *IEEE International Conference on Robotics and Automation*, 2007.

[2] A. Angelova, L. Matthies, D. Helmick, and P. Perona. Learning and prediction of slip using visual information. *Jour. Field Robotics*, 2007.

[3] M. Bajracharya, B. Tang, A. Howard, M. Turmon, and L. H. Matthies. Learning long-range terrain classification for autonomous navigation. *IEEE International Conf. on Robotics and Automation*, 2008.

[4] C. Brooks and K. Iagnemma. Self-supervised classification for planetary rover terrain sensing. *Proc. IEEE Aerospace Conf.*, March 2007.

[5] E. Coyle and E. G. Collins. A comparison of classifier performance for vibration-based terrain classification. *26th Army Science Conf.*, 2008.

[6] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski. Self-supervised monocular road detection in desert terrain. *Robotics: Science and Systems Conference*, 2006.

[7] C. Dima, N. Vandapel, and M. Hebert. Classifier fusion for outdoor obstacle detection. *IEEE Internat. Conf. Robotics & Automation*, 2004.

[8] P. Giguere and G. Dudek. Clustering sensor data for autonomous terrain identification using time-dependency. *Autonomous Robots*, pp. 171–186, March 2009.

[9] R. Granat, G. Aydin, M. Pierce, Z. Qi, and Y. Bock. Analysis of streaming GPS measurements of surface displacement through a web services environment. *Proc. IEEE Sympos. on Computational Intelligence and Data Mining*, 2007.

[10] M. Happold, M. Ollis, and N. Johnson, Enhancing supervised terrain classification with predictive unsupervised learning. *Robotics: Science and Systems Conference*, 2006.

[11] A. Howard. Real-time stereo visual odometry for autonomous ground vehicles. *IEEE/RSJ Conf. on Intelligent Robots and Systems*, 2008.

[12] A. Howard, M. Turmon, L. Matthies, B. Tang, A. Angelova, and E. Mjolsness. Towards learned traversability for robot navigation: From underfoot to the far field. *Journal of Field Robotics*, 2006.

[13] D. Kim, J. Sun, S. Oh, J. Rehg, and A. Bobick. Traversability classification using unsupervised on-line visual learning for outdoor robot navigation. *International Conference on Robotics and Automation*, 2006.

[14] A. Lacaze, K. Murphy, and M. DelGiorno. Autonomous mobility for the Demo III experimental unmanned vehicles. *AUVSI Conference on Unmanned Vehicles*, 2002.

[15] H. Lütkepohl, *New Introduction to Multiple Time Series Analysis*, Springer, 2007 (corr. ed.).

[16] M. Maimone, J. Biesiadecki, E. Tunstel, Y. Cheng, and C. Leger. Surface navigation and mobility intelligence on the Mars exploration rovers. *Intelligence for Space Robotics*, 2006.

[17] R. Manduchi, A. Castano, A. Talukder, and L.H. Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous Robots*, 18:81–102, 2005.

[18] G. J. McLachlan and D. Peel, *Finite Mixture Models*, Wiley, 2000.

[19] M. Procopio, J. Mulligan, and G. Grudic. Long-term learning using multiple models for outdoor autonomous robot navigation. *International Conference on Intelligent Robots and Systems*, 2007.

[20] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2): 257–286, 1989.
[21] S. Roweis. Constrained hidden Markov models. *Advances in Neural Information Processing Systems 12*, 782–788, MIT Press, 2000.
[22] D. Silver, J. A. Bagnell, and A. Stentz. Applied imitation learning for autonomous navigation in complex natural terrain. *Field and Service Robotics*, July, 2009.
[23] B. Sofman, E. L. Ratliff, J. A. Bagnell, J. Cole, N. Vandapel, and A. Stentz. Improving robot navigation through self-supervised online learning. *Journal of Field Robotics*, 23(12), December 2006.
[24] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.
[25] C. Ward and K. Iagnemma. Speed independent vibration-based terrain classification for passenger vehicles. *Vehicle System Dynamics*, 47(9): 1095–1113, September 2009.
[26] C. Wellington and A. Stentz. Learning predictions of the load-bearing surface for autonomous rough-terrain navigation in vegetation. *International Conference on Field and Service Robotics (FSR)*, 2003.
[27] J.Y. Wong. *Theory of Ground Vehicles*. Wiley, 2001.