Journal of Machine Learning Research 1 (2005) 1-48 Submitted 4/00; Published 10/00

Learning for Autonomous Navigation: Extrapolating from Underfoot to the Far Field

Larry Matthies, Michael Turmon, Andrew Howard,
Anelia Angelova, Benyang Tang FIRSTNAME.LASTNAME@JPL.NASA.GOV
Jet Propulsion Laboratory
California Institute of Technology, Pasadena, CA, 91109, USA

Eric Mjolsness EMJ@UCI.EDU
Department of ICS, University of California, Irvine, CA, 92697, USA

Editor: Jane Mulligan, Greg Grudic

Abstract

Autonomous off-road navigation of robotic ground vehicles has important applications on Earth and in space exploration. Progress in this domain has been retarded by the limited lookahead range of 3-D sensors and by the difficulty of preprogramming systems to understand the traversability of the wide variety of terrain they can encounter. Enabling robots to learn from experience may alleviate both of these problems. We define two paradigms for this, *learning from 3-D geometry* and *learning from proprioception*, and describe initial instantiations of them we have developed under DARPA and NASA programs. Field test results show promise for learning traversability of vegetated terrain, learning to extend the lookahead range of the vision system, and learning how slip varies with slope.

1. Introduction

Robotic ground vehicles for outdoor applications have achieved some remarkable successes, notably in autonomous highway following (Dickmanns, 1992), planetary exploration (Maimone, 2004), and off-road navigation on Earth (Lacaze, 2002). Nevertheless, major challenges remain to enable reliable, high-speed, autonomous navigation in a wide variety of complex, off-road terrain. 3-D perception of terrain geometry with imaging range sensors is the mainstay of off-road driving systems. However, the stopping distance at high speed exceeds the effective lookahead distance of existing range sensors. Moreover, sensing only terrain geometry fails to reveal mechanical properties of terrain that are critical to assessing its traversability, such as potential for slippage, sinkage, and the degree of compliance of potential obstacles. Rovers in the Mars Exploration Rover (MER) mission have got stuck in sand dunes and experienced significant downhill slippage in the vicinity of large rock hazards. Earth-based off-road robots today have very limited ability to discriminate traversable vegetation from non-traversable vegetation or rough ground. It is impossible today to preprogram a system with knowledge of these properties for all types of terrain and weather conditions that might be encountered. The 2005 DARPA Grand Challenge robot race, despite its impressive success, faced few of these issues, since the route was largely or completely on smooth, hard, relatively low-slip surfaces with sparse obstacles and no dense, vegetated ground cover on the route itself.

Learning may alleviate these limitations. In particular, 3-D geometric properties of obstacle vs. drivable terrain are often correlated with terrain appearance (e.g., color and

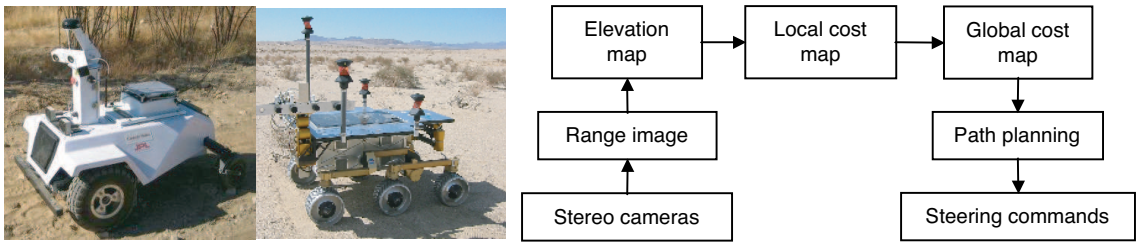

Figure 1: LAGR robot (left), Rocky 8 robot (center), and a simple view of their baseline navigation software architecture (right). Both robots are just over 1 meter long.

texture) in 2-D imagery. A close-range 3-D terrain analysis could then produce training data sufficient to estimate the traversability of terrain beyond 3-D sensing range based only on its appearance in imagery. We call this *learning from 3D geometry* (Lf3D). In principle, information about mechanical properties of terrain is available from low-level sensor feedback as a robot drives over the terrain, for example from contact switches on bumpers, slip measurements produced by wheel encoders and other sensors, and roughness measurements produced by gyros and accelerometers in the robot's inertial measurement unit (IMU). Recording associations between such low-level traversability feedback and visual appearance may allow the learning of mechanical properties from visual appearance alone; we call this *learning from proprioception* (LfP). While learning-related methods have a long, extensive history of use for image classification and robot road-following, work in the paradigms described here is quite limited. LfP has been addressed recently in formulations aimed at estimating where the ground surface lies under vegetation and closely related work (Wellington, 2005). We are unaware of work closed related to the Lf3D paradigm.

This paper outlines some key issues, approaches, and initial results for learning for off-road navigation. We describe work in the DARPA-funded Learning Applied to Ground Robotics (LAGR) program and the NASA-funded Mars Technology Program (MTP). Both use wheeled robotic vehicles with stereo vision as the primary 3-D sensor, augmented by an IMU, wheel encoders, and in LAGR, GPS; they also use closely related software architectures for autonomous navigation (Figure 1). Section 2 outlines these architectures and how they need to change to address Lf3D and LfP. Sections 3, 4, and 5 present results of our initial work on Lf3D and two flavors of LfP, one aimed at learning about vegetation and the other aimed at learning about slip. Our work to date necessarily stresses fairly simple methods with real-time performance, due to the demonstration-oriented nature of the LAGR and MTP programs; nevertheless, the results justify the value of our approaches and their potential to evolve to more sophisticated methods.

2. Architectures and issues

The baseline navigation software architecture in both the LAGR and MTP programs operates roughly as follows (Figure 1). Stereo image pairs are processed into range imagery, which is converted to local elevation maps on a ground plane grid with cells roughly 20 cm square covering 5 to 10 m in front of the vehicle, depending on camera height and resolution. Geometry-based traversability analysis heuristics are used to produce local,

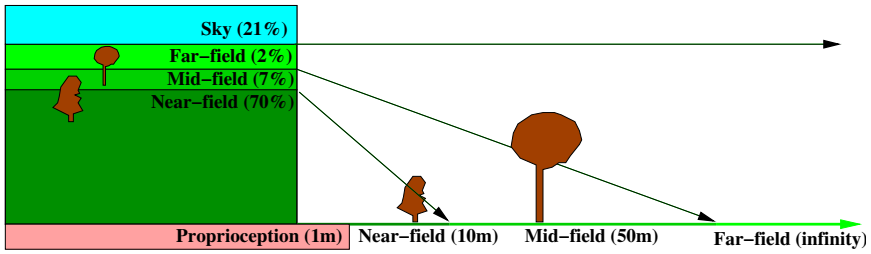

Figure 2: Typical information zones from proprioception and stereo (image space, left; map space, right), with specific numbers for the LAGR robot. See text for discussion.

grid-based, "traversability cost" maps over the same area, with a real number representing traversability in each map cell. The local elevation and cost maps are accumulated in a global map as the robot drives. Path planning algorithms for local obstacle avoidance and global route planning are applied to the global map; the resulting path is used to derive steering commands sent to the motor controllers.

Figure 2 gives a deeper look at information available from stereo vision and how this relates to Lf3D, LfP, and richer local map representations. The near-field region is where stereo vision gets range data of sufficient resolution to build a local elevation map; color and texture information (collectively, "appearance") is also available for insertion in the map. The mid-field region is where the range data samples the ground too sparsely to create a useful elevation map; image appearance is also available. The far-field region is beyond the range of stereo vision (i.e. zero disparity), so only image appearance is available. To make things concrete, in LAGR, the near-field is about 70% of the image, the mid-field is 7%, and the far-field is 2%; conversely, on the ground plane, the near-field extends to about 10 m, the mid-field from 10 to 50 m, and the far-field from 50 m to infinity (right side of Figure 2). The inset at the bottom of the image in Figure 2 represents information available from the proprioceptive sensors: the bumper, IMU, and wheel encoders.

Given this view, our problem can be cast as transferring knowledge between the adjacent distance regimes in Figure 2: (1) between underfoot and near-field (proprioception vs. appearance plus rich geometry), (2) between near-field and mid-field (appearance plus rich geometry vs. appearance plus poor geometry, and (3) between mid-field and far-field (appearance plus poor geometry vs. appearance only). In this process, the effective lookahead distance of the sensors is extended by using the learned correlations to ascribe properties sensed proprioceptively or geometrically in the closer zones to regions sensed just by appearance or weaker geometric perception in the more distant zones. The same obstacle classes, and sometimes the same obstacle, will be present across all zones. Our ultimate goal is to jointly estimate terrain traversability across zones, unifying the Lf3D and LfP concepts, and encompassing slippage, sinkage, and obstacle compliance in the notion of traversability.

Proxies and learned estimators of traversability

Traversability T is a random variable associated with a certain site s, either a pixel in the scene or a cell in the map. $T(s)$ always takes values in the unit interval, but depending on context, we may take it to be binary (e.g., bumper hits) or real-valued (e.g., wheel slip). In making the link to path planning, it may be helpful to define $T(s)$ as the probability that the robot can successfully move out of a map cell s after deciding to do so. We could imagine

a physics-based simulation that would determine this *self-transition probability* given vehicle and terrain parameters. Accumulating this T over a path would then yield the cumulative probability of a successful sequence of moves.

Lacking such a model, we view T as a random variable to be predicted from correlated information, where the predictor is in turn learned from training data. Our learning strategy is to use high-quality (typically, nearer the robot, as in Figure 2) input examples to produce training labels $\tilde{T}$, which serve as proxies for the unknown T. The proxy labels are given to a learning algorithm which trains a regression model $\hat{T}$ that approximates $\tilde{T}$. The regression model is then used to drive the robot. In Lf3D, object location statistics, specifically elevation changes, are used to provide the proxy $\tilde{T}$, which is estimated using appearance information (normalized color in our case). In LfP, the proprioceptive inputs (e.g., bumper hits and slip) are used to generate the proxy $\tilde{T}$, which is then estimated using the available appearance and geometry information from stereo images.

3. Learning near-field traversability from proprioception

In the LAGR program, we are using the LfP paradigm to address the key problem of learning about traversability of vegetation. For robots in general, the bumper, IMU, and slip measurements ultimately will all be important for this. In practice, for the robot and terrain used in the LAGR program to date, the bumper provides most of the information, so we currently model the proxy $\tilde{T}$ as a 0/1 quantity. Operationally, we gather samples of $\tilde{T}$ by recording the location statistics of the brush we can and cannot push through.

A technical problem of *blame attribution* arises because roughly six map cells are overlapped by the bumper at any time, so the nontraversable samples are contaminated with data from traversable cells. Heuristics alone may prove sufficient to narrow down blame to one cell, or a constrained clustering approach may be needed to separate these two classes. To avoid solving the blame attribution problem, initially we obtained training data from hand-labeled image sequences: a human identifies a mix of traversable and untraversable map cells.

Terrain representation

Elevation maps *per se* do not adequately capture the geometry of vegetated and forested terrain. Three-dimnensional voxel density representations have been used successfully in the past with range data from ladar (Lacaze, 2002). We are experimenting with such a representation for range data from stereo vision. The space around the robot is represented by a regular three-dimensional grid of $20\,\text{cm} \times 20\,\text{cm} \times 10\,\text{cm}$ high voxels. Intuitively, we expect that only low-density voxels will be penetrable (see Figure 3). The voxel density grid is constructed from range images by ray-tracing: for each voxel, we record both the number of *passes* (rays that intersect the voxel) and the number of *hits* (rays that terminate in the voxel). The per-voxel density ρ equals the ratio of hits to passes. Since the ground may be non-planar, we also identify a *ground voxel* g in each voxel column; we assume that this voxel represents the surface of support for a robot traversing this column. The ground voxel is determined using a simple heuristic that locates the lowest voxel whose density exceeds some preset threshold. Although calculating it is relatively complex, in practice the density estimate is robust and rich in information.

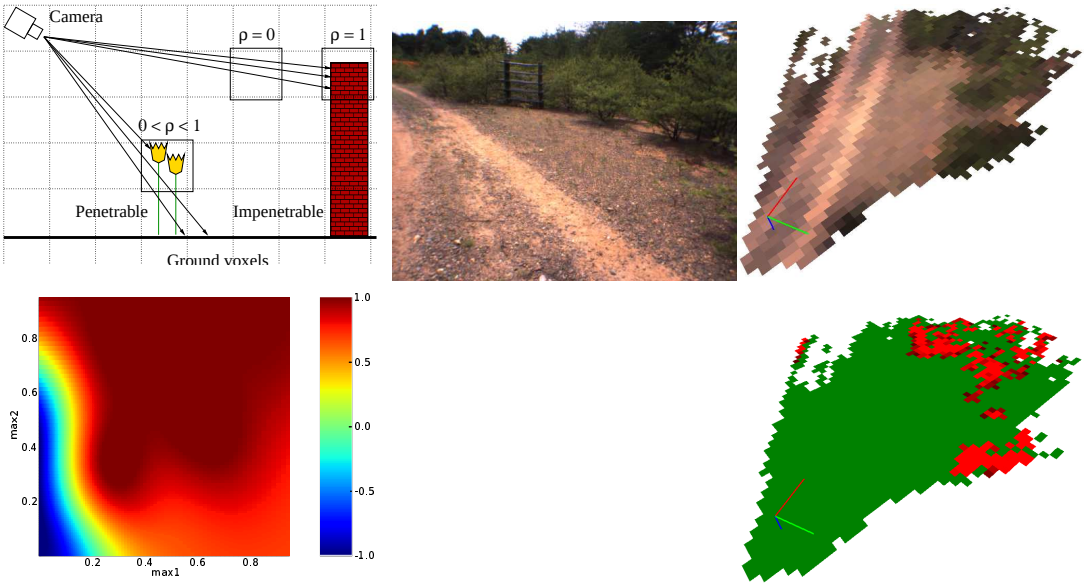

Figure 3: Learning from proprioception. Top row: schematic illustrating the voxel density map representation (left), sample camera image (middle), and its projection onto a local map (each map cell is colored with the mean of all pixels projecting into it). Bottom left: learned cost lookup table as a function of ρ^* and ρ^{**}. Bottom right: cost map computed from voxel densities; green is traversable, red is not.

Each map cell s has an above-ground density column $[\rho_{g(s)+1}\, \rho_{g(s)+2} \cdots \rho_{32}]$. For simplicity, we have started with the following reduced feature set: ρ^*, the maximum density; ρ^{**}, the next-highest; i^*, the height of ρ^*; and i^{**}. We have used ρ^* and ρ^{**} below, but ρ^* and i^* provide similar performance when used together. The average color within a map cell would also be a good feature, but we have not used this in classifications yet.

Learning algorithm

Initially, we wanted to validate the use of the density features and to replace our existing hand-coded, geometry-based traversability cost heuristic with a learned value. In this offline context, training time is not an issue so we use a Support Vector Machine (SVM) classifier. We used a radial basis function kernel, with the SVM hyper-parameters estimated by cross-validation. The training data consisted of 2000 traversable and 2000 non-traversable examples, and the resulting model has 784 support vectors (SVs). The large number of SVs for a relatively modest two-dimensional problem indicates a considerable degree of overlap between the classes which is borne out in scatter plots. Tests were performed on an independent image sequence which contains roughly 2000 examples. We achieved a classification error rate of 14% on the test set, again indicative of strong class overlap from these limited features.

Classification is done at frame rates of 2–5 Hz, so SVM query time would be prohibitive. We therefore coded the SVM into a lookup table (LUT) for speed and simplicity, but a reduced-set SVM would be easy to substitute (Schölkopf, 1999). The continuous output f of the SVM is turned into a traversability classification by thresholding at zero. The results

of SVM classification and the LUT are shown in Figure 3. This system was used for path planning in LAGR test 5.

4. Learning mid and far-field traversability from near-field 3-D geometry

To address another goal of the LAGR program, we are using the Lf3D paradigm to extend near-field range-based proxies $\tilde{T}$ to mid-field and far-field traversability estimates $\hat{T}$. Here $\tilde{T}$ is a function of the heights of all pixels landing in a $(20\,\mathrm{cm})^2$ map cell. When at least 10 pixels land in one cell, their height standard deviation σ_z becomes resolvable: a large value indicates rough ground or obstacles. We compute a traversability proxy $\tilde{T} = f(\sigma_z)$ for the cell, which is associated with the appearance u of all pixels mapping into that cell, thus providing a training set $\mathcal{T}$ of $(u, \tilde{T})$ pairs. We use this $\mathcal{T}$ to select an extrapolating function $\hat{T} = \hat{T}(u)$ from appearance to traversability.

We now use only two appearance-based features: the mean RGB color within a map cell, normalized to sum to unity. These are referred to as normalized R and G. Training data for the experiments below was gathered from 10 consecutive frames ($\sim$4500 examples). We would like to retrain classifiers at each frame, so speedy training and evaluation are required, prompting the reduction of $\mathcal{T}$ to a parameterized model. Below, we consider two approaches: unsupervised clustering using k-means, and supervised discriminant analysis with Mixtures of Gaussians (MoG); a close variant of the first was used in LAGR test 7.

Unsupervised K-means

The geometry-based proxy is itself heuristic, so we might prefer to use $\tilde{T}$ somewhat weakly. We had success with unsupervised clustering of the input pixel appearance, followed by deducing the per-cluster traversability from the average proxy value within each cluster. That is, we discard the $\tilde{T}$ labels within $\mathcal{T}$ and perform a k-means clustering with $K = 5$. Figure 4 shows such a clustering; each cluster is annotated with its mean σ_z in mm (low values correspond to traversable sites). The coherence of these values shows that the appearance-based clusters capture the traversability structure. The traversability estimate is a weighted average of per-cluster traversabilities

$$\hat{T}(u) = \sum_{k=1}^{K} \tilde{T}_k \exp(-\|u - \mu_k\|^2/\lambda^2) \Big/ \sum_{k=1}^{K} \exp(-\|u - \mu_k\|^2/\lambda^2),$$

where $\tilde{T}_k$ is the average traversability proxy value per cluster and μ_k is the kth cluster center. To form a classifier and compute error rates, pixels with $\hat{T}(u) \geq \tau$ are considered traversable, otherwise not; τ is chosen so that 65% of the data is traversable. The parameter λ is selected using cross-validation. Classification error is 7% with $\lambda = 0.2$. (Note: we report within-sequence error rates because extrapolation is done within a single frame.)

Supervised MoG-based discriminants

It may be preferable to constrain the cluster membership a priori (using $\tilde{T}$ up front) rather than extracting clusters after the fact. At the expense of some reliance on a prior rule about association based on $\tilde{T}$, we may extract more stable and homogeneous clusters. We have experimented with three approaches: introducing $\tilde{T}$-based cannot-link constraints into k-means (Wagstaff, 2001), stratifying the cluster memberships according to $\tilde{T}$ within the EM algorithm in a semi-supervised framework (McLachlan, 2000, sec. 2.20) and adopting a two-class discriminant-based approach with populations determined by $\tilde{T}$.

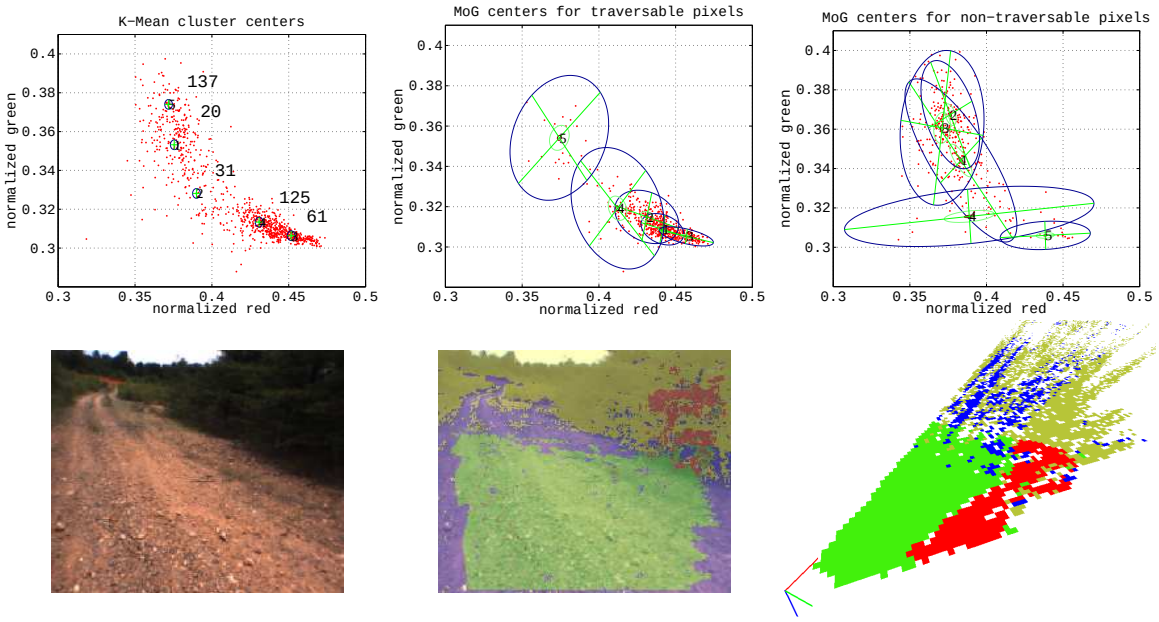

Figure 4: *Top row:* k-means clustering annotated by within-cluster height variance (left); Gaussian mixture models for $p_1(u)$ (center) and $p_0(u)$ (right). *Bottom row:* image from the training set (left), same image with k-means classification overlay (center), and the corresponding map (right). The map extends through the mid-field, nearly 50 m. Training pixels are in light green (traversable, $\tilde{T} = 1$) and red ($\tilde{T} = 0$). Classified pixels are in blue ($\hat{T} = 1$) and olive green ($\hat{T} = 0$).

In the discriminant approach, rough thresholds are used to form sets of negative and positive examples: $\mathcal{T}_0 = \{(u, \tilde{T}) : \tilde{T} \leq \tau\}$, $\mathcal{T}_1 = \{(u, \tilde{T}) : \tilde{T} > \tau\}$. We selected $\tau = 0.4$ so that 65% of $\mathcal{T}$ belongs to $\mathcal{T}_1$. It could be helpful to define a "don't-care" traversability class by using separate thresholds. Two separate MoGs $p_0(u)$ and $p_1(u)$ are fit to the two training sets, and we declare a pixel traversable if $p_1(u)/p_0(u) > 0.65/0.35$. We have used $K = 5$ component, full-covariance Gaussian mixtures to parameterize each of the two distributions, and fit the parameters by maximum-likelihood using the EM algorithm (Figure 4). As with k-means, there is good separation between the two classes. The test error achieved is 6%. Training time for $N = 1000$ is about 40 ms in an unoptimized code. Evaluation time for 5000 pixels (about 9% of a 192×256 pixel image) is less than 10 ms, which easily permits training and evaluation at our path-planning rates of 2–5 Hz.

Putting traversability in the map

Note that both approaches classify terrain in the *image* space: each pixel is assigned a traversability estimate $\hat{T}$. To use this result for navigation, these values must be projected into the map. There are two issues: how to combine traversability estimates and proxies, and how to determine the 3D location of image pixels. For data fusion, we currently allow traversability proxies derived from geometry (the near-field training set) to override traversability estimates inferred from appearance (the mid- and far-field query set). To project pixels from the image into the map, when the pixel has a non-zero disparity (near- and mid-field), 3D locations are computed by triangulation. Because of range uncertainty,

this leads to maps with more blur with increasing range; at present, this is unavoidable. When disparity is zero (far-field), pixels can in principle be projected onto a nominal ground plane; currently, we ignore these pixels.

Sample classifications for k-means Lf3D are in the bottom row of Figure 4 (MoG results were similar). The technique can extend the range of look-ahead that can be used in path planning by up to a factor of five. The main limitation of these methods at present is the low information content of the appearance features used. Introducing local texture measures should substantially improve classification accuracy and robustness, at which point new algorithmic tradeoffs could be evaluated.

5. Learning slip from proprioception

While the main terrain unknown so far in LAGR has been the compliance of vegetation, on Mars it has been slippage on slopes. Slip measures of lack of progress of the vehicle and can be defined as the difference between the commanded velocity and the actual velocity, which is estimated here using visual odometry ("VO") (Helmick, 2004). Slip influences total traversability cost: the robot's mobility on certain terrains significantly degrades, especially as slope angle increases (Lindemann, 2005). Thus, we seek to improve path planning by predicting slip before entering a given terrain. We ultimately intend to address compliance and slip in a unified framework, but for now we are addressing them separately in each domain.

Slip depends on terrain slope, but the precise relationship varies with terrain type, so both geometry and appearance must be observed. Slip learning fits into our proprioceptive learning framework: information about the terrain geometry s and appearance u of pixels within a map cell, collectively referred to as $\{(u, s)\}$, is measured from stereo imagery. At training time, this information is correlated to the traversability proxy, the robot's slip, as the robot traverses the cell. At query time, slope and appearance alone are used to estimate slip.

Slip learning framework

Slip is learned in a Mixture of Experts framework: terrain is classified first using appearance information and then slip, as a function of terrain slopes, is learned (Angelova, 2006). The rationale for doing that is: 1) terrain type and appearance are approximately independent of slope; 2) introducing this structure helps constrain learning to better balance limited training data and a potentially large set of texture features. Here we focus on learning slip as a function of slopes when the terrain type is known. We briefly outline the terrain type classification algorithm, which is a subject of our current work.

Slip is a nonlinear function of terrain slopes. We use the Locally Weighted Projective Regression method (Vijayakumar, 2005), preferring it to other nonlinear approximation methods, like Neural Networks, because it can be easily extended to online learning. The slip $\hat{T}$ is estimated in terms of input slopes $\mathbf{x}$ via

$$\hat{T}(\mathbf{x}) = \sum\nolimits_{c=1}^{C} K(\mathbf{x}, \mathbf{x}_c)\big(b_0^c + \sum\nolimits_{i=1}^{r} b_i^c \left\langle \mathbf{p}_i^c, \mathbf{x} \right\rangle\big)$$

where $K(\mathbf{x}, \mathbf{x}_c) = \exp(-\|\mathbf{x} - \mathbf{x}_c\|^2/\lambda)$ is a receptive field centered about $\mathbf{x}_c$, controlling the dominant local linear regression model, and r is the number of linear projections (here

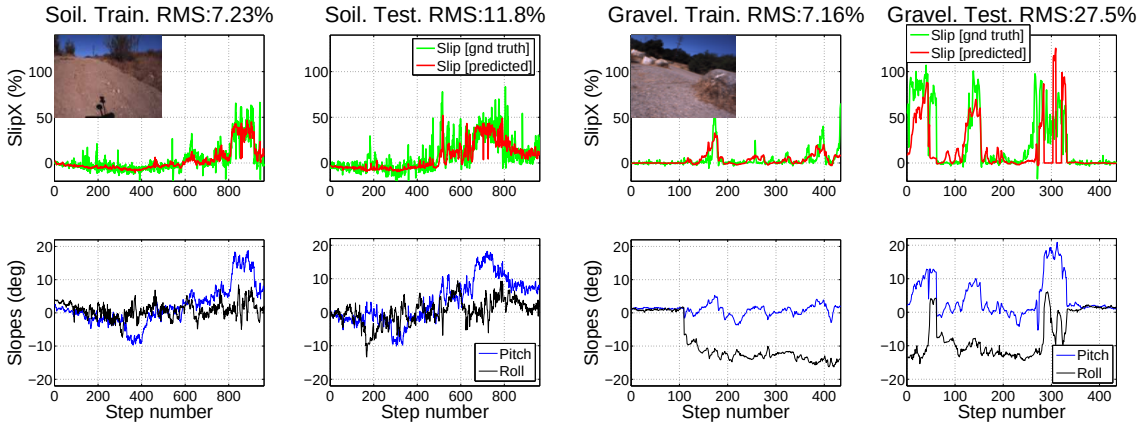

Figure 5: Learning slip as a function of slopes: soil (left), gravel (right).

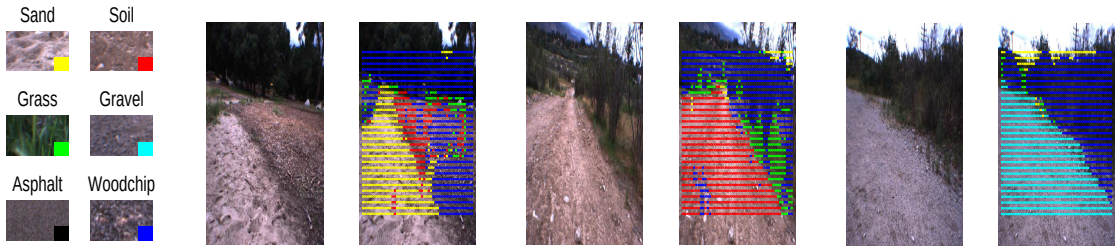

Figure 6: Example terrain classification results: sand, soil, and gravel.

≤ 2). Learning proceeds by assigning receptive fields to cover populated regions of the input domain and then fitting a linear regression (i.e., estimating factor loadings b_i^c and directions $\mathbf{p}_i^c$) locally in each receptive field. This fit weights all training points with their corresponding distance to the receptive field center $\mathbf{x}_c$, thus giving more influence to the nearest points (Angelova, 2006, has details). The receptive field size λ is selected using a validation set, and varies dependent on the dataset.

We compute slopes within a 2D map cell representation. Cells are $0.2\,\mathrm{m}\times 0.2\,\mathrm{m}$ and the map looks 15m ahead. The minimum-square-error plane fit at cell s is computed using the mean elevations of cells in the 6×6-cell neighborhood of s. The slope is then decomposed into a longitudinal (along the direction of motion) and a perpendicular lateral component, corresponding respectively to the pitch and roll of the vehicle. VO is used for localization and the vehicle's attitude (received from the IMU) gives an initial gravity-leveled frame to retrieve correct longitudinal and lateral slope angles from the terrain (Angelova, 2006).

Figure 5 shows learning and prediction of longitudinal slip as a function of slopes for soil and gravel terrains. Training is performed on the first portion of the traverse, and testing on a later, nonoverlapping, portion. The average root mean squared (RMS) error is given atop each plot. Slip is normalized by the average velocity per step to get the results in percent. It is apparent that the right qualitative relationship between slope and slip has been captured. Note that there are (roll,pitch) angle combinations in the gravel test data which were not seen during training, which requires good generalization. The results are very promising given the noise level and the limitations of the training data. Figure 6 shows preliminary terrain classification results using a texton-based approach (Varma, 2005) in the image plane. Despite some classification errors, the method is successful in discriminating

visually similar terrains at close range which serves the purposes of slip prediction. For now, the system is working offline, but we are exploring methods to speed up the terrain classification algorithm and integrate it into the navigation system.

6. Discussion

The Lf3D and LfP results show that there is a high degree of usable correlation between the appearance of obstacles and their 3D geometry, which validates the concept of knowledge transfer between sensing regimes. In future work, we believe there will be good return from augmenting the feature set to provide more error-free knowledge transfer. Other areas for future work include blame attribution, confidence assessment, joint estimation of traversability across all regimes, and strategic navigation (choosing to push an obstacle across a regime boundary to gain appearance or geometric information about it).

Acknowledgments

The research described here was carried out by the Jet Propulsion Laboratory, California Institute of Technology with funding from the DARPA LAGR and NASA MTP programs. We thank Nathan Koenig for data collection and the rest of the JPL LAGR team.

References

Angelova, A., Matthies, L., Helmick, D., Sibley, G., Perona, P., Learning to predict slip for ground robots, *submitted to ICRA*, 2006

Dickmanns, E., Mysliwetz., Recursive 3-D road and relative ego state recognition, *IEEE Trans. PAMI*, Vol. 14, No. 2, 1992

Helmick, D., Cheng, Y., Clouse, D., Matthies, L., Path following using visual odometry for a Mars rover in high-slip environments, *IEEE Aerospace Conf.*, Big Sky, MO, March 2004
using neural networks, *International Journal of Neural Systems*, vol. 8 (1), 1997

Lacaze, A., Murphy, K., DelGiorno, M., Autonomous mobility for the Demo III experimental unmanned vehicles, *AUVSI Conf. on Unmanned Vehicles*, July 2002

Lindemann, R., Voorhees., C., Mars exploration rover mobility assembly design, test and performance, *IEEE International Conference on Systems, Man and Cybernetics*, 2005

McLachlan, G., Peel, D., *Finite Mixture Models*, John Wiley & Sons, 2000

Maimone, M., Johnson, A., Cheng, Y., Willson, R., Matthies, L., Autonomous navigation results from the Mars Exploration Rover mission, *Int'l Symp. Experimental Robotics*, Singapore, June 2004

Schölkopf, C., Mika, S., Burges, C., Knirsch, Ph., Müller, K., Rätsch, G., Smola, A., Input space versus feature space in Kernel-based methods, *IEEE Trans. Neural Networks*, vol.10, 5, 2002

Varma, M., Zisserman, A., A statistical approach to texture classification from single images, *Int. Journal of Computer Vision*, Vol. 62, 2005

Vijayakumar, S., D'Souza, A., Schaal, S., Incremental online learning in high dimensions, *Neural Comp.*, 2005

Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S., Constrained K-means clustering with background knowledge, *Proc. ICML*, 2001

Wellington, C., Courville, A., Stentz, A., Interacting Markov Random Fields for simultaneous terrain modeling and obstacle detection, *Robotics Science and Systems*, 2005