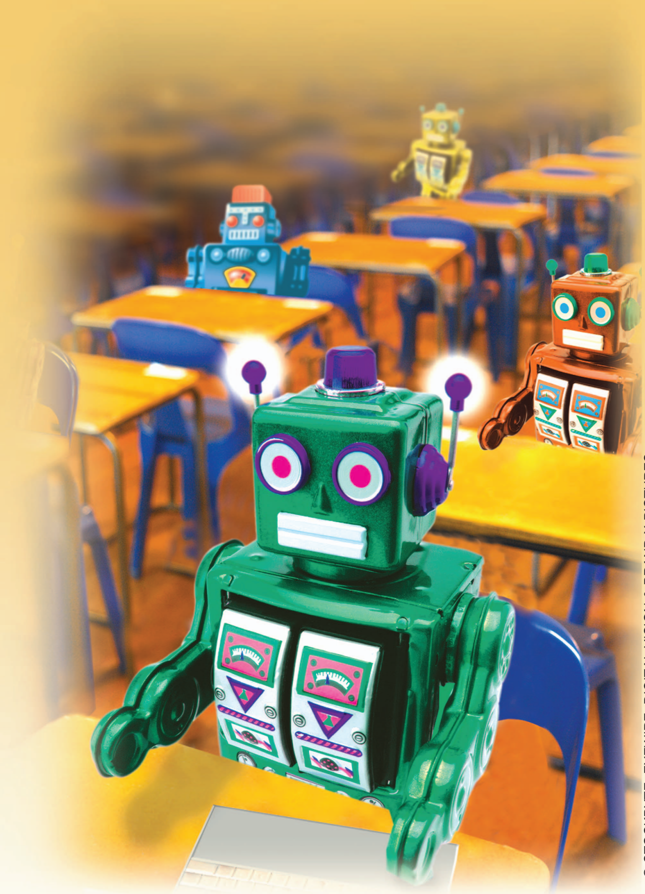
Learning for Autonomous Navigation

Advances in Machine Learning for Rough Terrain Mobility



© STOCKBYTE, EYEWIRE, DIGITAL VISION & BRAND X PICTURES

Robot Learning

Autonomous navigation by a mobile robot through natural, unstructured terrain is one of the premier challenges in field robotics. Tremendous advances in autonomous navigation have been made recently in field robotics [1]–[3]. Machine learning has played an increasingly important role in these advances. The Defense Advanced Research Projects Agency (DARPA) UGCV-Perceptor Integration (UPI) program was conceived to take a fresh approach to all aspects of autonomous outdoor mobile robot design, from vehicle design to the design of perception and control systems with the goal of achieving a leap in performance to enable the next generation of robotic applications in commercial, industrial, and military applications. The essential problem addressed by the UPI program is to enable safe autonomous traverse of a robot from Point A to Point B in the least time possible given a series of waypoints in complex, unstructured terrain separated by 0.2–2 km. To accomplish this goal, machine learning techniques were heavily used to provide robust and adaptive performance, while simultaneously reducing the required development and deployment time. This article describes the autonomous system, Crusher, developed for the UPI program and the learning approaches that aided in its successful performance.

Development on the UPI program was driven by extensive field testing on unrehearsed terrain to meet key metrics in off-road speed and required human interventions. These tests were conducted in extremely challenging terrains, including everything from temperate forests to high deserts, and required negotiating steep slopes, large boulders, deep washes, and dense vegetation. In the complex terrain considered, it was recognized that some human intervention was inevitable, although performance metrics strictly limited that availability. A more in-depth discussion of the program as a whole as well as the autonomy components of the program can be found in [4] and [5].

Digital Object Identifier 10.1109/MRA.2010.936946

Autonomous operation under these conditions stands in contrast to the aims of that of the DARPA Grand Challenge [6]. Although the grand challenge significantly advanced the state of the art in reliable, high-speed autonomy, it did not address the challenges posed by truly unstructured environments. Such challenges include distinguishing between traversable obstacles, such as vegetation, and nontraversable positive obstacles, such as rocks and trees, while accounting for potential negative obstacles, such as ditches and cliffs. In addition, grand challenge vehicles were significantly constrained in their areas of operation, simplifying the problem of a long-range planning. The UPI program is more similar to the Robotics Collaborative Technology Alliance (RCTA) program [7], which has also incorporated learning-based terrain classification in challenging off-road environments. In the tradition of the DARPA PerceptOR [8] and DoD DEMO III [9] programs, the UPI program focuses on navigating in unconstrained areas of operation and with little to no-time for mission preplanning.

BY JAMES ANDREW BAGNELL, DAVID BRADLEY, DAVID SILVER, BORIS SOFMAN, AND ANTHONY STENTZ

 1070-9932/10/$26.00©2010 IEEE

UPI demanded new approaches to meet speed and intervention goals to an order of magnitude greater than anything achieved by autonomous robotics in such a complex terrain. The diversity of obstacles, vegetation, and terrain meant that the core of the program was extracting maximal benefit from a limited number of expert man-hours during the system design process. The traditional algorithmic tools of field robotics [8], [9] require time-consuming parameter tuning difficult even for experts. Instead, the UPI team combined proven engineering developed within the field, including model-predictive control [1], [10], three-dimensional (3-D) laser detection and ranging (LADAR) analysis [1], [11], and fast motion replanning [12] with data-driven, machine-learning approaches. This synthesis enabled the program-by-demonstration of many challenging aspects of the problem, including the automatic interpretation of ambiguous sensor data and adapting the system's performance automatically to novel, previously unseen terrain.

Machine learning techniques were heavily used to provide robust and adaptive performance, while simultaneously reducing the required development and deployment time.

In this article, we lay out the elements of this approach to robotics and the role played by machine learning in improving the performance by enabling nonexperts to modify the system and leading to dramatic reductions in engineering time and effort. The learning approaches that enabled this performance draw broadly from machine learning, including classification and regression, self-supervised learning, imitation learning, online (no-regret) learning, and iterative learning control. Learning techniques are nearly ubiquitous in our approach, touching the full array of subsystems on the Crusher platform, including near-range LADAR analysis, far-range monocular and stereo image interpretation, overhead imagery and height map analysis, local and global motion planning, motor control, and vehicle positioning.

The adaptive techniques presented here, when combined with the nonadaptive algorithmic approaches and inherent vehicle mobility, have arguably achieved the state of the art in outdoor, complex terrain autonomy. We discuss some of the extensive experimental testing and validation of these techniques. We conclude with a discussion of the key lessons learned on the integration of learning into autonomous systems and the many challenges that remain in achieving high-performance autonomy while maximizing the value of collected data and minimizing engineering time and effort.

Crusher Autonomous System

The Crusher unmanned ground vehicle (UGV) of the UPI program (shown in Figure 1) was designed to navigate autonomously through complex, outdoor environments using a combination of LADAR and camera sensors onboard the vehicle as well as satellite or other prior overhead data when available [5]. Figure 2 outlines the high-level data flow within Crusher's autonomy systems.

(a) (b)

Figure 1. Crusher robots used in the UPI program. The natural terrain shown here is the representative of the domains they operate within.

Figure 2 hides many details of the system, including positioning, expiration of data, and interprocess communications and timing but retains the core elements of the system important for a discussion of the role of machine learning. The various modules and heuristics used in the autonomy system contain a large number of settings and parameters. Engineering these parameters is difficult and time consuming, and, hence, we choose to learn the parameters of the system from data or human demonstration wherever possible. Space limits our ability to discuss all learning modules in the system, and so we focus on a few critical to overall performance.

Crusher's onboard perception system can be split into two. Near-range perception uses a combination of onboard camera and actuated LADAR sensors to produce a continuously updating high-resolution model out to a range of 20 m. These models consist of various geometric or appearance-based features assigned to individual 3-D voxels. Supervised learning (see the Terrain Perception—Near-Range Terrain Classification) is used to classify 3-D voxels into ground, obstacle, and vegetation classes from a set of labeled terrain examples. A far-range perception system produces a low-resolution model out to a range of at least 60 m. Limited sensor resolution and sparse LADAR returns make properly interpreting far-range data difficult. The "Bootstrapped Far-Range Terrain Perception" section describes a powerful self-supervised learning technique that was developed to bootstrap far-range perception using the output of the near-range system. In addition, both near and far-range modules maintain an estimate of the local terrain supporting surface (see the "Terrain Surface Estimation" section).

A useful and common abstraction for navigation is to represent the world as a two-dimensional (2-D) horizontal grid around

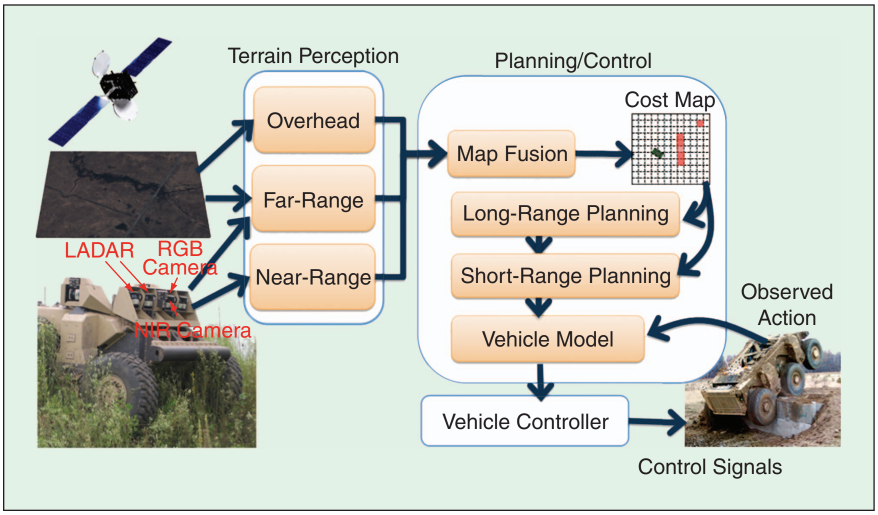


Figure 2. *Crusher's autonomous navigation system combines data from camera and LADAR sensors onboard the robot with previously collected satellite imagery to find safe and efficient paths through complex off-road environments.*

the robot (see Figure 3) [1]. Navigation through the environment is achieved by first producing a traversal cost for each cell in the 2-D grid and then planning a minimum cost path (green lines) through the grid to the goal (red X). These costs are generated as a function of the features associated with each 3-D voxel within the appropriate 2-D column. In practice, it is remarkably difficult to hand engineer an effective mapping from perceptual features to traversal costs that produces plans and vehicle behaviors that correspond to our expectations. Identifying such cost functions from limited and ambiguous perception data is effectively a black art for mobile robotics. The UPI program developed novel techniques for imitation learning that allow a cost function to be learned directly from examples of preferred driving behavior (see the "Coupling Perception and Planning" section). These costs, along with costs generated from available prior overhead data, are then merged and passed on the Crusher's planning system. The processing of overhead data mirrors that of Crusher's onboard perception system and is described in the "Overhead Terrain Perception" section.

The planning system uses a hybrid global/local approach [1] that continuously replans to account for new information. A long-range global path to the goal is computed by finding a minimum cost path through the map using the Field D* algorithm [12]. The global plan is refined by the local planner in a kinematically constrained search over a small area around the robot where it is able to plan sophisticated actions and maneuvers that incorporate both mobility and dynamics. Such maneuvers require an accurate model of the vehicle motion produced by a given set of control signals. Because Crusher is a six-wheeled skid-steer vehicle, its motion is difficult to model; self-supervised learning was applied (see the "Vehicle Modeling" section) to learn response to control inputs and improve following of commanded paths.

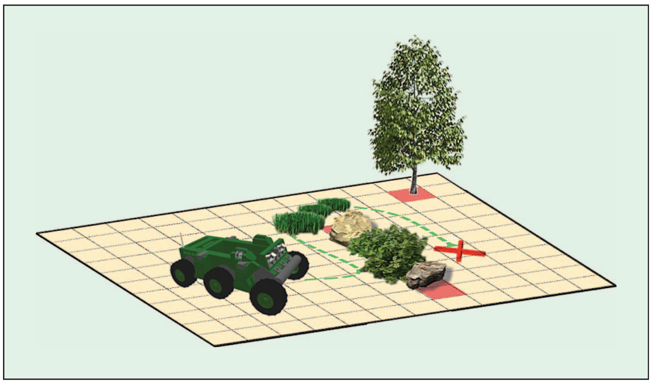


Figure 3. *Abstraction of the mobile robot navigation problem.*

Terrain Perception

Crusher employs learning-based perception modules for spatial regions close to the vehicle, farther from the vehicle, and areas that are outside of the range of the vehicles onboard sensors but for which there is available satellite or aerial data.

Near-Range Terrain Classification

A set of LADAR scanners collects 3-D range data, whereas cross-calibrated color and near infrared (NIR) cameras determine appearance data from the vehicle's surroundings. Within 20 m of Crusher, these sensors provide abundant data for terrain analysis. The 3-D range data processing approach builds on [7] by incorporating multispectral color features and image texture features, leading to improved accuracy in many terrains. The near-range perception uses 3-D points from the LADAR that have been projected into the camera images and tagged with local properties of the image, such as color and image texture. This system discretizes the space surrounding the robot into a 3-D grid of voxels and computes features of the tagged points over each voxel. Examples of some of these features are shown in Figure 4, including the averages of the point tags, eigenvalues of the local point cloud scatter matrix [11], the surface normal (third eigenvector of the scatter matrix), and the probability and strength of laser pulse reflection from a voxel. A particularly useful feature for vegetation detection is the average normalized difference vegetation index (NDVI) value of a voxel, which is computed from the red and NIR color tags of each point [13]. NDVI was developed in the satellite imaging community to exploit the unique spectral properties of small water-filled plants cells that contain chlorophyll. Figure 5 highlights bright green vegetation detected by NDVI color and NIR images in our system. The engineered features are classified with a multiclass logistic regression (MaxEnt) classifier into ground, vegetation, or obstacle classes using a labeled data set of more than 4 million voxels collected from a variety of test environments. Collection of this large data was made feasible by the development of an intuitive image-based voxel labeling tool.

Figure 6 visualizes the output of the near-range classification system on a forest scene in Colorado. These classifications are used in the ground height estimation algorithm described later. The terrain classification of each voxel is combined with the ground height estimate for each 2-D cell to produce a traversal cost estimate for each cell, using the imitation learning techniques described in the "Coupling Perception and Planning" section. It should be noted that these classes were chosen to simplify labeling, and there is no evidence that they are the optimal terrain classes for the overall system. Rather, including this classification phase provides a small set of useful input features to the cost producer and simplifies the task of cost production. In some terrains, the wide variety of examples included in the labeled training set hurts classification performance. For example, the NDVI feature improves performance in terrains (and seasons) where chlorophyll-rich vegetation is abundant, but dependence on NDVI hurts classification performance in areas with dead or desert-type vegetation. To fix this problem, a method was developed in [14] to automatically adapt the terrain classifier to the environment the robot is currently operating in. Unlabeled voxels that the robot is trying to classify are used to give more weight to similar examples from the labeled data set, and the classifier is retrained on a reweighted data set.

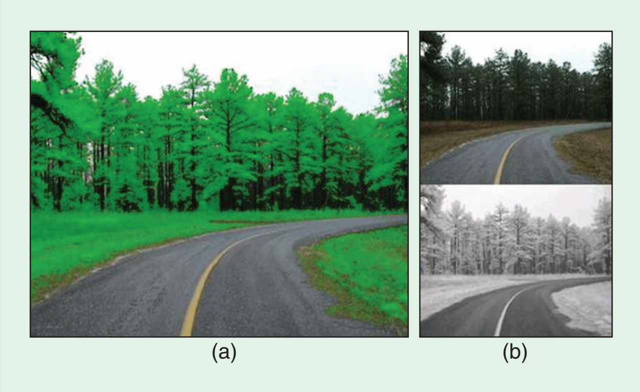


Figure 5. *(a) Vegetation detected with the NDVI feature is highlighted in bright green, and (b) the original color and NIR images. Note how bright vegetation appears in NIR.*

Terrain Surface Estimation

A central element of successful rough terrain navigation systems is the estimation of a terrain-supporting surface. Identifying the surface enables the autonomy system to detect hazards because of its shape, high grades, ditches, holes, or high-centering hazards. Further, the ability to separate positive obstacles from the surface allows both for an accurate estimation of their height as well as proper interpretation of terrain classifications. Both of these abilities in turn are crucial for robust traversability estimation. Identifying the ground surface is complicated by vegetation, occlusion, and sparsity in LADAR point density (see Figure 7). Identifying this surface is naturally viewed as an estimation problem rather than one of machine learning.

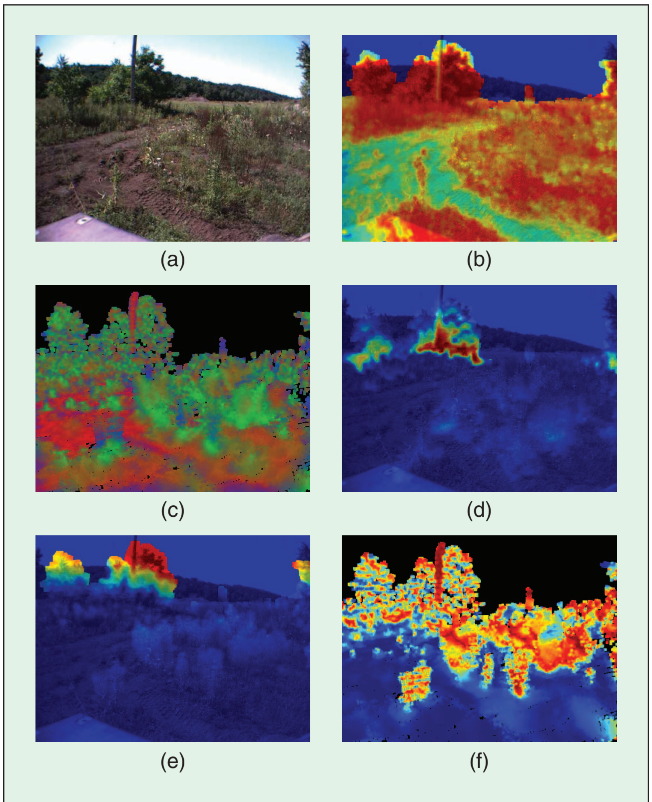


Figure 4. *The camera and laser data captured by the sensors onboard the mobile robot are used to compute a set of engineered features. Here, a few of the features are shown projected into the image from one of the onboard cameras.*

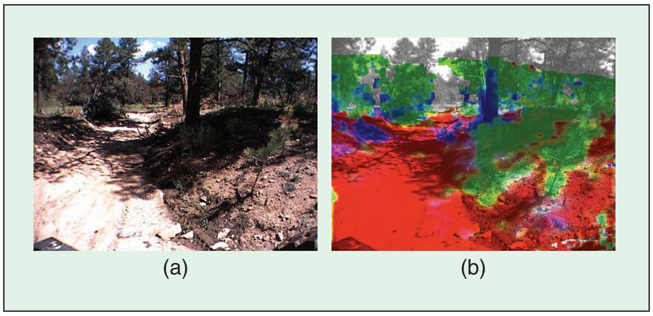


Figure 6. *Classified voxels in a forest scene.*

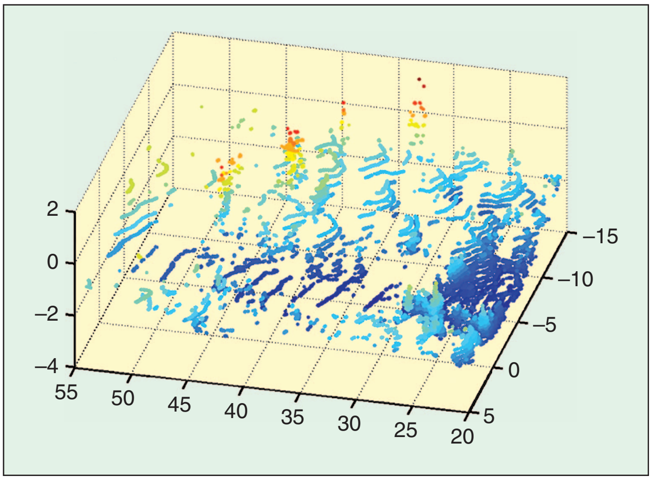


Figure 7. *LADAR scans of rough terrain often have complex structure and highly variable resolution that decays rapidly with distance.*

Autonomous navigation by a mobile robot through a natural, unstructured terrain is one of the premier challenges in field robotics.

Indeed, although our approach builds on past work in this area [1], [15], we determined that a method from the online learning community provided a remarkably fast and effective algorithm. Our approach to estimation leverages the notion, developed in [1], that each LADAR point from the sensor is truly a ray from the sensor to a point in the world and that this ray defines a space-carving constraint on the ground supporting surface—at any 2-D discretized grid location in the world the surface must lay beneath that ray. This approach makes sparse LADAR data dramatically more effective for upper bounding the ground at each location than if only the reflection points were considered [16]. Additionally, it maintains knowledge of areas that have not yet been directly observed and may contain negative obstacles. However, in practice, this approach is insufficient in areas where data are especially sparse and particularly where there is vegetation.

In [15], the authors developed a Markov random field (MRF)-based technique over 3-D voxels that estimates terrain-supporting surface by leveraging contextual information from neighboring cells using a computationally demanding Markov-chain Monte-Carlo procedure. On the UPI program, these ideas are combined in a 2-D MRF formed with a random variable representing height at each location. This random field uses both LADAR points and terrain classification to establish potentials between the random variables, e.g., if terrain classification detects bare Earth, the random field strongly enforced the ground height at that location. More complete details of the approach, including a detailed description of the random field and its interaction with terrain classification, can be found in [4].

The connection with machine learning is perhaps a bit surprising: the fast, anytime estimates of the ground plane necessary for autonomous navigation were made possible using the online projected gradient convex optimization described in [17] to iteratively infer the most probable height of the supporting surface at each 2-D location. At every iteration, the optimization takes a gradient step by moving each cell's ground estimate along the vertical axis toward a position suggested by the neighbors of that cell, effectively anisotropically smoothing the ground heights. Each cell is then projected onto the constraint implied by the LADAR rays; if the estimate goes above the lowest point along any ray that intersects a cell, the estimate is clamped back onto that value. Related work has considered more sophisticated discretization-free estimators of terrain [18] using Gaussian processes and has recently included the space-carving bound constraint as well [16].

Overhead Terrain Perception

When the length of an autonomous traverse greatly exceeds the range of a robot's onboard sensors, prior knowledge of the environment can improve autonomous safety and efficiency by guiding the robot over more easily navigable terrain [19]. The overhead vantage point is an attractive source of such prior knowledge. Satellite imagery can now be commercially purchased at high resolution and can be augmented with digital elevation maps or aerial LADAR scans. Although the wide variety of available data sources increases the available prior knowledge, the heterogenous nature of the data can make processing and interpretation difficult.

To reduce the degree of engineering and human effort that must go into processing overhead data sets from multiple sources, learning techniques were used extensively. At a high level, the processing of overhead data mirrors the approach used for near-range perception (see the "Near-Range Terrain Classification" section), and a typical result is given in Figure 8. First, the available raw data sets are processed into a set of feature maps (e.g., color, texture, NDVI for imagery and slope, canopy cover, and other geometric features for LADAR). Along with these features, human labeling of terrain is used as input into a supervised semantic classification (specifically a convolutional neural network) [19]. Appropriate classes are defined by the labeler for each environment. This approach has the advantage that it is easily adaptable to different environments without additional feature engineering; it requires only a brief human labeling effort. Additionally, when aerial LADAR scans are available, a terrain surface estimation procedure is performed based on the same basic algorithm as

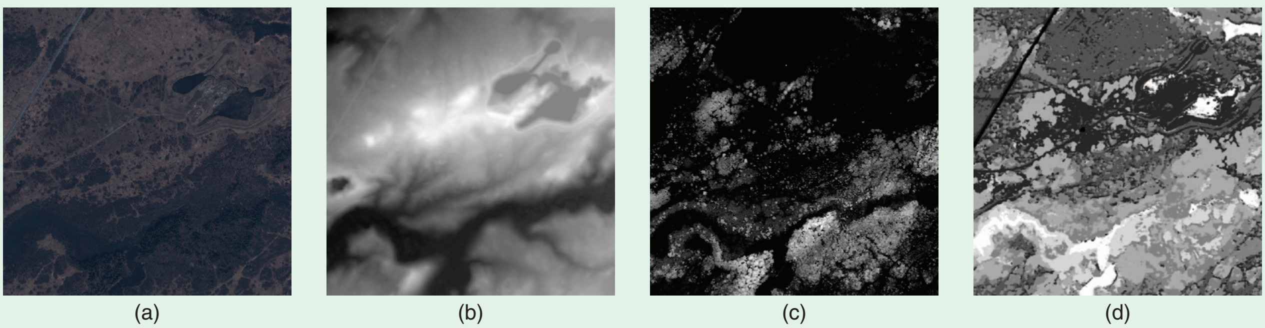


Figure 8. Examples of overhead data more than a 1-km^2 area. (a)–(d) Satellite imagery, extracted ground surface height, tree canopy, and semantic classification. (Quickbird satellite imagery courtesy of Digital Globe, Inc.)

discussed in the "Terrain Surface Estimation" section. As with near-range perception, terrain classification is a useful but not necessary input to this procedure. A notable difference is that the sensor origin is often not provided with commercial aerial LADAR. However, generally the low angle of incidence means that the location of each point in space is a sufficient constraint. After processing, refined overhead data (visual and geometric features plus semantic classifications) are converted into traversal costs to be fused with onboard data. The learning procedure for this conversion is described in the "Coupling Perception and Planning" section. The procedure for cost fusion almost universally favors onboard data as overhead data can sometimes be months or even years old.

Bootstrapped Far-Range Terrain Perception

Interpreting terrain data far from an onboard sensor helps a robot to identify hazards as quickly as possible, leading to more efficient trajectories and reduced risk. Unfortunately, reliable and effective automated interpretation of far-range data in diverse terrain, environmental conditions, and sensor varieties is challenging. As a result, a system that needs to perform reliably across many domains without reengineering by hand or supervised training must rely on the subset of available information that generalizes well across many domains. Because of the accuracy, consistency, and density required for such features, this often limits a system to use only onboard sensor data within a short proximity to the vehicle.

The UPI program employed a bootstrapping system to be able to leverage potentially useful data sources. In [20], we introduced a Bayesian probabilistic framework for learning and inferring between heterogeneous data sources that vary in density, accuracy, and scope of influence. Here, we present a specific instance of this framework that takes advantage of accurate and robust near-range perception system to extend the perception range of the robot by interpreting any combination of overhead and far-range sensor data with online, self-supervised learning. When dealing with overhead data, our algorithm will be referred to as map online learning (MOLL) and when dealing with far-range sensor data, our algorithm will be referred to as far-range online learning (FROLL). This concept of near-to-far learning was pioneered by Wellington and Stentz [21] and independently advanced for the DARPA grand challenge by Thrun et al. [2], as well as later systems developed by several teams from the DARPA Learning Applied to Ground Robots (LAGR) program [22]–[24].

A sample scenario is visualized in Figure 9. Our goal is to produce traversal cost estimates for the environment we are operating in. The near-range perception system generates traversal cost estimates in proximity to the robot (blue region) that are used to learn the mapping from difficult to interpret locale-specific features to traversal cost. This learned model can then be applied elsewhere to produce traversal cost predictions where no near-range perception estimates are available. Throughout the robot's traverse, it observes a sequence of features generated from overhead data and far-range sensor data (which we call our locale-specific features x). Initially, it does not know how to interpret these features. The robot also observes a sequence of traversal cost estimates $\tilde{c}$ generated from the near-range perception system for a small subset of the environment that we assume are the true traversal costs c with some amount of uncertainty. These estimates are gathered online and are only available for a subset of the locations in the world, whereas locale-specific features are available for many more locations.

By remembering the locale-specific features x seen during traversal, the subset of these locations for which the near-range perception system has generated estimates $\tilde{c}$ are used to learn a mapping from x to c. As the robot traverses through the environment, it refines this model online.

We choose a simple model for traversal costs, given locale-specific features that model the distribution of the traversal costs as a Gaussian with constant variance and a mean produced by a linear function of the features. This allows us to develop the inference for this linear-Gaussian model in an online fashion by revising the posterior distribution of the model in light of a new Gaussian likelihood that takes into account noise from all features. Empirically, Gaussian distributions with a generous selection of variance on features and cost estimates learned within minutes but remained robust to the presence of outliers. This approach offers many advantages including reversible learning (useful when multiple traversal cost estimates of different quality become available for a single location), feature selection, and confidence-rated prediction.

Although the system's near-range perception system was effective for most situations, the extended perception range resulted in significantly faster and safer navigation (see the "Experiments" section). The successful use of this system removed any necessity for engineering an independent far-range perception system. While we did not operate MOLL and FROLL concurrently, nothing prevents a system from doing so if adequate computational resources were available.

Vehicle Planning and Control

Coupling Perception and Planning

The coupling of Crusher's perception and planning systems is achieved through the notion of traversal or mobility costs. Scalar cost values assigned to patches of terrain concretely describe the relative preference between different areas. As the planning system seeks to minimize accrued cost, the cost

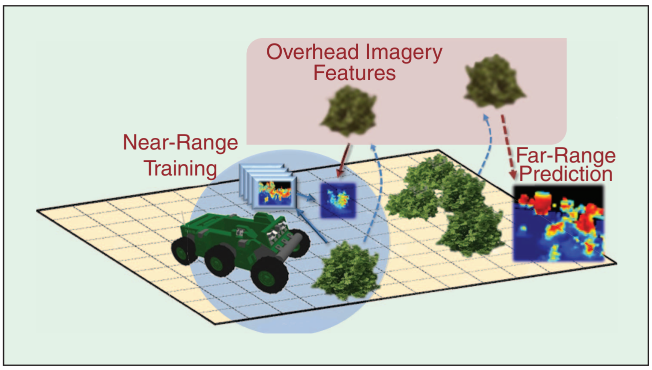


Figure 9. *Abstraction of the online bootstrapped far-range perception system.*

The UPI program focuses on navigating in unconstrained areas of operation and with little to no-time for mission preplanning.

function implicitly defines Crusher's driving style, e.g., how aggressive it will be. A well-designed cost function is essential to robust autonomous performance.

Despite the importance of robust cost functions, the task of developing them has previously received little effort. As a result, cost function design is something of a black art with functions manually engineered on a case-by-case basis. Although such cost functions can and have produced high-performance systems, they often require an enormous investment of human resources to continually design, tune, and validate; further, it is often unclear if an optimal coupling has been achieved. Such effort must be reinvested with every modification to the perception or planning subsystems.

The design of cost functions for both near-range perception and overhead data originally involved manually engineering, whereas later efforts focused on more formal approaches. Specifically, a learning-from-demonstration framework was developed that allowed cost functions to be learned from example vehicle behavior produced from human domain experts (i.e., the expert must be sufficiently familiar with the environment and the vehicle capabilities to produce good example behavior but need not to be an autonomy expert). This approach is based on the concept of inverse optimal control in which the metric for an optimal controller is determined from examples of optimal trajectories. Our learning-from-demonstration approach uses an algorithm we call *learning to search* (LEARCH) [25], which we believe to be the first application of inverse optimal control to any physical system. The input to the algorithm is a set of example paths representing the expert's preferred path between a specific start and goal location. LEARCH searches for a cost function that meets the constraint that every example path should be the optimal path between the start and end location. As there is usually no cost function that will satisfy the constraints of a large set of examples, this search takes the form of a soft-constrained optimization, trying to minimize the difference between the cost of each example path and the corresponding planned path.

This objective function can be minimized by computing the gradient with respect to the costs of individual patches of terrain. As the cost of a path is simply the cost of the terrain it traverses, the local gradient is simply made up of the locations along each path. Hence, the objective can be minimized by raising the cost of locations along the current plan and lowering the cost of locations along the expert example. By using a gradient boosting procedure [26], a cost function can be constructed using standard classification or regression techniques. Subsequent improvements to this procedure have significantly increased robustness to imperfect or noisy demonstration. Figure 10 provides a visual example of this procedure in action with overhead data as an input. By accounting for the dynamic and partially unknown nature of onboard perceptual data, a similar procedure can learn a cost function for the near-range perception system from expert teleoperation of Crusher [27].

Vehicle Modeling

An important component of the local planning system is predicting the motion that the vehicle will experience for a given set of motor commands. As Crusher is a skid-steered vehicle with complex interactions with slippery, unknown terrain, it proves difficult to construct a traditional high-fidelity, physics-based vehicle model. When going up hills, the vehicle's wheels often slip, reducing the along-track velocity that the vehicle is trying to achieve in its intended direction of travel and often introducing cross-track velocities perpendicular to the vehicle's heading. Turning on slopes is particularly prone to slip, and failure to predict the vehicle's motion correctly can result in collision with obstacles. To account for wheel slip, the UPI system predicts the vehicle's motion out to a 3-s time horizon using a 24-layer deep neural network [28]. This neural network predicts the along-track and cross-track velocities

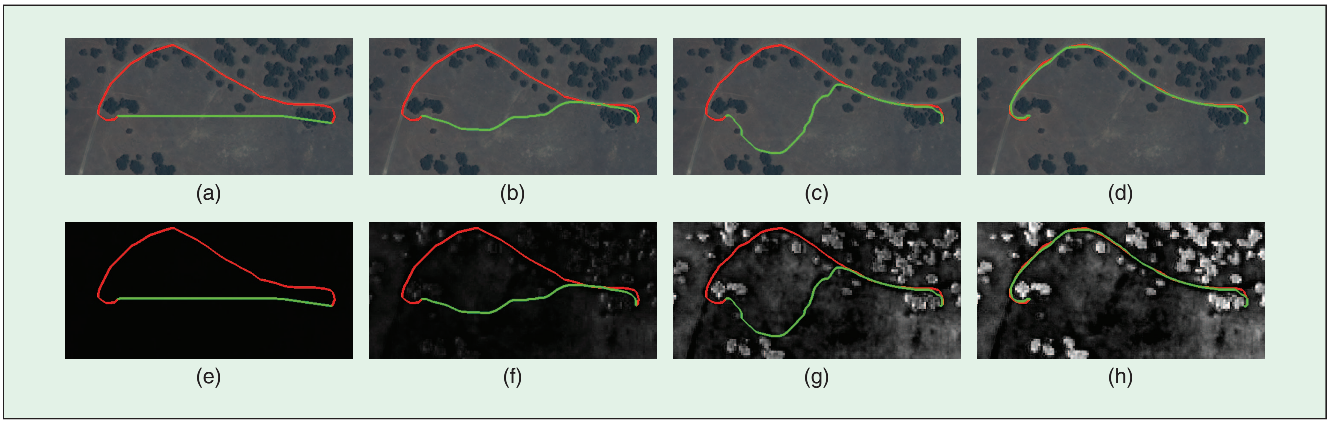


***Figure 10.** An example of the LEARCH algorithm learning to interpret [(a)–(d)] satellite imagery as [(e)–(h)] costs. The brighter pixels indicate higher cost (e)–(h). As the cost function evolves, the current plan (green) recreates more and more of the example plan (red).*

that the vehicle will experience as well as the vehicle yaw. The velocity estimates are integrated to produce a predicted path for the vehicle. The current pose (pitch and roll) and velocities provided by the vehicle's inertial measurement unit (IMU) as well as the commands that will be provided to the motors in the 3-s window are used as input. The ground surface estimate is used to predict the pitch and roll that the vehicle will experience in future locations.

The predictor is trained from logs of the commands sent to the vehicle motors and the actual motion of the vehicle from the inertial navigation system. Such deep neural networks are considered difficult to train because prediction errors often cascade into large output errors and because of the difficult credit assignment required to determine how the many parameters in initial layers of the network influence the final output. This network was successfully learned by adding local information to each hidden layer of the network based on the difference between the predicted and actual velocities of the vehicle at each 0.25-s interval, leading to a 59% reduction in error over a previous conventional physics-based vehicle model.

Experiments

Numerous, large, unrehearsed, field tests were conducted at various sites across the continental United States with no prior exposure of either the robot or the autonomy team to the test site, and only minimal time was provided for any additional training of the system. Despite these constraints, the Crusher and Spinner vehicles autonomously traversed more than 1,000 km across extreme terrain, averaging human intervention once every 20 km. Because the focus of testing was overall system performance, it was not always possible to quantify the effect of various learning techniques; however, for a subset of the implemented learning techniques, it was possible to perform specific comparison experiments.

The bootstrapped perception system approach was tested with both overhead data and far-range sensor data to measure its impact on navigation performance compared with a system using only the near-range perception system. The test environments contained a large variety of vegetation, various-sized dirt roads (often leading through narrow passages in dense vegetation), hills, and ditches. The vehicle traversed several courses defined by a series of waypoints using only its onboard perception system for navigation and then traversed the same courses with the help of our bootstrapping approach. First, MOLL was applied using 40-cm resolution overhead imagery and elevation data to supplement the onboard perception system with traversal cost updates computed within a 75-m radius once every 2 s. Next, FROLL was used to interpret and make predictions from far-range sensor data every 1 s. The algorithm was initialized for each course with no prior training to simulate its introduction into a novel domain.

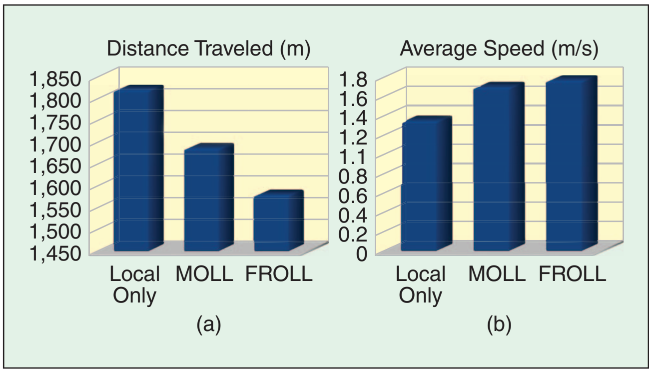


Figure 11. *Over many comparison runs, the bootstrapped far-range perception system was shown to outperform the baseline system using only near-range perception in both distance of travel required and average speed.*

Quantitative results are shown in Figure 11, and a sample scenario where such an approach is especially valuable is shown in Figure 12. In Figure 12(a), the course started at the top right and ended at the left. Predictions of terrain traversal costs for the environment by our bootstrapping perception system at the times the vehicle chose to avoid the large cul-de-sac in front of it are shown for MOLL in Figure 12(b) and for FROLL in Figure 12(c). Traversal costs are color-scaled for improved visibility. Blue and red correspond to lowest and highest traversal cost areas, respectively, with roads appearing in black. In Figure 12(b), MOLL helped the vehicle avoid the area of dense trees by executing a path that is 43% shorter in

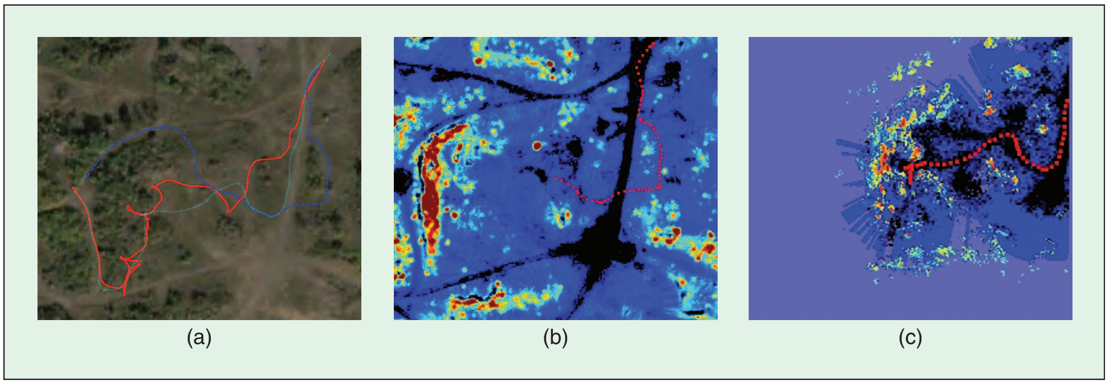


Figure 12. *(a) Comparison of paths executed for shown situations when using only onboard perception (in solid red) and with MOLL (in dashed blue) and FROLL (in dotted cyan) are shown. The predicted traversal costs for the environment are shown for MOLL and FROLL in (b) and (c), respectively.*

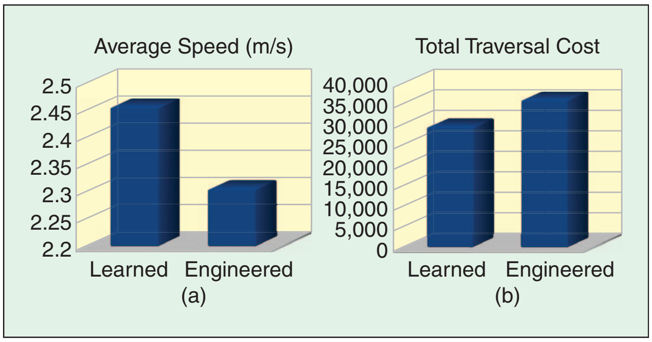


Figure 13. *Comparing performance over an approximately 12-km course, Crusher drover faster and went over safer terrain (as scored independently by onboard perception) when using a learned interpretation of prior data.*

73% less time, whereas in Figure 12(c), FROLL helped the vehicle identify a lack of opening and avoid traversing deep into dense and hazardous vegetation.

In general, we found that both techniques not only improved the quality of the paths chosen by the vehicle but also allowed higher speed navigation by increasing the time the vehicle had to react to upcoming obstacles and identifying safer terrain such as roads. By learning online to leverage potentially powerful, but difficult to generalize, features from overhead and far-range sensor data, our system is able to use all potentially useful data sources and to adapt to changing conditions without the necessity of human-supervised retraining or engineering. Additional results and applications are presented in [20].

At the beginning of the UPI program, manually engineered and tuned cost functions were used for near-range perception and overhead data. Creating cost functions for overhead data was especially time consuming; as each test site consisted of data sets of varying source and quality, a new cost function was generally required for each site. Creating and validating these cost functions usually involved several days of an engineer's time. Midway through the program, this process was changed to make use of the learning from demonstration approach of the "Coupling Perception and Planning" section. In contrast, the learning approach would only require several hours of expert interaction to produce a training set and resulted in cost functions that produced more desirable long-range plans [29]. In addition, the online performance of the system improved. Crusher was shown to drive faster and through safer terrain when using a learned interpretation of overhead data than when using an engineered interpretation (see Figure 13) [30].

Throughout a majority of the UPI program, a manually engineered cost function was used for near-range perception. Although this resulted in a high-performance system, it came at a high cost, requiring hundreds of engineer-hours of development spread over a three-year period. Once an effort was undertaken to apply learning from demonstration, another significant time savings was realized. The final training set used to train the system consisted of less than a full day of examples collected piecemeal over terrains. Comparison experiments demonstrated that the learned interpretation of near-range perceptual data provided a level of safety equivalent to the engineered system while slightly increasing route efficiency (see Figure 14) [29].

Future of Machine Learning in Mobile Robotics

The UPI program yielded lessons for applying machine learning to autonomous robotic systems. Given the role that learning played throughout the Crusher system, we should have better designed the system to facilitate learning approaches, both at the architectural and component levels. For example, not exposing all relevant data signals in modules from the onset often prevented the fullest application of learning. There was often significant inertia preventing new learning applications, most often due to the high upfront cost of implementing necessary infrastructure. It was simply assumed that it would be easier to engineer solutions one time. Again and again, this proved insufficient as changes to one component or subsystem would cascade to other components. Although implementing a learning infrastructure resulted in a higher upfront cost, we found that it quickly paid dividends as machine learning allowed the entire system pipeline to be retrained using existing logged data to adapt to changes elsewhere.

Currently, intermediate modules in Crusher's autonomy, such as terrain classification, are not directly trained to improve the end performance of the system but rather have been trained to solve local tasks, i.e., classification of voxels as road, obstacle, or vegetation, which we believe will produce effective intermediate representations. It is unclear how to define those terrain classes for any particular vehicle or even the best number of terrain classes. Worse still, improved performance on an intermediate task does not always translate into improved navigational performance; the metrics on performance are too weakly coupled. This was observed several times during development, e.g., when a large improvement in

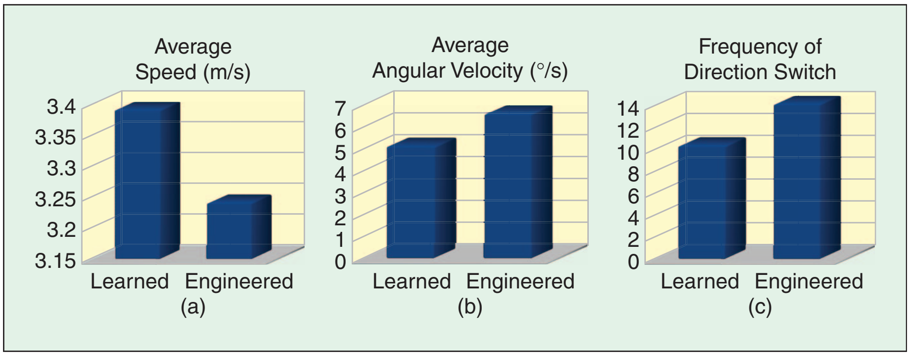


Figure 14. *Comparing performance over approximately 35 km of autonomous traverse, Crusher drove faster, turned softer, and changed direction less when using a learned cost function. In addition, there was no statistically significant difference in safety between the two systems [29].*

classification performance produced no improvement in overall autonomy performance. Therefore, we contend end-to-end learning approaches that jointly learn all modules of the system from sets of labeled and unlabeled data will best leverage expensive field tests, designer time, and computing hardware. Current work [14] is applying methods from the emerging field of deep learning [22], [31], [32] to learn parameters for intermediate modules that directly optimize the final output of the system, i.e., providing global coordination of the entire system, while benefiting from local training data. Ultimately, the goal of this learning approach is to use human experts, where they are most effective to provide an overall architecture for the solution, and to define the problem in intuitive but mathematically precise terms, and manage the detail work of tuning this architecture with automated machine learning algorithms. More generally, we contend there is important opportunity in developing a science of design for learnability: baking in the hooks to facilitate learning throughout an autonomy system.

Beyond this, we believe transitioning mobile robotic systems to real-world use requires an extraordinary degree of reliability with even a relatively rare rate of failure viewed as unacceptable. Current approaches develop and train robots in areas representative of those they are expected to operate in, only to discover new situations that are not in an experience base that can lead to unexpectedly poor performance or even complete failures. Since it is impossible to prepare for the unexpected, one must assume that such situations will arise during real-world operation. To mitigate this risk, a UGV must be able to identify situations that it is likely untrained to handle before it experiences a major failure.

This problem can be framed as novelty detection: identifying when perception system inputs differ dramatically from prior inputs. We have achieved initial results in exploring online algorithms that can compactly represent the past experiences of the robot to identify potentially dangerous situations such as those shown in Figure 15 [33]. With this ability, the system can either avoid novel locations to minimize risk or stop and enlist human help via supervisory control or teleoperation.

Finally, we concede that the complexity and unpredictability of the real-world forces robotic systems to adapt online. A key is to identify feedback that allows online training in as many ways as is feasible whether a part of the system serving as an expert to train another or a human operator serving as the expert at opportune times. We contend that using such techniques completely will enable robots to adapt to and improve their performance in diverse environments.

Acknowledgments

This work was made possible by DARPA under contract "UGCV-PerceptOR Integration" (MDA972-01-9-0005). We especially thank Dr. Lawrence Jackel and his team for their support. We gratefully acknowledge the entire UPI autonomy and vehicle teams for their tremendous efforts. We especially thank Cris Dima, Cliff Olmstead, Michael Bode, Tom Pilarski, and Carl Wellington for their invaluable efforts in the design and implementation of the described work.

Figure 15. *The chain-link fence identified here as novel has characteristics of sparse vegetation in many of its features and would therefore get assigned a dangerously low traversal cost.*

Keywords

Field robotics, machine learning.

References

[1] A. Kelly, A. Stentz, O. Amidi, M. Bode, D. Bradley, A. Diaz-Calderon, M. Happold, H. Herman, R. Mandelbaum, T. Pilarski, P. Rander, S. Thayer, N. Vallidis, and R. Warner, "Toward reliable off road autonomous vehicles operating in challenging environments," *Int. J. Robot. Res.*, vol. 25, no. 5–6, pp. 449–483, 2006.

[2] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley: The robot that won the DARPA grand challenge," *J. Field Robot.*, vol. 23, no. 9, pp. 661–692, June 2006.

[3] C. Urmson, J. Anhalt, J. A. Bagnell, C. Baker, R. Bittner, M. N. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. M. Howard, S. Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y.-W. Seo, S. Singh, J. Snider, A. Stentz, W. R. Whittaker, Z. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, B. Litkouhi, J. Nickolaou, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms, and D. Ferguson, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 425–466, 2008.

[4] A. Stentz, "Autonomous navigation for complex terrain," Carnegie Mellon Robot. Inst., Tech. Rep., submitted for publication.

[5] A. Stentz, J. Bares, T. Pilarski, and D. Stager, "The crusher system for autonomous navigation," in *Proc. AUVSIs Unmanned Systems*, Aug. 2007.

[6] M. Buehler, "Summary of DGC 2005 results," *J. Field Robot.*, vol. 23, no. 9, pp. 465–466, 2006.

[7] J.-F. Lalonde, N. Vandapel, D. F. Huber, and M. Hebert, "Natural terrain classification using three-dimensional ladar data for ground robot mobility," *J. Field Robot.*, vol. 23, no. 10, pp. 839–861, Oct. 2006.

[8] E. Krotkov, S. Fish, L. Jackel, B. McBride, M. Perschbacher, and J. Pippine, "The DARPA PerceptOR evaluation experiments," *Auton. Robots*, vol. 22, no. 1, pp. 19–35, 2007.

[9] C. Shoemaker and J. Bornstein, "The DEMO III UGV program: A testbed for autonomous navigation research," in *Proc. IEEE Int. Symp. Intelligent Control (ISIC)*, Sept. 1998, pp. 644–651.

[10] T. Howard and A. Kelly, "Optimal rough terrain trajectory generation for wheeled mobile robots," *Int. J. Robot. Res.*, vol. 26, no. 2, pp. 141–166, 2007.

[11] N. Vandapel, D. Huber, A. Kapuria, and M. Hebert, "Natural terrain classification using 3-D ladar data," in *Proc. IEEE Int. Conf. Robotics and Automation*, Apr. 2004, pp. 5117–5122.

[12] D. Ferguson and A. Stentz, "Using interpolation to improve path planning: The field D* algorithm," *J. Field Robot.*, vol. 23, no. 2, pp. 79–101, Feb. 2006.

[13] D. Bradley, R. Unnikrishnan, and J. Bagnell, "Vegetation detection for driving in complex environments," in *Proc. IEEE Int. Conf. Robotics and Automation*, Apr. 2007, pp. 503–508.
[14] D. M. Bradley, "Learning in modular systems," Ph. D. dissertation, Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, Aug. 2009.
[15] C. Wellington, A. Courville, and A. Stentz, "Interacting markov random fields for simultaneous terrain modeling and obstacle detection," in *Proc. Robotics Science and Systems*, June 2005.
[16] R. Hadsell, J. A. Bagnell, and M. Hebert, "Accurate rough terrain estimation with space-carving kernels," in *Proc. Robotics Science and Systems*, June 2009.
[17] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. 20th Int. Conf. Machine Learning (ICML)*, 2003.
[18] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Gaussian processing modeling of large scale terrain," *J. Field Robot.*, vol. 26, no. 10, pp. 812–840, 2009.
[19] D. Silver, B. Sofman, N. Vandapel, J. A. Bagnell, and A. Stentz, "Experimental analysis of overhead data processing to support long range navigation," in *Proc. IEEE/JRS Int. Conf. Intelligent Robots and Systems*, Oct. 2006, pp. 2443–2450.
[20] B. Sofman, E. L. Ratliff, J. A. Bagnell, J. Cole, N. Vandapel, and A. Stentz, "Improving robot navigation through self-supervised online learning," *J. Field Robot.*, vol. 23, no. 12, pp. 1059–1075, Dec. 2006.
[21] C. Wellington and A. Stentz, "Learning predictions of the load-bearing surface for autonomous rough-terrain navigation in vegetation," in *Proc. Int. Conf. Field and Service Robotics (FSR'03)*, July 2003, pp. 49–54.
[22] R. Hadsell, P. Sermanet, J. Ben, A. Erkan, M. Scoffier, K. Kavukcuoglu, U. Muller, and Y. LeCun, "Learning long-range vision for autonomous off-road driving," *J. Field Robot.*, vol. 26, no. 2, pp. 120–144, 2009.
[23] M. Happold, M. Ollis, and N. Johnson, "Enhancing supervised terrain classification with predictive unsupervised learning," in *Proc. Robotics: Science and Systems*, 2006.
[24] L. Jackel, E. Krotkov, M. Perschbacher, J. Pippine, and C. Sullivan, "The DARPA LAGR program: Goals, challenges, methodology, and phase I results," *J. Field Robot.*, vol. 23, no. 11, pp. 945–973, 2006.
[25] N. D. Ratliff, D. Silver, and J. A. Bagnell, "Learning to search: Functional gradient techniques for imitation learning," *Auton. Robots*, vol. 27, no. 1, pp. 25–53, July 2009.
[26] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Boosting algorithms as gradient descent," in *Advances in Neural Information Processing Systems 12*. Cambridge, MA: MIT Press, 2000, pp. 512–518.
[27] D. Silver, J. A. Bagnell, and A. Stentz, "Perceptual interpretation for autonomous navigation through dynamic imitation learning," in *Proc. Int. Symp. Robotics Research*, Aug. 2009.
[28] M. W. Bode, "Learning the forward predictive model for an off-road skid-steer vehicle," Robot. Inst., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-07-32, Sept. 2007.
[29] D. Silver, J. A. Bagnell, and A. Stentz, "Applied imitation learning for autonomous navigation in complex natural terrain," in *Proc. Field and Service Robotics*, July 2009.
[30] D. Silver, J. A. Bagnell, and A. Stentz, "High performance outdoor navigation from overhead data using imitation learning," in *Proc. Robotics Science and Systems*, June 2008.
[31] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
[32] Y. LeCun, U. Muller, J. Ben, E. Cosatto, and B. Flepp, "Off-road obstacle avoidance through end-to-end learning," in *Proc. Neural Information Processing Systems (NIPS'05)*, Cambridge, MA, MIT Press, 2005, pp. 739–746.
[33] B. Sofman, J. A. Bagnell, and A. Stentz, "Anytime online novelty detection for vehicle safeguarding," in *Proc. IEEE Int. Conf. Robotics and Automation*, May 2010.

James Andrew Bagnell received a B.Sc. degree in electrical engineering with highest honors from the University of Florida in 1998 and M.S. and Ph.D. degrees from Carnegie Mellon University (CMU) in 2002 and 2004, respectively. He is an associate research professor at CMU's Robotics Institute and National Robotics Engineering Center (NREC) and is cross-appointed with the Machine Learning Department. He has spent the last decade working in machine learning and robotics, has written more than 50 peer-reviewed publications, regularly serves on the program committee, and is an associate editor for conferences and journals in the field. His research focuses on machine learning for automated decision making, machine perception, adaptive control, planning under uncertainty, and robotics.

David Bradley received a B.S.E. degree in electrical engineering from Princeton University in 2003 and an M.S. and Ph.D. degrees in robotics from CMU in 2005 and 2010, respectively. He was a National Defense Science and Engineering Graduate Fellow. He is a commercialization specialist at the NREC, part of CMU. He is currently researching machine learning techniques for mobile robot perception.

David Silver received a B.S. degree in computer science and an M.S. degree in robotics from CMU. He is a doctoral candidate in robotics at CMU. He is currently researching robust mobile robot navigation systems for complex unstructured environments.

Boris Sofman received a B.S. degree in computer science and electrical and computer engineering and an M.S. degree in robotics from CMU in 2005 and 2007, respectively. He is a doctoral candidate in robotics at CMU, being supported by a Sandia National Laboratories Excellence in Engineering Fellowship. He is currently researching online machine-learning techniques for improving the effectiveness and reliability of mobile robot navigation.

Anthony Stentz received his B.S. degree in physics from Xavier University, Ohio, in 1982 and his M.S. and Ph.D. degrees in computer science from CMU in 1984 and 1989, respectively. He is a research professor at the Robotics Institute, CMU and an associate director of the Robotics Institute's NREC. He has published more than 150 journal articles, conference and workshop papers, books, technical reports, and has patents to his credit. He is a recipient of the Alan Newell Award for Research Excellence for his agricultural automation work and a NASA Board Award for developing software used on the Mars Exploration Rovers. His research expertise includes unmanned ground vehicles, unmanned air vehicles, dynamic planning, multivehicle planning and coordination, perception for mobile vehicles, robot architecture, and artificial intelligence in the context of fieldworthy systems.

Address for Correspondence: James Andrew Bagnell, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA. E-mail: dbagnell@ri.cmu.edu.