

# Learning Predictions of the Load-Bearing Surface for Autonomous Rough-Terrain Navigation in Vegetation

Carl Wellington

Robotics Institute  
Carnegie Mellon University  
Pittsburgh, PA 15201 USA  
cwellin@ri.cmu.edu

Anthony (Tony) Stentz

Robotics Institute  
Carnegie Mellon University  
Pittsburgh, PA 15201 USA  
axs@ri.cmu.edu

## Abstract

*Current methods for off-road navigation using vehicle and terrain models to predict future vehicle response are limited by the accuracy of the models they use and can suffer if the world is unknown or if conditions change and the models become inaccurate. In this paper, an adaptive approach is presented that closes the loop around the vehicle predictions. This approach is applied to an autonomous vehicle driving through unknown terrain with varied vegetation. Features are extracted from range points from forward looking sensors. These features are used by a locally weighted learning module to predict the load-bearing surface, which is often hidden by vegetation. The true surface is then found when the vehicle drives over that area, and this feedback is used to improve the model. Results using real data show improved predictions of the load-bearing surface and successful adaptation to changing conditions.*

## 1 Introduction and Related Work

Automated vehicles that can safely operate in rough terrain hold the promise of higher productivity and efficiency by reducing the need for skilled operators, increased safety by removing people from dangerous environments, and an improved ability to explore difficult domains on earth and other planets. Even if a vehicle is not fully autonomous, there are benefits from having a vehicle that can reason about its environment to keep itself safe. Such systems can be used in safeguarded teleoperation or as an additional safety system for human operated vehicles.

To safely perform tasks in unstructured environments, an automated vehicle must be able to recognize terrain interactions that could cause damage to the vehicle. This is a difficult problem because there are complex dynamic interactions between the vehicle and the terrain that are often unknown and can change over time, vegetation is com-

pressible which prevents a purely geometric interpretation of the world, there are catastrophic states such as rollover that must be avoided, and there is uncertainty in everything. In agricultural applications, much about the environment is known, but unexpected changes can occur due to weather, and the vehicle is often required to drive through vegetation that changes during the year. In more general off-road exploration tasks, driving through vegetated areas may save time or provide the only possible route to a goal destination, and the terrain is often unknown to the vehicle.

Many researchers have approached the rough terrain navigation problem by creating terrain representations from sensor information and then using a vehicle model to make predictions of the future vehicle trajectory to determine safe control actions [1, 2, 3, 4]. These techniques have been successful on rolling terrain with discrete obstacles and have shown promise in more cluttered environments, but handling vegetation remains a challenge.

Navigation in vegetation is difficult because the range points from stereo cameras or a laser range-finder do not generally give the load-bearing surface. Classification of vegetation and solid substances can be useful for this task, but it is not sufficient. A grassy area on a steep slope may be dangerous to drive on whereas the same grass on a flat area could be easily traversable. Researchers have modeled the statistics of laser data in grass to find hard objects [5], assigned spring models to different terrain classes to determine traversability using a simple dynamic analysis [4], and kept track of the ratio of laser hits to laser pass-throughs to determine the ground surface in vegetation [3].

The above methods all rely on various forms of vehicle and terrain models. These models are difficult to construct, hard to tune, and if the terrain is unknown or changing, the models can become inaccurate and the predictions will be wrong. Incorrect predictions may lead to poor decisions and unsafe vehicle behavior. In this work, we investigate model learning methods to mitigate this problem.

Other researchers have investigated the use of parameter identification techniques with soil models to estimate soil parameters on-line from sensor data [6, 7], but these methods only determine the terrain that the vehicle is currently traversing. We are interested in taking this a step further and closing the loop around the vehicle predictions themselves by learning a better mapping from forward looking sensor data to future vehicle state. This allows the vehicle to use its experience from interacting with the terrain to adapt to changing conditions and improve its performance autonomously.

Our vehicle test platform is described in section 2 and our model-based approach to safeguarding in rough terrain is given in section 3. Section 4 explains the general approach of learning vehicle predictions and then describes how this is used to find the load-bearing surface in vegetation. Experimental results are given in section 5 and conclusions and future work are given in section 6.

## 2 Vehicle Platform and Terrain Mapping

Our project team [8] has automated a John Deere 6410 tractor (see figure 1). This vehicle has a rich set of sensors, including a differential GPS unit, a 3-axis fiber optic vertical gyro, a doppler radar ground speed sensor, a steering angle encoder, four custom wheel encoders, a high-resolution stereo pair of digital cameras, and two SICK laser range-finders (ladar) mounted on custom actively controlled scanning mounts. The first ladar on the roof of the vehicle is mounted horizontally and is scanned to cover the area in front of the tractor. The ladar on the front bumper is mounted vertically and is actively scanned in the direction the tractor is steering. We are currently experimenting with a near-infrared camera and a millimeter-wave radar unit as well.

The approach described in this work builds maps using range points from multiple lasers that are actively scanned while the vehicle moves over rough terrain. The true ground surface is then found when the tractor drives over that area a number of seconds later. To make this process work, it is important that the scanned ladars are precisely calibrated and registered with each other in the tractor frame, the timing of all the various pieces of sensor data is carefully synchronized, and the vehicle has a precise pose estimate. Our system has a 13 state extended Kalman filter with bias compensation and outlier rejection that integrates the vehicle sensors described above into an accurate estimate of the pose of the vehicle at 75Hz. This pose information is used to tightly register the data from the ladars into high quality terrain maps.

The information from the forward looking sensors represents a massive amount of data in its raw form, so some form of data reduction is needed. One simple approach is



Figure 1: Automated tractor test platform.

to create a grid in the world frame and then combine the raw data into summary information such as average height for each grid cell. This approach makes it easy to combine range information from the two ladars on our vehicle and to combine sensor information over time as the vehicle drives. Figure 3 shows the type of terrain we tested on and a grid representation of this area using the average height of each cell.

## 3 Rough Terrain Navigation

The goal of our system is to follow a predefined path through rough terrain while keeping the vehicle safe. Path tracking is performed using a modified form of pure pursuit [8]. The decision to continue is based on safety thresholds on the model predictions for roll, pitch, clearance, and suspension limits. These quantities are found by building a map of the upcoming terrain and using a vehicle model to forward simulate the expected trajectory on that terrain [2].

If the vehicle is moving relatively slowly and the load-bearing surface of the surrounding terrain can be measured, these quantities can be computed using a simple kinematic analysis. The trajectory of the vehicle is simulated forward in time using its current velocity and steering angle. A kinematic model of the vehicle is then placed on the terrain map at regular intervals along the predicted trajectory, and the heights of the four wheels are found in order to make predictions of vehicle roll and pitch. The clearance under the vehicle is important for finding body collisions and high centering hazards. It is found by measuring the distance from the height of the ground in each cell under the vehicle to the plane of the bottom of the vehicle. Our vehicle has a simple front rocker suspension, so checking the suspension limits involves calculating the roll of the front axle and comparing it to the roll of the rear axle. For smooth terrain with solid obstacles, this approach works well because accurate predictions of the load bearing sur-

face can be found by simply averaging the height of the range points in the terrain map.

If there is vegetation, finding the load-bearing surface can be difficult because many laser range points hit various places on the vegetation instead of the ground. Simply averaging the points in a grid cell performs poorly in this case. One possible solution is to use the lowest point in each grid cell instead. This correctly ignores the range points that hit vegetation, but because there is inevitable noise in the range points (especially at long distances), this results in the lowest outlier in the noise distribution being chosen, thus underestimating the true ground height.

## 4 Learning Vehicle Predictions

To overcome the difficulties associated with creating vehicle and terrain models for a complex environment that may be unknown or changing, a learning method is proposed. At the highest level, this approach is about closing the loop around vehicle predictions, as shown in figure 2. A vehicle prediction is a mapping from environmental sensor information and current vehicle state to future vehicle motion. This mapping is learned by observing actual vehicle motion after driving over a given terrain. During training and execution, the vehicle makes predictions about the future state of the vehicle by reasoning about its current state and the terrain in front of the vehicle. Then, when the vehicle drives over that terrain, it compares its predictions to what actually happened. This feedback is used for continual learning and adaptation to the current conditions.

By closing the loop around vehicle predictions and improving the system models on-line, tuning a system to a given application is easier, the system can handle changing or unknown terrain, and the system is able to improve its performance over time.

The learning vehicle predictions approach has been applied to the problem of finding the load-bearing surface in vegetation. The system makes predictions of the load-bearing surface from features extracted from the laser range points. Then it drives over the terrain and measures the true surface height with the rear wheels. These input-output pairs are used as training examples to a locally weighted learner that learns the mapping from terrain features to load-bearing surface height. Once the load-bearing surface is known, parameters of interest such as roll, pitch, clearance, and suspension limits can easily be computed using a kinematic vehicle model as described in section 3.

This combination of kinematic equations with machine learning techniques offers several advantages. Known kinematic relationships do not need to be learned, so the learner can focus on the difficult unknown relationships. Also, the learned function can be trained on flat safe areas, but is valid on steep dangerous areas. If we learned the

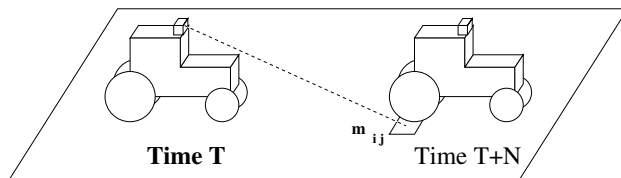


Figure 2: Learning vehicle predictions. Features from map cell  $m_{ij}$  extracted at time  $T$  are used to make a prediction. Then, at time  $T + N$  the vehicle traverses the area and determines if its prediction is correct. This feedback is used to improve the model.

roll and pitch directly, we would need to provide training examples in dangerous areas to get valid predictions there.

### 4.1 Feature Extraction

As described in section 2, the range points from the ladars are collected over time in a world frame grid. In addition to maintaining the average and lowest height of points in each cell, we use an approach similar to [3] to take advantage of the added information about free space that a laser ray provides. We maintain a scrolling map of 3D voxels around the vehicle that records the locations of any hits in a voxel, as well as the number of laser rays that pass through the voxel. Each voxel is 50cm square by 10cm tall. We use a cell size of 50cm because that is the width of the rear tires on our tractor, which are used for finding the true ground height.

Four different features are extracted from each column of voxels in the terrain map. The average height of range points works well for hard surfaces such as roads and rocks. The lowest point may provide more information about the ground height if there is sparse vegetation. Voxels that have a high ratio of hits to pass-throughs are likely to represent solid objects, so the average of the points in these voxels may help determine the load-bearing surface. As shown in figure 4, the standard deviation from a plane fit provides a good measure of how “smooth” an area is, and works well as a discriminator between hard things like road and compressible things like weeds. We are currently working on other features that use color and texture information in addition to laser range points.

### 4.2 Learning

By closing the loop around vehicle predictions, this approach produces a large amount of input-output pairs of training data. The system extracts features from the sensor data when making predictions and then records the true value when it drives over that area. This happens continuously, so the more the vehicle interacts with the environment, the more training data the learning system has to work with.



Figure 3: Top: Test area showing dirt roads and vegetation. Bottom: Map of test area using average height.

The mapping between the laser point features and the true ground height is unknown and potentially complex, so we use a general purpose function approximator for this task. Among the many possibilities, locally weighted learning [9] was chosen because it can accurately fit complex functions, it produces confidence estimates on its predictions, and there are online versions available.

A common form of locally weighted learning is locally weighted regression (LWR). Training with this algorithm simply involves inserting input-output pairs into memory. Then, when a new prediction is requested, the points in memory are weighted by a kernel function of the distance to the new query point, and a local multi-dimensional linear regression is performed on these weighted data points to produce a prediction. For good results, the kernel width must be chosen properly so that the resulting function does not over-fit or over-smooth the data. We use global leave-one-out cross validation to find the kernel width.

Standard statistical techniques for computing prediction bounds have been adapted to be used with this algorithm [10]. The size of the prediction bound depends both on the density of data points in the area, and on the noise in the outputs of nearby data points that cannot be explained by the model. The prediction bounds assume that the linear model structure is correct and the noise is zero-mean Gaussian. These assumptions can rarely be verified, but because they only have to be satisfied locally, the prediction bounds still give useful information in practice about

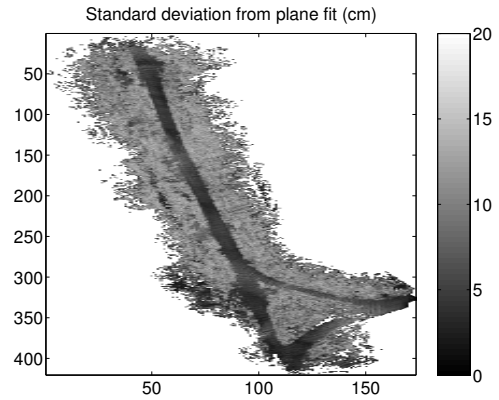


Figure 4: The standard deviation from a plane fit is found for each cell in the terrain map to discriminate between hard things like road and compressible things like weeds.

the confidence in the prediction.

Locally weighted learning stores all of its training data, so predictions take longer to compute as more training data is collected. This is not practical for systems such as ours that receive a continuous stream of data. Schaal [11] has described an on-line incremental version of LWR called receptive field weighted regression (RFWR). Instead of postponing all computations until a new prediction is requested, RFWR incrementally builds a set of receptive fields, each of which has a local regression model that is incrementally updated. The data points are then discarded, and predictions are made from a combination of nearby receptive fields. A forgetting factor is used to slowly discount old experience as it is replaced with new data.

## 5 Experimental Results

A set of experiments were performed to test the capabilities of this system. The first experiment shows the improved performance of the learned predictions and the usefulness of prediction bounds. The second experiment shows the system's ability to adapt to a change in the environment.

Although the analysis for the following experiments was performed off-line, the data used was collected in realistic conditions at the test site shown in figure 3. We collected approximately 25 minutes of data traveling on dirt roads and driving through varied vegetation often over a meter tall. Traveling at a speed of 1 m/s, we logged input-output pairs of laser features and corresponding rear wheel height at a rate of 4 examples per second, giving a total of 5700 data points. The data from the section shown in figure 3 was used as the test set, and the remaining 75% of the data was used for training.

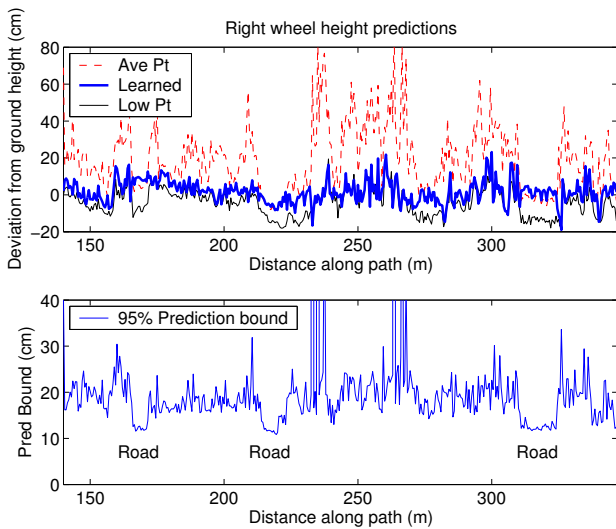


Figure 5: Top: Height prediction deviations for right wheel (should be zero). Average point has trouble in grass, lowest point underestimates, learned predictions do better. Bottom: Learner produces prediction bounds on its estimate that are lower for the smooth road areas and high for some of the tall vegetation.

### 5.1 Performance Improvement

The locally weighted regression technique described in section 4 was used to find a model for the training data described above. Leave-one-out cross validation was used to choose the kernel bandwidth. Figure 5 shows height prediction results for a section of the test set. The top graph shows the errors in predicted heights using the average height, the lowest point, and the learned result using all the features described in section 4.1. It can be seen that using the average point does quite poorly in vegetation and using the lowest point can result in outlier points being chosen. This especially occurs on roads where many laser points are recorded at long range (10m), resulting in higher uncertainty. The learned result has smaller errors than either of these because it is able to combine the different features in an appropriate way. The lower graph shows the 95% prediction intervals produced by LWR. The three times that the vehicle drives on the road are clearly shown by the tighter prediction bounds, and the learner has very little confidence in its predictions for some of the points in deep vegetation. These bounds are important because they could be used to modify the behavior of the vehicle. On simple terrain such as roads where the learner is confident, the vehicle could drive faster or attempt more extreme angles. In difficult terrain with tall vegetation where the learner has low confidence, the vehicle could approach the area with caution or avoid it completely.

Figure 6 shows that combining all the features using

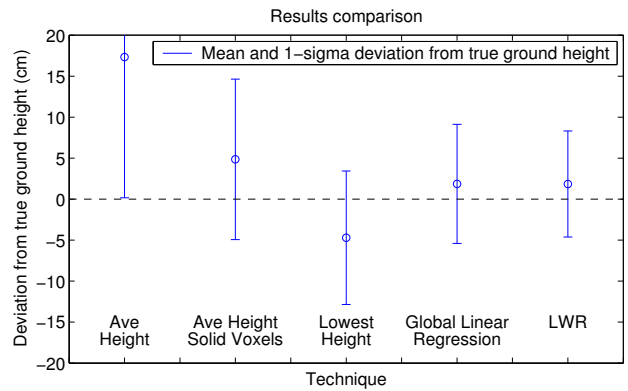


Figure 6: Comparison of the performance of the different features along with global and local regression on the test set.

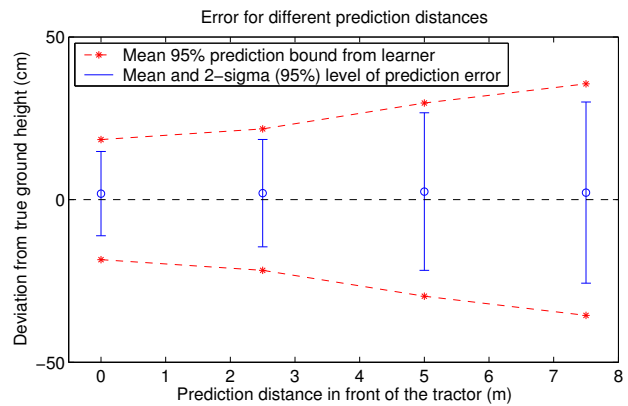


Figure 7: Error of predictions on the test set at different distances using LWR. The prediction bounds from the learner are similar to the actual prediction errors.

global or local regression performs better than using any of the features individually. LWR can represent the small non-linearities in this problem and performs slightly better than global linear regression. More importantly, LWR produces local prediction bounds that reflect the confidence in the prediction at that particular location in the feature space, allowing the system to be cautious in areas it hasn't experienced before.

The height predictions in figures 5 and 6 use all the laser points that are collected for a given patch of terrain. In practice, this would only happen if the system made predictions of future vehicle motion just in front of its current location. This is not useful if the vehicle cannot react in this distance. Figure 7 shows the effect of making predictions further ahead of the vehicle. The plot shows that the spread of the prediction errors increases when the predictions are made farther into the future. The figure also shows the mean 95% prediction bound the learner produces for dif-

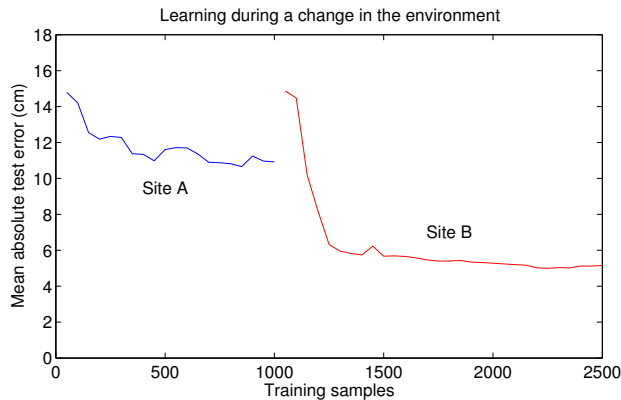


Figure 8: After learning a model for Site A using RFWR, the system is transported to Site B. The error rises initially, but then the system adapts to the new environment.

ferent prediction distances. This curve is similar to the  $2\sigma$  (95%) level on the error, again showing the usefulness of the prediction bounds.

## 5.2 Adapting to Environmental Change

The above tests were performed in batch mode using locally weighted linear regression. In this experiment, the on-line receptive field weighted regression algorithm was used to test the system’s ability to adapt to a changing environment. We collected more data in another test site that had dense vegetation approximately 0.75m high. This data was split into a training and test set. The training data was presented to the RFWR learning algorithm, and the prediction error on the test set was periodically computed.

Figure 8 shows the algorithm learning the characteristics of this site and reducing prediction error on the test set. Then, starting with sample 1000, the system was presented data used in the previous experiments. After being trained in the first site with dense vegetation, the algorithm did poorly in the new site with roads and more varied vegetation. However, the learner quickly adapted to the new environment and ended up with a prediction error similar to the batch mode version of LWR on the test set. This experiment shows the importance of using an adaptive approach to be able to handle a changing environment.

## 6 Conclusions and Future Work

We have described the general approach of learning vehicle predictions for local navigation, and we have applied the technique to the problem of finding the load-bearing surface in vegetation. The locally weighted learning solution to this problem performed better than using a particular feature or performing global linear regression. An-

other benefit of this technique is that it produces prediction bounds on its estimate, and these were shown to be fairly accurate. Finally, the ability of the system to adapt to changing environmental conditions was shown.

Future work in this area will include an investigation into other features such as color and texture from camera data, and the use of the prediction bounds for better vehicle control. We will also investigate whether the technique of learning vehicle predictions can be applied to other rough-terrain navigation problems such as finding terrain friction characteristics or dynamic vehicle response.

## Acknowledgments

This work has been supported by John Deere under contract number 476169. The authors would like to thank Jeff Schneider for the use of his Vizier code for locally weighted learning, and Stefan Schaal for the use of his code for incremental locally weighted learning.

## References

- [1] M. Daily, J. Harris, D. Keirse, D. Olin, D. Payton, K. Reiser, J. Rosenblatt, D. Tseng, and V. Wong, “Autonomous cross-country navigation with the ALV,” in *IEEE Int. Conf. on Robotics and Automation (ICRA 88)*, vol. 2, pp. 718–726, April 1988.
- [2] A. Kelly and A. Stentz, “Rough terrain autonomous mobility - part 2: An active vision, predictive control approach,” *Autonomous Robots*, vol. 5, pp. 163–198, May 1998.
- [3] A. Lacaze, K. Murphy, and M. DelGiorno, “Autonomous mobility for the Demo III experimental unmanned vehicles,” in *Assoc. for Unmanned Vehicle Systems Int. Conf. on Unmanned Vehicles (AU-VSI 02)*, July 2002.
- [4] A. Talukder, R. Manduchi, R. Castano, K. Owens, L. Matthies, A. Castano, and R. Hogg, “Autonomous terrain characterisation and modelling for dynamic control of unmanned vehicles,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 02)*, pp. 708–713, October 2002.
- [5] J. Macedo, R. Manduchi, and L. Matthies, “Ladar-based discrimination of grass from obstacles for autonomous navigation,” in *Int. Symposium on Experimental Robotics (ISER 00)*, December 2000.
- [6] A. T. Le, D. Rye, and H. Durrant-Whyte, “Estimation of track-soil interactions for autonomous tracked vehicles,” in *IEEE Int. Conf. on Robotics and Automation (ICRA 97)*, vol. 2, pp. 1388–1393, April 1997.
- [7] K. Iagnemma, H. Shibly, and S. Dubowsky, “On-line terrain parameter estimation for planetary rovers,” in *IEEE Int. Conf. on Robotics and Automation (ICRA 02)*, May 2002.
- [8] A. Stentz, C. Dima, C. Wellington, H. Herman, and D. Stager, “A system for semi-autonomous tractor operations,” *Autonomous Robots*, vol. 13, pp. 87–104, July 2002.
- [9] C. Atkeson, A. Moore, and S. Schaal, “Locally weighted learning,” *AI Review*, vol. 11, pp. 11–73, April 1997.
- [10] S. Schaal and C. G. Atkeson, “Assessing the quality of learned local models,” in *Advances in Neural Information Processing Systems (NIPS 94)*, pp. 160–167, 1994.
- [11] S. Schaal and C. G. Atkeson, “Constructive incremental learning from only local information,” in *Neural Computation*, vol. 10, pp. 147–184, 1998.