

# Online Adaptive Rough-Terrain Navigation in Vegetation

Carl Wellington and Anthony (Tony) Stentz  
Robotics Institute  
Carnegie Mellon University  
Pittsburgh PA 15201 USA  
{cwellin, axs}@ri.cmu.edu

**Abstract**—Autonomous navigation in vegetation is challenging because the vegetation often hides the load-bearing surface which is used for evaluating the safety of potential actions. It is difficult to design rules for finding the true ground height in vegetation from forward looking sensor data, so we use an online adaptive method to automatically learn this mapping through experience with the world. This approach has been implemented on an autonomous tractor and has been tested in a farm setting. We describe the system and provide examples of finding obstacles and improving roll predictions in the presence of vegetation. We also show that the system can adapt to new vegetation conditions.

## I. INTRODUCTION AND RELATED WORK

Automated vehicles that can safely operate in rough terrain would benefit many applications in agriculture, mining, and the exploration of hazardous areas. Operating in the unstructured environments common in these applications requires a vehicle to recognize untraversable areas and terrain interactions that could cause damage to the vehicle. This is a challenging task due to the complex interactions between the vehicle and the terrain, an environment that is often unknown or changing, and the limitations of current sensing technologies to provide measurements of important quantities, such as the the load-bearing surface of the upcoming terrain.

Vegetation further complicates the situation by covering and hiding the load-bearing surface, preventing a purely geometric interpretation of the world. In many agricultural applications, the vehicle is required to drive through vegetation, and in more general off-road exploration tasks, driving through vegetated areas may save time or provide the only possible route to a goal. Vegetation also changes based on the season and weather.

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 challenging problem.

Navigation in vegetation is difficult because the range points from forward looking sensors such as stereo cameras or a laser range-finder do not generally give the load-bearing surface. Classification of vegetation and solid substances [5] 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 [6], assigned spring models to different terrain classes to determine traversability using a simple dynamic analysis [4], kept track of the ratio of laser hits to laser pass-throughs to determine the ground surface in vegetation [3], and used a set of time and space locality constraints to filter out vegetation [7].

The above methods all rely on fixed vehicle and terrain models that can be 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.

Other researchers have investigated the use of parameter identification techniques with soil models to estimate soil parameters on-line from sensor data [8], but these methods only determine the terrain that the vehicle is currently traversing. We would like to make predictions of the terrain in front of the vehicle so that the system can take appropriate action before it reaches these areas.

To make predictions of upcoming terrain, we have proposed a method of learning the mapping from forward looking sensors to future vehicle state automatically by using the vehicle's experience from interacting with the terrain [9]. In sparse vegetation, we showed improved predictions of the load-bearing surface using this method running offline with simple features.

This paper describes the implementation of the method presented in [9] on an autonomous farm vehicle (see figure 1)



Fig. 1. Automated tractor test platform in vegetation.

and shows results of the system running online in more difficult and varied terrain. The method has also been extended to use a richer set of features, to automatically learn the distance dependence of the features, and to include training on untraversable terrain.

Our vehicle test platform and general approach to rough terrain navigation is given in section II. Section III describes the online adaptation of the system. Experimental results are given in section IV and conclusions and future work are given in section V.

## II. ROUGH TERRAIN NAVIGATION

Autonomous navigation in rough terrain requires consideration of how the vehicle will interact with upcoming terrain in order to keep the vehicle safe. Our project is focusing on agricultural applications where the goal is to follow a predefined path or reach a goal destination while staying within various safety constraints. Depending on the specific application, the vehicle should avoid an unsafe area on the path or stop and call for help. We consider terrain that is common on a farm including slopes, ditches, and vegetation, as well as relevant obstacles such as people, equipment, posts, and buildings.

### A. Vehicle Platform and Terrain Mapping

Our project team has automated a John Deere 6410 tractor (see figure 1). The tractor computes its current pose using a 13 state extended Kalman filter with bias compensation and outlier rejection that integrates data from a differential GPS unit, a 3-axis fiber optic vertical gyro, a doppler radar ground speed sensor, a steering angle encoder, and four custom wheel encoders. To sense upcoming terrain, the vehicle has a high-resolution stereo pair of digital cameras, a near-infrared camera, 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.

The cameras and scanned ladars are precisely calibrated and registered with each other in the tractor frame, and the information from the different sensors is tightly synchronized with the vehicle pose to be able to accumulate data into a high quality global terrain map (see figure 2).

Our goal is to find the load-bearing surface, so the system maintains a grid representation of the area around the vehicle to hold the data from the forward looking sensors. We use an approach similar to [3] to take advantage of the added information about free space that a laser ray provides. Each grid cell contains a column of voxels that record the locations of any hits in that area of space, as well as the number of laser rays that pass through the voxel. Fixed length buffers in each voxel let new hit or pass-through data replace old data and allow the system to handle simple dynamic obstacles. Although the results presented in this paper only use ladar data, this representation is fairly general and can accept any



Fig. 2. Map of farm test site buildings using both ladars as the vehicle drove on a path to the field. The brightness of the range points represents the laser reflectance value.



Fig. 3. Elevation map of farm test site buildings (tractor is in lower right of figure 2) using the mean height in each grid cell. Model predictive control showing chosen arc along path and unsafe arcs to the left due the steep slope.

range sensor as an input. To reduce the dimensionality of the data, features can be computed for each grid cell. For example, figure 3 shows a map that uses the average height of all the points in each cell.

### B. Model Predictive Control

Autonomous navigation in rough terrain has been successfully implemented within a model predictive control framework as a search through speed and steering angle commands over a fixed path length [2]. In this framework, a vehicle model is used with a terrain model to predict the behavior of the vehicle for different control inputs, and then the predictions are evaluated to choose the best control. Figure 3 gives an example of this approach. A set of controls are sampled from the space of possible speed and steering angle commands, and the vehicle and terrain models are used to predict the result of using these controls. The vehicle model includes system delays and the dynamics of the steering actuation to produce feasible arcs.

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 the vehicle roll and pitch and ensure that the front rocker suspension is within limits. The heights of the terrain cells under the vehicle are used to check for clearance hazards. We have extended our tractor model to include various implements and they are checked for clearance hazards as well.

Once a set of possible arcs are found that satisfy the safety constraints, a cost function is used to evaluate the arcs to find the best control. For path tracking, the cost is the error between the arc and the desired path. If a destination does not have a specific path associated with it, the cost is defined as the minimum distance between the arc and the goal point. By choosing the lowest cost arc that satisfies the safety constraints, the vehicle is able to smoothly avoid obstacles or other dangerous terrain conditions and then recompute its path. The system checks speed choices from fastest to slowest, which results in a graceful slowdown as the vehicle approaches an obstacle.

For smooth terrain with solid obstacles, this approach works well because accurate predictions of the load bearing surface can be found by simply averaging the height of the range points in the terrain map. However, this performs poorly in vegetation since many laser range points hit various places on the vegetation instead of the ground. Using the lowest point in each grid cell correctly ignores the scattered range points that hit vegetation, but many times the lowest laser point does not penetrate thicker vegetation, and using the lowest point will cause the system to ignore positive obstacles. Clearly, a more advanced criteria than simply using the average or lowest height in a cell is needed to find a useful estimate of the ground plane in vegetation. However, it is difficult to come up with such a rule, and tuning to a given environment is time consuming and challenging.

### III. ONLINE ADAPTATION

To overcome the difficulties associated with creating terrain models for a complex environment that may be unknown or changing, we close the loop around vehicle predictions as shown in figure 4 by making predictions from sensor data and then observing actual behavior when the vehicle drives over that area. This feedback is used for continual learning and adaptation to current conditions.

The system makes predictions of the load-bearing surface from features extracted from the laser range points and stores these features in that cell. Features may change depending on how far away the sensors are, so this process is repeated at different distances to build up a set of feature sets for the cell. Then the vehicle drives over the terrain and measures the true surface height with the rear wheels. All the stored feature sets are correlated with the true height and 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. This process happens continuously, so the more the vehicle interacts with the environment, the more training data the learning system receives.

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 II-B. This combination of kinematic equations with learning techniques offers several advantages. Known kinematic relationships do not need to be learned, so the learner can focus on the difficult unknown relationships.

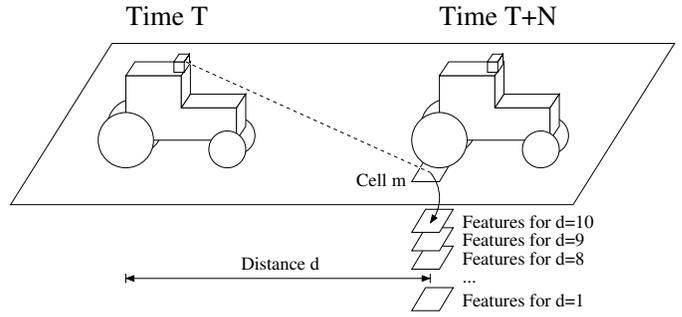


Fig. 4. Learning terrain height predictions. At time  $T$ , features from map cell  $m$  at distance  $d$  are used to make a prediction and then stored. As the vehicle gets closer to the cell, feature sets for the cell computed at those distances are stored. Then, at time  $T + N$  the vehicle traverses the cell and finds its true height. The learner is trained with the set of feature sets computed at different distances and the true height found from the wheel.

Also, the learned function can be trained on flat safe areas, but is valid on steep dangerous areas. If we learned the roll and pitch directly, we would need to provide training examples in dangerous areas to get valid predictions there.

Features for obstacles such as buildings and other vehicles cannot be found by driving over them, so they need to be entered manually. For these untraversable areas, feature sets are stored for different distances just like other areas, but we must manually select them in our user interface to assign a truth value for learning (often highest or lowest point).

#### A. Feature Extraction

As described in section II-A, sensor information is accumulated over time in a global voxel representation. Each voxel is a 15 cm cube, and maintains the laser hits and pass-throughs for that voxel. Features are extracted over 3x3 neighborhoods of columns of voxels. The small voxel size is needed because laser pass-throughs in vegetation become more unlikely for larger voxel sizes. However, 15cm is too small to get meaningful results for many of features based on the distribution and shape of points, so we combine multiple columns of voxels together. The resulting 45cm x 45cm patch on the ground is approximately the same size as the rear wheel contact surface we use to measure true ground height.

1) *Distance*: Many features are dependent on how far away from the vehicle they are when they are observed and how many laser range points are in the cell. Figure 5 shows a plot of a learned surface and the associated 95% prediction intervals for the difference between true ground height and the 'lowest point' feature using a small dataset taken in vegetation. The plot shows that the performance of using the 'lowest point' feature to predict ground height is dependent on the distance and number of points. The surface was learned using the techniques given below in section III-B. The surface shows that the lowest point becomes a worse predictor of the true ground height at greater distances. The prediction intervals show that predictions using lowest point can be more certain at closer distances and when there are more points in the cell. Because of observations such as these, we include the number

of points in a cell and the distance from the tractor of a cell as features for the learner to use. This allows the learner to find relationships such as those in figure 5 to produce better predictions and more accurate prediction bounds.

2) *Simple statistics on height:* We extract several simple features based on the height of points in a column of voxels, including the mean height and robust lowest and highest point (5% and 95% height value).

3) *Shape of the point cloud:* Analysis of the eigenvalues of the covariance matrix of the points in a cell gives information about the shape of the point cloud, which can be used to differentiate different terrain types. We incorporate the three shape features discussed in [5] that are able to separate different types of point clouds (line, plane, scatter). These features are meant to be applied over a neighborhood, so for each set of voxel columns, we find the 3x3x3 cube of voxels with the highest point density and find the shape features from this data. Full classification and grouping of all 3D data as in [5] could increase performance further.

4) *Voxel hit and pass-through information:* Several features are computed from the voxel hit and pass-through information that are useful to determine if an area is solid. In general, areas with a mixture of hits and pass-throughs are vegetation, whereas solid objects have a higher percentage of hits. As a vegetation indicator, we compute the ratio of pass-throughs in voxels with hits to the total number of pass-throughs. To detect solid objects, we compute the percentage of hits for each voxel and then sum the result. Finally, we compute the difference in the number of hits in adjacent columns of voxels to detect vertical solid surfaces.

5) *Appearance:* The maximum laser reflectance value in a cell is used as a feature that can help differentiate between different materials. We can also project values from our IR camera and color cameras into the grid representation, and we are currently looking into additional appearance features such as temperature, color, and texture that use this data.

## B. Learning

We use locally weighted learning [10] to learn the unknown and potentially complex mapping between laser features and the true ground height. 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.

Standard statistical techniques for computing prediction bounds have been adapted to be used with this algorithm [11]. 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.

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

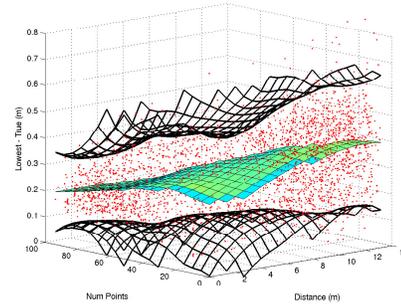


Fig. 5. Learned surface and prediction intervals showing that the performance of using the lowest point to predict ground height is dependent on distance and number of points.

that receive a continuous stream of data. Schaal [12] has described an on-line incremental version of LWR called locally weighted projection regression (LWPR). Instead of postponing all computations until a new prediction is requested, LWPR 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. LWPR handles high dimensional data by taking local projections on the data instead of using all the dimensions.

Our online system is currently using LWPR as its function approximator. Figure 5 shows an example of LWPR for a simple two dimensional problem.

## IV. EXPERIMENTAL RESULTS

We have tested the system described above at a nearby working farm and an undeveloped area with tall weeds. Results show that the system can find obstacles in sparse vegetation and improve predictions of vehicle safety quantities such as roll. We also show the benefits of adapting online when the vehicle encounters challenging new terrain.

### A. Obstacle in vegetation

Figure 6 shows an example of a person kneeling among tall sparse vegetation. The angle of figure 6(a) makes it difficult to see the taller weeds, but the highest point statistic shown in figure 6(b) clearly shows that the kneeling person is beside various vegetation of a similar height. Because the vegetation is sparse, the laser was able to penetrate through the tall weeds and the lowest point feature in figure 6(c) gives a good representation of the ground plane, but it makes the person disappear along with the vegetation. From these figures, it is clear that using a simple statistic such as the highest or lowest point will lead to false positives or false negatives in the presence of vegetation.

Instead of hand-crafting a set of rules and thresholds for how to combine our features to accomplish this task, we used the learning method described above to find this automatically. We drove around in similar types of vegetation for approximately 10 minutes to let the system learn what types of features

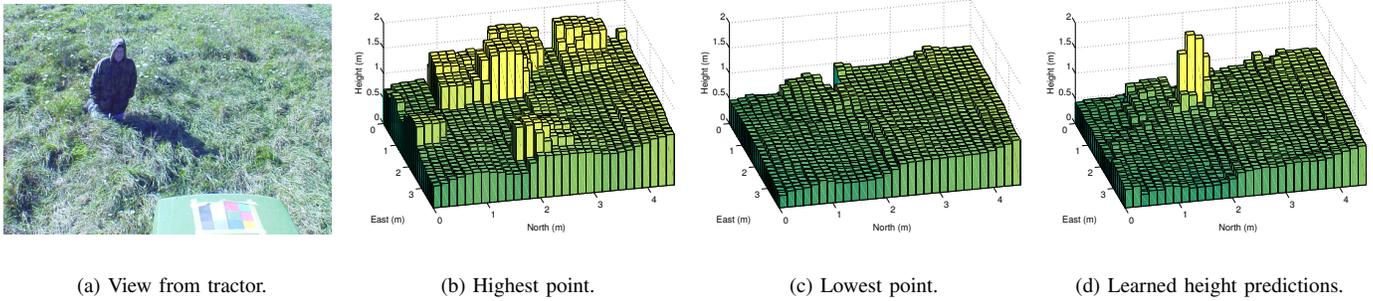


Fig. 6. Person kneeling next to similarly sized sparse vegetation. Learned result removes the vegetation but keeps the person.

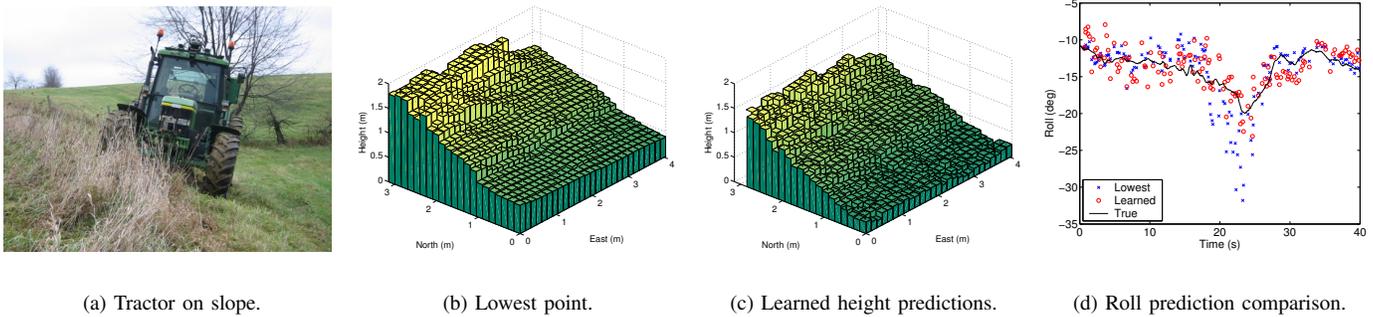


Fig. 7. Tractor driving through weeds on a slope. Learned result produces more accurate roll predictions.

represent compressible objects. To teach it about solid objects, we drove up to a set of walls, posts, and hay bales and then manually selected them to be trained using the highest point as the truth value.

After this simple training procedure, the learning system was able to produce the results shown in figure 6(d). The vegetation has been removed, thus reducing false positives, but the person remains and would trigger a clearance hazard.

### B. Roll predictions in vegetation

Figure 7 shows the vehicle traversing a slope with vegetation. Using the lowest point as shown in figure 7(b) results in overestimating the ground surface in some of the vegetation on the left. After training our system by driving it through some similar vegetation for approximately 10 minutes, it produces more accurate height predictions as shown in figure 7(c). From the system level, we really care about predictions of safety parameters such as roll and pitch. Figure 7(d) shows that the improved height estimates result in better roll predictions when compared to the true value found when we drove over that area. The plot shows that predictions using lowest point are poor for the area shown in figures 7(b) and 7(c) that is encountered 20 seconds into the test.

If the vegetation was on the bottom of the slope instead of the top, then making this mistake could be dangerous because the system would believe that the vehicle could drive on top of

the vegetation instead of predicting that it would drop below the vegetation and perhaps cause a roll-over hazard.

### C. Online adaptation

We also performed a set of experiments in much thicker vegetation that the laser cannot penetrate beyond short distances. To make predictions ahead of the vehicle in these circumstances, the system must use its prior experience with similar terrain to determine an estimate of the ground plane. Figure 8(a) shows the tractor entering dense vegetation over 1m tall. The three graphs in figure 8 show the results of three different ways to approach this difficult task. Each graph shows the error from the true ground height of the lowest point feature and the predicted ground height at a distance around 5m in front of the vehicle as it drives into the tall vegetation. At the beginning, when it is driving through relatively short grass, the lowest point feature works well, but after 10 seconds into the run when the vehicle enters the tall vegetation, the laser cannot penetrate the vegetation 5m in front of the vehicle, and the predictions must adjust.

The first case in figure 8(b) shows a system that was trained in other vegetation in a nearby area for approximately 10 minutes, and then continued to adapt as it drove into this new vegetation. It's prior training allows it to handle the transition from short grass to tall weeds, and it also does a decent job of finding the true ground height despite the limited data available.

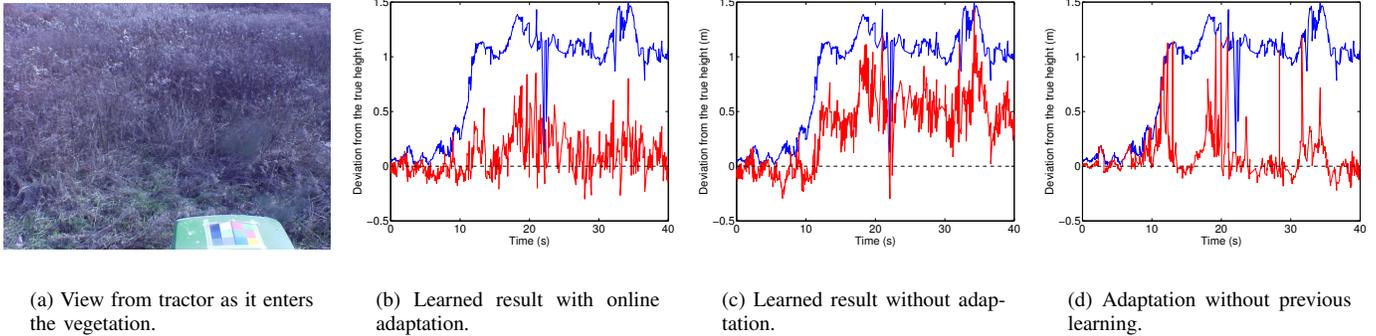


Fig. 8. Benefits of online adaptation when vehicle enters new vegetation. Blue line (top) shows lowest point. Red line (bottom) shows learned result.

Figure 8(c) shows what happens if the vehicle is trained just as in the first case, but then the learning algorithm is turned off. Using only its prior experience it is able to handle the transition, but it then has significantly higher error in the vegetation than the adaptive case, which would result in many false positives for the system.

The third case in figure 8(d) shows the predictions of the system when it is started fresh with no prior training but is allowed to adapt during the run. Without any experience, it fails to recognize the transition to weeds and continues using the lowest point. However, once it collects some data it adapts and starts doing better. Due to its limited training, it continues to make errors about when to use lowest point and when to drop below it, but the adaptive capability will allow it to get better over time.

These results show the benefit of adapting online to the current conditions and giving the system some context. Our system makes predictions by looking at a single grid cell in isolation. Without the context that humans use so effectively, the system is at a great disadvantage because a patch of short grass and a patch of tall grass may have very similar laser feature signatures, especially at a distance. Continually adapting online is one way to give the system context by exploiting the local similarity of the world. Of course, it is also important for the algorithm to detect when the upcoming terrain is different than what is around it, and then it must fall back on prior experience or use caution if the area is unknown.

## V. CONCLUSIONS AND FUTURE WORK

We have shown a system that produces improved predictions of vehicle safety parameters by learning predictions of the load-bearing surface in vegetation while still finding positive obstacles. The entire system runs online on our autonomous tractor test platform, and can improve and adapt over time. A key benefit of this approach is that the vehicle can generate labeled training examples of features at different distances just by driving through an area of interest. This makes it easy to collect massive amounts of training data quickly and let the system figure out how to interpret it.

In addition to more systematic testing of our system, we are currently looking at using the learner prediction intervals

for better vehicle speed control so it can slow down in areas where it is not confident. We are also experimenting with more appearance based features using the cameras on the vehicle. Finally, we will apply the same learning approach to be able to predict upcoming slippery conditions.

## ACKNOWLEDGEMENTS

This work has been supported by John Deere under contract 476169. The authors would like to thank 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, April 1988, pp. 718–726.
- [2] A. Kelly and A. Stentz, "Rough terrain autonomous mobility - part 2: An active vision, predictive control approach," *Autonomous Robots*, vol. 5, no. 2, 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 (AUVSI 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)*, October 2002, pp. 708–713.
- [5] N. Vandapel, D. Huber, A. Kapuria, and M. Hebert, "Natural terrain classification using 3-d lidar data," in *IEEE Int. Conf. on Robotics and Automation (ICRA 04)*, April 2004.
- [6] J. Macedo, R. Manduchi, and L. Matthies, "Lidar-based discrimination of grass from obstacles for autonomous navigation," in *Int. Symposium on Experimental Robotics (ISER 00)*, December 2000.
- [7] A. Castano and L. Matthies, "Foliage discrimination using a rotating lidar," in *IEEE Int. Conf. on Robotics and Automation (ICRA 03)*, vol. 1, September 2003, pp. 1–6.
- [8] 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.
- [9] C. Wellington and A. Stentz, "Learning predictions of the load-bearing surface for autonomous rough-terrain navigation in vegetations," in *Proceedings of the Int. Conf. on Field and Service Robotics*, July 2003.
- [10] C. Atkeson, A. Moore, and S. Schaal, "Locally weighted learning," *AI Review*, vol. 11, pp. 11–73, April 1997.
- [11] S. Schaal and C. G. Atkeson, "Assessing the quality of learned local models," in *Advances in Neural Information Processing Systems (NIPS 94)*, 1994, pp. 160–167.
- [12] S. Vijayakumar and S. Schaal, "Locally weighted projection regression : An  $O(n)$  algorithm for incremental real time learning in high dimensional spaces," in *Proceedings of the Seventeenth Int. Conf. on Machine Learning (ICML 2000)*, 2000, pp. 1079–1086.