An Integrated System for Autonomous Off-Road Navigation

D. Langer, J.K. Rosenblatt, M. Hebert

The Robotics Institute,
Carnegie Mellon University,
Pittsburgh, PA 15213

Abstract

In this paper, we report on experiments with a core system for autonomous navigation in outdoor natural terrain. The system consists of three parts: a perception module which processes range images to identify untraversable regions of the terrain, a local map management module which maintains a representation of the environment in the vicinity of the vehicle, and a planning module which issues commands to the vehicle controller. Our approach uses reactive planning for generating commands to drive the vehicle along with "early traversability evaluation," in which the perception module decides which parts of the terrain are traversable as soon as a new image is taken. We argue that our approach leads to a robust and efficient navigation system. We illustrate our approach by an experiment in which a vehicle travelled autonomously for one kilometer through unmapped cross-country terrain.

1 Introduction

Autonomous navigation missions through unmapped open terrain are critical in many applications of outdoor mobile robots. To successfully complete such missions, a mobile robot system needs to be equipped with reliable perception and navigation systems capable of sensing the environment, of building environment models, and of planning safe paths through the terrain. In that respect, autonomous cross-country navigation imposes two special challenges in the design of the perception system. First, the perception must be able to deal with very rugged terrain. Second, the perception system must be able to reliably process a large number of data sets over a long period of time.

Several approaches have been proposed to address these problems. Autonomous traverse of rugged outdoor terrain has been demonstrated as part of the ALV [11] and UGV [10] projects. JPL's Robby used stereo vision [9] as the basis of its perception system and has been demonstrated over a 100 m traverse in outdoor terrain. Other efforts include: The VAP project which is also based on stereo vision [2]; the PANORAMA Esprit project [15]; the MIT rovers which rely on simple sensing modalities [1]. Most of these perception systems use range images, from active ranging sensors or passive stereo, and build a map of the terrain around or in front of the vehicle. The planning systems use the maps to generate trajectories. The main questions in building such systems are: What should be in the map, and when should the map be computed?

In this paper, we argue that relatively simple methods of obstacle detection and local map building are sufficient for cross-country navigation. Furthermore, when used as input to a reactive planner, the vehicle is capable of safely traveling at significantly faster speeds than would be possible with a system that planned an optimal path through a detailed, high-resolution terrain map. Moreover, we argue that an accurate map is not necessary because the vehicle can safely traverse relatively large variations of terrain surface.

For these reasons, we propose an approach based on "early evaluation of traversability" in which the output of the perception system is a set of untraversable terrain regions used by a planning module to drive the vehicle. The system relies on "early evaluation" because the perception module classifies regions of the terrain as traversable or untraversable as soon as a new image is taken. As we will show, early traversability evaluation allows for a more reactive approach to planning in which steering directions and speed updates are generated rapidly and in which the vehicle can respond to dangerous situations in a more robust and more timely manner.

The goal of this paper is to present and discuss the performance of the overall system. Detailed descriptions of its components may be found in [7] for the local map module, [12] for the planning component, and in [8] for the complete system description.

2 System Overview

To illustrate our approach, we will describe a set of perception and navigation modules which constitute the core of a cross-country navigation system. The goal of this system is to enable the vehicle to travel through unmapped rugged terrain at moderate speeds, typically two to three meters per second. We arranged the system modules in a self-contained navigation system which we demonstrated on a one kilometer path through unmapped open terrain. In the next sections, we will use this result to illustrate our approach and to discuss the system performance and the implementation details of each module.

The perception and navigation system was developed as part of the Unmanned Ground Vehicle (UGV) project. The

. This research was partly sponsored by ARPA, under contracts "Perception for Outdoor Navigation" (contract number DACA76-89-C-0014, monitored by the US Army Topographic Engineering Center) and "Unmanned Ground Vehicle System" (contract number DAAE07-90-C-R059, monitored by TACOM). J. Rosenblatt was supported by a Hughes Research Fellowship. The views and conclusions expressed in this document are those of the authors and should not be interpreted as representing the official policies of ARPA or the US government.

support vehicle is a retrofitted HMWVV suitable for cross-country navigation. The sensor is the Erim laser range finder which acquires 64x256 range images at 2 Hz. An estimate of vehicle position is available at all times by combining readings from an INS system and from encoders.

Figure 1: The testbed vehicle

In the current system, the output of the perception system is a set of untraversable terrain regions which is maintained by a local map manager and is used by a planning module to drive the vehicle. The location of the untraversable regions with respect to the vehicle is maintained by a local map manager. Untraversable regions are terrain features such as high slopes, ditches, or tall objects which would endanger the vehicle.

In the remainder of the paper, we first describe the performance of the perception and navigation system on an example, then describe the three components of the system (Figure 2): perception, local map management, and planning, with an emphasis on the perception component. Finally, we discuss the pros and cons of the early evaluation approach.

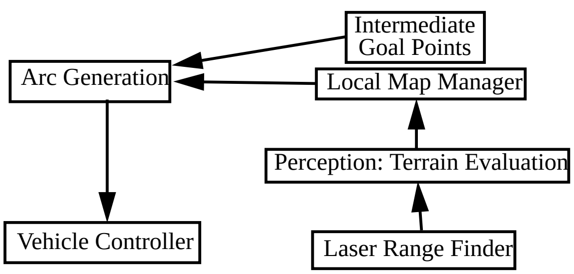

Figure 2: Architecture of the navigation system.

3 Autonomous Navigation: An Experiment

Figure 3 to Figure 7 show a typical run of the perception and navigation system. Figure 3 shows the environment used in this experiment. The terrain includes hills, rocks, and ditches. The white line superimposed on the image of the terrain shows the approximate path of the vehicle through this environment. The path was drawn manually for illustrative purpose.

Figure 4 shows the actual path recorded during the experiment projected on the average ground plane. In addition to the path, Figure 4 shows the obstacle regions as black dots and the intermediate goal points as small circles. In this example, the vehicle completed a one kilometer loop without manual intervention at an average speed of 2 m/s. The input to the system was a set of 10 waypoints separated by about one hundred meters on average. Except for the waypoints, the system does not have any previous knowledge of the terrain. Local navigation is performed by computing steering directions based on the locations of untraversable regions in the terrain found in the range images. An estimated 800 images were processed during this particular run. Figure 5 shows close-ups of three sections of the loop of Figure 3 with the approximate path of the vehicle.

Figure 3: View of terrain and approximate path.

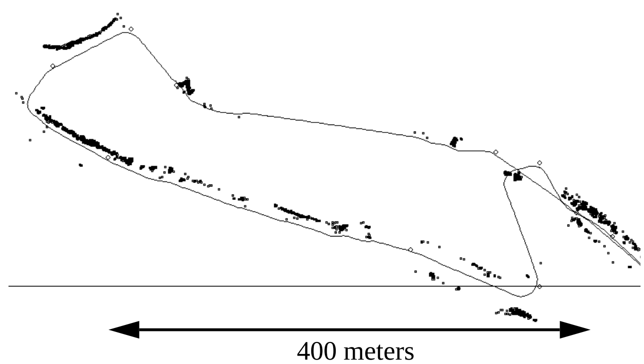

Figure 4: Exact path of vehicle; the obstacle regions are shown as black dots; the intermediate goal points are shown as small circles.

Figure 6 shows the elevation map obtained by pasting together the images taken along the paths. In each figure, the grey polygons are the projections of the fields of view on the ground, the curved grey line is the path of the vehicle on the ground, and the white dots indicate locations at which images were taken. The images are separated by approximately two meters in this case. The paths shown in Figure 6 are the actual paths followed by the vehicle. It is important to note that these maps are included for display purposes only and that the combined elevation maps are not actually used in the system. Finally, Figure 7 shows a display of the local map maintained around the vehicle. The squares correspond to 40x40 cm patches of terrain classified as untraversable regions or obstacles. This local map is computed from the position shown in Figure 5(c) and Figure 6(c).

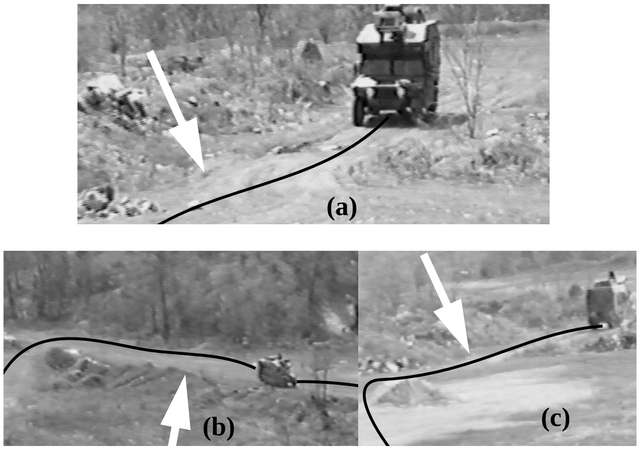

Figure 5: Local path of vehicle in three sections of the loop of Figure 3. The arrows indicate the locations at which the local maps are displayed in Figure 6 below.

All the displays, timings, and results used in the rest of the paper are from the site of Figure 3, although they may have been collected over different runs of the system. The system was configured to run on three SparcII workstations. Each workstation was dedicated to one of the three components of the system: perception, planning, and local map management. Communications between processes were established through Ethernet. Controller and sensor interface have their own dedicated processors running a real-time operating system while the other processes are Unix-based.

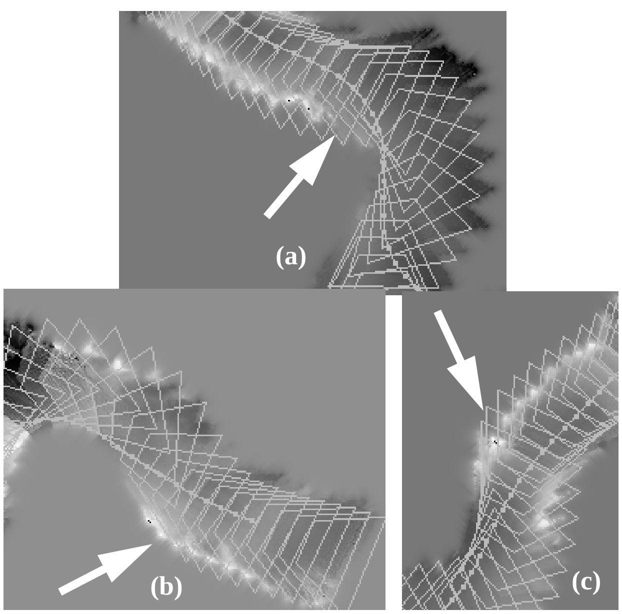

Figure 6: Display of the terrain as elevation maps for the sections shown in Figure 5. The grey line shows the path followed by the vehicle in this section. The grey dots show the positions at which the images were taken.

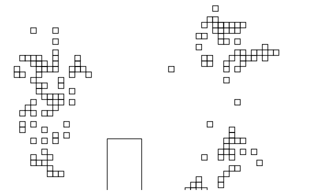

Figure 7: Display of the local traversability map at the location indicated Figure 5(c) and Figure 6(c).

4 Perception

The range image processing module takes a single image as input and outputs a list of regions which are untraversable. After filtering the input image, the module computes the (x,y,z) location of every pixel in the range image in a coordinate system relative to the vehicle's current position. The transformation from sensor to vehicle takes into account the orientation of the vehicle read from an INS system. The points are then mapped into a discrete grid on the (x,y) plane. Each cell of the grid contains the list of the coordinates of the points which fall within the bounds of the cell. The size of a cell in the current system is 20 cm. This number depends on the angular resolution of the sensor, in this case 0.5^o, and on the size of terrain features which need to be detected. The terrain classification is first performed in every cell individually. The criteria used for the classification are the height variation of the terrain within the cell, the orientation of the vector normal to the path of terrain contained in the cell, and the presence of a discontinuity of elevation in the cell. To avoid frequent erroneous classification, the first two criteria are evaluated only if the number of points in the cell is large enough. In practice, a minimum of five points per cell is used.

This range image processing algorithm has several important properties. First, it does not build a complete, high-resolution map of the terrain, which would require interpolating between data points, an expensive operation. Instead, the algorithm evaluates only the terrain for which there is data. Second, the algorithm processes each image individually without explicitly merging terrain data from consecutive images. Instead, it relegates the task of maintaining a local map of untraversable regions to a separate local map module. The importance of this is that the local map module deals only with a few data items, the cells classified as untraversable, instead of with raw terrain data. As a result, maintaining the local map is simpler and more efficient. Because of these two features, range image processing is typically on the order of 200ms on a conventional Sparc II workstation.

It is clear that the range image processing module may miss untraversable regions of the terrain because the terrain is evaluated only where data is present in the image and because the data may be too sparse to provide complete coverage of the terrain at long range. However, because of the processing speed, a region that is missed in a given image will become visible in subsequent images quickly enough for the vehicle to take appropriate action. Although this problem effectively reduces the maximum detection range of the perception system, we argue that the other possible solutions would reduce the maximum range even further and would introduce additional problems.

The most obvious alternative is to merge data from a few images before committing to a terrain classification. This solution effectively reduces the maximum detection range because the system has to wait until enough overlapping images are taken before a terrain region is evaluated. In addition, merging images is in itself a difficult problem because it requires precise knowledge of the transformation between images. In particular, even a small error in rotation angles between two images may introduce enough discrepancy between the corresponding elevation terrain maps to create artificial obstacles at the interface between the two maps. (We refer the reader to [5] for a more quantitative description of this problem.) Therefore, it is preferable to not merge images explicitly and to rely on fast processing to compensate for the sparsity of the data. In practice, 30cm terrain features are detected at a range of ten meters from the sensor when vehicle speed is on average 2m/s. By comparison, the maximum range of the scanner is 18 meters with a distance between pixels of about one meter at that range.

It would be advantageous to vary the field of view of the range sensor dynamically as a function of speed and turning radius. Our range sensor does not have this capability at this time so the parameters are tuned for a nominal maximum speed instead of being adjusted dynamically.

5 Local Map Management

The purpose of the local map module is to maintain a list of the untraversable cells in a region around the vehicle. In the current system, the local map module is a general purpose module called Ganesha [7]. In this system, the active map extends from 0 to 20 meters in front of the vehicle and 10 meters on both sides. This module is general purpose in that it can take input from an arbitrary number of sensor modules and it does not have any knowledge of the algorithms used in the sensor processing modules.

The core of Ganesha is a single loop in which the module first gets obstacle cells from the perception modules, and then places them in the local map using the position of the vehicle at the time the sensor data was processed (Figure 8). The sensing position has to be used in this last step because of the latency between the time a new image is taken, and the time the corresponding cells are received by the map module, typically on the order of 600ms.

At the end of each loop, the current position of the vehicle is read and the coordinates of all the cells in the map with respect to the vehicle are recomputed. Cells that fall outside the bounds of the map are discarded. Finally, Ganesha sends the list of currently active cells in its map to the planning system whenever the information is requested.

Because the map module deals only with a small number of terrain cells instead of with a complete model, the map update is rapid. In practice, the update rate can be as fast as 50 ms on a SparcII workstation. Because of the fast update rate, this approach is very effective in maintaining an up-to-date local map at all times.

One advantage of Ganesha's design is that it does not need to know the details of the sensing part of the system because it uses only information from early terrain classification. In fact, the only sensor-specific information known to the map module is the sensor's field of view which is used for checking for consistency of terrain cells between images as described below.

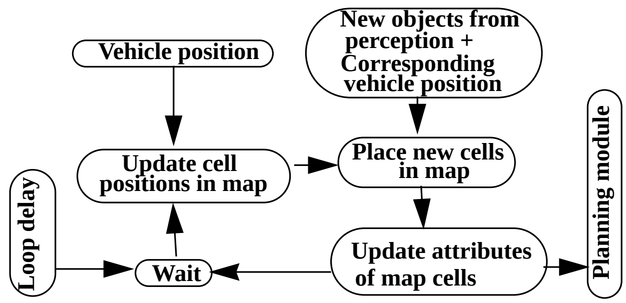

Figure 8: Main loop in Ganesha.

In this design of the navigation system, the local map and planning modules do not have access to the original sensor data and therefore cannot correct possible errors in the output of the perception. In particular, a region which is mistakenly classified as traversable will never be reclassified because the local map module cannot go back to the original data to verify the status of the region. It is therefore important to use conservative values for the detection parameters in order to ensure that all the untraversable regions of the terrain are classified as such.

The drawback of this approach is that the perception module may generate terrain regions which are incorrectly classified. Because the perception processes images individually without explicitly building maps, it cannot detect that the classification is inconsistent with previous observations. This problem is solved by the map maintainer which does maintain a history of the observations. Specifically, an untraversable map cell which is not consistent across images is discarded from the local map if it is not reported by the perception module as untraversable in the next overlapping images. Because the terrain classification is fast compared to the speed of the vehicle, many overlapping images are taken during a relatively short interval of distance travelled. As a result, an erroneous cell is deleted before the vehicle starts altering its path significantly to avoid it.

6 Path Planning

The third component of the system is the Distributed Architecture for Mobile Navigation (DAMN), which generates commanded steering radius and velocity with a high update rate. DAMN (Figure 9) is a behavior-based architecture [1]. In contrast to more traditional centralized AI planners that build a world model and plan an optimal path through it, a behavior-based architecture consists of specialized task-achieving modules that operate independently and are responsible for only a very narrow portion of vehicle control, thus avoiding the need for sensor fusion. A distributed architecture has several advantages over a centralized one, including greater reactivity, flexibility, and robustness [12]. The figure below shows the organization of the DAMN system in which individual behaviors such as seek goal and obstacle avoidance vote for or against a set of possible steering commands. A command arbitration module periodically computes a weighted sum of all the votes from each behavior and issues a command to the vehicle controller corresponding to the highest summed vote.

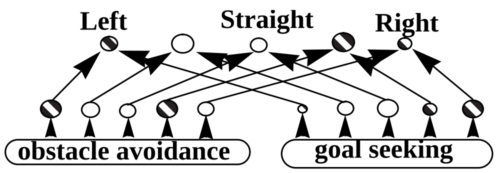

Figure 9: Architecture of the DAMN planning system.

The current implementation uses both an obstacle avoidance behavior and a goal seeking behavior. The obstacle avoidance behavior computes the votes based on the distribution of untraversable terrain cells around the vehicle as reported by the local map module. Arcs that steer the vehicle away from the untraversable regions are given a high vote, while arcs that would cause the vehicle to travel through a forbidden region are given a low vote. Every arc is evaluated by checking the intersection of each obstacle cell with an expanded version of the vehicle along the arc. A safety distance of 0.3 m around the vehicle is used in the evaluation. If an arc intersects a cell at a distance less than a pre-set minimum distance then the arc is assigned a vote of -1; if an arc intersects a cell at a distance greater than a maximum distance, the vote is set to 1. In between the min and max distances, the vote is set to an intermediate value in a continuous manner.

The second behavior, goal seeking, gives higher votes to arcs that steer the vehicle toward intermediate goal points. This second behavior ensures that the overall path of the vehicle follows the desired global trajectory. The votes are between -1 and +1. A set of 15 possible arcs is used in the experiment reported in Section 3. The last module of the trajectory planner is an arbiter which combines the votes from the two behaviors and sends the arc with the highest vote to the vehicle controller.

This implementation of goal seeking is the simplest one that would allow us to demonstrate the system over a significant distance. More sophisticated modules, e.g., the global route planner D* [14], can be integrated in the system. The only requirement is that the higher-level planning modules be able to generates arrays of votes of the format required by DAMN at regular interval.

Because the trajectory planner generates only local arcs based on compact local information, the obstacle cells, it has a high update rate and allows for rapid correction of small errors due to system delays or isolated perception errors.

To illustrate the operation of the arc generation system, Figure 11 shows the distribution of votes at five points of interest along the path, indicated by capital letters in Figure 10. Each graph depicts the votes issued for each turn choice by the obstacle avoidance and heading behaviors, as well as the weighted sum of these votes as computed by the arbiter. The horizontal axis of each graph shows the possible turn radius choices. There are 15 possible turn radii in this example ranging from -8 to +8 meters.

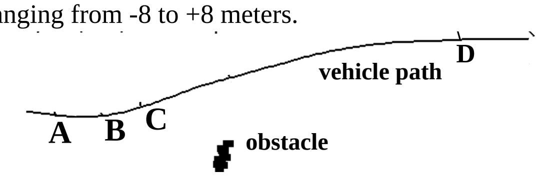

Figure 10: Example vehicle path around an obstacle.

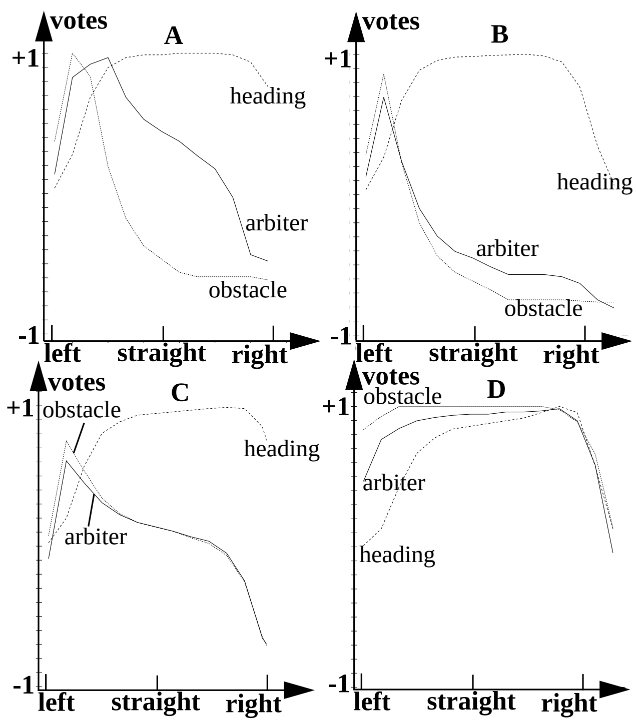

Figure 11: Distribution of votes in the arbiter (thick line), obstacle avoidance (thin line) and heading (dotted line) behaviors.

At point *A*, the obstacle is first reported and the obstacle avoidance behavior generates high votes for the left turns to go around the obstacle and inhibits the right turns with negative votes. At the same time, the heading behavior's vote distribution is relatively flat around the straight direction since the vehicle is currently headed in the desired direction. Because of the small relative weight of the heading behavior, the combined vote distribution in the arbiter, shown as a thicker solid line in the graph, is dominated by the votes received from the obstacle avoidance behavior; a left turn is therefore commanded.

At point *B*, the obstacle is still close to the vehicle and the votes distributions are similar to the ones at *A*, thus keeping the vehicle to the left of the obstacle. At point *C*, the obstacle avoidance behavior is still voting in favor of a sharp left turn, but the votes for the softer left turns are now not as low as they were at *A* or *B*, since the vehicle is now clearing the obstacles. At the same time, the heading behavior is starting to shift its votes towards turning right in order to bring the vehicle back to the target heading. The summed votes are still at a maximum for a left turn because of the greater weight of the obstacle avoidance behavior's votes.

Finally, at point *D*, the vehicle has completely cleared the obstacles and they are no longer in the local map, so that the votes from the obstacle avoidance behavior are mostly +1. The heading behavior now dominates and a right turn is commanded.

7 System Performance and Future Work

In summary, early evaluation of terrain traversability allows us to achieve continuous motion at moderate speeds by: reducing the amount of computation required by the perception system; simplifying local map management and path planning; hiding the details of sensing from all the modules except perception; and avoiding the problems caused by merging multiple terrain maps using inaccurate position estimates. The drawback of this approach is that an error in the perception system cannot be corrected later in the system because only the perception module has access to the sensor data. This problem is eliminated by using a fast reactive path planner, an error-correction algorithm in the local map manager, and a simple perception algorithm with fast cycle time relative to vehicle speed, both of which allow the system to correct quickly for occasional perception errors.

In the current system, the vehicle can tolerate terrain discontinuities of 20cm at a speed of 3 m/s. With a range resolution of 7cm and an angular accuracy of 0.5^o, such a discontinuity can be detected in time to avoid it with an arc of radius greater than the minimum turning radius of 7.5 m, assuming a 2Hz image acquisition rate and an additional 0.5 seconds latency in the system. Although recent hardware improvements have raised the maximum range to 40m. The acquisition rate, maximum range, and resolution of the sensor are the numbers that set hard limits on the speed.

Another limitation of the system is the image-by-image nature of the perception module. Specifically, objects are reported to the appropriate behavior only after the entire image is processed. As a result, an object close to the vehicle is reported at the same time as objects at the maximum distance in the image. We have implemented a different version of the perception module which processes the range data scanline per scanline and reports new obstacles every time a new scanline is processed. We describe such an approach in a separate paper in these proceedings [3].

The other major factor in overall system performance is the distributed and asynchronous nature of the system. In particular, the message traffic between perception, Ganesha, and the behaviors may become heavy enough to introduce significant delays in the system. This delay, on the order of 0.5 seconds, is small enough to permit reliable navigation at relatively low speeds. However, the situation worsens when a more efficient scanline-by-scanline processing of the range data is used, to the point where the system cannot keep up with the volume of data shipped between modules. A solution is to combine the range processing, the local map, and the corresponding behavior into a single module. In doing this, we retain the basic functionality of early traversability evaluation and of reactive planning while eliminating the network communication problems. Preliminary results on combining line processing and single module operation are also reported in [3].

References

[1] R. Brooks and A. Flynn. Fast, Cheap, and Out of Control: A Robot Invasion of the Solar System. *J. British Interplanetary Society*, 42(10):478-485, 1989.

[2] G. Giralt and L. Boissier. The French Planetary Rover VAP: Concept and Current Developments. In *Proc. IEEE Intl. Workshop on Intelligent Robots and Systems*, pp. 1391-1398, Raleigh, 1992.

[3] M. Hebert. Pixel-Based Range Image Processing for Autonomous Driving. *Proc. IEEE Robotics and Automation*. May 1994.

[4] M. Hebert, E. Krotkov. 3D Measurements from Imaging Laser Radars. *Image and Vision Computing 10(3)*, April 1992.

[5] A. Kelly, T. Stentz, M. Hebert. Terrain Map Building for Fast Navigation on Rugged Outdoor Terrain. In *Proc. of the SPIE Conference on Mobile Robots*, 1992.

[6] I.S. Kweon. *Modeling Rugged Terrain by Mobile Robots with Multiple Sensors.* Ph.D. thesis, Robotics Institute, Carnegie Mellon University, January, 1991.

[7] D. Langer and C. Thorpe. Sonar-Based Outdoor Vehicle Navigation and Collision Avoidance. In *Proc. IROS '92*. 1992.

[8] D. Langer, J.K. Rosenblatt, M. Hebert, "Off-Road Navigation", submitted to Special Issue of *IEEE Transactions on Robotics and Automation*, 1993.

[9] L. H. Matthies. Stereo Vision for Planetary Rovers: Stochastic Modeling to Near Real-Time Implementation. *International Journal of Computer Vision*, 8:1, 1992.

[10] E. Mettala. Reconnaissance, Surveillance and Target Acquisition Research for the Unmanned Ground Vehicle Program. In *Proc. Image Understanding Workshop*, Wash., D.C., 1993.

[11] K. Olin, D.Y. Tseng. "Autonomous Cross-Country Navigation". *IEEE Expert*, 6(4), August 1991.

[12] D.W. Payton, J.K. Rosenblatt, D.M. Keirsey. Plan Guided Reaction. *IEEE Transactions on Systems Man and Cybernetics*, 20(6), pp. 1370-1382, 1990.

[13] J.K. Rosenblatt and D.W. Payton. A Fine-Grained Alternative to the Subsumption Architecture for Mobile Robot Control. in *Proc. of the IEEE/INNS International Joint Conference on Neural Networks*, Washington DC, vol. 2, pp. 317-324, 1989.

[14] T. Stentz. Optimal and Efficient Path Planning for Partially-Known Environments. *Proc. IEEE Robotics and Automation*. May 1994.

[15] T. Van Bogaert, P. Lemoine, F. Vacherand, S. Do. "Obstacle Avoidance in Panorama Esprit II Project". *Proc IFAC Workshop on Intelligent Autonomous Vehicles*. Southampton, 1993.