

A REACTIVE SYSTEM FOR OPEN TERRAIN NAVIGATION: PERFORMANCE AND LIMITATIONS

N94-30566

D. Langer, J. Rosenblatt, M. Hebert

The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh PA 15213

207663
P-7

Abstract

We describe 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 is to use the concept of "early traversability evaluation," and on the use of reactive planning for generating commands to drive the vehicle. 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: France's VAP project which is also based on stereo vision [2]; 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 approaches used in the existing planning systems range from purely reactive to fully proactive, depending on the type of maps. 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. We start by giving an overview of the approach and of the system architecture in Section 2; we then describe the performance of the system in an actual experiment in Section 3. We focus on the individual components of the system in Sections 4 to 6. More detailed descriptions of the components may be found in [5] for the local map module, [12] for the planning component, and in [8] for the complete system description.

2 Early Evaluation of Traversability: Overview

The perception and navigation system was developed as part of the Unmanned Ground Vehicle (UGV) project. The support vehicle is a retrofitted HMWVV suitable for cross-country navigation (Figure 1). 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. 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.

Because of the speed requirement, the perception system must update the local terrain map fast enough to keep up with vehicle motion. For that reason, it is impractical to

Copyright 1993 by the American Institute of Aeronautics and Astronautics, Inc. All rights Reserved.

build a detailed, high-resolution terrain map every time a new image is taken. Moreover, an accurate map is not necessary because the vehicle can safely tolerate relatively large variation of terrain surface. For these reasons, we used in this example an approach based on "early evaluation of traversability" in which the output of the perception system is a set of untraversable terrain regions which is used by a planning module to drive the vehicle. Untraversable regions are terrain features such as high slopes, ditches, or tall objects which would endanger 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. This has the advantage of reducing the amount of data passed to the planner for path generation and reducing the amount of computation needed in later stages of planning.

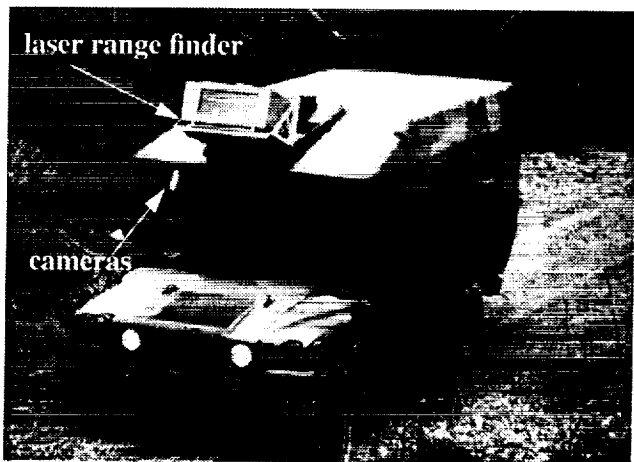


Figure 1: The testbed vehicle.

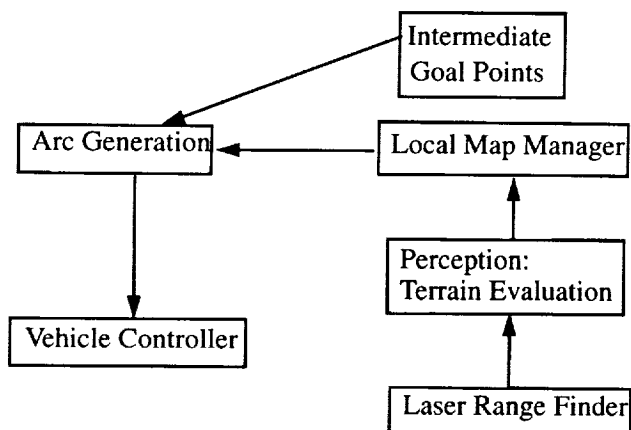


Figure 2: Architecture of the navigation system.

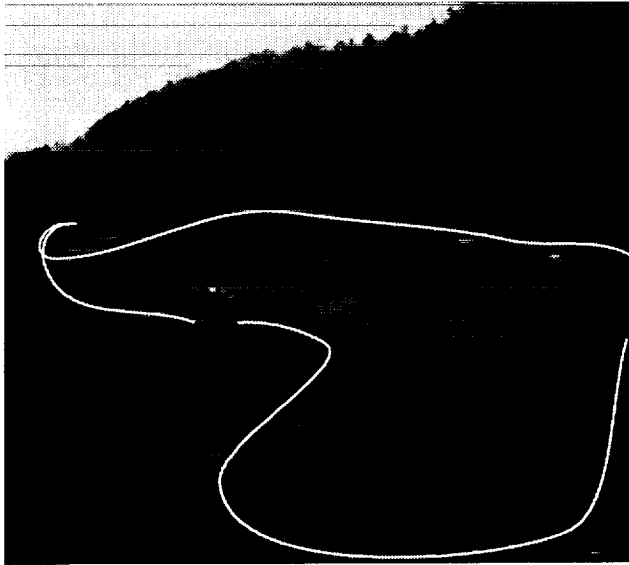
Figure 2 summarizes the system developed based on the idea of "early traversability". The perception component of the system consists of a terrain evaluation module which takes images from a range scanner and outputs untraversable regions to a local map manager. The local map manager maintains a consistent description of the terrain around the vehicle as it travels and send periodically a description of the untraversable regions in the vicinity of the vehicle to

an arc generation module. The arc generation module rates each arc out of a finite set of arcs between forbidden if the arc hits an obstacle and clear if the arc does not pass close to any obstacle region. The arc generation module generates traversability votes for each of the arcs rather than the best arc to follow next. This permits the combination of these votes with votes from other modules. For example, we have used a goal-seeking module which steers the vehicle toward the next goal point. In practice, any navigation module could be substituted to the goal-seeking module.

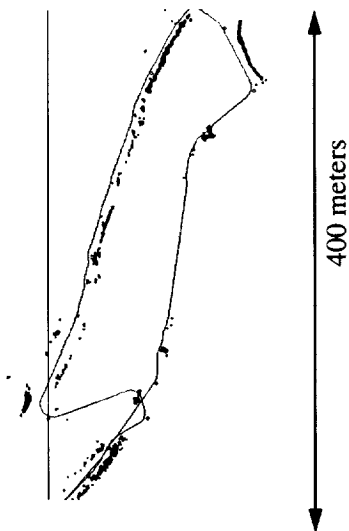
3 System Operation: A Typical Mission

Figure 3 and Figure 4 show a typical run of the perception and navigation system. Figure 3 (a) 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 3 (b) shows the actual path recorded during the experiment projected on the average ground plane. In addition to the path, Figure 3 (b) 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 4 shows close-ups of three sections of the loop of Figure 3. The black lines show the approximate paths followed by the vehicle in these three sections. Figure 5 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 5 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 6 shows displays of the local map which is maintained at all times around the vehicle. The squares correspond to 40x40 cm patches of terrain classified as untraversable regions or obstacles. These local maps are computed from the positions shown in Figure 4 and Figure 5 by the white arrows. The trajectories are planned using this compact representation rather than the detailed maps of Figure 5.



(a) View of terrain and approximate path.



(b) Exact path of vehicle; the obstacle regions are shown as black dots; the interme-

Figure 3: A Loop through natural terrain.

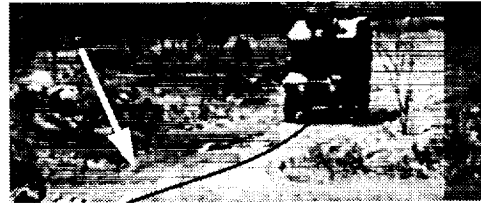


Figure 4: Local path of vehicle in three sections of the loop of Figure 3. The arrows indicate the locations at which the local maps are displayed in Figure 5 below.

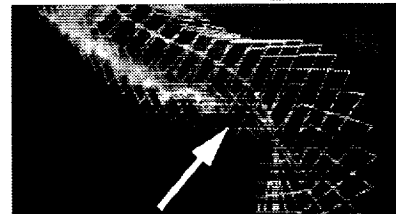
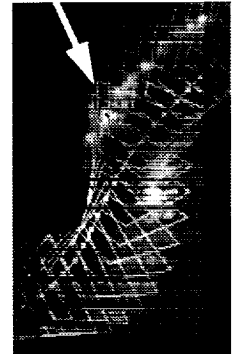
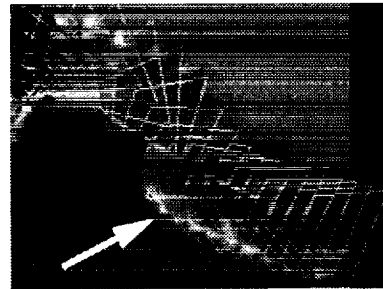


Figure 5: Display of the terrain as elevation maps for the sections shown in Figure 4. The polygons indicate the projection of the field of view of the sensor on the ground. The white line shows the path followed by the vehicle in this section. The white dots show the positions at which the images were taken. The arrows are placed at the same locations as in Figure 4.

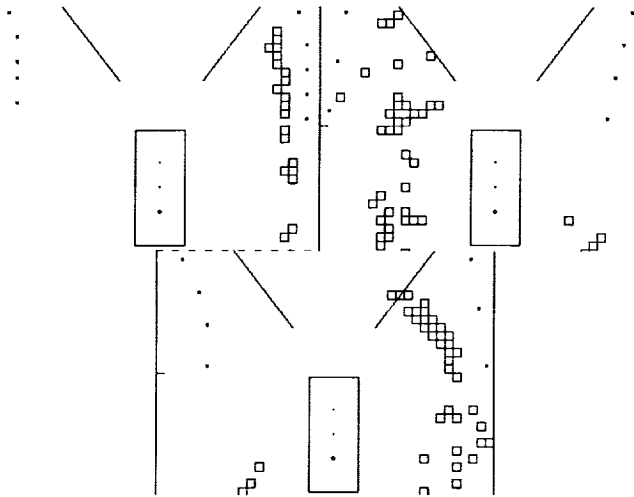


Figure 6: Display of the local traversability map at the locations marked by arrows in Figure 4 and Figure 5. Only the portion of the map in the immediate vicinity of the vehicle is displayed here. The vehicle is depicted by a rectangle. The untraversable regions are shown as squares.

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 coordinate system is defined so that the z axis is vertical with respect to the ground plane, and the y axis is pointing in the direction of travel of the vehicle. It is convenient to center the coordinate at the point used as the origin for vehicle control, in this case between the two rear wheels, rather than at the origin of the sensor. The transformation 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 (x,y,z) coordinates of the points which fall within the bounds of the cell in x and y . The size of a cell in the current system is 20 cm in both x and y . This number depends on the angular resolution of the sensor, in this case 0.5° , and on the size of terrain features which need to be detected. The terrain classification as traversable or untraversable 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. Once individual cells are classified, they are grouped into regions and sent to the local map maintainer.

Figure 7 shows the operation of the perception module in a typical outdoor scene. Figure 7(a) shows a video image of the scene and Figure 7(b) shows the corresponding range image used for evaluating terrain traversability. Figure 7(c) shows the elevation map obtained by converting the range

pixels to a Cartesian coordinate system in which z is approximately the vertical direction with respect to the ground plane. The maximum elevation with respect to the reference plane is one meter in this example. Figure 7(d) shows the result of the traversability evaluation. In this display, the traversable parts of the map are set to 0, the untraversable parts are set to 1. The set of bushes and rocks on the left side of the scene are correctly identified as untraversable. The classification of Figure 7(d) is converted to a list of obstacle patches and sent to the local map manager.

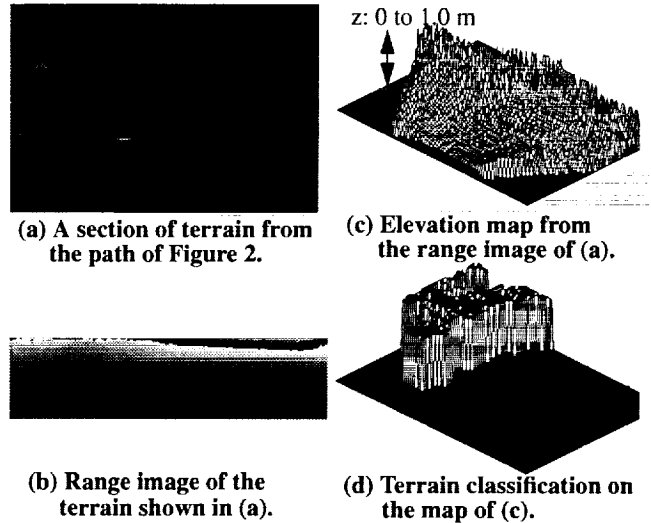


Figure 7: Example of range image processing.

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 as in [7], 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 very fast, typically on the order of 200ms on a conventional Sparc II workstation. The main limitation is the 2 Hz acquisition rate of the sensor, not the processing time.

It is clear 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 so-

lution 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 [6] for a more quantitative description of this problem.) Therefore, unless the vehicle and position estimation systems are designed to produce very accurate pose estimates, it is preferable to not merge images explicitly and to rely on fast processing to compensate for the sparsity of the data.

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, developed by Dirk Langer [5]. 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 (Figure 8) 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 was processed. 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 active region are discarded from the map. 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 last 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.

A different design of the local map module would be to maintain a much larger map with more information than just a list of terrain cells which would theoretically allow the navigation system to use data recorded from earlier images. There are two problems with this approach, however. First, the local map module is now forced to maintain a

much larger amount of data, most of which is never used, introducing additional delays in the system. Second, errors in vehicle position accumulate to a point at which most of the map becomes useless. These two problems offset the occasional gain in additional information in the map.

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. For example, this may occur because of noise in the image or because of an erroneous reading of vehicle pose. Because the perception processes images individually without explicitly building maps, it cannot detect that this erroneous 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.

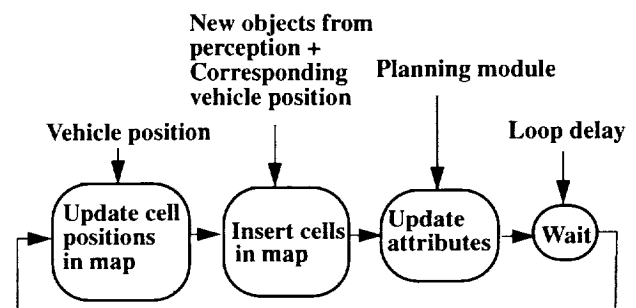


Figure 8: Local map loop.

6 Path Planning

The last piece of the system is a trajectory planner which generates commanded steering radius and velocity with a high update rate. The trajectory planner, developed by Julio Rosenblatt [12][13], is composed of several modules. A set of two behaviors generates votes for every possible arc at the current vehicle position. An 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 low votes. A second behavior 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 last module of the trajectory planner is an arbitrator which combines the votes from the two behaviors and sends the arc with the highest weight to the vehicle controller. Although we describe the architecture for trajectory planning strictly in the context of rugged terrain navigation, the architecture is very general in that it can accommodate a variety of behaviors, it is sensor-independent, and it can implement different strategies for combining weights.

Figure 9 illustrates the operation of the arc generation system. Figure 9(a) shows a display of the local map in the vicinity of the vehicle. The untraversable regions are displayed as before as squares corresponding to 40cm by 40cm terrain patches. Figure 9(b) shows the distribution of votes computed from this local map. The votes are between -1.0 and 1.0. The votes are computed for a list of 39 arcs with turning radii ranging from -8 to +8 meters.

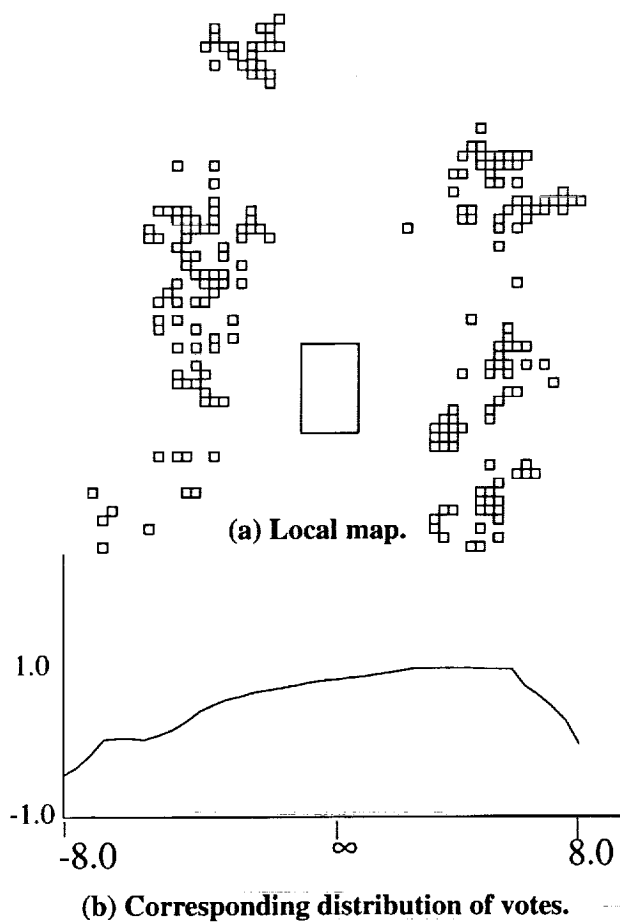


Figure 9: (a) Example of local traversability map; (b) Distribution of votes in this example. The votes between -1.0 (forbidden arc) and 1.0 (clear arc) are computed for 39 arcs with radii between -8 and 8 meters.

The computation of the vote for a particular arc is controlled by three parameters: a maximum and minimum collision distance, and a near miss factor. These parameters are used as follows: Any arc for which the vehicle would collide with an obstacle cell at a distance less than the mini-

mum distance is assigned a vote of -1.0; an arc which does not collide with an obstacle at a distance less than the maximum distance is assigned a vote of 1.0; and any arc which intersect an obstacle cell at an intermediate distance is assigned a negative vote weighted by the distance so that the vote increase as the collision occurs further along the arc. Finally, the near miss factor is used for penalizing the arcs which does not have any direct collisions but which pass close to obstacle cells. The votes decrease as the obstacle cells are closer to the arc.

This algorithm realizes a good compromise between the need to avoid obstacle regions, the need handle near-misses when an arc does not collide with an obstacle in order to take into account the uncertainty in the control system, and the need for limiting the lookahead distance of the planner in order to avoid situations in which the vehicle would be blocked by obstacles that are very far away and therefore do not pose any threat.

Because the trajectory planner generates only local arcs based on compact local information, the obstacle cells, it has a very high update and allows for rapid correction of small errors due to system delays or isolated perception errors. This is in contrast to the trajectory planner alternative in which a sequence of arcs is planned ahead instead of a single steering direction. In this case, trajectory planning is considerably slower and therefore introduces significant latency in the navigation system. A side-effect is that the system cannot recover from an error in the terrain map until it has already started executing a significant portion of the path through this map. This can be avoided by using more precise map building algorithms, but only at the cost of additional latency in the system. We refer the reader to [6] and [3] for a more precise description of the performance and limitations of this type of approach.

7 Conclusion

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 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.

While appropriate in many instances, this approach is not suited for all vehicles. In particular, we have made the assumption that the vehicle can safely negotiate terrain variations which are detectable far enough in advance that the vehicle is able to modify its path appropriately. For example, this vehicle at these speeds can tolerate terrain discontinuities of 20cm. With a range resolution of 7cm and an angular accuracy of 0.5° , such a discontinuity can be detected in time to avoid it with an arc of radius less 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. Sensor acquisition rate and resolution are the two numbers that set hard limits on the speed.

We have described the navigation system as a distributed system composed of three modules. Recently, we have improved our approach by merging all three modules into a single integrated modules. The integrated modules processes range images one scanline at a time, extracting obstacle regions, and maintaining its own local map internally. At regular interval, the module evaluates votes for a fixed set of arcs based on the current local map, much in the same way as the arc generation described in Section 6, and sends the votes to an arbiter which combines them with votes from external modules. This integrated approach allows for better performance by eliminating some of the latency due to the distributed nature of the system, and by ensuring that obstacle regions are reported as soon as they are detected by the perception processing.

Acknowledgments

The UGV project is supported by ARPA under contracts DACA76-89-C-0014 and DAAE07-90-C-R059 and by the National Science Foundation under NSF Contract BCS-9120655. Julio Rosenblatt is supported by a Hughes Research Fellowship. The authors wish to thank Jay Gowdy, Bill Ross, Jim Moody, Jim Frazier and Mike Blackwell for their help in the cross-country navigation experiments.

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]B. Brummit, R. Coulter, A. Stentz. Dynamic Trajectory PLanning for a Cross-Country Navigator. In *Proc. of the SPIE Conference on Mobile Robots*, 1992.
- [4]M. Hebert, E. Krotkov. 3D Measurements from Imaging Laser Radars. *Image and Vision Computing* 10(3), April 1992.
- [5]D. Langer and C. Thorpe. Sonar-Based Outdoor Vehicle Navigation and Collision Avoidance. In *Proc. IROS '92*. 1992.
- [6]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.
- [7]I.S. Kweon. *Modeling Rugged Terrain by Mobile Robots with Multiple Sensors*. Ph.D. thesis, Robotics Institute, Carnegie Mellon University, January, 1991.
- [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*, Washington, 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. Keirse. 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, June 1989