SPEED-RANGE DILEMMAS FOR VISION-BASED NAVIGATION IN UNSTRUCTURED TERRAIN

Pierre Sermanet^{1,2} **Raia Hadsell**^{1} **Jan Ben**^{2}
Ayse Naz Erkan^{1} **Beat Flepp**^{2} **Urs Muller**^{2} **Yann LeCun**^{1}

(1) Courant Institute of Mathematical Sciences,
New York University
(2) Net-Scale Technologies, Morganville, NJ 07751, USA

Abstract:
The performance of vision-based navigation systems for off-road mobile robots depends crucially on the resolution of the camera, the sophistication of the visual processing, the latency between image and sensor capture to actuator control, and the period of the control loop. One particularly important design question is whether one should increase the resolution of the camera images, and the range of the obstacle detection algorithms, at the expense of latency and control loop period. We first report experimental results on the resolution-period trade-off with a stereo vision-based navigation system implemented on the LAGR mobile robot platform. We propose a multi-agent perception and control architecture that combines a sophisticated long-range path detection method operating at high resolution and low frame rate, with a simple stereo-based obstacle detection method operating at low resolution, high frame rate, and low latency. The system combines the advantages of the long-range module for strategic path planning, with the advantages of the short-range module for tactical driving.

Keywords: robotics, vision, autonomous navigation, LAGR, control architecture

1. INTRODUCTION

Vision-based navigation systems in off-road mobile robots often face a dilemma concerning the expenditure of limited resources. Fancy image processing schemes often result in robots with superior long range vision, but these systems yield poor driving performance and frequent collisions due to slow reaction times. Simpler navigation systems benefit from lower latency and a shorter control loop, thus affording nimble driving behavior and efficient avoidance of obstacles. However, the trade-off leads to myopic navigation: the simpler system will drive into dead-ends and miss obvious paths, thus reducing the overall performance. How can the accuracy and range of the visual processing be increased without paying a high cost in speed, reaction time, and flexibility?

We propose a solution to this dilemma that is based on experiments performed with the LAGR (Learning Applied to Ground Robots) mobile robot platform. After confirming, through field experiments, the deleterious effects of increasing the control loop period, we investigate the advantages of using a multi-agent architecture for perception and control. The architecture that we propose combines two navigation modules: a sophisticated long-range obstacle detector, and a simpler, short-range stereo obstacle detector. The long-range module provides information about distant obstacles and pathways, enabling *strategic* path planning, thus avoiding dead-ends and efficiently navigating toward goals. The short-range module operates on a fast frame-rate and performs *tactical* obstacle avoidance.

The short-range module follows a standard approach to vision-based obstacle detection (for similar examples, see Kelly and Stentz (1998); Kriegman et al. (1989); Goldberg et al. (2002)). On each frame, the module constructs a traversability map of the environment using stereo vision. A stereo matching algorithm is applied to images from a pair of stereo cameras, producing a *point-cloud* in which the most visible pixels are given an XYZ position relative to the robot. A traversability map can then be derived using various heuristics, such as counting the number of points that are above the ground plane. Maps from multiple frames are assembled in a global map in which path-finding algorithms are run. The performance of such stereo-based methods is limited, because stereo-based distance estimation is often unreliable above 10 meters (for typical camera configurations and resolutions), and if a very low latency is required, the range is reduced further.

The long-range module must balance the shortcomings of the short-range module. It can afford a higher resolution image and longer latency if it gives valuable information about distant terrain. The experimental results given here were obtained using as a long-range module a stereo module with a higher resolution and longer range than the short-range module. A more sophisticated obstacle detector could be used, but for testing purposes, a higher resolution stereo module was sufficient.

The actual long-range module being developped but not used here uses stereo information to train offline and online classifiers on large windows containing contextual information as well as shape, color and texture, allowing extended vision beyond stereo range, up to 30 meters. This module is described in more details in Hadsell et al. (2007).

Path planning is critical to a goal-driven mobile robot. The use of a multi-agent control architecture allows for both strategic and tactical planning decisions that are merged using a ray-casting planning algorithm. The integration of the short and long range modules was tested in the field. Crashes, avoidance of cul-de-sacs, and overall speed were used to evaluate different configurations of the multi-agent architecture.

2. PREVIOUS WORK

Considerable progress has been made over the last few years in designing autonomous off-road vehicle navigation systems. One direction of research involves mapping the environment from multiple active range sensors and stereo cameras, for example in the work of Krotkov and Hebert (1995) and Matthies et al. (1995). Kriegman et al. (1989) and Thrun (1998) present systems that simultaneously navigate and build maps and classify objects. Estimating the traversability of an environment constitutes an important part of the navigation problem, and solutions have been proposed by many; see Bellutta et al. (2000); Huertas et al. (2005); Pagnot and Grandjea (1995); Rieder et al. (2002); Singh et al. (2000); Vandapel et al. (2004).

Fig. 1. The robot weighs 70kg, measures 1 meter high, and can reach a maximum speed of 1.3 meters/second.

“Layered” architectures, combining multiple control agents or multiple sensory modules, have been designed for autonomous vehicles that need real-time responsiveness plus goal-driven performance. Bayouth et al. (1998) suggests a human-computer layered architecture for highway driving. Beetz (2001) describes a method for integrating perception, planning, and control in a uniform framework and demonstrates the framework with a service robot. Low et al. (2002) presents a method for unified planning and motion control for a mobile robot. By contrast, our research is conducted using only vision-based, off-road robots in order to focus on the difficult problems of image processing in unknown outdoor terrain.

3. THE LAGR VEHICLE: OVERVIEW OF PATH PLANNING AND LOCAL NAVIGATION

This section gives an overview of the full navigation system developed for the LAGR robot. Experiments were run on the LAGR (Learning Applied to Ground Robots) robot platform (see Figure 1). Both the robot and the reference “baseline” software were built by Carnegie Mellon University and the National Robotics Engineering Center. The LAGR platform is composed of two stereo pairs of color cameras, a GPS receiver, an inertia measurement unit (IMU), wheel encoders and a front bumper. Four onboard Linux machines (2Ghz Pentium 4 equivalent with 1Gb of RAM) are connected via a 1Gb ethernet network. One computer is dedicated to path planning, one to low-level functions and two to the stereo pairs.

Although reference “baseline” software was provided, none was used in our system. Our navigation system consists of 5 major components, described below (see Figure 2).

- **Obstacle Detection.** The obstacle detection module(s) use camera inputs to identify traversable and non-traversable regions. The module(s) then populate the vehicle map with the traversability information in the form of cost and confidence values. We used two obstacle detectors for this research. One is a fast, short-range stereo module (FAST-OD), and the other is a slower, long-range vision module (FAR-OD).
- **Vehicle Map.** The vehicle map is a local map in polar coordinates that is fixed relative to the robot position. It is 180 degrees wide and has a

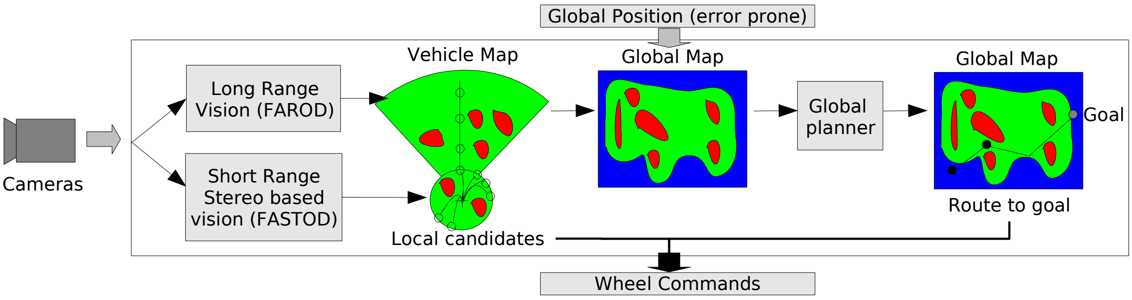

Fig. 2. A flow chart of the full LAGR navigation system. The long-range and stereo obstacle detectors both populate the vehicle map, where local navigation is done. The local map gets written to the global map after every frame, where route planning is done with the global planner.

30 meter radius (actually 15 meters for our experiment). It stores cost and confidence data which is delivered by the various obstacle detectors.

- **Local Navigation.** The local navigation is based on the vehicle map. It determines a set of candidate waypoints based on cost, confidence, and steering requirements. The candidate waypoint is picked which lets the vehicle progress toward the goal. Driving commands are issued based on this choice.
- **Global Map.** The global map is a Cartesian grid map into which cost and confidence information from the vehicle map is copied after each processed frame. The global map is the system's "memory".
- **Global Planner.** The global planner finds a route to the goal in the global map, starting with candidate points proposed by the local navigation module. The algorithm is a modified A* algorithm which operates on rays rather than grid cells.

The traditional approach for autonomous vehicle navigation consists of obstacle detection, map building, path planning, and driver modules. Obstacles are discovered and inserted into the global map, then driving commands are issued based on the populated global map. However, this approach relies on accurate position and orientation information to place the obstacles into the global map correctly and consistently. Unfortunately, the POSE is frequently erroneous because of poor GPS data and wheel slip, causing both jumps and drift in the POSE.

Our system is designed to be less sensitive to POSE errors. As shown in Figure 2 the path planning is broken up into two steps, a local navigation and a global planning step. Local planning is carried out entirely in the vehicle map, based on what the robot sees at that particular time. Because the vehicle map is completely independent of POSE, the POSE errors have no effect on local planning at all.

4. THE SPEED-RANGE DILEMMA

The design of our navigation system separates goal-driven global planning from local navigation, as was described in Section 3. This allows the main navigation control loop to focus on obstacle avoidance and *tactical* driving. It also frees the local navigation system from the errors propagated by faulty POSE data. However, we are still confronted with the speed-range dilemma in the local navigation system. Reducing the control loop period increases the responsiveness of the system, but restricts the range of vision because the resolution of the incoming images must be lowered and the processing algorithm must be simplified.

In order to quantify the relationship between control loop period and system responsiveness, we set up a field experiment that would compare the responsiveness of the system as a function of different control loop periods. The experiment and its results are described in Section 4.1. Motivated by these findings, we designed an experimental, multi-agent control architecture and subjected it to a series of field tests to understand the speed-range dilemma. This architecture is described in Section 4.2. Path planning using this architecture is discussed in Section 4.3.

4.1 Control loop period experiment

These experiments investigated the effect of increasing the control loop period on the responsiveness of the system, as measured by the number of collisions with obstacles on a fixed course. To ensure that the performance of the system was only affected by the control loop period, we use a single, fixed, mid-range obstacle avoidance module. For each test, the control loop period was artificially increased, but the obstacle avoidance processing and the test course was kept fixed.

The LAGR Vehicle was tested in a picnic area at Holmdel Park, NJ. The course had 8 large trees, 2 barrels, and 2 picnic tables, and the goal was 30m away. The start position faced 90 degrees away from the goal and there was a barrel 3m from the start. When a run was started, the barrel would not be seen until the robot turned toward the goal, at which point the robot would have to quickly turn to avoid the barrel. The vehicle was tested using 4 different control loop periods (time from frame timestamp to issuing drive commands): 360ms, 510ms, 660ms, and 960ms.

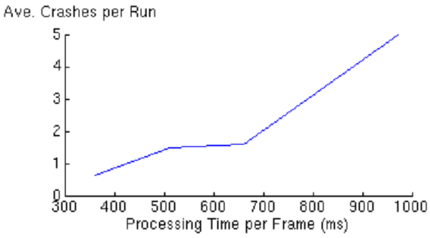

Fig. 3. Experimental results: performance of the system, as measured by the number of collisions per run, for different control loop periods. The performance of the system is roughly linearly correlated with the length of the control loop.

The experiment results are summarized in Figure 3. The shortest loop period performed significantly better than any of the other test modes, particularly in avoiding the first obstacle. The intermediate loop periods produced poorer performance: they usually crashed into the first barrel, but could often navigate around other obstacles (although not as consistently as the shortest loop). The longest period (960ms) produced abysmal driving, repeatedly crashing into obstacles.

4.2 Architectural solution to optimize speed-range dilemma

From the field experiments of previous section, we conclude that an OD module requires a low control loop period and low latency to be responsive to real-time obstacles. On the other hand, long range vision does not require high frequency and low latency. A human walking down a street can look far away infrequently but still easily maintain a global direction, but has to be continuously aware of near surroundings and react quickly to avoid obstacles successfully. This intuitive idea is confirmed by Wagner et al. (1980), who investigated the gaze behavior of walking humans in an outdoor environment. The results indicated that the human gaze was directed at objects close to the observer a majority of the time.

To keep advantages of both long range vision and short-range obstacle avoidance, we hypothesize that those two processes should be independent, each module running at its own pace. Note that this frees the short-range module to use a lower image resolution than traditional vision modules since it only has to be reliable within a very short range. This further reduces the control loop period of the fast module.

The fast object detection module is termed FAST-OD for the remainder of this discussion; i.e. it does not have to see far but must be fast. The long-range module, termed FAR-OD, has to see far but not necessarily fast. Note that when running those processes on the same CPU, the FAST-OD requires a higher priority to make sure it has all CPU cycles available in order to minimize latency. This can be achieved by using priority scheduling features of a real-time operating system or with a regular operating system by giving control of the FAR-OD process to the FAST-OD. When receiving a new frame to process the FAST-OD pauses the FAR-OD process and resumes it when done processing. Then the FAST-OD can pause itself for a fixed time to allow some processing time to FAR-OD. A trade-off must be found depending on the available CPU budget, running speed and vision distance. At the risk of losing reactivity to moving obstacles, one could extend this idea by giving more CPU cycles to FAR-OD when going straight and more to FAST-OD when the robot turns.

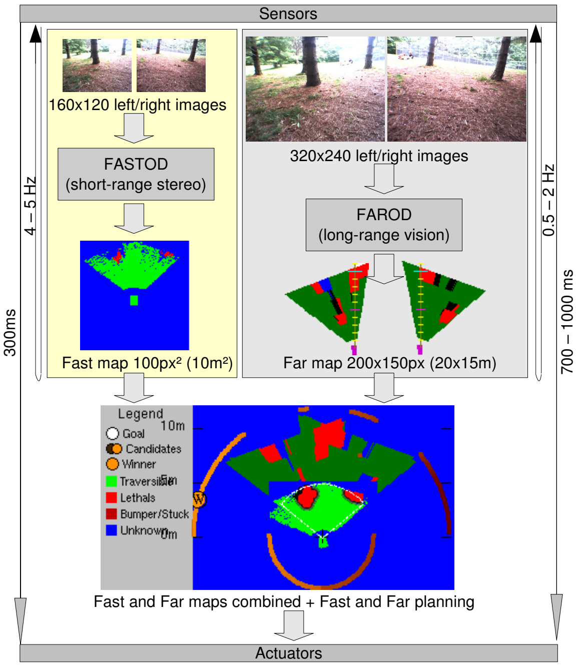

Fig. 4. This figure (best seen in color) shows the chain of input maps top to bottom from image acquisition to wheel actuation for each independent control-loop. Note that since FAST-OD is faster than FAR-OD and the final planning decision is using both modules, it does not wait for FAR-OD to respond but rather uses its last available map. FAST-OD and FAR-OD eventually merge their obstacle maps into a single vehicle-centered map. Now our 2-step planning algorithm takes place. Those candidate waypoints are the orange to dark orange points around the vehicle center (brighter corresponds to lower cost). The chosen candidate is the bright orange “W”, which here allows the vehicle to avoid a tree and follow the global direction.

In our current system, FAST-OD uses stereo input images at a resolution of 160x120 to produce a traversability map with a radius of 5 meters. When run alone on the robot CPU, it reaches 10Hz and a latency from the acquired image to the wheel commands of about 300 ms. The FAR-OD uses a stereo module at input resolution 320x240. When run together with a reasonable CPU trade-off between the modules, FAST-OD has a frequency of about 4Hz and latency of 300ms and FAR-OD has a frequency between 0.5Hz and 1Hz.

4.3 Planning in a multi-agent architecture

As FAST-OD and FAR-OD do not run at the same pace (see Figure 4), their respective maps will be available at different times but planning will be executed

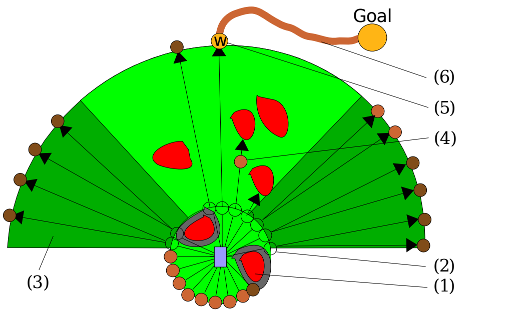

Fig. 5. Path planning in combined FAST-OD/FAR-OD map: (1) obstacle, (2) intermediate candidate, (3) short term memory, (4) candidate between 2 relatively distant obstacles, (5) winning candidate, (6) global route that influences FAST-OD and FAR-OD candidates.

every time a new frame comes from FAST-OD. Thus the last available FAR-OD has to be shifted and rotated according to the robot's movement since the time it was issued. It can then be combined with the new FAST-OD frame. For each OD, left and right frames are merged together with a short term memory using a running average. The FAST-OD module has to be reliable. It may produce more reliable maps than the FAR-OD. In that case, planning should be done in two steps, first in FAR-OD where the best direction is determined and only then in FAST-OD using FAR-OD information to influence the decision. This way the influence of the FAR-OD can be modulated based on its reliability. Similarly to our global/local planning separation (see Section 3), where what we see prevails over the global map content because what we see directly is more reliable, our local planning is itself split and FAST-OD is given the final say.

Figure 5 shows how this two-step planning is done. In the FAST-OD range (5m), rays are shot in each potential direction and candidate waypoints are placed and given a cost based on several parameters (proximity to an obstacle, distance to the global route, heading, etc). Then the ray casting algorithm is ran again in the FAR-OD map from the candidates previously found in FAST-OD. The lowest cost candidate is selected and appropriate driving commands are issued.

Another advantage of separating the planning process is that we can easily use different planning algorithms for each map. For example planning is currently done by casting rays in each direction and stopping when an obstacle is encountered. An on-going project integrates the vehicle's dynamics in the planner by casting curves instead of rays in the FAST-OD map. This algorithm does not need to be used at more than 5 meters radius with our robot, thus we can keep this high dimensional algorithm small enough to be efficient and keep the ray casting for FAR-OD at the same time.

5. RESULTS

In order to show the gains of our architecture, the following test was designed (see Figure 6):

- To test reactivity, the robot starts a short distance from an obstacle and facing 90 degrees away from it. The goal is also 90 degrees away, so that the robot will turn right into the obstacle and avoid it or not depending on how fast it reacts to new obstacles.
- To test long range vision, a cul-de-sac of haybales deeper than the FAST-OD radius is placed after the first obstacle and on the straight line to the goal. This way, a robot with no long-range or low frequency long-range vision will enter the cul-de-sac.

The system to perform best in this test will combine fast reactivity and long-range vision. Three different modes are tested:

- FAST-OD only, fast reactivity but no long-range vision.
- FAST-OD+FAR-OD in parallel, moderate reactivity and moderate long-range reactivity.
- FAST-OD+FAR-OD in series, poor reactivity but good long-range reactivity.

The result table in Figure 7 indeed shows that FAST-OD alone performed very well in reactivity but not in long-range vision, and FAST-OD+FAR-OD in series performed well in long-range but not in reactivity. The FAST-OD+FAR-OD in parallel combination did not individually perform as well on each task but overall provided the best compromise between them.

Note that a reasonable balance between each module must be found to insure good results. One cannot, for example, let the FAR-OD control loop latency and period be too high to keep good long-range reactivity. Depending on the speed of the vehicle, the processing time and maximum distance of each vision module, a trade-off must be found. One could also use more than two layers of modules. For example, a mid-range vision module with a slow stereo system could be placed in between FAR-OD and FAST-OD, thus allowing even more computation time for FAR-OD.

6. SUMMARY AND FURTHER WORK

This paper describes an architecture that effectively allows reactivity and long-range vision at the same time, suitable for outdoor vision-based mobile robots. The simple but non-trivial hypothesis of using a fast, low-resolution stereo module for short-range vision, running independently of a slow and sophisticated long-range vision module, appears necessary to build a robust navigation system.

The architecture used proved to be effective, rather easy to implement and allowed more computation and consequently more sophistication for a long-range object detection module.

The robot's driving performance was improved by adding a low resolution but fast stereo in parallel to our higher resolution and slow long-range vision module.

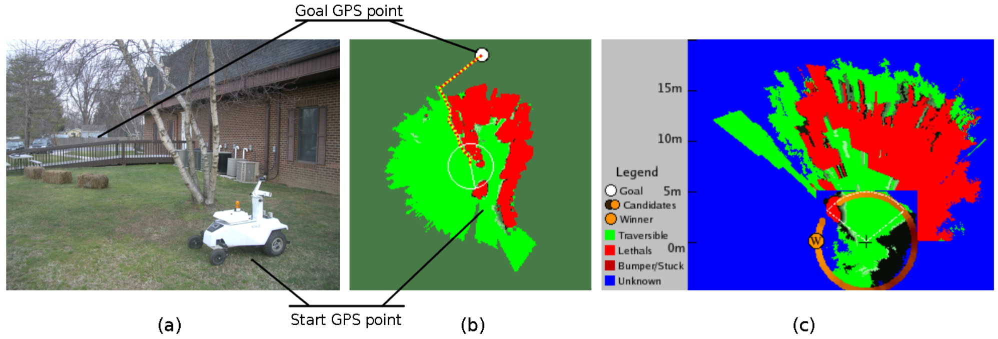

Fig. 6. Test description. *(a)* Site of the field test. *(b)* Global map of the test produced by the robot. A tree serves as the responsiveness test and a hay-bale cul-de-sac as long-range vision test. *Note (1):* a mid-range stereo was used as FAR-OD instead of our long-range module which is still under development. *Note (2):* distances appear bigger than reality in our map due to calibration issues. Our architecture however is rather robust to this imprecision. *(c)* Vehicle map produced by the robot at the entrance of the cul-de-sac. The vehicle map combines FAST-OD and FAR-OD maps, thus we can see that, thanks to the FAR-OD, the winning candidate "W" of the FAST-OD map is chosen to avoid the cul-de-sac.

Test Mode	**Runs**	**Frequency**	**Latency**	**Total Time**	**Crashes /Run**	**Into Cul- -de-sac/Run**
(1) FAST-OD only	5	FAST-OD: 10Hz	280ms	**26.2s**	0	1
(2) FAST-OD+FAR-OD in parallel	5	FAST-OD: 4-5Hz FAR-OD: 2Hz	300ms	**19s**	0.2	0.2
(3) FAST-OD+FAR-OD in series	5	FAST-OD +FAR-OD:3Hz	650ms	**32.1s**	1.4	0

Fig. 7. Results of test. Each configuration was tested for 5 runs and an average was taken for each metric and configuration. *(1)* had the highest frequency and lowest latency and never crashed but consistently went into the cul-de-sac. *(2)* had low frequencies and good latency and crashed once in an obstacle. This crash was probably because FAST-OD was running at a lower frequency compared to (1). It went once into the cul-de-sac and this is probably due to a bit lower frequency in FAR-OD. This shows the limit of the low frequencies of FAST-OD and FAR-OD and that a trade-off must be found. (2) reached the goal the fastest by having fewer crashes and going less into the cul-de-sac. *(3)* had a moderate frequency and low latency and many crashes were observed. However it consistently avoided the cul-de-sac. Note: latency is determined by sensors to actuators time and is not necessarly a factor of frequency, for example on the LAGR platform, the eye machines can run at 4Hz but frame latency be more than 250ms because it includes time of sending data to planner machine and planning. Thus using FAR-OD only (4Hz) would give a good frequency but poor latency, resulting in poor performance; however FAST-OD at 4Hz does maintain a good latency: latency and frequency have to be maintained together.

However, lower control-loop latency and period are not enough to obtain the best driving when ignoring the vehicle's dynamics. Modifying our short-range planning to take in account the vehicle's inertia and capabilities will complete our navigation system.

Acknowledgements

This work was supported in part by DARPA under the LAGR program.

REFERENCES

M Bayouth, I Nourbakhsh, and C Thorpe. A hybrid human-computer autonomous vehicle architecture. May 1998.

Michael Beetz. Structured reactive controllers. *Autonomous Agents and Multi-Agent Systems*, 4(1-2):25–55, 2001.

P Bellutta, R Manduchi, L Matthies, K Owens, and A Rankin. Terrain perception for demo iii. 2000.

S B Goldberg, M Maimone, and L Matthies. Stereo vision and robot navigation software for planetary exploration. March 2002.

R. Hadsell, P. Sermanet, A. Erkan, J. Ben, J. Han, B. Flepp, U. Muller, and Y. LeCun. On-line learning for offroad robots: Using spatial label propagation to learn long-range traversability. In *Proc. of the Robotics Science and Systems Conference 07 (RSS 2007)*, 2007.

A Huertas, L Matthies, and A Rankin. Stereo-based tree traversability analysis for autonomous off-road navigation. 2005.

A. Kelly and A. Stentz. Stereo vision enhancements for low-cost outdoor autonomous vehicles. *Int'l Conf. on Robotics and Automation, Workshop WS-7, Navigation of Outdoor Autonomous Vehicles*, May 1998.

D J Kriegman, E Triendl, and T O Binford. Stereo vision and navigation in buildings for mobile robots. *IEEE Trans. Robotics and Automation*, 5(6):792–803, 1989.

E Krotkov and M Hebert. Mapping and positioning for a prototype lunar rover. pages 2913–2919, May 1995.

K. H. Low, W. K. Leow, and M. H. Ang Jr. Integrated planning and control of mobile robot with self-organizing neural network. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'02)*, volume 4, pages 3870–3875, 2002.

L Matthies, E Gat, R Harrison, B Wilcox, R Volpe, and T Litwin. Mars microrover navigation: Performance evaluation and enhancement. volume 1, pages 433–440, August 1995.

R Pagnot and P Grandjea. Fast cross country navigation on fair terrains. pages 2593 –2598, 1995.

A Rieder, B Southall, G Salgian, R Mandelbaum, H Herman, P Render, and T Stentz. Stereo perception on an off road vehicle. 2002.

S Singh, R Simmons, T Smith, A Stentz, V Verma, A Yahja, and K Schwehr. Recent progress in local and global traversability for planetary rovers. pages 1194–1200, 2000.

S Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, pages 21–71, February 1998.

N Vandapel, D F Huber, A Kapuria, and M Hebert. Natural terrain classification using 3d ladar data. 2004.

M Wagner, J. C. Baird, and W. Barbaresi. The locus of environmental attention. *J. of Environmental Psychology, 1:195-206*, 1980.