FIELD REPORT

Prospect Eleven: Princeton University's Entry in the 2005 DARPA Grand Challenge

Anand R. Atreya, Bryan C. Cattle, Brendan M. Collins, Benjamin Essenburg, Gordon H. Franken, Andrew M. Saxe, Scott N. Schiffres, and Alain L. Kornhauser*
Princeton University
E-407 Engineering Quadrangle
Princeton, New Jersey 08544
e-mail: alaink@princeton.edu

Received 29 January 2006; accepted 21 June 2006

This paper describes Princeton University's approach to the 2005 DARPA Grand Challenge, an off-road race for fully autonomous ground vehicles. The system, Prospect Eleven, takes a simple approach to address the problems posed by the Grand Challenge, including obstacle detection, path planning, and extended operation in harsh environments. Obstacles are detected using stereo vision, and tracked in the time domain to improve accuracy in localization and reduce false positives. The navigation system processes a geometric representation of the world to identify passable regions in the terrain ahead, and the vehicle is controlled to drive through these regions. Performance of the system is evaluated both during the Grand Challenge and in subsequent desert testing. The vehicle completed 9.3 miles of the course on race day, and extensive portions of the 2004 and 2005 Grand Challenge courses in later tests.

1. BACKGROUND

Prospect Eleven was Princeton University's entry in the 2005 DARPA Grand Challenge, a competition for autonomous vehicles, held on October 8, 2005, on an off-road course in the vicinity of Primm, Nevada. The race was organized by the Defense Advanced Research Projects Agency (DARPA) to promote research in autonomous ground vehicles. This was the second instance of the race, the first having been held a year earlier. Princeton University did not participate in the first race, and no entrants completed the course that year.

The Princeton team consisted entirely of undergraduates under the direction of Professor Alain Kornhauser. Among Prospect Eleven's unique features are its inexpensive and simple design and its reliance on stereo vision as its only means of obstacle detection. What follows is a high-level description of

*To whom all correspondence should be addressed.

Published online in Wiley InterScience (www.interscience.wiley.com). • DOI: 10.1002/rob.20141

Prospect Eleven's main systems, the lessons learned while developing these systems, and a discussion of the vehicle's accomplishments to date.

2. MECHANICAL SYSTEMS

The Princeton University team received a stock 2005 GMC Canyon truck to serve as a development platform. The throttle, braking, and steering systems were modified to allow for drive-by-wire operation.

Like many modern cars, the gas pedal in the Canyon is fully electronic so it was not necessary to establish a mechanical linkage to the engine throttle. Instead, a computer-generated voltage simulates the behavior of the physical pedal. The brake pedal is mechanically controlled by two independent systems: A custom-built linear ball-screw actuator used under normal operation, and a pneumatic piston capable of applying 670 N of force for emergency use. These are connected to the brake pedal with a sheathed steel cable, and either system may be in operation without preventing the operation of the other. An inline tension sensor is used to measure the degree of brake application. Steering control is accomplished via a DC motor mounted under the steering column and attached to the steering wheel with a set of gears. An optical rotary encoder, also attached to the steering wheel, provides precise position feedback.

3. VEHICLE CONTROL

The vehicle's steering wheel angle and vehicle speed are maintained with modified Proportional-Integral-Derivative (PID) control loops. During normal autonomous operation, each of the PID controllers runs at approximately 20 Hz.

Steering control uses the optical encoder as input, and controls the steering motor as needed. A two-layer system of PID controllers regulates Prospect Eleven's speed. In the first PID control layer, the reference input is the car's velocity and the output is a throttle voltage if the output is positive; or a desired brake tension, if negative. The desired brake tension is then monitored by the second PID controller. This controller takes the current brake tension as input and controls the braking motor. Figure 1 depicts the speed control block diagram. Figure 2 shows desired and actual velocity during Prospect Eleven's first run at the National Qualification Event (NQE).

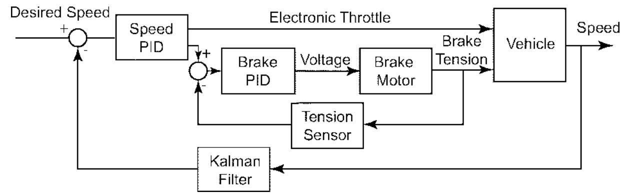

Figure 1. Speed control functional block diagram.

4. COMPUTING

All computing is performed by two standard desktop computers. These are mounted in a shock-isolated rackmount case behind the passenger seat. The rack sits on four fluidic shock mounts designed to attenuate high-frequency vibrations, which can cause hard drive failure. To provide further protection, each computer has a RAID array which mirrors the contents of the primary hard drive.

Vision processing and obstacle detection algorithms were written in C++. All other car control, data acquisition, and decision making control systems were implemented in C# ("C-sharp") on the other computer. Both systems run Microsoft Windows. The C# language proved particularly effective as a development platform; the extensive Microsoft .NET libraries and intuitive object-oriented structure allowed rapid debugging, while permitting the implementation of advanced functionality, such as multithreading and low-level input/output.

5. STEREO VISION

Prospect Eleven relied solely on stereo vision to detect and range obstacles. In doing so, it was unique among the contestants present at the Grand Challenge finals.

Obstacle detection using stereo vision can be broken down into three problems: (1) Obtaining an accurate depth map of the scene ahead from pixel disparities between the two cameras, (2) identifying obstacles in the scene, and (3) calculating the range of detected obstacles so they can be avoided.

This process is susceptible to many environmental sources of error, including unfavorable lighting conditions and irregular terrain. Central to the approach of this paper is the assumption that most of this error is random, and hence can be averaged out

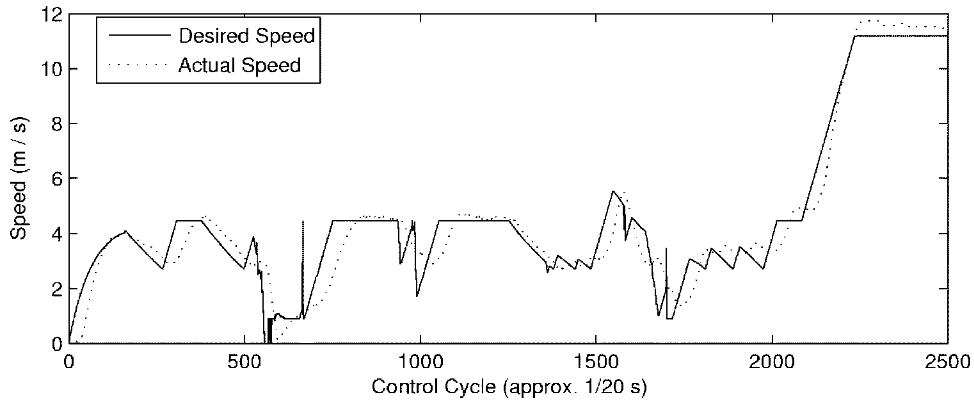

Figure 2. Speed versus time at start of NQE Run 1.

by filtering many measurements of obstacle position over a period of time. This approach suggests simple and fast algorithms, so that many samples may be obtained.

A Point Grey Research (Vancouver, Canada) Bumblebee, a commercially-available stereo camera pair, captures simultaneous images from two black and white charge-coupled devices (CCDs). The baseline separation is 12 cm. The included libraries process images from the camera by comparing each pixel in one image to the corresponding feature in the other image. The difference in pixel location (disparity) is inversely proportional to the pixel's depth in the scene. From this, a depth map is calculated. A depth map is simply an image in which each pixel value corresponds to the disparity value of that pixel in the scene. The included software libraries perform validation of the depth map. In order to simplify the overall algorithm and ensure fast performance, subpixel interpolation is not used. As a result, disparities may only take on integer values. For example, at a range of 20 m, the difference in range between two adjacent disparity values is 3 m.

Several strategies were found to be effective in improving the number of accurate and validated matches—particularly in poor lighting conditions. Red photographic filters mounted in front of each lens were used to increase contrast by blocking blue light and reducing ultraviolet haze, mitigating problems such as CCD "bleeding" on bright days, and boosting the brightness of the ground—the area of interest in each frame. Results were further improved by custom camera gain control which was designed to optimize the exposure in the ground plane at the expense of the upper half of the frame.

5.1. Obstacle Detection

As mentioned, it is important that Prospect Eleven ranges an obstacle many times, such that many measurements may be filtered to increase accuracy. This is only possible with a fast obstacle detection algorithm. This section presents an algorithm which is very fast and well-suited for heavily quantized data. When faced with conditions such as those encountered on the Grand Challenge course, the system performs sufficiently well for obstacle avoidance.

Several authors have also examined the problem of fast obstacle detection. One approach, adopted by Matthies & Grandjean (1994), is to consider the slope of a pixel relative to a ground plane. It is supposed that a significant obstacle will have a slope greater than some threshold. Similarly, Broggi, Caraffi, Fedriga & Grisleri (2005) simply search each column in the depth map for large intervals at similar disparities. Indeed, for a camera mounted nearly parallel to the ground plane, as was the case for Prospect Eleven and Broggi et al. (2005) this approach approximates thresholding vertical slope over some window. The algorithm in this paper parallels that of Broggi et al. (2005).

Were disparity values not so heavily quantized, a logical measure of similarity might be variance. However, as a result of quantization, the algorithm simply looks for a contiguous span within the column, for which the disparity is the same value

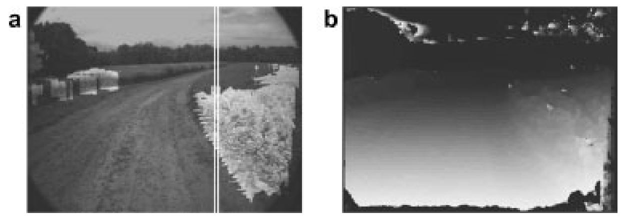

Figure 3. (a) Sample scene image, with detected obstacle pixels highlighted and sample column outlined, as analyzed in Figure 4. (b) The corresponding disparity map.

throughout the span. Such spans occur in flat scenes in the image, so it must be required that they be of at least a certain length, l, to be classified as an obstacle. In contrast to Broggi et al. (2005), this paper's approach is to make l dependent on the extent to which the current interval is above the row-wise median disparity. The justification for this is simple: For a relatively flat scene, the disparity of a pixel which belongs to an obstacle should be above the median disparity in its row. Setting l to be lower for pixels above the median enables information from the entire image to be considered, while detecting obstacles in a single column. We set l to be 12 for pixels, which are above the median by 2; 20 for pixels above the median by 1; and 35 for pixels at the median disparity value. Use of the median may not be effective in unstructured environments, but since the Grand Challenge course was graded, it was assumed that most rows would be roughly homogeneous over the traversable region.

The algorithm can be structured as a finite state machine which scans each column from top to bottom and has states `IN OBSTACLE` and `NOT IN OBSTACLE`. When the state is `NOT IN OBSTACLE`, the code simply looks for an interval satisfying the above criteria. In state `IN OBSTACLE`, it grows the interval downward until it first fails to meet a slightly weaker form of the above criteria. The weaker form differs only in the constants used, and is employed because the base of an obstacle is more similar to the background than the top. Figure 3 shows a scene image, with obstacle pixels highlighted, as well as the corresponding depth map. One column in Figure 3(a) is highlighted. Figure 4 shows depth values and the algorithm's output on this highlighted column.

The time complexity of the algorithm is linear. Each pixel need only be examined once, and median calculation can be performed efficiently using a radix sort. Figure 5 shows the computation times of 193 images at 640×480 resolution versus the proportion of pixels in the image which were identified as obstacles.

Once each pixel is classified as being an obstacle or not an obstacle, bounding boxes are constructed around each connected obstacle region. In doing so, a central point and width are computed for each box. A confidence measure is computed based on the

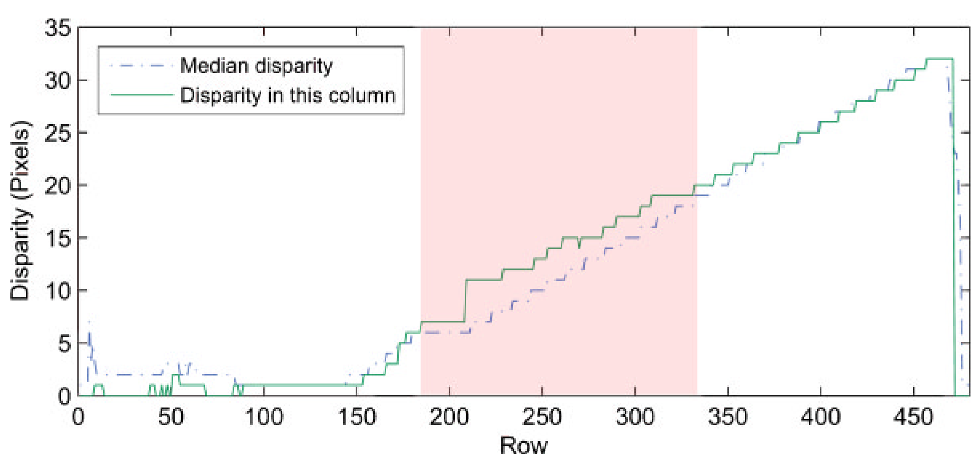

Figure 4. Disparity values and obstacle detection in the highlighted column of Figure 3(a). The shaded region is detected as an obstacle.

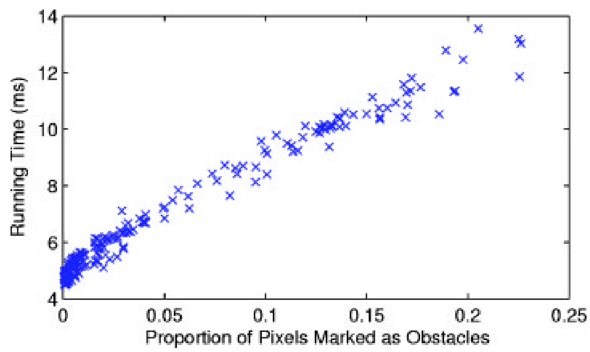

Figure 5. Computation time versus proportion of obstacle pixels over 193 640×480 images.

detected size of the obstacle and the similarity of disparity values within the obstacle. A box is classified as an obstacle if its confidence measure exceeds a threshold. Table I gives the performance of this algorithm on several obstacles at various distances. A ✓ indicates successful detection, and an × indicates that the obstacle was not detected.

As can be seen in Table I, the range at which short obstacles can be reliably detected is quite limited. Fortunately, the primary function of obstacle detection during the Grand Challenge was to detect graded berms on either side of the course. Data recorded during the Grand Challenge indicate that Prospect Eleven was able to do so at ranges of approximately 8 m, which was adequate for navigation during the race. The limited detection distance for small but dangerous obstacles capped the vehicle's maximum speed.

5.2. Tracking in the Time Domain

Detected obstacles are tracked in the time domain to improve accuracy in positioning and limit false positives. When a new image is processed, the list of obstacles from that image is compared to the list of currently tracked obstacles. Each new obstacle is matched to the closest existing one, or declared a new obstacle if no suitable match exists. A confidence measure is maintained for each obstacle, based on the aforementioned confidence of each detection, number of frames in which it was detected, and the number of frames in which it was *not* detected—despite being in detection range. Only obstacles whose confidence measure exceeds a threshold are used in path planning. A Kalman filter maintains the estimate of the obstacle's location.

Matching is effective in improving localization, particularly for obstacles at ranges above 8 m. The measurement error for localization decreases quadratically as a function of range. Though ground-truth data are not available, a direct approach to obstacles at randomized positions is simulated with a wide variety of parameters. Figure 6 gives the mean precision of localization at various ranges over 10,000 simulations. Values simulated include vehicle speeds in the range from 6 m/s to 13 m/s, minimum detection ranges between 1.5 m and 5 m, maximum detection ranges between 15 m and 20 m, detection frequencies between 6 Hz and 10 Hz, and disparity calculations with unbiased Gaussian error with σ^2 between 0.05 pixels and 0.5 pixels. The error model assumes that error is caused entirely by miscalculation and quantization of disparity values. The periodic behavior of the measurement error is a result of quantization: For ranges which correspond to a nearly integer disparity value, quantization causes very little error. Though there are certainly many more sources of error than those modeled, and actual error is much greater, the simulation demonstrates that tracking is effective at reducing error in localization.

Table I. Detection of objects at various ranges (in m).

Object	4.5	6	7.5	9	10.5	12	13.5	15	16.5
Downturned cinderblock (19.5 cm)	✓	✓	✓	×	×	×	×	×	×
Upright cinderblock (40 cm)	✓	✓	✓	✓	✓	×	×	×	×
Shelves (65 cm)	✓	✓	✓	✓	✓	✓	✓	✓	×
Trashcan (69 cm)	✓	✓	✓	✓	✓	✓	✓	✓	×

Note: ✓=Successful detection; ×=Obstacle not detected.

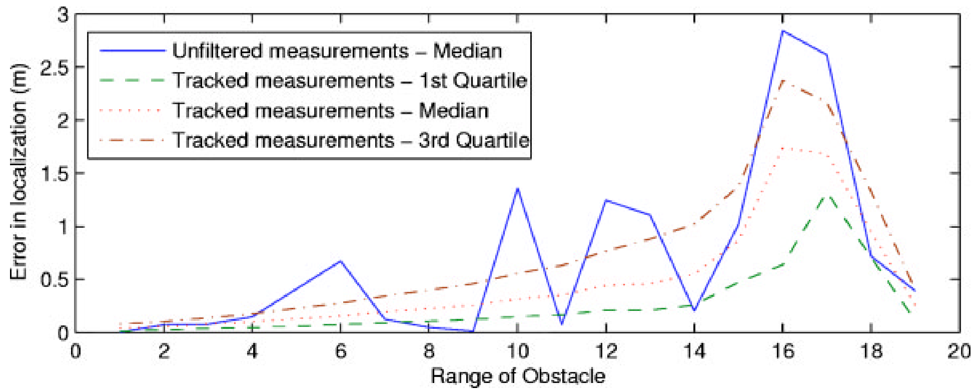

Figure 6. Error in localization versus range in meters.

6. NAVIGATION

The choice of navigation scheme was governed by the specific structure of the competition. DARPA's rules and a review of the 2004 competition suggested the following:

1. DARPA would provide a high-level high-detail global positioning system (GPS) path from start to finish,
2. A traversable path would exist within course boundaries, and
3. The course would be narrow and lie on desert roads.

An algorithm which chooses a steering angle at each instant without preplanning a path ahead is sufficient for the Grand Challenge. Global convergence problems are largely mitigated due to the provided GPS path. This path performs the function of a high-level planned path. For these reasons, the navigation software implements a reactive algorithm that processes a region limited to the detection range of the stereo vision system.

The algorithm implemented is a nearness diagram (ND) approach (Minguez & Montano, 2004) modified to suit the structure of the competition. The ND navigation scheme, as implemented by Minguez and Montano, begins by forming a polar plot of the distance from the vehicle to the nearest obstacle at each angle. This plot is manipulated to identify available gaps in the obstacles surrounding a robot, and a control vector is selected from among five different control strategies based on the particular structure of available openings.

Our implementation differs by: (1) Changing the gap calculation to accommodate road boundaries, (2) changing the gap representation from angular width to physical end points and width in meters, and (3) utilizing the physical dimensions of the gap to collapse ND's five control schemes into one. These modifications adapt the ND approach to travel along roads.

6.1. World Representation

The navigation system maintains an internal model of the world composed of the DARPA-defined GPS course and detected obstacle locations. The model is a Cartesian plane with the origin located at the first course waypoint. Terrain elevation is neglected. Approximating the globe as a plane was found to be sufficient throughout the race. The course representation was geometric, with course segments represented as rectangles capped with semicircles, and obstacles represented as circles of varying diameter and location. Approximating obstacles as circles leads to inaccuracies in representing planar obstacles, such as when walls appear as a string of small circles. However, the nature of the competition lessened the impact of this shortcoming as frequently encountered objects, such as gate posts, tank traps, and bushes, are all well approximated by circles.

This internal representation stands in contrast to cost map approaches. The geometric model requires less memory than large cost maps, and in general can calculate intersections and other quantities analytically. However, a drawback of this method is that

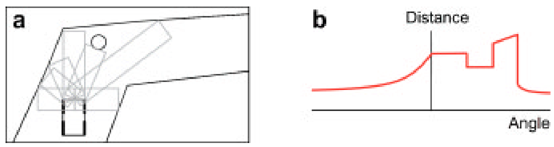

Figure 7. (a) A sample world configuration consisting of GPS course boundaries, a circular obstacle, and the vehicle. Projected tubes are shown in gray. (b) The resulting polar tube plot.

its obstacle representation is binary—space must either be fully traversable or blocked. Since the stereo vision system cannot distinguish between obstacles of different severity, such as rocks which one would prefer to avoid versus a parked car that one must avoid, the binary representation simply reflected a constraint of our stereo vision system, and was not itself a limitation.

6.2. Gap Calculation

The central task of the navigation system is to extract the location of "gaps" in the terrain in front of the vehicle, where gaps are defined as physical openings wide enough for the vehicle to travel through. The first step in locating these is the construction of a polar tube plot, which closely parallels the ND. Whereas the ND graphs distance to the nearest obstacle in an angular sector, the tube plot graphs the distance to the nearest obstacle in a rectangular region (a tube), extending at a given angle off of the car's heading (Figure 7). Using tubes instead of sectors preserves the actual width of gaps at different distances from the vehicle. Next, discontinuities in the plot are identified, and gaps are formed from a pair of adjacent left and right endpoints. Because road borders are continuous lines, no discontinuities are present to form appropriate gaps between road edges and obstacles. Another mechanism was developed to achieve suitable gap formation on roads or in corridors. Additional "corridor" gaps are formed between an unpaired endpoint and a point along the adjacent corridor boundary. The appropriate contact point along the boundary is taken to be the point of intersection of the boundary with a line passing through the unpaired endpoint perpendicular to the vehicle's direction of travel (Figure 8).

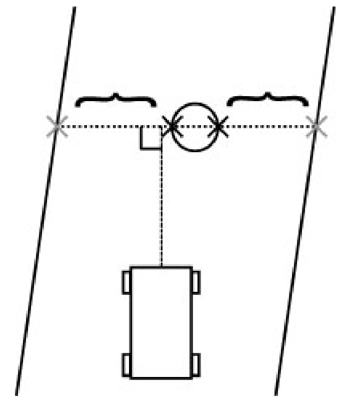

Figure 8. Corridor gap geometry. Black ×'s mark unpaired endpoints; gray ×'s mark corridor endpoints; and braces show the resulting corridor gaps.

6.3. Control Action Selection

A control action comprising desired heading and speed is calculated based on the available gaps. First, a target gap is selected based upon width and angle from the vehicle's heading. Wider gaps are more desirable, as are those which require the least deviation from current heading. Once chosen, a gap is biased to be targeted in subsequent time steps to prevent alternating between gaps. This is a serious problem, because such indecision effectively takes the average between the two gaps, where an obstacle lies. Next, a pursuit point along the course centerline is selected. In the absence of obstacles, aiming at this pursuit point will guide the vehicle along the GPS course. To cause the vehicle to dodge obstacles by a margin of d where possible, the target gap endpoints are each moved toward each other by $d+w/2$, where w is the width of the vehicle. If the gap is too narrow to move each endpoint the required distance, both endpoints are placed at the middle of the gap, so the vehicle will still pass through narrower openings if d is unavailable. The desired heading is calculated to be the heading within the narrowed target gap closest to the heading of the pursuit point. This method allows the vehicle to dodge obstacles by precise distances.

The desired speed is computed as the minimum of three terms: (1) The DARPA mandated speed limit, (2) a safety speed limit based on the average curvature of the track ahead of the car, and (3) a reactive term proportional to the length of the tube

Table II. Results at the National Qualification Event.

Result	Run 1	Run 2	Run 3	Run 4	Run 5
Obstacles avoided (of 5)	5	0	5	2	5
Gates passed (of 50)	48	0	50	8	50
Time	13:03	DNF	12:34	DNF	12:11

Note: DNF=did not finish.

projected from the front of the vehicle. This last term slows the vehicle while dodging obstacles.

7. RESULTS

7.1. Site Visits

To earn an invitation to the NQE in Fontana, California, teams had to demonstrate basic GPS-following and obstacle avoidance capabilities during a site visit by DARPA officials. During the first site visit in May, Prospect Eleven failed to evade several trashcans, and was not originally placed in the top 40 contestants. However, our team did earn a second site visit in June, and performed well enough to earn an invitation to the NQE as one of three alternates.

7.2. National Qualification Event

The NQE consisted of five runs over a 3.5 km course. The course included a 100-ft tunnel under which robots lost GPS fix, rumble strips, four parked cars, a tank trap, hay bales, a simulated mountain pass, and tire stacks. Each run was judged on time, evasion of obstacles, and completion of course gates. Though Prospect Eleven performed admirably on three runs, its poor performance on the other two demonstrated serious software reliability issues. Table II shows the results of the five NQE runs. During Runs 1–3, the vehicle's GPS system was misaligned. Despite this, during Runs 1 and 3, the vision system was able to keep Prospect Eleven within boundaries by detecting physical course markings. In Run 2, the misalignment caused Prospect Eleven to collide with the first obstacle. During Run 4, slow software performance resulted in unstable steering control. This was a result of several extraneous processes left running on the vehicle, as well a systemic software bug, discussed in the next section. The fifth run was essentially perfect and within 2 min of the course record.

7.3. Grand Challenge Event

The top 23 robots from the NQE were invited to participate in the Grand Challenge Event on October 8, 2005 in Primm, Nevada. Prospect Eleven was seeded tenth. The race started smoothly, and Prospect Eleven appeared on schedule at the 8 mile mark. Shortly thereafter, at approximately 9.4 miles, steering control became unstable. The DARPA chase vehicle disabled Prospect Eleven, and Prospect Eleven's race was over.

An analysis of the logs made it clear that the culprit was again slow software performance. Normally, Prospect Eleven's control loops run at 16–20 Hz; however, at the time the vehicle was disabled they were only running at 0.3 Hz. Further analysis revealed that this was caused by a bug in the obstacle tracking code, as obstacles were never entirely cleared from the list of tracked obstacles when passed. Tracking the position of thousands of irrelevant obstacles overwhelmed the processor, and starved critical code.

7.4. Post-Grand Challenge

In order to evaluate Prospect Eleven's performance without this software bug, the course was attempted once more on October 31, 2005. By this point, course conditions had changed considerably from the day of the Grand Challenge: Several formerly dry lake beds had been filled with rain in the intervening weeks. Also, the heavy rains cut a number of deep washouts across the path. In addition, some parts of the course had been altered following the Grand Challenge, with ramps bulldozed, and a short stretch of track deliberately rendered impassable. In all of the above cases, Prospect Eleven was removed from autonomous operation, and manually driven around the impasse. In addition, Prospect Eleven suffered a communications failure between the GPS unit and the guidance computer just before Beer Bottle Pass, a mountain pass near the end of the course, that would have ended a fully autonomous attempt. This was fixed en route. Nevertheless, Prospect Eleven drove the rest of the course autonomously, successfully navigating two tunnels, multiple gates, and descending winding Beer Bottle Pass at night without any intervention.

To assess repeatability, Prospect Eleven ascended and descended Beer Bottle Pass again the following day. Although traversal of the pass itself was uneventful, the vehicle blew out its left front tire at the base of the pass on the descent, following a collision with a small sharp rock. The front wheels were also jarred out of alignment. This failure demonstrates a limit of the vehicle's stereo vision technology, which could not detect small but crucial features of this size. The descent occurred at night. The vehicle headlights provided sufficient illumination for robust obstacle detection.

As a final test, Prospect Eleven attempted the 2004 Grand Challenge course backward, from Primm, NV to Barstow, CA. Again, the vehicle was unable to complete a fully autonomous traversal of the course due to environmental factors. Manual control had to be taken back a number of times to guide the vehicle around washouts and, in one case, to divert the vehicle around an underpass that had filled with silt. Also, three hardware failures would have ended a fully autonomous attempt of the course: A communications cable came loose, the steering position encoder became jammed with sand, and the vehicle's spare tire, installed to replace the old left front tire, was eventually destroyed by the terrain. The failure of the spare was expected due to the misalignment of the wheels after the previous collision. Despite these issues, Prospect Eleven navigated a substantial distance autonomously. The vehicle traversed the three mountain passes of the course without incident, including a descent of Daggett Pass in total darkness.

8. CONCLUSION

Over the course of the Grand Challenge qualification, competition, and subsequent testing, Prospect Eleven performed well. Though the system was designed to exploit the specific structure of the Grand Challenge race, it showed the capability to perform in some environments more complicated than those originally envisioned. This demonstrates the promise of a simple approach to autonomous systems. Though it is useful to consider implementation as a system of interconnected modules, each of which may be individually optimized, Prospect Eleven shows that overall performance relies on carefully considering the integration of components in the system as a whole. For instance, the use of a binary obstacle representation, though in itself suboptimal, contains all information that the stereo vision system can provide and more accurately reflects the limitations of the detector. Prospect Eleven shows the feasibility of a simple autonomous design that can operate effectively in difficult environments.

ACKNOWLEDGMENTS

This work was supported by the CSX Transportation Research Endowment Fund, Eric Huber, Katherine Kornhauser, General Motors, Applanix Corporation, Tom Haines, and ALK Technologies, Inc.

The students would particularly like to thank Professor Alain Kornhauser for his steadfast support of this project.

REFERENCES

Broggi, A., Caraffi, C., Fedriga, R.I., & Grisleri, P. (2005). Obstacle detection with stereo vision for off-road vehicle navigation. Paper presented at the IEEE Workshop on Machine Vision for Intelligent Vehicles, San Diego, CA (pp. 65–72).

Matthies, L., & Grandjean, P. (1994). Stochastic performance modeling and evaluation of obstacle detectability with imaging range sensors. IEEE Transactions on Robotics and Automation, Special Issue on Perception-based Real World Navigation, 10(6), 783–792.

Minguez, J., & Montano, J. (2004). Nearness diagram (ND) navigation: Collision avoidance in troublesome scenarios. IEEE Transactions on Robotics and Automation, 20(1), 45–59.