RASCAL – An Autonomous Ground Vehicle for Desert Driving in the DARPA Grand Challenge 2005

Reinhold Behringer, William Travis, Rob Daily, David Bevly, Wilfried Kubinger, Wolfgang Herzner, Victor Fehlberg

Abstract—**The DARPA Grand Challenge is a competition of autonomous ground vehicles in the Mojave desert, with a prize of $2M for the winner. This event was organized in 2004 and will be held annually at least until 2007, until a team wins the prize. The teams are coming from various background, but the rule that no US government funding or technology that was created with US government funding could be used for this competition, prevented some of the well established players to participate. The team SciAutonics/Auburn-Engineering continues their effort to build a system for participation in this challenge, based on the 2004 entry RASCAL. The main focus in the system design is on improvements of the design from 2004. Novel sensing modalities the team plans to use in 2005, are a stereo vision system and a radar system for obstacle detection. Offline simulation allows to analyze situations in the laboratory and to replay recordings from sensors. The Grand Challenge 2005 will take place on October 8, and the SciAutonics/Auburn team intends to compete with the improved RASCAL system.**

I. Introduction And Context

THE DARPA Grand Challenge (DGC) was organized for the first time in 2004. Fifteen teams competed in a challenge to design, build, and run a ground vehicle over a distance of 142 miles completely autonomously, from Barstow (near Los Angeles) to Primm (near Las Vegas). As observers of this event know, nobody completed the course in 2004, and the prize was not awarded to any team. The vehicle that came furthest was "Sandstorm" from the "Red Team" by CMU – it drove 7.4 miles before getting stuck. At that competition, SciAutonics, LLC, was sponsoring two participating teams. The team SciAutonics-2 had been the result of collaboration with Elbit/Elop who fielded their vehicle AVIDOR and achieved "2nd place" (6.7 miles). The team SciAutonics-1 achieved "rank #7" with the vehicle RASCAL: a temporary hard drive failure disabled the vehicle control system, and RASCAL had to be stopped by DARPA after 0.75 miles [6]. A summary about the DARPA Grand Challenge 2004 and an overview on the participating teams is given in [1], and there are several publications about the teams themselves (e.g. [2][3][4][5]).

The team SciAutonics/Auburn is comprised of volunteer engineers from Thousand Oaks and surroundings (Conejo Valley) who work for Rockwell Scientific and other high-tech enterprises. The team is the result of a collaboration between SciAutonics, LLC, and Auburn University (Auburn, AL). Also collaborating in this team are employees of ESRI and Seibersdorf research (Vienna, Austria).

In 2005, the DARPA Grand Challenge will be held again – on October 8, 2005. The team SciAutonics/Auburn intends to continue to work towards building an autonomous system on the base of the previous work. This paper describes the new approaches and the improvements that the SciAutonics/Auburn team implemented.

II. System Architecture

The emphasis of the system design is on asynchronously coupled components which can continue to run even after partial component failure. Therefore, special emphasis was placed on a design in which components were not too much dependent on each other. Due to constraints and budget, the design does not contain redundant computing systems. The vehicle control module is the most crucial one – it needs to be designed to be very robust. The modules communicate with each other through multicast or UDP. This allows asynchronous communication without explicit connection.

A. System Concept

The system architecture and the different modules are shown in Figure 1. This architecture maps to a heterogeneous cluster of computers, which perform the various module functionalities. In order to have synchronized time available on each processing node, NTP demons are running in the background.

The OS of the RASCAL system in the previous DGC in 2004 had been Windows, but the software has been ported to Linux. There has been some consideration of using a hard real-time OS for the vehicle control system in order to keep the 20 ms control loop, but it has been shown that this is not necessary and that both Windows and Linux were able to keep the timing constraints.

The system relies on a regular series of closely spaced waypoints. Since the waypoints that DARPA provides to denote the course to be driven, are not at a given fixed spacing, the employed approach is to use extensive mapping to generate a set of waypoints that is more closely and uniformly spaced. The waypoint spacing chosen is 1 m.

Manuscript received March 1, 2005.

R. W. Behringer is Program Manager at Rockwell Scientific and President of SciAutonics, LLC, Thousand Oaks, CA 91360 (805-373-4435, fax 805-373-4862, e-mail: rbehringer@rwsc.com).

W. Travis, R. Daily, and David M. Bevly are with Auburn University, Auburn, AL 36849 ({traviwe,dailyrl,dmbevly}@eng.auburn.edu).

W. Kubinger and W. Herzner are with Seibersdorf research, Austrian Research Centers, TechGate Vienna, Donau-City-Str. 1, Vienna 1220 AUSTRIA ({wilfried.kubinger,wolfgang.herzner}@arcs.ac.at).

V. Fehlberg is systems engineer at Amgen Corporation, Thousand Oaks, CA 91360 (fehlberg@amgen.com).

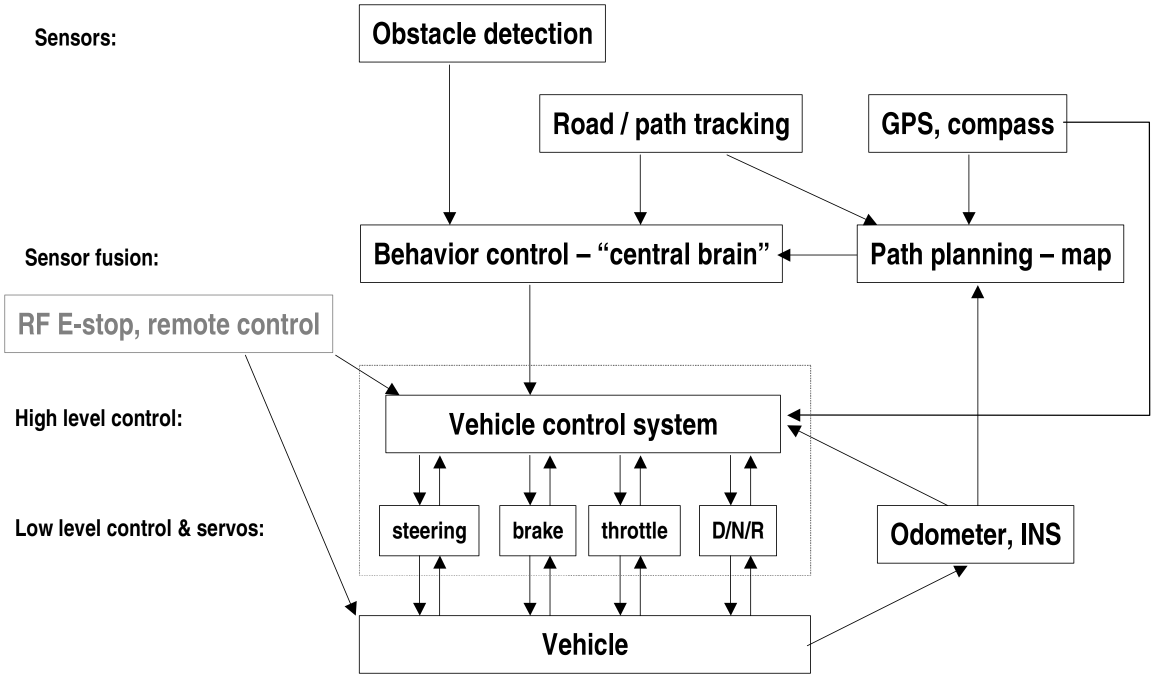

Figure 1. System architecture of RASCAL system for autonomous driving.

B. Data Processing

Position updates from the GPS – a Navcom Starfire – are processed at 5 Hz. A Rockwell Collins inertial measurement unit GIC-100 provides yaw rate at 50 Hz. A Kalman filter is used to smooth the location data and to bridge GPS outages [6]. The vehicle control system is then computing the actuating values for keeping the vehicle on the course towards the next waypoint.

The sensor suite produces "auxiliary, optional" data. In case these sensors or their processing units fail, RASCAL continues to follow the waypoints. If the failure of these sensors or their modules is detected, the speed of the vehicle is reduced to minimize the impact of a potential collision.

If the sensors return measurements of objects within their measurement range, they create feature descriptors, translating their direct measurements into spatial descriptions of either "safe" areas or "dangerous" areas. These features are placed into a 2D birds-eye view of the world in a R-Tree representation.

The path planner monitors this R-Tree and plots the course waypoints into this representation. If an obstacle in this path is encountered, a re-planning occurs by shifting the waypoints into safe areas where a collision is avoided.

III. Sensor Suite

The concept behind the SciAutonics sensor suite is redundancy. Multiple sensors with different functionalities, ranges, and operating principles are covering basically the same region of interest and providing the capability of exhibiting intelligent behavior and avoid collisions with obstacles.

A. Lidar Scanners

Figure 2. Lidar sensors mounted in experimental configuration with high-positioned sensor (on roof).

The SICK Lidar scanners provide reliable object detection under a large variety of environmental conditions. RASCAL employs four of such sensors: two for horizontal scanning of road and objects, and two for vertical scanning for detection of cliffs and "negative obstacles" (ditches). Each Lidar unit provides a maximum look-ahead distance of 80 m. Various configurations of these Lidars have been examined. Figure 2 shows a configuration with one of the horizontally scanning Lidars mounted on the vehicle roof. This provides the possibility to use the high mounted scanner to track road edges, which are often indicated by small berms or ditches.

An alternative mounting location for a horizontal Lidar scanner is at the front of RASCAL. In Figure 3 the Lidars can be seen with metal shields – to prevent from external light disturbing the measurement process. However, experiments showed that these shields produce significant internal reflection through scattered light inside the shields so that the signal-noise ratio is reduced. Therefore, these shields were abandoned for the RASCAL system.

Figure 3. Shields on the Lidar to avoid perturbations.

B. Stereo Vision System

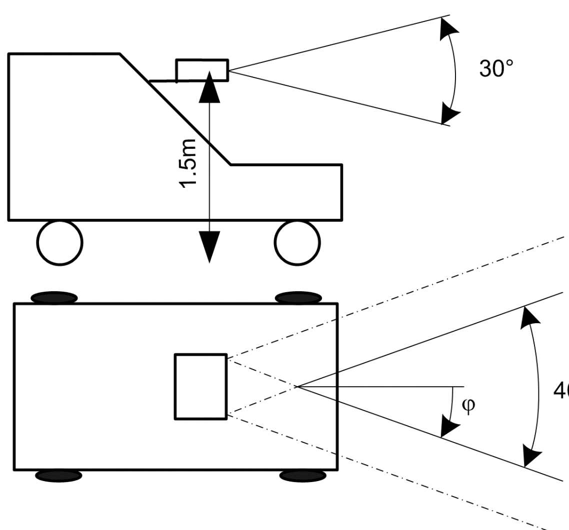

Figure 4: Mounting of the Stereo Vision Sensor

The Stereo Vision Sensor for RASCAL, developed by Seibersdorf research, is mounted in front of the windshield in a height of 1.5m. The field of view is shown in Figure 4.

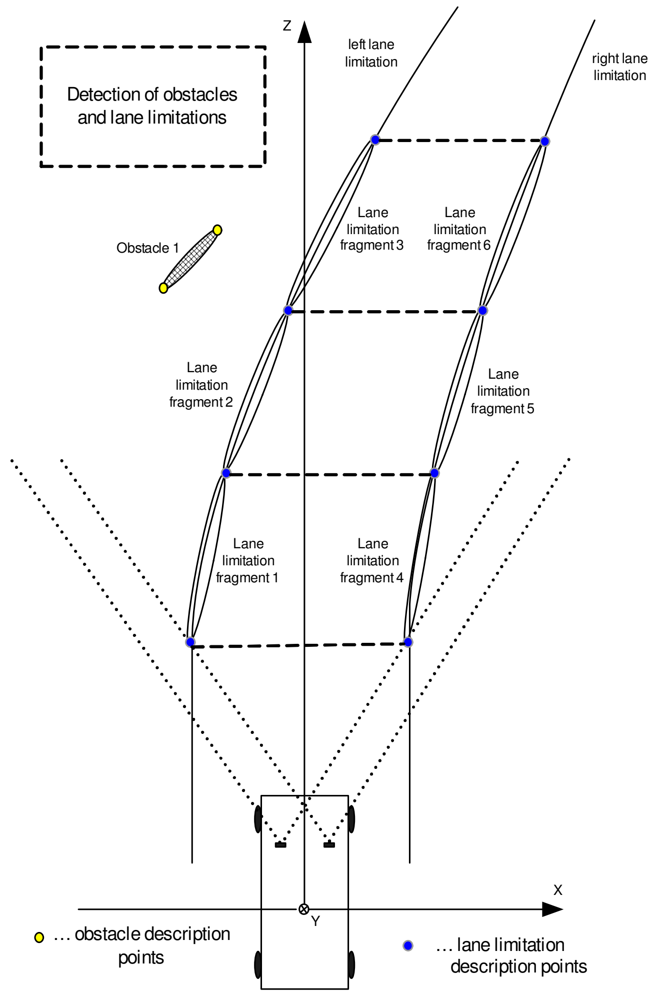

Figure 5. Representation of obstacles and lane borders.

The Stereo Vision Sensor consists of a pair of Basler A601f monochrome cameras with a resolution of 656 (H) x 491 (V) and a quantization of 8 bits/pixel [7]. The focal length of the lenses is 8.5mm and the baseline of the stereo head is 0.3m. The cameras are connected by two 400Mbit-Firewire-cables [9] to an embedded system, called VisionBox, which is placed in one of the electronic boxes. The VisionBox is based on a Texas Instruments TMS320C6414 DSP running at 1GHz and offers two 400MBit-Firewire- and one 100MBit-Ethernet-interface. It is responsible for the synchronous acquisition of both images, for the execution of the computer vision algorithm, and for the communication with RASCAL brain via the Ethernet interface using UDP-sockets. The operating system is DSP/BIOS II from Texas Instruments and the code of the computer vision algorithm is automatically generated from the corresponding MATLAB/ SIMULINK model using Real-Time Workshop Embedded Coder [8]. This model-based design approach enables us to significantly reduce the final test-debug-redesign cycles. The whole sensor is protected against dust and sun-light by a special housing and a sun-roof.

The main task of the stereo vision sensor is detection of obstacles and lane limitations in front of RASCAL. For lane limitation detection, left and right camera image are divided into different regions-of-interest (ROI's). Each ROI is median-filtered and afterwards, a linear gradient filter is used to extract edges. By applying the Hough transformation [13], line segments are identified in each ROI and grouped together over the whole image to form left and right lane limitation. For obstacle detection, the maximally stable extremal regions (MSER) method is used [12].

As presented in Figure 5, obstacles are described by their very left and right border. Lanes are split into fragments, which itself are marked as obstacles. Therefore, integration of this sensor system into RASCAL is easy, since all sensor systems use the same kind of obstacle representation.

An example of detected lane limitations from a test run near Böheimkirchen, Austria, is shown in Figure 6. Lane limitations are detected in both stereo images. After calculation of stereo correspondence, the six lane limitation segments are identified and the description points reported to RASCAL brain.

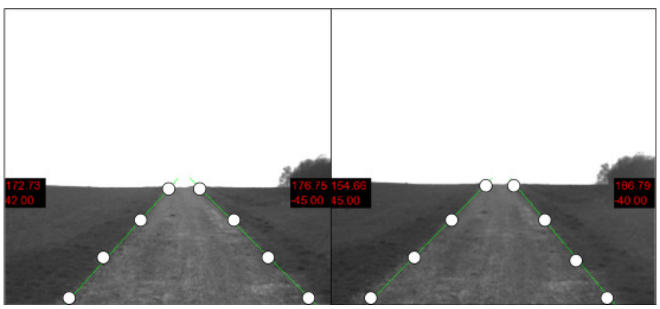

Figure 6: Results of test runs in Austria, left and right stereo image are shown. Detected lane limitations are marked and description points are shown.

C. *Other Sensors*

In addition to the main sensing systems described above, additional sensors are placed on RASCAL to provide redundant sensing by different methods. Radar allows detecting metal objects such as wire mesh fencing, which is otherwise not detectable by any sensor. RASCAL will employ an Epsilon-Lambda radar system which performs a mechanical scanning within a 12 deg horizontal wedge. It also is able to resolve vertical bins within 4 deg. The current status of this sensor integration is still work-in-progress, and we cannot yet report any performance data. However, initial tests looked quite promising.

Another active sensing system is a set of ultrasound sensors. These are spread around the vehicle, to provide emergency object detection. Due to their limited range (6 m), they are only suitable for slow motion, but they provide an additional sensing path in case of specific situation, such as driving in a tunnel / under a bridge, or backing up. For rearward sensing, these ultrasound sensors are the only object sensors. We do not anticipate backing up with a high speed, so they are suitable for this situation.

IV. Vehicle Control

Autonomous capability requires a reliable and robust navigation and control system. The primary goal of the controller is to take in waypoints from the path planner and accurately guide the vehicle to a desired area at or near the waypoint. Vehicle control algorithms were developed at Auburn University. Three main parts comprise the controller: navigation, throttle control, and steering control. Each part operated at a 50 Hz update rate. Please see [6] for a detailed description of the controller.

At the Grand Challenge 2004, the maximum speed of RASCAL had been deliberately set to a very low level, due to concerns regarding object recognition system and a lack of a robustly functioning collision avoidance module. Since then, work was performed on the RASCAL vehicle control system to optimize the behavior and achieve a higher speed than before. At Auburn University's National Center for Asphalt Technology (NCAT), there is a 1.7 mile test track that was used from July 2004 to March 2005 for carrying out experiments and control characterizations.

The main improvements that have been made this year are due to the development of a vehicle model by executing many system identification tests. This model has allowed Auburn to more accurately tune the controller, thus raising the overall performance level of the entry vehicle.

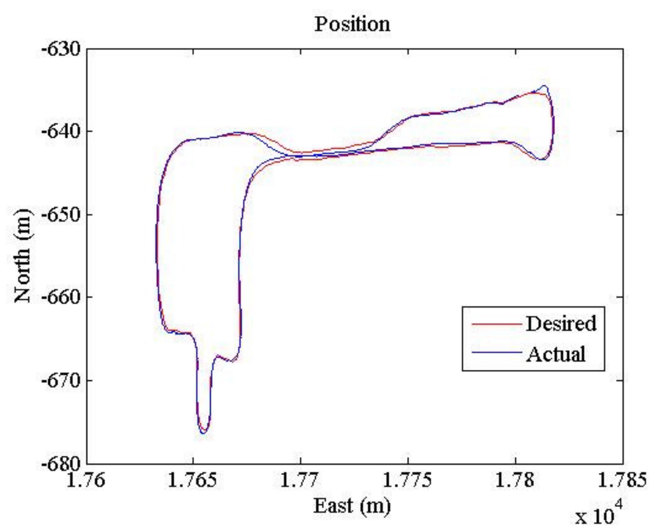

Figure 7. Test path made at Auburn University.

A. *Control Repeatability*

The test path created at Auburn (Figure 7) was used to carry out many autonomous runs of RASCAL. The path segments were designed to test high and low speed maneuverability.

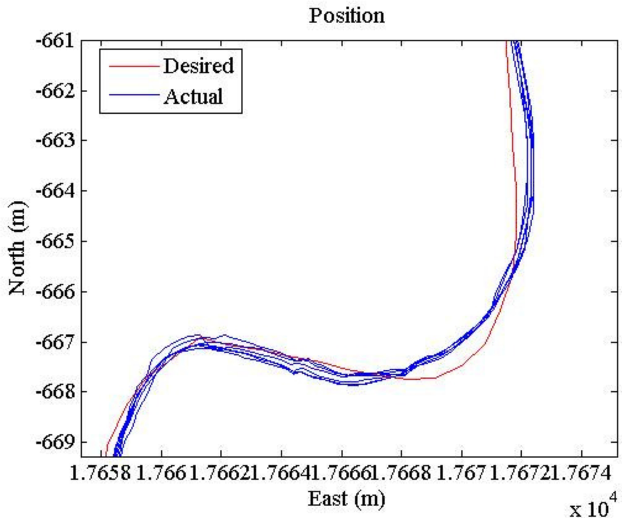

Figure 8. Control repeatability, driven by RASCAL in autonomous mode.

In Figure 8 several of these plots are overlaid for one segment of the course. This plot shows the repeatability of both the GPS and the control algorithms. The repeatability error is primarily due to error in GPS, but also contains some error due to the deviation of the path as a result of controller inaccuracy. This error is mainly due to the controller's look-ahead scheme which causes it to cut corners when entering a turn, as well as exit them wide, and vehicle slip.

B. Lateral Controls

In order to judge the quality and smoothness of the lateral control, a plot is shown in Figure 8 for steer angle and yaw rate.

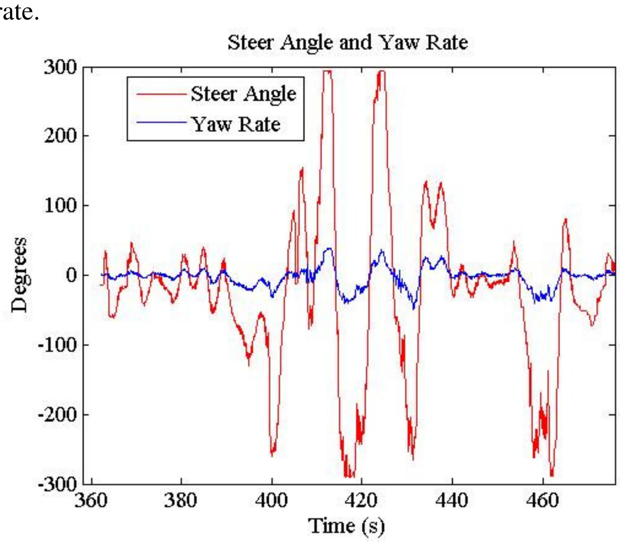

Figure 9. Steer angle and yaw rate.

There are some small oscillations, but they are very small, indicating that the control parameters have been chosen appropriately.

C. Longitudinal Control

The longitudinal control determines the vehicle speed for driving to the next set of waypoints. This speed depends on the curvature of the waypoint path and on an absolute limit within the waypoint segment.

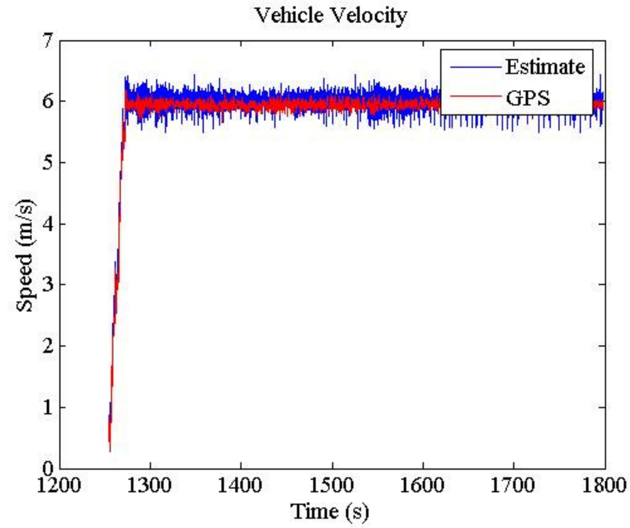

Figure 10. Constant velocity run. Red: GPS measurement. Blue: estimated speed in control loop.

In Figure 10, the speed is shown during an autonomous run, with a constant target velocity set. It can be seen that there are oscillations in the measured speed (obtained from the GPS). This indicates that the integrator needs to be slightly tuned to match the reference. This may be caused by the fact that a reference point too far ahead is being picked up that has a desired speed outside of the limit put on the integrator. Another issue is the quality of sensors. For immediate speed measurement, wheel speed sensors are used, and they are the largest contributor to noise.

V. Collision Avoidance

The collision avoidance task needs to collect all relevant sensor information, information about the target path to be driven (DARPA waypoints), all possible boundary conditions (e.g., corridor width), and the current vehicle state (location, speed). The approach is to collect all features from the sensors in an R*-Tree data structure for analysis.

Sensor processors identify basic shapes that describe the input the sensors are receiving. This procedure utilizes edge detection algorithms and other object identification techniques to determine what might potentially block the ideal path of the vehicle. The shapes are then passed as part of a feature report to a separate system responsible for making sense of all of the different sensor feedback. Here, the shapes are placed into an R*-Tree for efficient querying. Although other structures exist, including Quad trees and k-d trees, they aren't as effective as R*-Trees when considering actual physical computer limitations, such as memory constraints. R*-Trees handle memory paging issues more efficiently than other structure implementations, while still providing the necessary spatial representation desired.

The R*-Tree view is multidimensional. At the present, we are utilizing a two-dimensional view of the world (birds-view), including only data that will be a potential hazard for the vehicle. In other words, objects in the third "Z" dimension (perhaps above the vehicle) are ignored if they are received. We believe the two-dimension representation will suffice for our purposes; however, extensive tests have not yet been performed as of the date of this paper. It is not difficult to switch to a three-dimensional model if such is desired, but performance would be compromised.

Using the R*-Tree, spatial queries are executed repeatedly to dynamically adjust the path of the vehicle based on new input. Simultaneously, the confidence level of previously "seen" obstacles is increased as repeated sittings are recorded. That is, obstacles that are not repeated at least a certain number of times are discarded as noise, whereas real objects will be repeatedly inserted into the tree by various sensors. Spatial queries utilize convex hull algorithms to combine smaller shapes into larger shapes that more accurately represent the real world. This spatial system allows easily formulated queries to select a path for the vehicle with no obstacles, and at the same time it minimizes deviation from the given target path.

The ability to avoid obstacles effectively is one of the most difficult tasks any autonomous vehicle will face. Our success in the Grand Challenge this year will rely heavily on our capacity to accurately model the real world and plan routes to avoid the obstacles that block our intended path.

VI. Testing

Theoretical analysis and simulation is being performed using the OpenSource simulation tool GAZEBO [10]. This tool allows simulation of a robotic vehicle in a 3D Virtual reality environment, by simulating situations and vehicle motion [11]. In order to use this tool, every module of the RASCAL software has an interface to the functions of GAZEBO. This allows easy configuration and porting of the system modules from laboratory simulation into the actual operating unit on RASCAL. At the time when the final version of this paper was submitted, we have successfully passed the DARPA site visit in May 2005, where RASCAL had to pass two obstacles (large garbage tons) which were placed at arbitrary locations within the drivable corridor. Our preliminary version of the obstacle detector and path planner managed successfully to avoid collisions with 5 out of 6 obstacles. A problematic scenario was when the obstacle was placed in a curve: the current implementation of the path planner does have trouble in correctly dealing with this situation. We are confident that we will have this problem solved by Sept. 2005 when the ITSC meeting (and later the Grand Challenge) takes place.

VII. Conclusion

The DARPA Grand Challenge has revived public interest in autonomous vehicles and in intelligent technology in general. The participation in that event provided a very good opportunity for teams to gain visibility. In terms of completing the course and demonstrate autonomous driving capability, the competition in 2004 was not very successful, due to a short preparation time and as a consequence of a lack of mature technology, but the 2005 Grand Challenge will see significant progress – the participants from 2004 can build on their expertise, and the new teams will also be able to leverage from the large body of research in this domain.

The team SciAutonics/Auburn is optimistic regarding its participation in this event. RASCAL has been upgraded by a robust system, and much of the ambitiously envisioned functionality that was not yet implemented in 2004, is now be ready in 2005. However, the prediction still is that also in 2005 no team will complete the course. But it could well be that several teams drive past the first 20 miles of the track.

Acknowledgment

This project would not have been possible without the up to volunteers in the SciAutonics/Auburn team, who worked on the design and implementation without pay. The authors thank Rockwell Scientific (RSC), Amgen Inc., ATV Corporation, Auburn University, and Seibersdorf research for their support of this project.

References

[1] R.Behringer, "The DARPA Grand Challenge – Autonomous Road Vehicles in the Desert," in *Proc. Intelligent and Autonomous Vehicles (IAV)*, Lisbon, 2004.

[2] C. Cavanaugh, "Design and Integration of the Sensing and Control Subsystems of CajunBot," in *Proc ITSC*, pp. 41-46, Washington DC, October 2004.

[3] A. Bacha, C. Reinholtz, A. Wicks, M. Fleming, A. N. Naik, M. Avitabile, and N. Elder, "The DARPA Grand Challenge: Overview of the Virginia Tech Vehicle and Experience," in *Proc ITSC*, pp. 481-486, Washington DC, October 2004.

[4] N. Howard, T. Patel, and J. C. Bebel, "An Autonomous System Used in the DARPA Grand Challenge," in *Proc ITSC*, pp. 487-490, Washington DC, October 2004.

[5] U. Ozguner, K. A. Redmill, A. Broggi, "Team TerraMax and the DARPA Grand Challenge: A General Overview," in *Proc. Intelligent Vehicles (IV)*, pp. 232-237, Parma, 2004.

[6] R. Behringer, S. Sundareswaran, B. Gregory, R. Elsley, B. Addison, W. Guthmiller, R. Daily, D. Bevly, "The DARPA Grand Challenge – Development of an Autonomous Vehicle," in *Proc. Intelligent Vehicles (IV)*, pp. 226-231, Parma, 2004.

[7] Basler Vision Technologies, "Basler A600f User's Manual," Document Number DA00056106, March 2004. Available: www.baslerweb.com

[8] The MathWorks, "Real-Time Workshop Embedded Coder User's Guide," Version 4.1, October 2004. Available: www.mathworks.com

[9] H. Yoshimoto, D. Arita, R. Taniguchi, "Real-Time Image Processing on IEEE1394-based PC Cluster," in *Proceedings of the 15th International Parallel and Distributed Processing Symposium*, 2001.

[10] http://playerstage.sourceforge.net/gazebo/gazebo.html

[11] B. Gerkey, R. T. Vaughan, and Andrew Howard, "The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems," in *Proc. of the 11th Int. Conf. On Advanced* Robotics, pp 317-323, Coimbra (Portugal), June 2003.

[12] J. Matas, O. Chum, M. Urban, T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions", in *Image and Vision Computing* 22 (2004) 761-767.

[13] R.C. Gonzalez, R.E. Woods, "Digital Image Processing", Prentice Hall, 2nd edition, 2002.