Autonomous Driving in Dynamic Environments

Sascha Kolski*, Dave Ferguson†, Cyrill Stachniss*, and Roland Siegwart*
*Swiss Federal Institute of Technology (ETHZ)
Zurich, Switzerland
{kolskis, scyrill, rsiegwart}@ethz.ch
†Carnegie Mellon University (CMU)
Pittsburgh, PA, USA
dif@cmu.edu

***Abstract*— Autonomous vehicles are being used increasingly often for a range of tasks, including automated highway driving and automated parking. These systems are typically either specialized for structured environments and depend entirely on such structure being present in their surroundings, or are specialized for unstructured environments and ignore any structure that may exist. In this paper, we present a hybrid autonomous system that recognizes and exploits structure in the environment in the form of driving lanes, yet also navigates successfully when no such information is present. We believe this approach is more flexible and more robust than either of its sub-components alone. We demonstrate the effectiveness of our system on both marked roads and unmarked lots under the presence of dynamic objects, such as pedestrians or other vehicles.**

I. Introduction

Every year, thousands of people are killed in road accidents, with millions more injured. The vast majority of these accidents are due to human error, with roughly 5% caused by vehicle defects [1]. Such staggering findings motivate the use of driver assistant systems and fully automated vehicles to increase driver and passenger safety.

Driver assistant systems can help drivers to identify dangerous vehicle states and traffic scenarios and reduce the risk of accidents. These driver assistant systems are widespread in all categories of vehicles and range from anti-lock brakes to radar based adaptive cruise control. The development of these systems has been accelerated by integrated drive-by-wire components such as electronic gas pedals, brakes, and steering systems.

The development of such components has also hastened the arrival of autonomous passenger vehicles. In 1997, the NavLab vehicles travelled 'no hands' across the United States, requiring only accelerator and brake pedal interaction from the driver [2]. In 2005, 23 autonomous vehicles started a race across the Nevada desert in the DARPA Grand Challenge race [3], with 5 of them finishing the 211.1 Km distance.

Most of these systems depend on environmental structure like driving lanes or dense sets of GPS points. However, in many common driving scenarios neither of these sources of information will be available, for example, when leaving a road and entering a parking lot.

Autonomous navigation in unstructured environments is an active research area in field robotics, and a number of effective approaches have been developed that address this task [4]–[7]. A common technique is to maintain a map of the environment and use this to plan safe paths to a desired goal location. As the vehicle traverses the environment, it updates its map and path based on its observations. Such an approach works well when dealing with reasonably small areas, but storing and planning over maps of the entire environment is impractical when traversing long distances. Further, without taking into account non-spatial information such as road markings, these approaches are unable to ensure that the vehicle stays within its lane (or even on the road) when navigating through highway or urban environments.

In this paper we present a hybrid navigation system that combines the benefits of existing approaches for driving in structured environments (e.g. roads) and unstructured environments (e.g. parking lots). When driving on detectable roads, the system uses visual lane detection and laser range data to generate a local map, which is processed by a local planner to guide the vehicle down the lane while avoiding obstacles. When driving in unstructured environments, the system employs a global map and planner to generate an efficient trajectory to a desired goal. The combined system is capable of navigating a passenger car to a given goal position without relying on road structures, yet it makes use of such structure when it is available. We also describe extensions to this approach capable of dealing with dynamic obstacles, such as pedestrians or other vehicles, that are commonly found in realistic driving scenarios.

We begin by briefly introducing our autonomous Smart vehicle and its onboard sensors. We then describe our system for navigating structured and unstructured environments, and go on to describe how this system can be used in environments containing dynamic obstacles. In Section IX we present results from runs performed in road and parking lot scenarios and we conclude with discussion.

II. Vehicle and Sensors

Our vehicle is a Smart fortwo passenger car that has been modified for autonomous operation. Firstly, we have interfaced the Smart's controller area network (CAN) bus to access data on the dynamic state of the vehicle, specifically the wheel speed and the steering angle. We have also added actuators to the brake pedal and interfaced the electronic gas pedal and power steering. Finally, a number of sensors (discussed below) have been added to provide vehicle and environmental information. A detailed description of the mechanical and architectural aspects of the vehicle can be found in [8].

Fig. 1. Our autonomous Smart car platform. There are three fixed laser range finders mounted on the front of the vehicle and on the sides of the roof, and two spinning laser range finders mounted together on the center of the roof. Inside the vehicle, mounted behind the windscreen, is an automotive camera used for lane detection.

A. Proprioceptive Sensors

As with many other passenger cars, the Smart is equipped with a variety of sensors which are linked using the vehicle's CAN bus. By interfacing this bus it is possible to access the sensor data and measure the vehicle's dynamic state precisely.

a) Wheel Encoders: The overall vehicle speed is derived from the four wheel encoders with a resolution of 0.5 revolutions/minute. The steering wheel angle is available with a resolution of $0.04°$.

b) IMU: We have added a 6 degree of freedom IMU to the Smart that is able to measure angular rates up to 100°/sec at a resolution of $0.025°$. Lateral accelerations in all three dimensions can be measured up to 2g with a resolution of 0.01 m/s^2.

B. Exteroceptive Sensors

c) Differential GPS system: (Omnistar Furgo 8300HP, rain proof antenna) This system provides an accurate position estimate together with its standard deviation when satellites providing the GPS drift correction are visible from the car. When no correction is available standard GPS is provided.

d) Laser Range Finders: We use five SICK laser range finders for sensing the spatial structure of the environment. These are configurable laser range finders based on time of flight measurements, with angular resolutions of 1 or 0.5°, angular ranges of 180° and measuring ranges up to 80 meters. Three of these lasers are kept at fixed angles—one at the front of the vehicle and two on the sides of the roof—to quickly detect upcoming obstacles and difficult terrain, and two of the lasers are mounted to a spinning platform—on the center of the roof—to provide full 3D information about the vicinity of the vehicle.

e) Monocular Camera: An automotive gray-scale camera is mounted inside the vehicle at the top of the windscreen for observing the area in front of the vehicle and detecting lane information. The resolution of the camera is 750 $\times$ 400 pixels and it delivers information at 25 frames per second.

III. Position Estimation

The localization algorithm used in our system is based on the information form of the Kalman filter, the Information filter. This filter has the property of summing information contributions from different sources in the update stage. This characteristic is very interesting when many sensors are involved, which is the case in our system. Our sensor fusion scheme is based on [9] [10] and [11]. To accurately localize the vehicle, four different sensors are used: DGPS, IMU, optical gyro and vehicle sensors (wheel encoders and steering angle sensor). The combination of their measurements allows the estimation of the vehicle's 6 degrees of freedom i.e. the 3D position (x, y, z) and the attitude (roll, pitch, heading).

A detailed description of this aproach can be found in [8].

IV. Traversability Estimation

Reliable estimates of the traversable area in the vicinity of the vehicle are crucial for autonomous driving. We currently use the three static laser range finders on our vehicle to estimate the traversability of the area in front of the vehicle.

Given a laser range observation, we first compute the end points of the individual beams. We then add the 3D points to the cells of a local two-dimensional grid map according to the x, y-coordinate of the beam (horizontal position). We then parse the cells and compute the mean and variance of the z-values (vertical position) for each cell. The traversability classification of a cell is performed locally based on these values. When adding observations from multiple laser range finders into a single grid, it is often the case that false obstacles are detected by some of the lasers (described as phantom obstacles by Thrun et al. [12]). These false obstacles are caused by small errors in the pitch estimate of the pose of the vehicle. To remove these artifacts, we compute the traversability estimate individually for each scan and merge the independently-estimated traversability values into a common grid structure. We found that this yields good results when moving on streets as well as on unpaved roads. An example traversability map produced by this approach is shown in Figure 2.

V. Traversability Prediction

If an autonomous vehicle has onboard far-range sensors, then the information from these sensors can be used by the vehicle to enable smooth following of winding roads and early reaction to obstacles in the vehicle's route. The smaller the sensing range, the more extreme the obstacle avoidance movements of the vehicle must be. This is also true for following the current road lane. To generate smoother motion

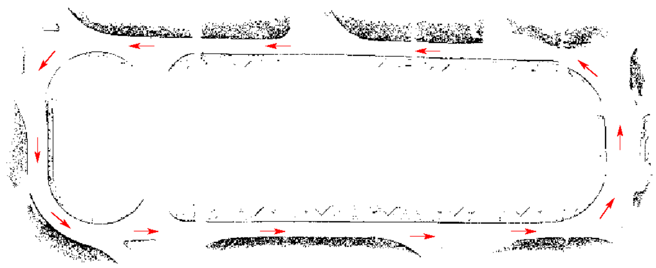

Fig. 2. A traversability map obtained using the three fixed laser range finders on our vehicle. Black areas are untraversable and the red/grey arrows illustrate the trajectory taken by the car.

of the vehicle using only our laser range finders, we perform a prediction of the traversable area at a far range by using the accurate near-range information provided by these sensors.

In order to perform the prediction, we compute a 1D pattern of average cost values. This pattern is generated by storing, for each cell in our 1D pattern, the average cost of all the already-observed cells in our local cost map that have their lateral offset position from the current route match the cell position in our 1D template. This process is illustrated by the image in the second row of Figure 3.

We then use this pattern to estimate unknown traversability cost values for cells far away from the vehicle along the current route (illustrated by the blue lines in the same figure). In this way, we obtain an estimate of the expected cost in the currently unobserved area. The image in the last row of Figure 3 provides an example result of this estimation. The area within the red rectangle is the measured cost map and the the area in the blue rectangle is the predicted cost based on the measured cost map.

This technique allows us to estimate the traversability of areas that have not been observed with the sensors. This prediction does not cope with unforseen obstacles but it does help the car to improve its estimate of the road projection. This is especially important if the localization is affected by GPS drift, since in this case it is not sufficient to simply follow the predefined route. With our approach we are able to robustly estimate the road and the traversable area, even when faced with GPS drift or outages.

VI. Driving in Structured Environments

When driving in structured environments such as roads or highways, it is important for safety that vehicles abide by traffic rules and stay in their own lanes. For autonomous vehicles, such structure is useful because it constrains the available actions of the vehicle and reduces the complexity of the navigation task. For instance, if an autonomous vehicle is traveling down a road, it knows it must stay within its current lane so the lane can be used as a guide for where the vehicle must travel to next. Such an approach can be coupled with a standard commercial navigation unit that provides higher-level guidance on when to turn down which street.

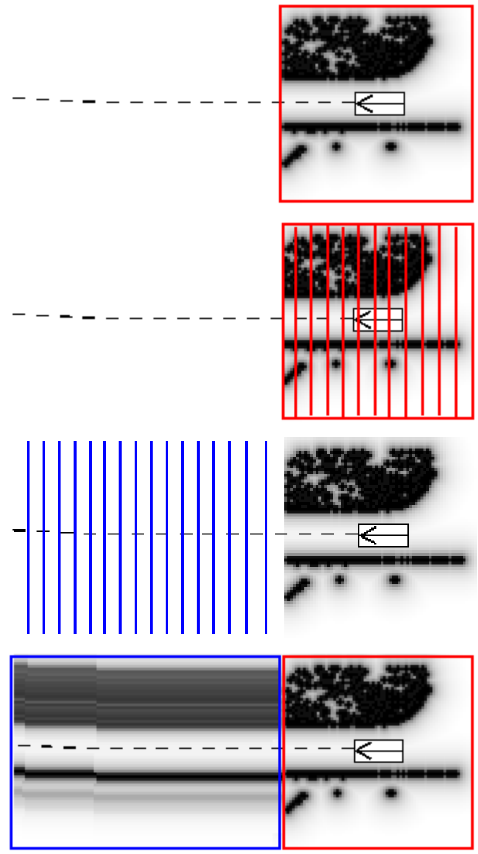

Fig. 3. The traversability prediction. The image in the first row shows a local cost map representing the traversable and non-traversable area. The dashed line illustrates the route. Based on this map and the route description, we compute a 1d cost pattern based on the cell labeled by the red lines in the image in the second row. We then use this pattern to predict the traversability in from of the car (illustrated by the blue lines in the third figure). Finally, we obtain a cost prediction for cells not observed to far by the robot (labeled by the blue rectangle in the last image).

The resulting combined system can autonomously navigate between arbitrary road locations.

However, to ensure safe navigation, it is not enough to just follow the current lane. The vehicle must be alert at all times and able to avoid other cars and objects that may unexpectedly place themselves in its path, such as cars pulling out from driveways or pedestrians crossing the street, for example. To achieve such behavior in our Smart, we construct a local map using the traversability estimation method described in the previous section and plan a collision-free path through this map. Both the map and the plan are updated frequently (at 20 and 10 Hz, respectively). With both the local obstacles and lane information encoded in the local map, the vehicle is able to plan trajectories that keep it within the current lane and also avoid any obstacles.

A. Lane Detection

To extract lane information, we use a monocular gray-scale camera designed for automotive use and a real-time lane detection algorithm running on a separate computer equipped

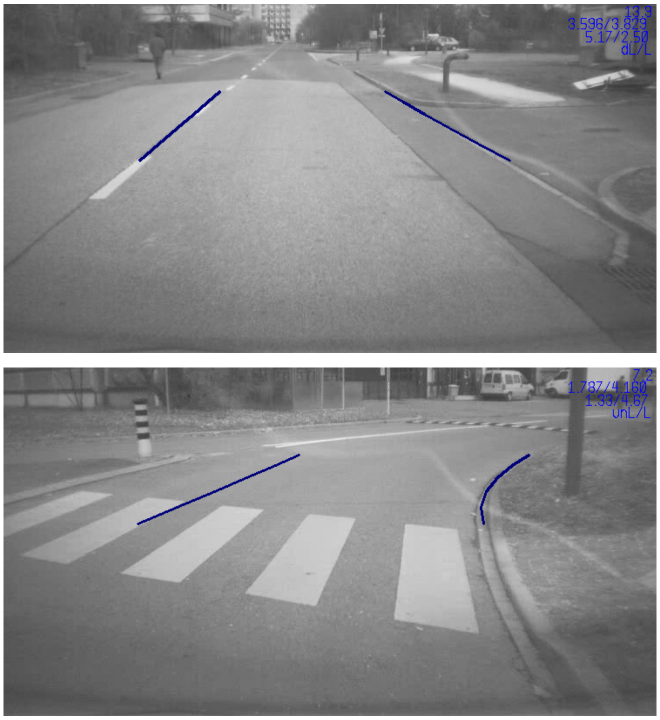

Fig. 4. Example results from our lane detection approach applied to images from a straight (top) and curved (bottom) section of road.

with a frame grabber. Our approach combines hypotheses from several lane detection algorithms, each designed to detect different types of lanes, such as the closest lane to the vehicle, straight lanes, or curved or symmetric lanes. These algorithms rely mainly on the spatial gradient of the image to extract their hypotheses. The results of the individual algorithms are then combined to determine the most probable lane. Example results from our lane detection algorithm are shown in Figure 4 and more details on the algorithm can be found in [13].

B. Local Planning

In order to follow the current lane safely and smoothly, we project a set of potential vehicle actions onto our traversability map and check the cost of these actions. For this, we use an approach similar to that used by the Stanford Racing Team in the Grand Challenge [12]. We take the centerline of the lane and use this to construct a set of possible trajectories for the vehicle to execute. These trajectories vary in their lateral offset from the nominal centerline path and provide a series of alternatives from which the best obstacle-free trajectory can be selected.

By exploiting the structure of the driving lane, this combined approach provides smooth, safe trajectories for the vehicle when it is operating on roads.

VII. Driving in Unstructured Environments

In unstructured environments where there is no lane information to guide or constrain the actions of the vehicle, we must use a more general approach for navigation. For instance, imagine our vehicle has arrived at its intended destination address and now wants to park in a specified goal location within the parking lot. To do this, we can still use the local planning component of our system, however we now need to compute a path for the planner to follow as we no longer have lane information to provide this for the vehicle. To generate these paths we use the Field D* algorithm, which has been incorporated into several fielded robotic systems [14]. This algorithm provides very low-cost 2D paths through grid-based representations of an environment and is able to repair these paths to account for new information as the vehicle observes obstacles during its traverse. These paths do not take into account the heading restrictions of the vehicle and instead approximate the least-cost path to the goal for a vehicle that can turn in place. Because Field D* does not encode the mobility constraints of the vehicle, it cannot be used alone for accurate trajectory planning for the vehicle. Consequently, we combine it with a local planner to provide feasible paths.

One way to do this is to use the Field D* path as the input to our local planner, which will then track this path to the goal. As the vehicle navigates through the environment, the global Field D* path is updated based on new information received through the onboard sensors, and the trajectories generated by the local planner are subsequently updated to reflect the new global path. This approach works well in static environments, where the Field D* path can be quite accurately tracked using the local planner. However, in dynamic environments such an approach may not be ideal, as discussed below.

VIII. Navigating in Dynamic Environments

Typical driving scenarios involve dynamic obstacles: there are usually pedestrians or other vehicles moving around within the environment that need to be avoided. These dynamic obstacles need to be accurately detected and reasoned about in order to produce safe paths for our vehicle to traverse. In the following two subsections we describe extensions to our navigation approach that enable us to model and reason about dynamic elements.

A. Mapping Dynamic Environments

To detect and predict the trajectories of moving objects, several approaches have been proposed in the robotics community. Feature-based approaches operate extract features from the raw data and then track these features to compute their motion parameters. Such approaches are suitable for a variety of sensor data, for example, vision, radar, and laser, and have been widely used [15]. However, these approaches typically require a priori knowledge of the features to track and are therefore only suitable for the detection of well defined classes of objects. Raw data-based approaches, on the other hand, detect motion from raw sensor data and do not depend on any model of the objects being observed. They are thus less accurate for predicting well-behaved, known object classes but perform well when confronted with a range of different dynamic elements.

Our vehicle uses a raw data-based scan alignment approach to detect moving objects in the environment. Based on work

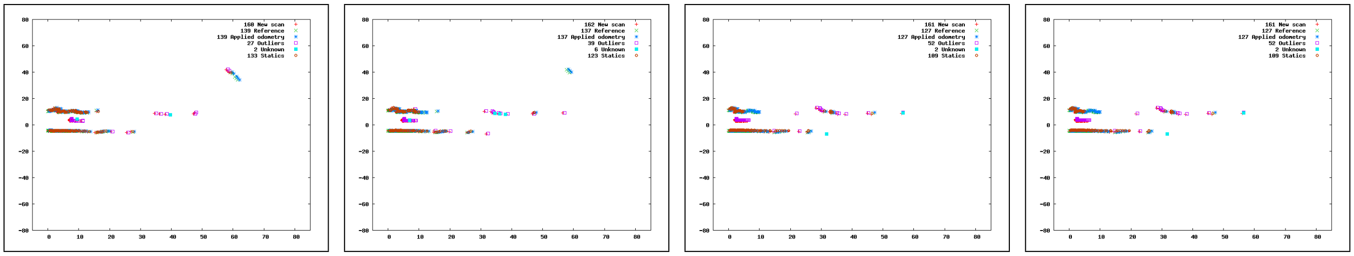

Fig. 5. Scan points taken during a test ride on campus. The static structure of the environment (curbs, buildings) is detected as static parts of the environment while the oncomming vehicle forms an L-shaped set of dynamic points.

introduced by Jensen [16], our algorithm extends the *iterative closest points* algorithm (ICP) [17]. The ICP algorithm aligns two sets of points by iteratively finding the set of points in one scan that are closest to a set of points in the other scan, and then computing a transformation that minimizes the distance between the two sets of points. Special care has to be taken to suppress outliers, which are points that are present in one scan, but not in the other, because they bias the alignment. The pose correction $d_x, d_y, d\Theta$ is computed as a weighted mean over all connected points. The link between a scan point (x_i, y_i) and a scan point (x_j, y_j) is expressed by a link variable $l_i = j$. With this the pose correction can be computed from the linked scan points and results in

$$d_x = \frac{1}{I} \sum_{i=I^*} (x_i - x_{l_i}) \tag{1}$$

$$d_y = \frac{1}{I} \sum_{i=I^*} (y_i - y_{l_i}) \tag{2}$$

$$d\Theta = \frac{1}{I} \sum_{i=I^*} (\phi_i - \phi_{l_i}) \tag{3}$$

Linking and correcting are repeated until the correction value is below a predefined threshold. The resulting transformation determines the displacement from the reference to the correspondence pose.

While in scan matching outliers are a disturbing factor and are filtered out in each iteration, they are very useful for the detection of dynamic obstacles. In our approach, the outliers found in each iteration of scan matching are collected and clustered. The resulting clusters are candidates for dynamic objects and are tracked to derive their motion parameters. Figure 5 provides an example illustrating this ability of this approach to filter static points from those in motion.

B. Planning in Dynamic Environments

When driving within road lanes, dynamic obstacles usually do not significantly interfere with the traverse of our vehicle because their behavior is well-defined. To ensure we don't collide with any of these obstacles, our local planner can estimate the trajectories of these other vehicles or pedestrians and then check that its intended trajectory does not intersect these objects at any point in time. Figure 6 shows a simple example of this reasoning. The local planner can then remove from contention any trajectories that intersect dynamic obstacles (or modify the velocity profile of the trajectory to avoid the obstacle). Since the local planner is generating a series of possible trajectories that span the current driving lane, at least one of these trajectories should still be obstacle-free if the dynamic obstacle is abiding by traffic rules.

For our unstructured driving scenario, the situation is complicated because the dynamic obstacles may interfere entirely with the global path being tracked. Thus, it may not be possible to track this path using our local planner. Instead, we may need to evaluate a more general set of possible local trajectories for the vehicle to execute, including some that do not follow the current path. For this, we use an approach that follows a large body of work on outdoor mobile robot navigation [4], which has the local planner project out a range of possible local trajectories and then evaluate each trajectory based on both the cost of the trajectory itself (in terms of curvature, terrain, distance, etc), as well as the cost of a global path from the endpoint of the local trajectory to the goal. Thus, rather than a single global path being planned from the current vehicle position to the goal, global paths are planned from each trajectory endpoint. Since Field D*, like most replanning algorithms, performs planning in a backwards direction out from the goal, computing these extra paths and their associated costs is very efficient (and often requires no extra planning at all).

Figure 7 shows an illustrative example of this combined approach. Here, a set of local arc-based trajectories are shown in red/gray, with the best trajectory shown in blue/black. Here, the best trajectory was selected based on a combination of the cost of the trajectory itself and the cost of a global path from the end of the trajectory to the goal (the goal is shown as a filled circle at the right of the figure). The global path from the end of the best trajectory to the goal is also shown in blue/black. In this example, a purely local planner would have selected the straight trajectory leading directly to the right, as this brings it closest to the goal in terms of straight-line distance. However, such a trajectory could cause it to get stuck behind the clump of obstacles in the middle of the map.

IX. Experiments

We have tested our system in both structured and unstructured environments. For structured environments, we had the vehicle drive down a road and record the resulting local maps.

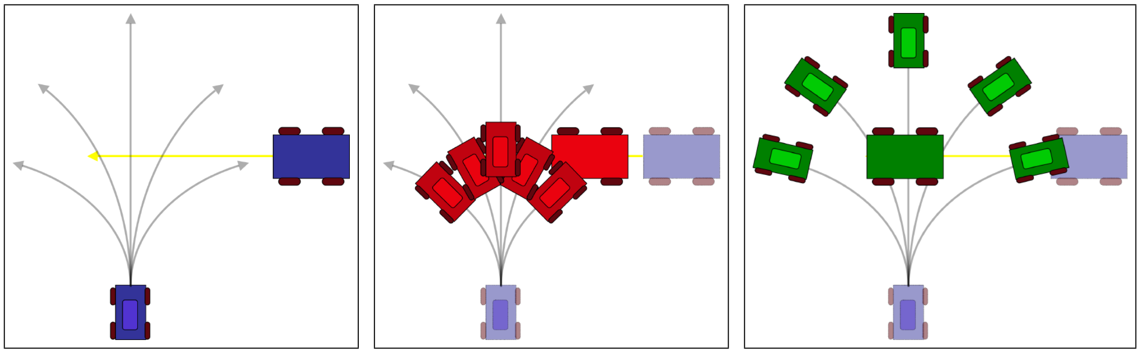

Fig. 6. Local planning amidst dynamic obstacles. If an agent (facing upwards) assumes the dynamic obstacle (traveling in from the right) is static when choosing its next action (potential actions shown as the arcs emanating out from the agent), it may select an action that will have it collide with the obstacle at some future point in time. Instead, it needs to estimate the position of the dynamic obstacle in the future and use these estimates to select an action that will avoid the obstacle at all times. The three images show the potential position of the agent based on its available actions, as well as the position of the dynamic obstacle, at three stages in time (the color of each agent reflects the time).

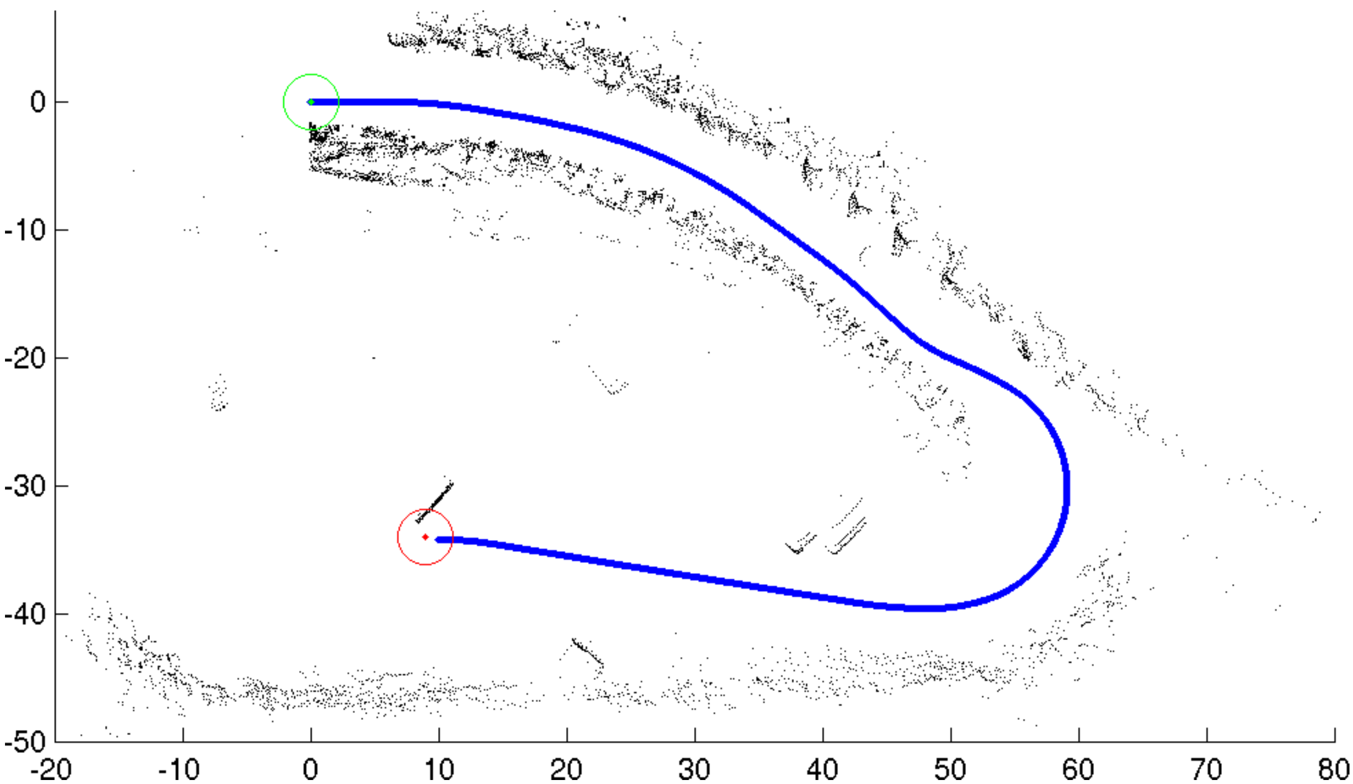

Fig. 8. Results from global planning and mapping in an unstructured environment. Shown here is the map created from the laser during an autonomous traverse from an initial position on a rural road to a goal position inside a large parking lot. Also shown is the path (in blue/black) traversed by the vehicle. The vehicle began from the position marked in green/gray at the top of the map, and navigated to the goal position marked in red/gray at the bottom.

Fig. 9. Snapshots from a video taken of the traverse in Figure 8.

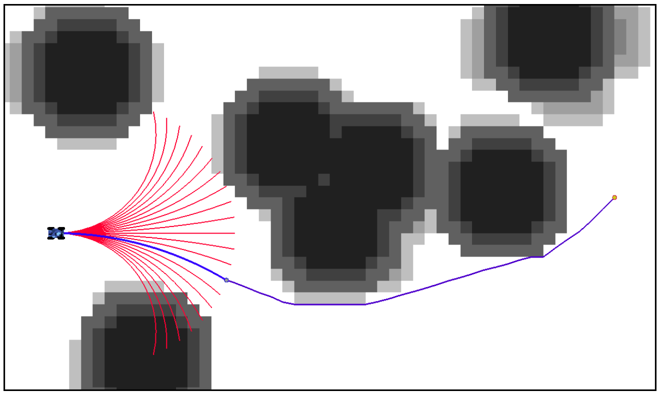

Fig. 7. **Global Planning in Unstructured Environments**. The vehicle projects a set of feasible local trajectories through the local map from its current position and orientation (trajectories for a single speed are shown in red/gray). The cost of each of these trajectories is computed based on the cost of the cells the trajectory travels through (darker areas are more expensive, with black cells representing obstacles). A global path is planned from the end of each trajectory to the goal (shown as a filled circle on the right side of the map) and the cost of this path is added to the cost of the trajectory. The best trajecotry is shown in blue/black, along with the global path from the end of this trajectory to the goal. The map here has been configuration-space expanded so that the vehicle can be treated as a single point during planning.

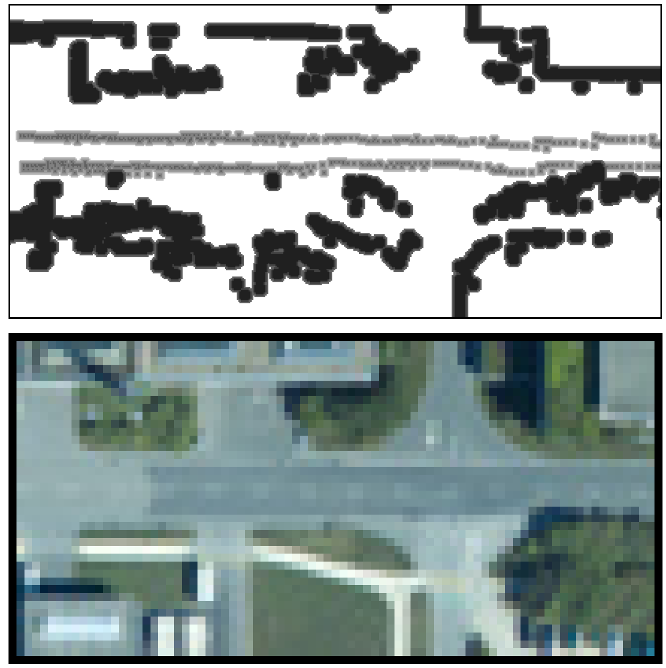

Fig. 10. Results from our lane detection and mapping in a structured environment. Data was gathered from roughly 100 meters of traverse down a road (traveling from left to right). The top image shows the combined local maps created by the vehicle during the traverse, with lane information shown as dark gray areas and obstacles shown in black. Notice that the obstacle information does not provide any real indication of the location of the lane or even road, and so does not suffice for safely guiding the vehicle. The bottom image shows a satellite map of the area.

Figure 10 shows the combined cost map constructed from the series of local maps and highlights both obstacles and lane information. Since the laser range data does not contain any information about the lane markings, the vision-based lane detection system is necessary to keep the vehicle in its lane.

To test our vehicle in unstructured environments, we gave it a more complex task. We began on a road and tasked it with autonomously navigating to a goal location in a nearby parking lot. Because there were large shrubs between its initial position and its goal, it was forced to travel down the road until it observed an opening through which it could enter the parking lot. At this point it entered the parking lot and navigated to its goal location.

Figure 8 shows the resulting map built by the vehicle and the vehicle's traverse. Figure 9 shows a series of images taken from a video of the traverse. Overall the vehicle travelled about 140 meters in 62 seconds, i.e. at average speed of roughly 2.3 m/s.

The vehicle trajectory seen in Figure 8 shows its ability to navigate in a scenario given a sparse set of waypoints and a combination of global and local path planning techniques.

Together these experiments illustrate our vehicle's ability to navigate through both road and non-road environments. Our vehicle effectively avoids obstacles to reach a defined goal position without relying on an a priori model of the environment.

X. Conclusion

In this paper we have presented a hybrid approach for autonomous navigation in structured and unstructured environments. Our approach exploits any lane structure present in the environment and combines this with local obstacle information to guide the vehicle along safe trajectories. When no structure is detected, the approach falls back on a global planner that generates efficient paths for the vehicle to desired goal locations. We have provided results demonstrating the operation of the vehicle in both structured and unstructured environments.

Acknowledgements

We thank the European Commission for partially funding this work through the SPARC project (Secure Propulsion using Advanced Redundant Control) under IST-507859 and the BACS Project under contract number FP6-IST-027140-BACS. Dave Ferguson is partially supported by a U.S. National Science Foundation Graduate Research Fellowship.

References

[1] M. Shell, "Final report of the european esafety working group on road safety, online available," 2003. [Online]. Available: http://europa.eu.int/informationsociety/activities/esafety/indexen.htm

[2] C. Thorpe, T. Jochem, and D. Pomerleau, "The 1997 automated highway demonstration," in *1997 International Symposium on Robotics Research*, 1997.

[3] "Darpa grand challenge race website." [Online]. Available: http://www.darpa.mil/grandchallenge

[4] A. Kelly, "An intelligent predictive control approach to the high speed cross country autonomous navigation problem," Ph.D. dissertation, Carnegie Mellon University, 1995.

[5] A. Stentz and M. Hebert, "A complete navigation system for goal acquisition in unknown environments," *Autonomous Robots*, vol. 2, no. 2, pp. 127–145, 1995.

[6] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999.

[7] 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," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2000.

[8] P. Lamon, S. Kolski, and R. Siegwart, "The smartter - a vehicle for fully autonomous navigation and mapping in outdoor environments," In Proceedings of CLAWAR 2006, Brussels, Belgium, 2006.

[9] P. Lamon and R. Siegwart, "3d position tracking in challenging terrain," Proceedings of the Field and Service Robotics FSR 2005, August 2005, 2005.

[10] E. N. G. Dissanayake, S. Sukkarieh and H. Durrant-Whyte, "The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications," *IEEE Transactions on Robotics and Automation, 2001*, 2001.

[11] E. N. S. Sukkarieh and H. Durrant-Whyte, "A high integrity imu/gps navigation loop for autonomous land vehicle applications," *IEEE Transactions on Robotics and Automation, Jun 1999*, 1999.

[12] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Winning the darpa grand challenge," *Journal of Field Robotics*, 2006, accepted for publication.

[13] M. Bellino, Y. Lopez de Meneses, P. Ryser, and J. Jacot, "Lane detection algorithm for an onboard camera," *SPIE proceedings of the first Workshop on Photonics in the Automobile*, 2004.

[14] D. Ferguson and A. Stentz, "Field D*: An Interpolation-based Path Planner and Replanner," in *Proceedings of the International Symposium on Robotics Research (ISRR)*, 2005.

[15] C.-C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas," in *IEEE International Conference on Robotics and Automation*, May 2003.

[16] B. Jensen, R. Philippsen, and R. Siegwart, "Motion detection and path planning in dynamic environments," in *Workshop Proc. Reasoning with Uncertainty in Robotics, International Joint Conference on Artificial Intelligence IJCAI'03*, 2003.

[17] P. J. Besl and N. D. Kay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.