

Spline Templates for Fast Path Planning in Unstructured Environments

Marcel Häselich, Nikolay Handzhiyski, Christian Winkens, Dietrich Paulus

Active Vision Group, AGAS Robotics

University of Koblenz-Landau

Universitätsstr. 1

56070 Koblenz, Germany

{mhaeselich, nikolay, cwinkens, paulus}@uni-koblenz.de

Abstract—This paper addresses the problem of navigating a heavyweight robot autonomously across unstructured environments. The data of a 3D laser range finder are partitioned in real time into equidistant grid cells to locate obstacles and to classify the negotiability of the surface terrain. In addition to the distinction between obstacles or free space, ground measurements are examined to determine the local terrain roughness for each cell.

We present a novel path planning algorithm operating on these classified grid cells. The algorithm is able to avoid positive and negative obstacles (cells without ground measurements), as well as regarding the roughness of the terrain. By applying a cost function, the robot is able to prefer routes across smooth terrain, like streets or trails, over routes across rough and muddy areas. We can determine the necessarily free terrain cells for each spline in the forefront because the splines and the dimensions of the robot are set. In this way we can access the resulting spline templates during runtime and connect them very fast for the path planning. The key feature of this approach is the low computation cost compared to existing approaches.

I. INTRODUCTION

Autonomous navigation in unstructured environments is a current and challenging research topic in robotics. In natural environments, the classification of the terrain surface is essential in order to plan an optimal route. Besides common obstacles like trees, rocks or bushes, the algorithms have to deal with potholes, slopes and ditches. Modern 3D laser range finders (LRFs) provide a detailed and thorough representation of the scenery surrounding the autonomous system. This representation allows a mobile system to extract precise terrain negotiability information as basis for path planning. Our approach uses the data acquired by a 3D LRF to navigate a heavyweight outdoor robot safely through difficult scenarios. In contrast to navigation in structured areas, such as the inside of buildings, factory environments or on urban roads, we developed a flexible and solid approach that is able to react quickly to changes and uncertainties of the environment. Furthermore, our system can detect and cruise terrain in real time and independent from given road material, and is able to detect the road directly from the sensor readings. This provides a great advantage in difficult terrains such as construction zones, dirt tracks or rural environments.

This paper is organized as follows. In Sec. II we present the platform and the deployed sensors, followed by a discussion of the state of the art in Sec. III. The terrain classification

in Sec. IV describes the laser data interpretation and provides the basis for the path planner. Global localization of the robot is depicted in Sec. V. The path planning algorithm itself is explained in Sec. VI. Finally, in Sec. VII we present the experimental results followed by a conclusion in Sec. VIII.

II. HARDWARE SETUP



Fig. 1. The autonomous outdoor robot *Mustang MK 1A*.

As shown in Fig. 1 we use a *Velodyne HDL-64E S2* mounted on top of a 500 kg robot, the *Mustang MK 1A*. The robot possesses an all-wheel steering and drive, can be directed by steering angle and velocity and achieves a top speed of 14 km/h. Considering global positioning, a *DGPS OmniSTAR 8400 HP* and a *Navilock NL-302U GPS* receiver are used. The head of the *Velodyne HDL-64E S2* consists of 64 lasers which are permanently gathering data of the environment as the head spins at a frequency from up to 15 Hz around the upright axis. In doing so, the sensor produces a rich dataset of 1.8 million distance measurements per second. The data of one full rotation are accumulated into one point cloud. As inertial measurement unit (IMU) an *xSens MTi* is used. The system has several other sensors at its disposal which are of minor interest for this approach.

III. RELATED WORK

Most robots have been designed for indoor tasks or navigation in urban areas. In recent years modern innovative

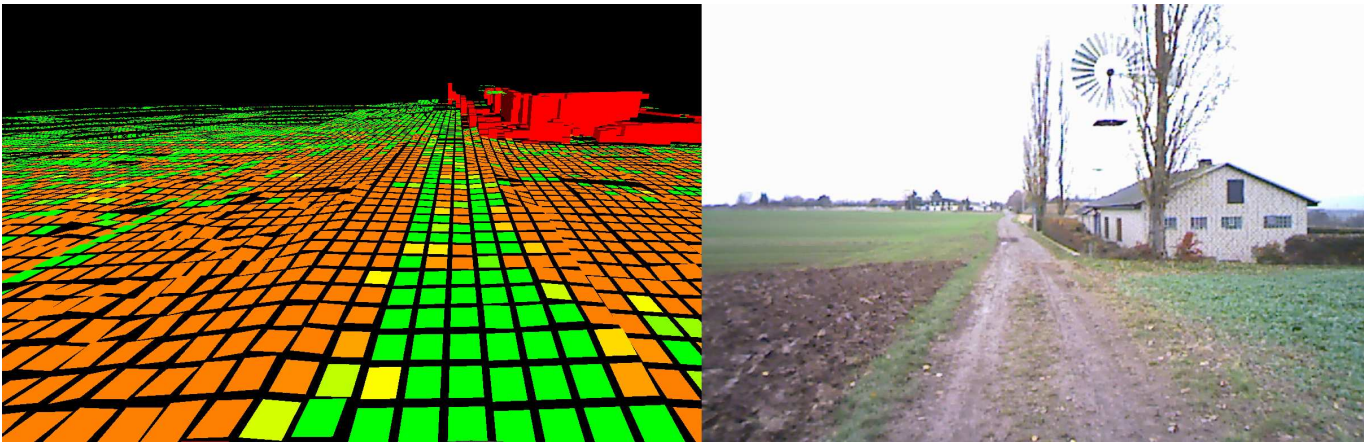


Fig. 2. Left: 3D visualization of the terrain classification. Obstacles are displayed in red and very flat cells in green. Less negotiable cells are vary from yellow to orange (roughness). Right: Camera image of the same scene.

3D sensors captured the market, providing rich datasets that can be processed by high-performance computers in real time. This enables autonomous systems to conquer natural environments by gathering and interpreting these new 3D data sets.

Wurm et al. [20] present a navigation approach which avoids vegetation using laser remission values because their robot is unable to cross unstructured terrain.

A generic approach for trajectory generation is introduced by Howard and Kelly [7]. The authors present an algorithm for planetary rovers, which regards among others terrain roughness, vehicle dynamics and wheel slippage. Their approach is tested for a simulated rover in an simulated environment where the elevation of each point on the terrain is available.

Lacaze et al. [10] present a path planning approach for autonomous vehicles over rough terrain. They use offline simulation results to generate trajectories to avoid obstacles in front of their vehicle. Therefore a grid map is used to determine between obstacles and non-obstacles but terrain negotiability is not taken into account. Philippsen and Siegwart [16] present a user-configurable path planning method which is also working on grid maps. Another approach which uses a grid to determine the negotiability of rough terrain is presented by Ye and Borenstein [22]. Their algorithms work with terrain cells to classify the ground, but only the height values of the cells are computed and used for navigation.

An autonomous off-road robot drives over rolling meadows at speeds up to 35 km/h in the approach presented by Coombs et al. [3]. The authors use clothoids to plan trajectories with a range of 20m to avoid collisions with positive and negative obstacles.

Fonseca et al. [5] use triangulated map overlays to apply graph searching algorithms in order to plan paths in outdoor environments. The map overlays enabled their approach to concern several simultaneous constraints, such as terrain roughness and mobile obstacles.

A control system for autonomous navigation in forested areas is described by Alberts et al. [1]. Their robot is able to

traverse a forest trail at low speed by analyzing the terrain data gathered by two cameras. Another solution to analyze terrain negotiability is stereo vision. Konolige et al. [9] use stereo vision and demonstrate an autonomous system for off-road navigation in unstructured environments. The authors state landmark detection as motivation to prefer a vision based approach over LRFs.

Huang et al. [8] present a method for path planning directly in the image space. Their approach fuses LRF measurements with image data which allows obstacle negotiation even if an object is out of the range of the LRF. An algorithm that learns from operator input and stereo cameras to adapt path planning to local terrain was developed by Bajracharya et al. [2] The authors use a low-capacity map based terrain classifier for short ranges and a higher capacity image-based classifier that is able to learn geometry to classify terrain in the far field.

An evaluation of seven different obstacle detection algorithms based on stereo vision is presented by Rankin et al. [17]. Otte et al. [14] also use stereo vision to find paths directly in the image plane by using a cost function and Yahja et al. [21] combine stereo vision with a quad-tree map representation for outdoor navigation.

IV. TERRAIN CLASSIFICATION

The purpose of the terrain classification [12] is to reduce and preprocess the vast amount of laser measurements of the *Velodyne HDL-64E S2* and to provide the necessary negotiability information about the environment. Therefore the principal component analysis (PCA) of the covariance matrix of each 3D point cloud is applied to determine the shape of the point cloud. As a first step, the large point cloud delivered by our sensor is subdivided into smaller chunks, which can then be analyzed with the help of the PCA. We use a simple 2D equidistant grid, centered around the origin of the sensor, to subdivide the point cloud. Our grid has a size of 100×100 m and each of the cells is scaled approximately 50×50 cm. The output of the algorithm is a classified grid, where each cell can either be *flat*, *obstacle* or *unknown*.

In addition, the *roughness* of each cell is computed, which provides a value from 0 (flat) to 1 (rough) that indicates the negotiability of the terrain surface and the plane normal of each cell is computed. Fig. 2 shows the result of the terrain classification in a rural environment. Obstacles are depicted in red, less negotiable regions are colored from yellow to orange and perfectly negotiable regions are colored in green. The picture shows that the field on each side of the farm road is still classified as less negotiable, yet the road is clearly to prefer since it is almost completely classified as perfectly negotiable. A detailed description of the terrain classification can be found in [12].

V. GLOBAL POSITIONING

For safe navigation an estimation of the current position as accurate as possible is essential. The GPS information gathered by our sensors drifts up to 10m from the actual position. Sometimes it even fails for a short period of time due to satellite concealment caused by huge trees or buildings. For this reason an Extended Kalman Filter (EKF) [18] is used, that fuses the GPS and IMU data. Supplementary and analogous to the optical navigation of a human, the data of the 3D LRF are used to determine the robot's current position. Therefore the Iterative Closest Point (ICP) algorithm is used to match two consecutive scans. Applying the ICP algorithm on the whole 3D point clouds is impossible in real time, instead only the cells classified by the terrain classification as obstacles are used. The obstacles contain the crucial description of the environment and provide a perfect data reduction. A detailed description of the global positioning and the used EKF implementation can be found in [19].

VI. PATH PLANNING

The path planning algorithm can now access the current position and speed of the robot as well as the surrounding terrain to plan a spline-based trajectory to the destination. Some of the computations about the splines are precomputed and used during runtime. This increases the speed immensely and gives us the opportunity to use the Dijkstra's algorithm [4]. To achieve this, we use different sets of templates, where each set belongs to a specific minimal and maximal speed.

A. Speed based Spline Templates

A spline template is an array that contains all boxes that need to be free from obstacles, so that the robot is able to drive the spline path without collision. In order to detect these cells, the broadness of the vehicle is considered. This is done by computing a corridor for the spline according to the broadness plus a small offset in case a steering command is not performed precisely enough by the robot, i.e. due to wheel slippage. Afterwards each cell within the corridor is marked as necessarily free. We created three classes of splines, each designed for a specific speed. The faster the robot drives, the less curvy are the splines the robot uses. Depending on the current speed, determined by the average of GPS and odometry data, the spline class is selected during

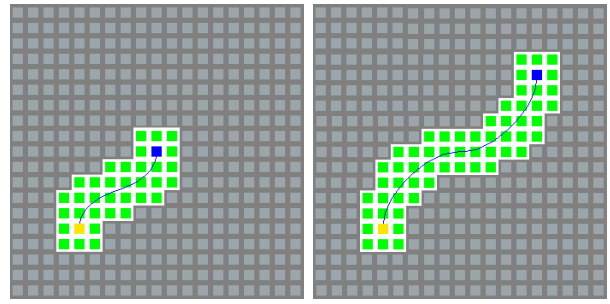


Fig. 3. Two example visualizations for different spline templates. The left side shows a spline for slow, curvy behavior while the right is designed for moderate speed an weaker steering commands.

runtime. An example of two splines of the first two speed classes is given in Fig. 3. The yellow points illustrate the starting points and the splines and destinations are blue. The cells that need to be free from obstacles and which will be regarded later for the terrain costs are marked in green. The first class covers a velocity up to 5 km/h and consists of 20 splines, the second from 5 to 9 km/h with 30 splines and the third one from 10 to 14 km/h with 40 splines. Please note that this selection is used for a vehicle with all-wheel steering.

B. Runtime Computations

The idea of our algorithm on the one hand is to find a way free from obstacles to a given destination from the current position of the robot. Terrain negotiability information acquired by the terrain classification of the 3D points on the other hand is used to prefer paths on smoother terrain. The destination is either within the range of the grid or out of reach. In this case we use the intersection point of the grid with the line of sight to the next destination to determine the direction. Whenever our LRF delivers a new dataset, the following algorithm is used:

- 1) Calculate terrain classification
- 2) Determine current speed and position and select the corresponding spline class
- 3) Use Dijkstra algorithm to construct an optimal path to the destination by connecting feasible splines
- 4) Compute the steering commands

For step 3 a cost function is used to rate the quality of each spline. The costs for an optimal spline \hat{S} are computed as

$$\hat{S} = \underset{S_k}{\operatorname{argmin}} \operatorname{cost}(S_k), \quad (1)$$

where S_k is one combined set of spline templates leading to the destination and k is the number of all possibilities to reach the destination. Let N be the necessary amount of spline templates to travel from the starting point to the destination point. Then the costs for a possible combined spline S_k are determined by the equation

$$cost(S_k) = \sum_{i=1}^N L_S(k_i)(\alpha cost_{St}(k_i) + \beta cost_{Te}(k_i)) . \quad (2)$$

Where $L_S(k_i)$ is the length of element i of the k -th spline. α and β represent the different weights of the particular cost functions and are respectively set to 0.3 and 0.7, because we want to prefer flat terrain unless the distance to drive around a rough region it is much larger.

We penalize the application of spline templates with a strong angular variance, which results in sharp steering commands, by the function $cost_{St}(k_i)$, that is defined as

$$cost_{St}(k_i) = \frac{1}{90} \max_l |S'_{k_i}(0) - S'_{k_i}(l)| , \quad (3)$$

where l , with $l = 1, \dots, m_i$, is a position on the i -th spline and m_i is the amount of possible positions on this spline. The orientation of the vehicle on a certain position on the spline is given by $S'(\cdot)$. The factor $\frac{1}{90}$ is applied to scale the variance of the angles, which reach from 0 to 90 degrees, of the spline to $[0, 1]$.

The driving on rough terrain is penalized by the function $cost_{Te}(k_i)$. To calculate the costs, the roughness $r \in [0, 1]$ of all relevant cells M_{k_i} for a spline template k_i are calculated as

$$cost_{Te}(k_i) = \frac{1}{M_{k_i}} \sum_{j=1}^{M_{k_i}} r_j . \quad (4)$$

Examples for the amount of different relevant cells M_{k_i} for two different splines are shown in green in Fig. 3.

C. Computation of Steering Commands

After the spline templates have been connected the next and last step is to compute the steering commands, consisting of steering angle and velocity, for the robot. Based on a top speed of 14 km/h we can expect, that the maximum distance our vehicle traveled between two full laser rotations, and thus two consecutive terrain classification results, will be 26 cm regarding an update rate of 15 Hz. Due to the possibility of delays like decreased rotation speed depending on temperature or general sensor variance, we chose to segment the spline \hat{S} into equidistant parts with a length of 50 cm. A visualization of the spline segmentation is given in Fig. 4. Between a spline segment d_x and its successor d_{x+1} the angle a_x is computed which forms the first component of the steering command. In order to determine the desired velocity the robot should drive, we consider not only the current steering angle, but also the following segments. Again regarding the maximum speed of our vehicle, we consider the next 9 segments to determine the velocity, thereby regarding the next 5 m for the current command. In this way we are able to slow down ahead of a curve and can stop the vehicle in front of an obstacle in time. The circle in green-white on the path planned in the example shown in Fig. 5 indicates this distance. Since our platform so far showed no recognizable discrepancy between the given commands and the realization of these, we have not implemented a regulation of the steering commands, yet.

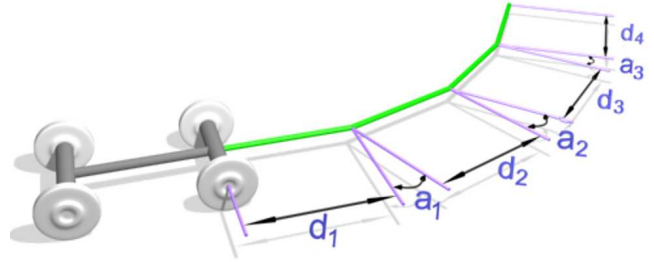


Fig. 4. Segmentation of a spline into equidistant parts to compute the steering commands. Here d_1 to d_4 are examples for the segments and a_1 , a_2 and a_3 are the angles between them.

D. Long Range Navigation

The path planning algorithm is able to avoid obstacles and to prefer certain routes, but is so far unable to proceed on an optimal route to a destination in a large distance. Due to the limited range of the sensors resulting in the grid size of the terrain classification and the missing high-level knowledge of the environment, another layer of route decision is needed. For example, the decision whether to drive around a huge forest or to drive directly through it cannot be efficiently realized without a more far-reaching knowledge. We plan to use different approaches for this task, of which the first is already implemented and tested with our robot. Analog to a car's navigation system, the idea is to use a map database and provide long range waypoints in order to solve situations as mentioned above. As map database we chose OpenStreetMap [13] because it allows a complete download of the whole road material of the planet. Using these maps, we can extract reasonable waypoints or directly import a GPX (also GPS eXchange Format) file.

For the next approach we are currently working on an exploration algorithm that is independent from a map database and improves the survey of the environment, e.g. by identifying points of interest or road courses from the video stream. In long range terms we plan to use our own 3D maps, which will be generated online while exploring at the same time to recognize places we have already visited and to improve the exploration.

VII. EXPERIMENTS

Our approach has been implemented and evaluated during several experiments. The following experiments are designed to demonstrate that our approach is suitable to allow an autonomous robot to reliably navigate in unstructured outdoor environments.

A. Runtime Experiments

Since our platform has no road license yet, we recorded the sensor data from a car's roof and performed the runtime test on the recorded sensor data, played back in real time. To prove the functionality directly on the system, we additionally performed several autonomous driving tests on our own compound. The runtime results of the path planning

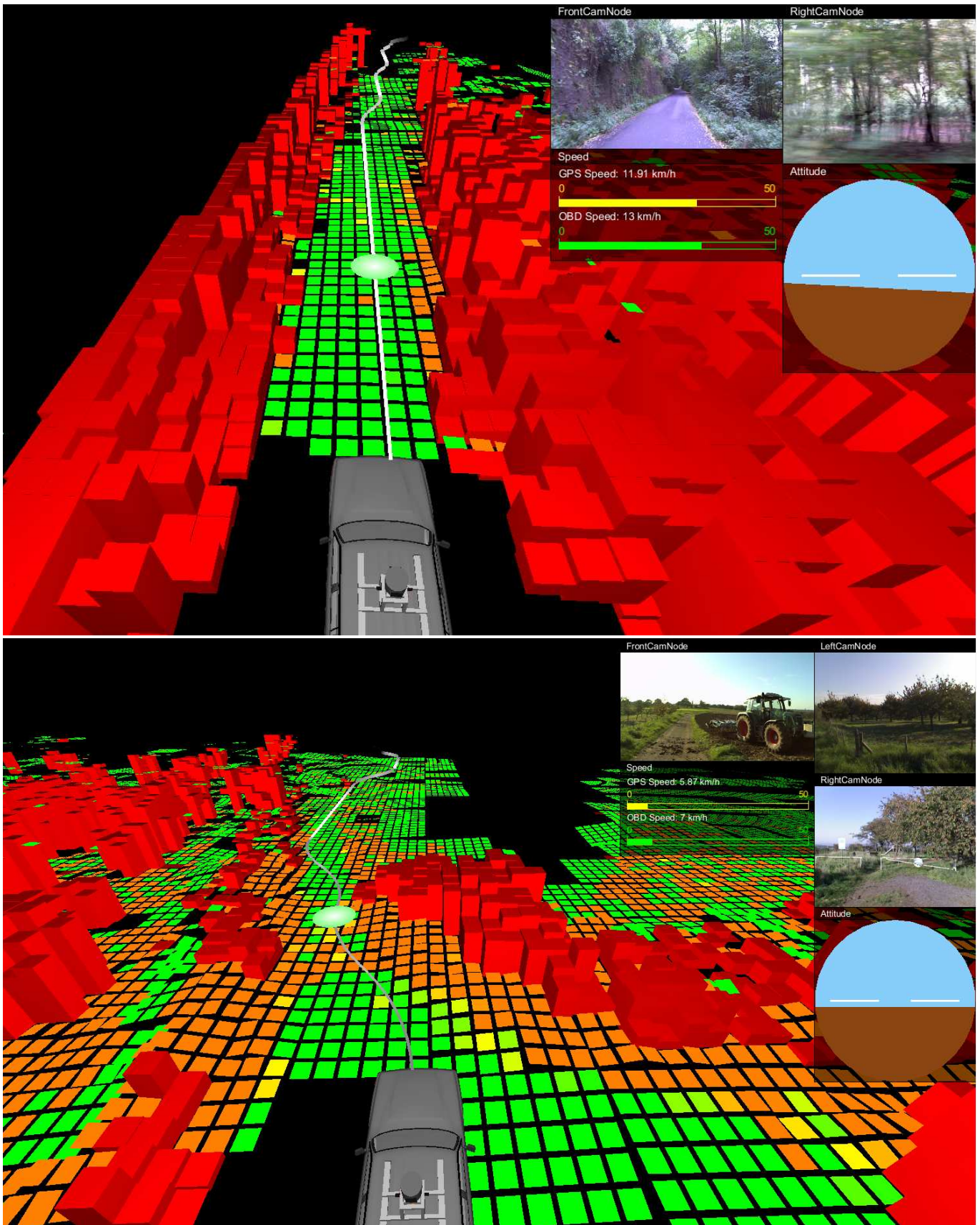


Fig. 5. Real time visualizations of the path (best viewed in color). The path (white) is planned on the negotiable surface (green and orange) between obstacles (red) in a forest region (upper image) and in around an acre. Additional Information, like camera images, alignment and speed, is displayed in the upper right corner of both images.

scenario	duration	preprocessing		path planning	
		mean	standard deviation	mean	standard deviation
campus	5:39 min	36.69 ms	4.59 ms	1.11 ms	0.38 ms
asphalt road	20:40 min	40.10 ms	5.89 ms	1.65 ms	0.92 ms
forest	12:45 min	52.73 ms	7.42 ms	1.38 ms	0.63 ms
farm road	11:34 min	43.93 ms	7.56 ms	1.61 ms	0.92 ms
city	26:53 min	39.62 ms	5.59 ms	2.42 ms	1.71 ms

TABLE I
COMPUTATION TIME IN DIFFERENT SCENARIOS.

are presented in Table I, which shows the results in different environments. The vehicle of course did not follow the commands, therefore most of the time a complete new path was planned every time a sensor update arrived. The system used for the computations is a single laptop, an Intel(R) Core(TM) i7 QM with 1.73 GHz and 8 GB RAM, which is either mounted on the robot or the car. Except for the terrain classification preprocessing for each laser scan no other data reduction is used.

B. Autonomous Navigation

Due to the missing road license we are currently restricted to our own compound, which is a more or less an urban region. However, the test runs we performed on the given terrain deliver promising results (see also our video attachment). We plan to transport the vehicle to a closed off forest region soon and to perform autonomous long time tests.

VIII. CONCLUSION AND FUTURE WORK

The path planner presented in this paper achieves very fast computation time results in all test environments. The approach is able to interpret the data of a 3D LRF in real time to avoid obstacles and to consider the negotiability of the surrounding terrain. It has a range of 50 m for each sensor update, which arrive at a rate of 66 ms. In several autonomous test drives with our robot, as well as on the recorded sensor data the presented approach delivered very good results.

In the future, we plan to travel on our own 3D maps [15] and we want to improve the terrain classification by fusing color and texture features from cameras with the laser data and by considering neighborhood information in a statistical network of the surrounding terrain. Dynamic obstacles will be detected and tracked and their trajectory will be regarded by the path planning to predict their position in the future in order to realize a foresighted behavior. We are also working on an advanced exploration algorithm considering visual pathdetection in combination with *Rapidly-exploring random trees* [11] for long range decisions. The splines will also receive extensions, e.g. the possibility of driving backwards. For further optimization we want to use the A* Algorithm [6] with a heuristic that prefers road segments.

IX. ACKNOWLEDGMENTS

This work was partially funded by Wehrtechnische Dienststelle 51 (WTD), Koblenz, Germany.

REFERENCES

- [1] J. Alberts, D. Edwards, T. Soule, M. Anderson, and M. O'Rourke. Autonomous navigation of an unmanned ground vehicle in unstructured forest terrain. In *Proceedings of the ECSIS Symposium on Learning and Adaptive Behaviors for Robotic Systems*, 2008.
- [2] Max Bajracharya, Andrew Howard, Larry H. Matthies, Benyang Tang, and Michael Turmon. Autonomous off-road navigation with end-to-end learning for the lagr program. *Journal of Field Robotics*, 26(1):3–25, 2009.
- [3] D. Coombs, K. Murphy, A. Lacaze, and S. Legowik. Driving autonomously offroad up to 35 km/h. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2000.
- [4] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerical Mathematics*, pages 269–271, 1959.
- [5] A. R. Fonseca, L. C. A. Pimenta, R. C. Mesquita, R. R. Saldanha, and G. A. S. Pereira. Path planning for mobile robots operating in outdoor environments using map overlay and triangular decomposition. In *Proceedings of the International Congress of Mechanical Engineering*, 2005.
- [6] P. E. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum-cost paths. *IEEE Trans. on Systems Science and Cybernetics*, SSC-4(2):100–107, 1968.
- [7] T. Howard and A. Kelly. Optimal rough terrain trajectory generation for wheeled mobile robots. *International Journal of Robotics Research*, 26(1):141–166, 2007.
- [8] Wesley H. Huang, Mark Ollis, Michael Happold, and Brian A. Stancil. Image-based path planning for outdoor mobile robots. *Journal of Field Robotics*, 26(2):196–211, 2009.
- [9] K. Konolige, M. Agrawal, R. C. Bolles, C. Cowan, M. Fischler, and B. Gerkey. Outdoor mapping and navigation using stereo vision. In *Proceedings of the International Symposium on Experimental Robotics*, 2006.
- [10] A. Lacaze, Y. Moscovitz, N. Declaris, and K. Murphy. Path planning for autonomous vehicles driving over rough terrain. In *Proceedings of the IEEE ISIC/CIRA/ISAS Joint Conference*, 1998.
- [11] S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. In, TR 98-11, 1998.
- [12] F. Neuhaus, D. Dillenberger, J. Pellenz, and D. Paulus. Terrain drivability analysis in 3d laser range data for autonomous robot navigation in unstructured environments. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation*, 2009.
- [13] OpenStreetMap. Openstreetmap - the free wiki world map, March 2011.
- [14] M. W. Otte, S. G. Richardson, J. Mulligan, and G. Grudic. Local path planning in image space for autonomous robot navigation in unstructured environments. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2007.
- [15] J. Pellenz, D. Lang, F. Neuhaus, and D. Paulus. Real-time 3d mapping of rough terrain: A field report from disaster city. In *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics*, 2010.
- [16] R. Philippsen and R. Siegwart. An interpolated dynamic navigation function. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.
- [17] A. Rankin, A. Huertas, and L. Matthies. Evaluation of stereo vision obstacle detection algorithms for off-road autonomous navigation. In *Proceedings of the AUVSI Symposium on Unmanned Systems*, 2005.
- [18] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.
- [19] A. Volk. Verwendung von 3d laserscan-daten zur verbesserung der selbstlokalisierung eines mobilen systems im outdoor-bereich. Master's thesis, University of Koblenz-Landau, Institute for Computational Visualistics, Active Vision Group, Koblenz, 2009.
- [20] K. M. Wurm, R. Kümmerle, C. Stachniss, and W. Burgard. Improving robot navigation in structured outdoor environments by identifying vegetation from laser data. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.
- [21] A. Yahja, S. Singh, and A. Stentz. An efficient on-line path planner for outdoor mobile robots. *Robotics and Autonomous Systems*, 33(1):129–143, 2000.
- [22] C. Ye and J. Borenstein. A method for mobile robot navigation on rough terrain. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004.