

Spline Templates for Fast Path Planning in Unstructured Environments

Marcel Häselich, Nikolay Handzhiyski, Christian Winkens,
Dietrich Paulus

mhaeselich@uni-koblenz.de

Active Vision Group, AGAS Robotics
University of Koblenz-Landau
robots.uni-koblenz.de

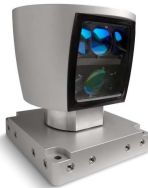
28th September 2011, IROS in San Francisco, USA



Platform and Deployed Sensors



Mustang MK IA



Velodyne HDL-64E S2



Cameras

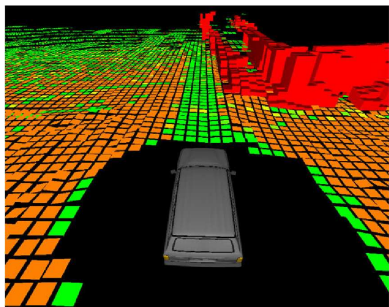
- ▶ **GOAL:** Create an efficient path planning algorithm for unstructured environments
 - ▶ fast computation
 - ▶ data reduction
 - ▶ obstacle avoidance
 - ▶ environment interpretation
 - ▶ regard surface negotiability

Algorithm steps:

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



CALCULATE TERRAIN CLASSIFICATION

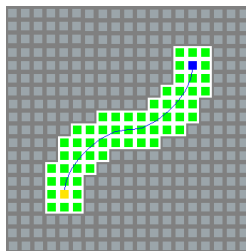
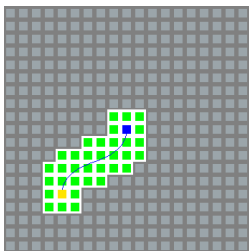


- ▶ Principal component analysis (PCA) [Neuhaus 2009] or Markov random field (MRF) [Häselich 2011] return a grid where a cell can either be obstacle or flat and provides cell negotiability information
- ▶ Subdivide the 3D point cloud into an 2D grid
 - ▶ 100 m × 100 m grid, each cell is 50 cm × 50 cm
 - ▶ centered around the origin of the sensor



Terrain classification video

Definition of a spline template



- ▶ 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 compute these cells, vehicle broadness is considered
- ▶ create different sets for different speeds

Global Positioning System

- ▶ GPS information drifts up to 10 m
 - ▶ Extended Kalman Filter [Volk 2009] fuses GPS, IMU data and wheel odometrie
 - ▶ Iterative Closest Point (ICP) on the reduced LRF data
-
- ▶ current speed and position are predicted, the corresponding spline class can be selected

Costs for an optimal spline \hat{S} are computed as

$$\hat{S} = \underset{S_k}{\operatorname{argmin}} \operatorname{cost}(S_k) \text{ with}$$

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

- ▶ S_k : combined set of spline templates leading to the destination
- ▶ k : number of all possibilities
- ▶ N : necessary amount of spline templates to destination
- ▶ $L_S(k_i)$: length of element i of the k -th spline
- ▶ $\alpha = 0.3$ and $\beta = 0.7$ (prefers flat terrain over distance)

Penalize spline templates with a strong angular variance:

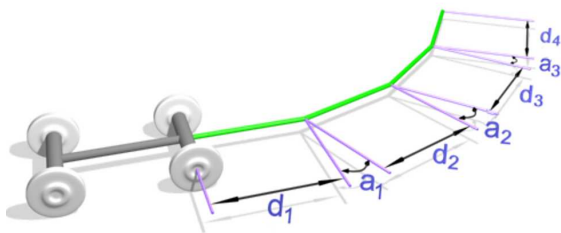
$$\text{cost}_{St}(k_i) = \frac{1}{90} \max_l |S'_{k_i}(0) - S'_{k_i}(l)| ,$$

- ▶ $l = 1, \dots, m_i$: a position on the i -th spline, m_i is the amount of possible positions on this spline.
- ▶ $S'(\cdot)$: orientation and position
- ▶ $\frac{1}{90}$: scales 0 to 90 degrees $\rightarrow [0, 1]$.

The roughness $r \in [0, 1]$ of all relevant cells M_{k_i} for a spline template k_i is calculated as

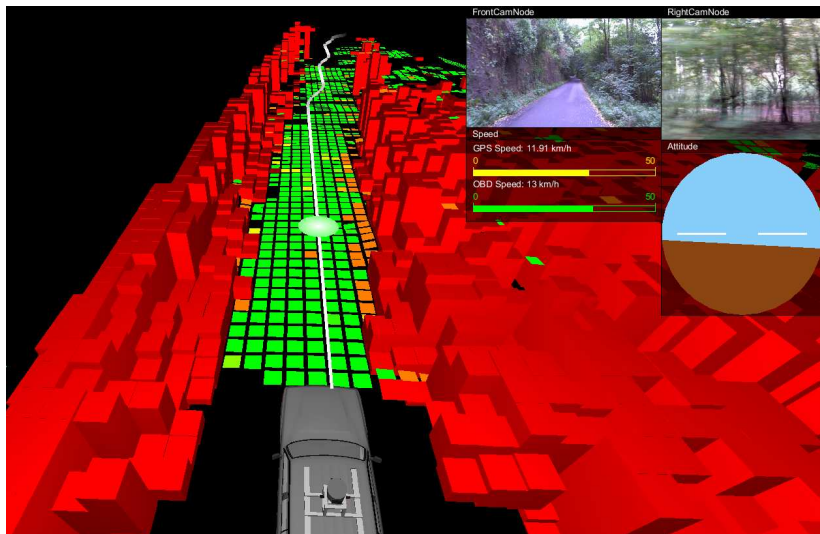
$$\text{cost}_{Te}(k_i) = \frac{1}{M_{k_i}} \sum_{j=1}^{M_{k_i}} r_j .$$

COMPUTE STEERING COMMANDS

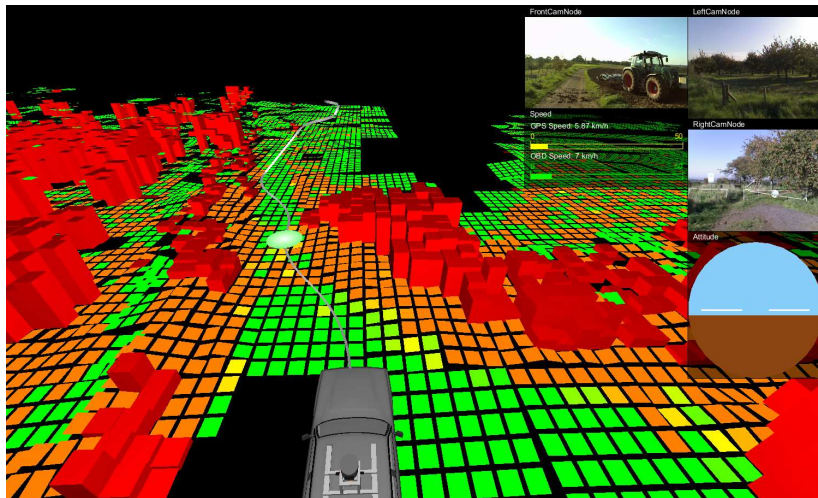


- ▶ segment the spline into equidistant parts
- ▶ d_1 to d_4 are examples for the segments
- ▶ a_1 , a_2 and a_3 are the angles between them

TESTS IN DIFFERENT SCENARIOS



TESTS IN DIFFERENT SCENARIOS



RUNTIME MEASUREMENTS

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: Computation time in different scenarios.

Spline template video

Achievements:

- ▶ very fast computation results in all test environments
- ▶ avoids obstacles and considers negotiability of the surrounding terrain
- ▶ range of 50 m for each sensor update, which arrive each 66 ms
- ▶ confirmed working for our autonomous outdoor robot

Limitations/future work:

- ▶ controller for adjustment on sloping regions
- ▶ dynamic obstacles
 - ▶ particle filter for tracking and prediction
- ▶ Rapidly-exploring Random Trees for long range decisions

Thank you for your attention!

Sources



Häselich, M. and Arends, M. and Lang, D. and Paulus, D.: Terrain Classification with Markov Random Fields on fused Camera and 3D Laser Range Data, 5th European Conference on Mobile Robotics (ECMR), 2011,
<http://www.uni-koblenz.de/~agas/Public/Haeselich2011TCW.pdf>



F. Neuhaus, D. Dillenberger, J. Pellenz and D. Paulus: Terrain Drivability Analysis in 3D Laser Range Data for Autonomous Robot Navigation in Unstructured Environments, 14th IEEE International Conference on Emerging Technologies and Factory Automation, 2009



A. Volk: Verwendung von 3D Laserscan-Daten zur Verbesserung der Selbstlokalisierung eines mobilen Systems im Outdoor-Bereich, master thesis, Universität Koblenz-Landau, Institut für Computervisualistik, Arbeitsgruppe Aktives Sehen, 2009