Markov Random Field Terrain Classification of Large-Scale 3D Maps

Marcel Häselich, Benedikt Jöbgen, Frank Neuhaus, Dagmar Lang and Dietrich Paulus
Active Vision Group
University of Koblenz-Landau
Universitätsstr. 1, 56070 Koblenz, Germany
{mhaeselich,bjoebgen,fneuhaus,dagmarlang,paulus}@uni-koblenz.de

***Abstract*— Simultaneous localization and mapping, drivability classification of the terrain and path planning represent three major research areas in the field of autonomous outdoor robotics. Especially unstructured environments require a careful examination as they are unknown, continuous and the number of possible actions for the robot are infinite. We present an approach to create a semantic 3D map with drivability information for wheeled robots using a terrain classification algorithm. Our robot is equipped with a 3D laser range finder, a Velodyne HDL-64E, as primary sensor. For the registration of the point clouds, we use a featureless 3D correlative scan matching algorithm which is an adaption of the 2D algorithm presented by Olson. Every 3D laser scan is additionally classified with a Markov random field based terrain classification algorithm. Our data structure for the terrain classification approach is a 2D grid whose cells carry information extracted from the laser range finder data. All cells within the grid are classified and their surface is analyzed regarding its drivability for wheeled robots.**

The main contribution of this work is the novel combination of these two algorithms which yields classified 3D maps with obstacle and drivability information. Thereby, the newly created semantic map is perfectly tailored for generic path planning applications for all kinds of wheeled robots.

We evaluate our algorithms on large datasets with more than 137 million annotated 3D points that were labeled by multiple human experts. All datasets are published online and are provided for the community.

I. INTRODUCTION

Autonomous navigation and exploration in unstructured environments is a major challenge in robotics and an active field of research. In order to fulfill complex outdoor tasks, a robot needs enhanced sensing and interpretation abilities to detect drivable regions and to avoid collisions. Our robot is equipped with a Velodyne HDL-64E, a GPS receiver, an inertial measurement unit and several other sensors. While driving, our robot can continuously gather sensor data and register consecutive laser scans in order to create large-scale 3D maps of the environment.

In this work, we combine two existing approaches: a 3D correlative scan matching algorithm that is able to create large 3D maps and a terrain classification algorithm using Markov random fields (MRFs) that is able to classify the data of 3D laser range finders (LRFs). The combination yields novel semantic 3D maps with obstacle and drivability information that are perfectly tailored for path planning applications. We created and annotated three different large maps of difficult environments, provide them for the community and evaluate our approach on each map.

This paper is structured as follows. Related work in the fields of scan matching and terrain classification is presented in Sec. II. Our mapping approach is briefly explained in Sec. III followed by a short introduction to the MRF based terrain classification in Sec. IV. The combination of both approaches and the resulting maps are described in Sec. V. Our evaluation is depicted in Sec. VI and the paper closes with a conclusion in Sec. VII.

II. RELATED WORK

State-of-the-art 3D simultaneous localization and mapping (SLAM) approaches are typically graph-based and consist of two principal components: the so-called front-and backend. The frontend locally aligns point clouds and stores them as a graph, where nodes correspond to the unknown robot poses, and edges correspond to successful scan-matching constraints. Since loop-closures can make the graph inconsistent, the backend is responsible for optimizing the graph globally in order to find a locally (and hopefully globally) optimal and consistent configuration of the graph's poses.

The core of the frontend is algorithm for locally aligning point clouds. In the area of 3D-SLAM, these approaches are often based on the well-known Iterative Closest Point (ICP)-algorithm [1] or his numerous extensions and adaptions which all require an initial guess of the transformation. The normal distribution transform either works in 2D [2] or 3D [11] and subdivides point clouds into cells. Each cell is assigned a normal distribution which locally models the probability of measuring a point. Scans are matched using these cells and Magnusson et al. [12] have shown that results are comparable to the ICP algorithm.

Censi et al. [4] present a scan matching algorithm that works in the Hough domain. The properties of the Hough domain allow for decoupling the estimation of the orientation from the estimation of the translation. In a subsequent work, Censi and Carpin extend the approach to 3D [3].

A method based on the correlation of (2D) laser scans is proposed by Konolige and Chou [10] and is refined by Olson [16] in a highly-successful method called real-time correlative scan-matching (cf. Sec. III).

Steder et al. [19] present an approach for 3D place recognition. The proposed algorithm also produces accurate pose transformations between two scans by comparing salient points.

Considering terrain classification, laser based approaches either work with a 2D LRF, a 2D LRF on stepper motors or a 3D LRF. Wurm et al. [23] use the laser remission values of a 2D LRF on a pan-tilt unit to classify the surface terrain based on the resulting 3D scans. In this way, they detect grass-like vegetation and prefer paved routes with their robot. Another approach for terrain classification is presented by Wolf et al. [22]. Their robot uses a 2D LRF which is oriented to the ground, records data while driving and produces 3D maps using hidden Markov models. The authors are able to differentiate flat areas from grass, gravel or other obstacles.

Vandapel et al. [20] segment 3D distance measurements and classify the segments into three different classes: terrain surface, clutter and wires. Their approach works with a stationary 3D sensor, which returns detailed data, as well as on a mobile platform with a rotating 2D scanning mount.

Ye and Borenstein [24] present an algorithm for terrain mapping with a 2D LRF. Their LRF is mounted at a fixed angle to the ground in front of their robot and creates an elevation map while driving.

A color stereo camera and a 2D LRF are used by Manduchi et al. [13] to detect obstacles. The authors present a color-based classification system and an algorithm that analyses the laser data in order to discriminate between grass and obstacles.

Schenk and Csatho [18] fuse aerial images with 3D point clouds to construct surfaces of urban scenes. Surfaces are represented in a 3D object space coordinate system by patches that store the shape and the boundary of the corresponding surface region.

A stereo pair of digital cameras, an infrared camera and two 2D LRFs on scanning mounts are used by Wellington et al. [21]. The authors apply multiple MRFs that interact through a hidden semi-Markov model that enforces a prior on vertical structures. Their results show that including the neighborhood structure significantly improves obstacle classification.

Coyle et al. [5] present another approach to reaction-based terrain classification for the detection of sand, grass, gravel and asphalt. The authors present a method of interpolating point clouds that uses singular value decomposition, matrix logarithms and Catmull-Rom splines to reduce the need to collect large datasets for algorithm training.

III. POINT CLOUD REGISTRATION

This section briefly describes our SLAM system consisting of a custom frontend solution with a more or less of-the-shelf SLAM backend. Our robot's LRF yields a total of 15 scans per second. We select about one of these scans every second to incorporate it into our map. The frontend needs to register this scan with the previous scan (and potentially other scans for loop closing purposes). As our robot does not have any wheel encoders or other odometry information, we require an algorithm that does not need an initial guess. We use an adaption of a 2D scan matching procedure originally presented by Olson [16] to 3D, published in [15], in order to perform this alignment.

Olson's method is probabilistically motivated and provides a way to efficiently maximize the typical SLAM measurement model

$$p(\boldsymbol{z} \,|\, \boldsymbol{x}_t, \boldsymbol{m}) \quad , \tag{1}$$

where $\boldsymbol{z}$ is the current scan, $\boldsymbol{x}_t$ is the pose of the robot at time t and $\boldsymbol{m}$ is a model of the world, e. g., the last scan. The method is not local, and is thus able to register two 2D scans without an initial guess.

A straight-forward extension of Olson's algorithm to 3D is infeasible because the search space would grow too large. To summarize, we reduce the 3D alignment problem to 2D by retrieving the known roll and pitch angles from our inertial measurement unit. Vertical structures are projected onto a the xy-plane which decouples the z-coordinate estimation from the estimation of x, y and of the yaw angle. Hence, the resulting steps of our frontend are:

1) Project of vertical structures on the xy-plane
2) Compute x, y and yaw angle using the correlative scan-matching approach
3) Estimate z (decoupled from x, y and yaw angle)
4) Perform the registration using ICP and the initial guess

These four steps are each described in detail in our previous publication [15] and form the core of our SLAM frontend. Loop-closures are identified probabilistically using the Dijkstra projection.

The final optimization of our graphs is done using a robust backend that is similar to what is described by Grisetti et al. [8], except that we also use max-mixtures [17] in order to provide more robustness to false loop-closures.

Having built a globally consistent map of the environment, the next step is to classify the individual scans using our terrain classification.

IV. MRF TERRAIN CLASSIFICATION APPROACH

The following section introduces our terrain data representation, the MRF model and the classification process. A 2D equidistant grid centered around the origin of our Velodyne HDL-64E LRF is used to represent the data and to process the large 3D point clouds. The grid has a variable size from $20 \times 20\,\mathrm{m}^2$ up to $100 \times 100\,\mathrm{m}^2$ and each cell c has a size of approximately $50 \times 50\,\mathrm{cm}^2$. The following section briefly introduces our previously published MRF terrain classification [9] which can be applied to single 3D scans. For brevity, features and details are omitted.

Our MRF works directly on the introduced terrain cells. We choose an MRF model introduced for image segmentation tasks by Deng and Clausi [6]. The neighborhood of each cell consists of the eight adjacent cells (except for border cells) whose neighborhood relations are modeled according to the Potts model. Cell features are computed from the data of the LRF. We introduce the cell classes *Road* and *Rough* to determine between easy or hard to pass cells. The class *Obstacle* represents impassable areas for our robot. Regions without measurements are also relevant, since they could contain potholes or descents. They are represented by the

class *Unknown*. All labels combined form our label set

$$\mathcal{L} = \{\textit{Obstacle}, \textit{Road}, \textit{Rough}, \textit{Unknown}\} \quad . \tag{2}$$

Let i, j index a cell position in the terrain grid. The neighborhood energy of a cell computed according to the Potts model was denoted by $\mathrm{E}_{\mathcal{N}_{i,j}}$ and the feature energy is denoted by $\mathrm{E}_{f_{i,j}}$. Then, the complete energy $\mathrm{E}_{i,j}$ of the cell at position i, j is computed as

$$\mathrm{E}_{i,j} = \mathrm{E}_{\mathcal{N}_{i,j}} + \mathrm{E}_{f_{i,j}} \quad . \tag{3}$$

The neighborhood of a terrain cell c at position i, j for the first-order neighborhood system is given by

$$\begin{aligned}\mathcal{N}_{i,j} = \{&c_{i-1,j-1}, c_{i-1,j}, c_{i-1,j+1}, c_{i,j-1}, \\ &c_{i,j+1}, c_{i+1,j-1}, c_{i+1,)}, c_{i+1,j+1}\}\end{aligned} \tag{4}$$

with respect to border cells. The neighborhood energy is computed as

$$\mathrm{E}_{\mathcal{N}_{i,j}} = \sum_{c_{m,n} \in \mathcal{N}_{i,j}} \delta(c_{i,j}, c_{m,n}) \quad , \tag{5}$$

where the function $\delta(c_{i,j}, c_{m,n})$ returns a value of -1 if the classes of the two cells are equal, and $+1$ otherwise, which is in compliance with the Potts model.

The feature energy $\mathrm{E}_{f_{i,j}}$ of a cell is based on our assumption that the features are normally distributed. Its computation is defined as

$$\mathrm{E}_{f_{i,j}} = \sum_k \left(\frac{(f_{i,j_k} - \mu_{i,j_k})^2}{2\sigma_{i,j_k}^2} + \log\left(\sqrt{2\pi}\sigma_{i,j_k}\right) \right) \quad , \tag{6}$$

where f_{i,j_k} is the k-th feature of $c_{i,j}$ and μ_{i,j_k} and σ_{i,j_k} are the mean and the standard deviation of the k-th feature of the class assigned to $c_{i,j}$. Mean and standard deviation of all features need to be computed for all classes which we solve by learning a prior from labeled data in advance. For example, we use the local height disturbance [14] of the laser measurements as a feature. A complete overview and explanation of all features is published in [9].

For classification, the sum of all computed energies $\mathrm{E}_{i,j}$ needs to be minimized, which leads to a maximization of the a posterior probability of the labeling of the terrain. These energies are minimized by finding a label for each cell which fits best for the computed features *and* the labels of the neighbor cells. We apply a Gibbs sampler as introduced by Geman and Geman [7] to solve this task.

V. COMBINED APPROACH

Up to this point, the scan registration and the classification of the scans have been described. The next step is to combine them in order to create a semantic map. Several possibilities exist to incorporate the terrain classification result into the registered map and will be described in the following.

Scan matching and terrain classification can operate simultaneously, each algorithm can process one scan at a time or one of the algorithms can be solely applied first before the second algorithm commences.

Furthermore, each 3D point of the map is rotationally invariant from a global reference frame while the terrain cells can either be aligned to the coordinate system of the local scan or globally. Our approach applies the SLAM algorithm first and the MRF terrain classification is applied afterwards to each scan of the newly created map. Our terrain grids are aligned to a global reference frame defined by the first scan of the map. The resulting semantic 3D map is defined as follows.

Maps of 2D and 3D data are often represented as a mixture of metrical and topological data structures. This kind of representation often does not take the classification results and their properties into account. Common-sense knowledge about the semantic information has to be incorporated into the map creation and classification process in order to enable high-level robotic tasks. We consider semantic mapping as a human-compatible understanding, interpretation and representation of the environment in the form of sensor data. Therefore, the relationship between the perceived environment and the common-sense knowledge has to be established. We define a semantic map for the environment limited to the task domain as

$$\mathcal{M}_{\mathrm{sem}} = \langle \mathcal{M}, \mathcal{V}, \mathcal{A} \rangle \quad , \tag{7}$$

where $\mathcal{M}$ is of a set of maps, $\mathcal{V}$ is a set of links and $\mathcal{A}$ is a structure representing knowledge. The environment is defined as a subset of the world and the task domain is defined as a subset of the environment describing the use of the semantic map for an intended application. In the context of this work, the set of maps $\mathcal{M}$ comprises a metrical map and a set of topological maps as

$$\mathcal{M} = \{M_T, M_{C_1}, \ldots, M_{C_n}\} \quad , \tag{8}$$

where M_T is a topological map representing the scans as nodes and the transformations between the scans as edges and where M_C are metrical maps each representing a single scan. The number of metrical maps is denoted by n. Entities of the semantic map are perceived by sensors and sensor data will be classified into the different classes defined in Eq. 2.

The set $\mathcal{V}$ contains links between corresponding entities of $\mathcal{M}$. Each entity can be assigned different attributes such as *Drivable*, *Non-Drivable* or *Less Drivable*. Attributes can be specialized, if required, e. g., to linguistic terms or real-valued measures.

The structure $\mathcal{A}$ represents knowledge about the relation between entities, classes and attributes and is also known as common-sense knowledge about the task domain. Generally, $\mathcal{A}$ can be defined in an arbitrary way, e. g., as a graph based structure or as an ontology and has to allow for inference. In our approach, we establish the link between entities and common-sense knowledge during the classification process with the MRF. The attributes of the entities are afterwards available for path planning applications.

A visual comparison of a human-annotated ground truth against one of our semantic maps for the Koblenz campus is presented in Fig. 1. The different classes from Eq. 2 are represented by different colors (*Obstacle*: red, *Road*: gray,

Rough: brown and *Unknown*: black) and the evaluation is described in the following section.

VI. RESULTS

The following evaluation aims to highlight strengths and limitations of the combined algorithms by applying them to scenarios which divert greatly from each other. Therefore, correctness and robustness of the algorithms are tested in the Koblenz city forest, on the campus of the University of Koblenz-Landau and on a farm road outside of Koblenz.

A. Datasets

Three datasets have been created and annotated by human experts in order to evaluate the presented terrain classification approach. The first dataset, *KoblenzForestMap*, has been recorded along a sloped forest road in the municipal forest of Koblenz and consists of 26 million 3D points with annotated ground truth. This map is characterized by many trees and the fact that the forest terrain surface is almost completely occluded. The second dataset, *KoblenzCampusMap*, consists of over 64 million 3D points from the university campus and is characterized by buildings, asphalt and road segments, some meadows and small vegetated areas. During the map creation, several loops have successfully been closed and the map is very compact due to the repetitive visits of locations. The first two scenarios motivate an environment where the class *Rough* is overrepresented. Hence, the third and last map, *KoblenzFarmMap*, has been created on a farm road with a lot of fields, meadows and vegetation. This scenario is the most complex one due to rainy weather, puddles, translucent vegetation and vibrations from the dirt road. Sensor misreadings arise from wet surfaces as some rays from the LRF are reflected and yield false measurements.

All datasets combined contain more than 137 million annotated and registered 3D points and are provided to the community. The viewing and annotation tool is published online together with all datasets[1].

B. Evaluation

For the classification process, the true positives (TP), the false positive (FP) and the false negative (FN) are determined by comparing the classification result against the annotated ground truth. Furthermore, precision is calculated as

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{9}$$

and recall is calculated as

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad . \tag{10}$$

The configuration of MRF terrain classification remains fixed for all experiments, e. g., without any variations of feature or neighborhood components. Two different types of evaluation are conducted. The first one is a point-wise evaluation, which means that every annotated point from the ground truth is compared to the class of the terrain cell that includes it. Since the result of the terrain classification is used for collision-free path planning, the algorithm should classify a terrain cell as an *Obstacle* if any *Obstacle* point lies within that cell. Any *Obstacle* point from the ground truth is considered a FP for the point-wise classification if the terrain cell is

[1] http://robots.uni-koblenz.de/datasets

TABLE I

EVALUATION OF THE TERRAIN CLASSIFICATION ON A LARGE FOREST ROAD MAP WHICH CONSISTS OF OVER 26 MILLION 3D POINTS. IN ALL THREE TABLES, HUMAN-ANNOTATED GROUND TRUTH IS COMPARED TO THE OUTPUT OF THE TERRAIN CLASSIFICATION ALGORITHM.

Number of Scans:	263	3D Points [No.]:	26,433,792
Grid Size [m×m]:	30×30	Annotated Points [No.]:	26,288,749
Cell Size [m×m]:	0.51×0.51	Classified Points [No.]:	21,700,151

Label	Point-wise				
	TP [No.]	FP [No.]	FN [No.]	Precision	Recall
Obstacle	14,950,332	1,199,927	372,315	0.93	0.98
Street	3,984,321	293,956	899,287	0.93	0.82
Rough	322,804	123,988	1,029,202	0.72	0.24

Label	Cell-wise				
	TP [No.]	FP [No.]	FN [No.]	Precision	Recall
Obstacle	333,438	21,493	6,035	0.94	0.98
Street	121,645	5,060	14,099	0.96	0.90
Rough	19,626	10,314	16,953	0.66	0.54

TABLE II

EVALUATION OF THE TERRAIN CLASSIFICATION ON A LARGE CAMPUS MAP WHICH CONSISTS OF OVER 64 MILLION 3D POINTS.

Number of Scans:	644	3D Points [No.]:	64,134,528
Grid Size [m×m]:	60×60	Annotated Points [No.]:	63,038,659
Cell Size [m×m]:	0.51×0.51	Classified Points [No.]:	60,486,169

Label	Point-wise				
	TP [No.]	FP [No.]	FN [No.]	Precision	Recall
Obstacle	29,408,823	3,641,866	1,865,557	0.89	0.94
Street	19,102,630	1,781,887	4,413,649	0.91	0.81
Rough	2,070,151	1,262,633	2,535,066	0.62	0.45

Label	Cell-wise				
	TP [No.]	FP [No.]	FN [No.]	Precision	Recall
Obstacle	692,713	124,771	50,980	0.85	0.93
Street	983,515	78,095	194,465	0.93	0.83
Rough	239,296	140,748	98,613	0.63	0.71

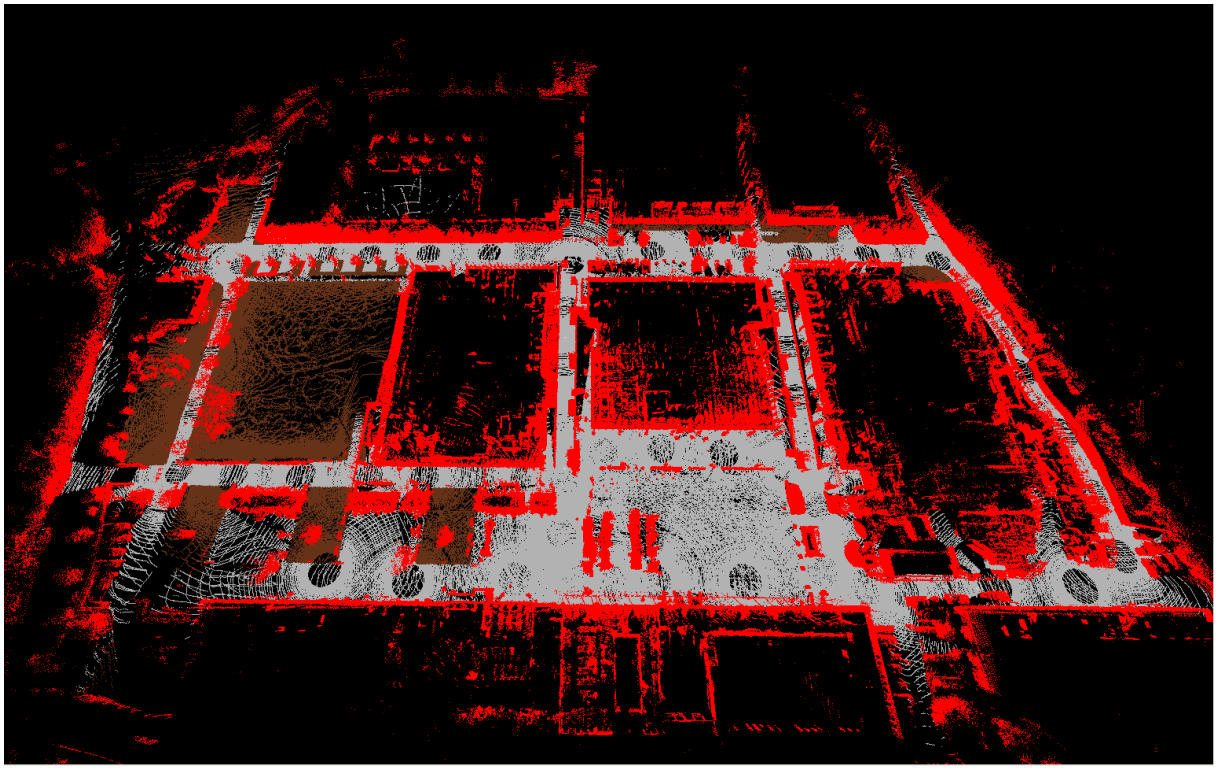
TABLE III

EVALUATION OF THE TERRAIN CLASSIFICATION ON A LARGE FARM ROAD MAP WHICH CONSISTS OF OVER 48 MILLION 3D POINTS.

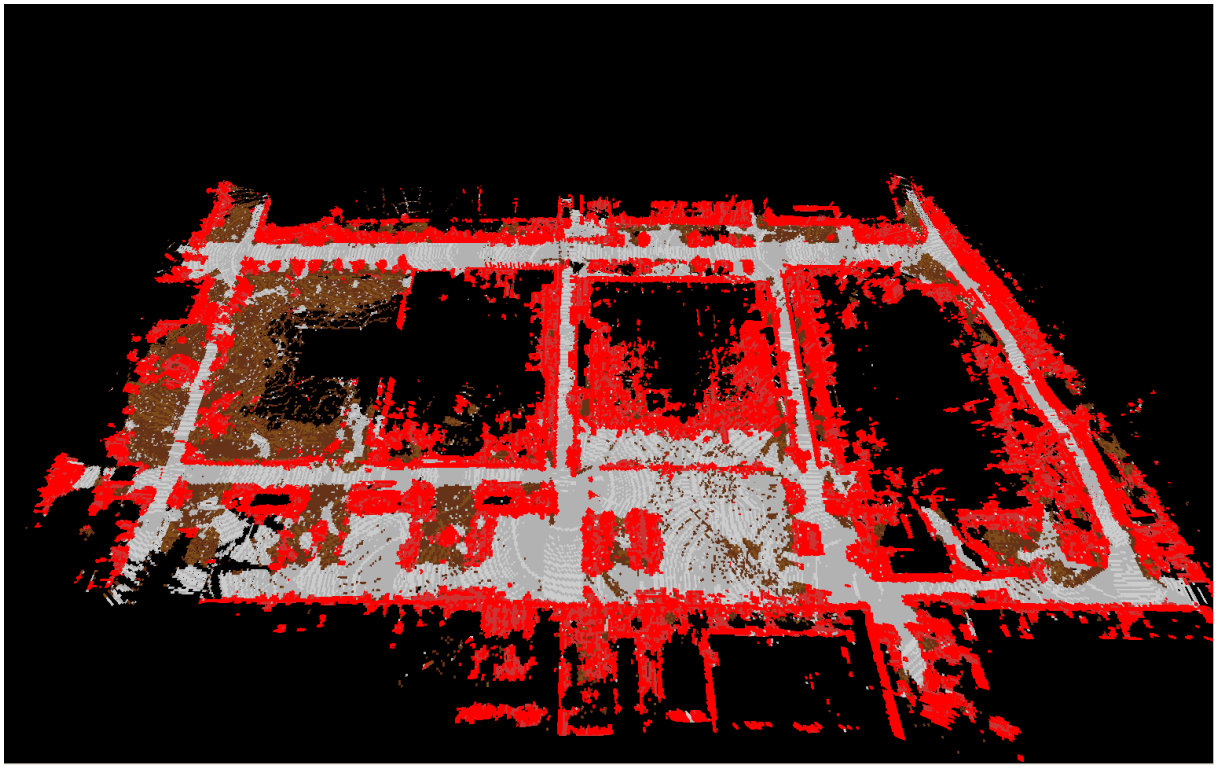
Number of Scans:	426	3D Points [No.]:	48,950,715
Grid Size [m×m]:	100×100	Annotated Points [No.]:	48,192,619
Cell Size [m×m]:	0.51×0.51	Classified Points [No.]:	46,751,857

Label	Point-wise				
	TP [No.]	FP [No.]	FN [No.]	Precision	Recall
Obstacle	18,908,187	2,836,187	1,412,879	0.87	0.93
Street	7,492,263	4,379,626	3,645,895	0.63	0.67
Rough	7,495,081	566,796	7,077,159	0.93	0.51

Label	Cell-wise				
	TP [No.]	FP [No.]	FN [No.]	Precision	Recall
Obstacle	785,886	138,567	76,399	0.85	0.91
Street	308,685	452,815	42,460	0.40	0.88
Rough	1,162,399	83,640	556,536	0.93	0.68



(a) Ground truth with 63 million 3D points that have been labeled by human experts.



(b) Resulting semantic 3D map with more than 2.2 million classified terrain cells.

Fig. 1. Example result of a semantic 3D terrain classification map (best viewed in color). The two images show a visual comparison between the ground truth (a) and the result from our approach (b) at the university campus in Koblenz. Scans are classified separately and the resulting map consists of overlapping terrain classification grids. The evaluation for this scenario is depicted in Table II.

not an *Obstacle*. Any *Non-obstacle* point from the ground truth is in turn not considered a FP if the terrain cell is an *Obstacle* and contains at least one *Obstacle* point from the ground truth. For all other cases, the class for the 3D point from the ground truth is compared to the class of the terrain cell it lies in. The second evaluation takes all other 3D points within a terrain cell into account and is therefore called the cell-wise evaluation. A terrain cell is considered a TP if the majority ($\geq 50\,\%$) of points from the ground truth share its class. The same exception for collision-free path planning from the point-wise evaluation applies here. Any *Obstacle* terrain cell is considered a TP for the cell-wise classification if the terrain cell contains at least one *Obstacle* point from the ground truth. Any *Non-obstacle* terrain cell is in turn not considered a FP if the terrain cell is an *Obstacle* and contains at least one *Obstacle* point from the ground truth. Results for varying grid sizes in the three different scenarios are shown in Tables I, II and III.

VII. CONCLUSIONS AND FUTURE WORKS

The presented MRF terrain classification approach for maps is able to create semantic 3D maps containing classified terrain cells and drivability information that are perfectly tailored for path planning applications. Our evaluation on more than 137 million 3D points revealed high precision and recall values. Especially obstacle classification with recall values of 0.98 (30 m grid), 0.94 (60 m grid) and 0.93 (100 m grid) prove its robustness. In the forest scenario, the terrain surface was mainly occluded by trees and vegetation and the class *Rough* was underrepresented and suppressed by the MRF neighborhood component. This indicates that the roughness feature used in this work requires a sufficient number of measurements within each terrain cell in order to work as a distinct measure.

Considering future work, our path planning algorithm is now able to operate directly on the map. Hence, our path planner can be extended to plan long trajectories in advance, before the robot starts a new task. An extension of the terrain classification algorithm to 3D is another useful development in order to enable navigation in structures like tunnels or multi-storied buildings and scenarios.

VIII. ACKNOWLEDGMENTS

This work was partially funded by the Deutsche Forschungsgemeinschaft (DFG) under research contract PA 599/11-1.

REFERENCES

[1] P. Besl and N. McKay. A Method for Registration of 3-D Shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.

[2] P. Biber and W. Straßer. The Normal Distributions Transform: A new Approach to Laser Scan Matching. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2743–2748, 2003.

[3] A. Censi and S. Carpin. HSM3D: Feature-Less Global 6DOF Scan-Matching in the Hough/Radon Domain. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1585–1592, 2009.

[4] A. Censi, L. Iocchi, and G. Grisetti. Scan Matching in the Hough Domain. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2739–2744, 2005.

[5] E. Coyle, E. Collins, and R. Roberts. Speed-independent Terrain Classification using Singular Value Decomposition Interpolation. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, pages 4014–4019, 2011.

[6] H. Deng and D. A. Clausi. Unsupervised Image Segmentation Using a Simple MRF Model With a New Implementation Scheme. *Pattern Recognition*, 37(12):2323–2335, 2004.

[7] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distribution and Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[8] G. Grisetti, R. Kuemmerle, C. Stachniss, and W. Burgard. A Tutorial on Graph-Based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2:31–43, 2010.

[9] M. Häselich, M. Arends, N. Wojke, F. Neuhaus, and D. Paulus. Probabilistic Terrain Classification in Unstructured Environments. *Journal of Robotics and Autonomous Systems*, 61(10):1051–1059, 2013.

[10] K. Konolige and K. Chou. Markov Localization using Correlation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1154–1159, 1999.

[11] M. Magnusson, T. Duckett, and A. Lilienthal. Scan Registration for Autonomous Mining Vehicles Using 3D-NDT. *Journal of Field Robotics*, 24:803–827, 2007.

[12] M. Magnusson, A. Nüchter, C. Lörken, A. Lilienthal, and J. Hertzberg. Evaluation of 3D Registration Reliability and Speed - A Comparison of ICP and NDT. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3907–3912, 2009.

[13] R. Manduchi, A. Castano, A. Talukder, and L. Matthies. Obstacle Detection and Terrain Classification for Autonomous Off-road Navigation. *Autonomous Robots*, 18(1):81–102, 2004.

[14] 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*, pages 1686–1689, 2009.

[15] F. Neuhaus, A. Mützel, and D. Paulus. Fast Registration of 3D Laser Scans without Initial Guess. *Journal of Imaging Science and Technology*, 2014. to appear.

[16] E. Olson. Real-Time Correlative Scan Matching. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4387–4393, 2009.

[17] E. Olson and P. Agarwal. Inference on Networks of Mixtures for Robust Robot Mapping. *International Journal of Robotics Research*, 32:826–840, 2013.

[18] T. Schenk and B. Csatho. Fusing Imagery and 3D Point Clouds for Reconstructing Visible Surfaces of Urban Scenes. In *Proceedings of the IEEE GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas*, pages 11–13, 2007.

[19] B. Steder, G. Grisetti, and W. Burgard. Robust Place Recognition for 3D Range Data based on Point Features. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1400–1405, 2010.

[20] N. Vandapel, D. Huber, A. Kapuria, and M. Hebert. Natural Terrain Classification Using 3-D Ladar Data. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5117–5122, 2004.

[21] C. Wellington, A. Courville, and A. Stentz. Interacting Markov Random Fields for Simultaneous Terrain Modeling and Obstacle Detection. In *Proceedings of Robotics Science and Systems*, pages 1–8, 2005.

[22] D. F. Wolf, G. Sukhatme, D. Fox, and W. Burgard. Autonomous Terrain Mapping and Classification Using Hidden Markov Models. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2026–2031, 2005.

[23] K. 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*, pages 1217–1222, 2009.

[24] 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*, pages 3863–3869, 2004.