

# Terrain Classification With Conditional Random Fields on Fused 3D LIDAR and Camera Data

Stefan Laible<sup>1</sup>, Yasir Niaz Khan<sup>2</sup> and Andreas Zell<sup>1</sup>

**Abstract**—For a mobile robot to navigate safely and efficiently in an outdoor environment, it has to recognize its surrounding terrain. Our robot is equipped with a low-resolution 3D LIDAR and a color camera. The data from both sensors are fused to classify the terrain in front of the robot. Therefore, the ground plane is divided into a grid and each cell is classified as either asphalt, cobblestones, grass or gravel. We use height and intensity features for the LIDAR data and Local ternary patterns for the image data. By additionally taking into account the context-sensitive nature of the terrain, the results can be improved significantly. We present a method based on Conditional Random Fields and compare it with a Markov Random Field based approach. We show that the Conditional Random Field model is better suited for our task. We achieve an average true positive rate of 94.2% for classifying the grid cells into the four terrain classes.

## I. INTRODUCTION

A mobile robot that operates in outdoor environments is faced with challenges quite different from those occurring in factories and office buildings. A major issue here is the changing and differently navigable terrain. On the one hand, the knowledge of the terrain around the robot is essential for safe and efficient navigation, on the other side, the terrain can give clues about where the robot is located. Thus, terrain classification is a fundamental ability for further tasks such as path planning and localization.

In previous work [LKBZ12] we classified the terrain using a low-resolution 3D LIDAR and a color camera separately, each sensor with its own advantages and drawbacks. For both types of sensor data we used a grid-based approach, which means that the image (in the case of the camera), and respectively the ground plane (in the case of the laser) are divided into a Cartesian grid, and then each grid cell is classified individually as either asphalt, cobblestones, grass or gravel. Now, in this work, in order to use the data of both sensors in the same coordinate system, we project the ground plane onto the image to get the corresponding pixels for feature extraction and classification.

Important for improving the classification results is the insight that terrain appears in contiguous areas — a fact that is ignored when the grid cells are considered only independently of each other. Only very rarely will one find terrain that varies greatly within a small range. To account for this, a suitable mathematical model is needed, which

exists in the form of a Markov Random Field (MRF) or a Conditional Random Field (CRF). An MRF is a generative model, which means that it models the joint probability of labels and features. In contrast, a CRF models the conditional probability of the labels given the features directly; such a model is called discriminative. Although MRFs have proven themselves to give good results for many tasks, including terrain classification, we will show that MRFs are not suitable for our specific problem, whereas a CRF achieves very good classification results.

In Sec. II we discuss related work where MRFs and CRFs are used. Our outdoor robot and its sensors can be seen in Sec. III. In Sec. IV we describe our terrain-classification method, that is, the LIDAR- and camera-based classification, the sensor fusion, and the context-sensitive classification of the grid cells, where we will look at MRFs and CRFs and their differences in more detail, as this is essential for our work. In Sec. V we present our experiments and results and conclude in Sec. VI.

## II. RELATED WORK

There exist several approaches for terrain classification that use range and image data in combination as mentioned in the Related Works section of [LKBZ12]. In our work, however, we not only differentiate between passable and non-passable terrain but also classify the type of terrain using a camera and a low-resolution LIDAR.

Our work is inspired by Häselich *et al.* [HALP11]. They use data from the high-resolution 3D LIDAR Velodyne HDL-64E S2 in addition to color and texture information from color cameras to classify the terrain into three classes: road, rough and obstacle. To take into account the context-sensitivity of the individual terrain grid cells, they apply an MRF and get a recall ratio of about 90%.

A couple of works show that CRFs are more appropriate than MRFs for many classification tasks. In [KH03] they compare both models for the task of detecting man-made structures in images, where they also divide the images into grids. The CRF yields higher detection rates with lower false positives. Multiscale CRFs are used in [HZCP04] for the task of image labeling. Their CRFs model local and global structures and yield better results than a MRF, which requires stricter independence assumptions.

A two-stage training is used in [FVS09] for identifying and localizing object classes in images. In the first stage, superpixels (small image regions obtained by segmentation) are trained with an SVM, and in the second stage they refine

<sup>1</sup>S. Laible and A. Zell are with the Chair of Cognitive Systems, Computer Science Department, University of Tübingen, Sand 1, 72076 Tübingen, Germany {stefan.laible, andreas.zell} at uni-tuebingen.de

<sup>2</sup>Y. N. Khan is with the Department of Machine Learning and Robotics, University of Stuttgart, Germany yasir.niaz at gmail.com

their results by using a CRF. A similar approach is used in [LEN10] for detecting shadows in images. They first use a decision tree to find potential shadow contours and then optimize the results with a CRF. The model we use for classifying the cells of the terrain grid comes closest to theirs.

An excellent introduction to CRFs is given in [SM12], and also in [NL11] with a focus on applications in Computer Vision.

### III. HARDWARE

Our outdoor robot *Thorin*, which can be seen in Fig. 1, was used for the experiments in this work. The robot is equipped with a Mini-ITX computer featuring a dual-core CPU and has, among other sensors, an AVT Marlin F-046 C Color Camera and a Nippon Signal FX6 3D LIDAR.



Fig. 1. Outdoor robot *Thorin* with a Nippon Signal FX6 3D LIDAR and an AVT Marlin F-046 C color camera

Marlin F-046 C Color Camera	
Vendor	Allied Vision Technologies GmbH
Resolution	638 x 480 pixels
Frame Rate	Max. 53 Hz
FX6 3D LIDAR	
Vendor	Nippon Signal Co., Ltd.
Resolution	29 x 59 data points
Frame Rate	8 or 16 Hz
Range	16 m
Scan Area	50° (hor.) and 60° (vert.)

The color camera is able to take pictures at a frame rate of up to 53 Hz. It has both manual and automatic white balance as well as an auto shutter and auto gain function. All three auto functions were enabled for the experiments.

The FX6 sensor uses a pulse laser in the near-IR range. It is lightweight and robust and largely illumination-independent, so that it works with ambient light of up to 100,000 Lux. A drawback of the sensor, however, is its low resolution with only 29 x 59 data points. In addition to the distance an intensity value is returned for each point, which indicates the proportion of the emitted light which arrives back at the sensor.

### IV. TERRAIN CLASSIFICATION

We use a two-stage approach for terrain classification. In the first stage, laser and image features are extracted for each terrain grid cell. Here, each grid cell is projected onto the image to get the corresponding image patch. Using these features Random Forests then assign to each cell a terrain label.

In the second stage we exploit the context-sensitive nature of the terrain grid cells to improve the classification results. We discuss Markov Random Fields (MRF) and Conditional Random Fields (CRF), which are probabilistic graphical models for context-sensitive classification. To find an optimal label configuration of the terrain grid, we use a Gibbs sampler with a simulated-annealing scheme.

#### A. 3D LIDAR- and Camera-Based Classification

In the first stage of our terrain classification method we train per-cell classifiers for the LIDAR and the image data separately. By converting the range values of the LIDAR into Cartesian coordinates we get a 3D point cloud wherein a RANSAC-based method finds the ground plane on which the robot drives (see Fig. 2(a)). This plane is divided into a grid and the grid cells are to be classified into the given terrain classes. As features we use the height of the points above the ground plane, and the distribution of the intensity values. Details can be found in [LKBZ12]. The cells in which the height exceeds a threshold are marked as obstacles and are not further classified. Since the LIDAR has a very low resolution, only for cells that are closer than about two meters to the front of the robot enough laser measurements are available to provide a meaningful analysis of the terrain (see Fig. 2(b)). For the small number of cells, however, the classification works very well, especially when distinguishing vegetation from non-vegetation.

In previous work [KKZ11] we used a grid-based method also for the camera-based classification, where the image is divided into equally-sized grid cells, and local features are computed across the grid. In this work, the grid cells of the ground plane are projected onto the image and determine the relevant image patches for feature extraction (see Fig. 2(c)). For the projection we need to know the transformation between the LIDAR and the camera coordinate system, which we compute using a calibration method with a checkerboard [Bou08], [UH05]. As features we use Local ternary patterns (LTP) [TT10], which are an extension of Local binary patterns (LBP) [OPH96]. LBPs are essentially histograms of binary-encoded differences in pixel brightness. While LBP is parameter-free, LTP has a parameter to threshold pixel differences into three values and yields a 512-dimensional feature vector.

For classification we use Random Forests [Bre01]. A Random Forest is a collection of multiple decision trees, in which each tree takes different samples of the training instances, and each node considers a different random subset of the features. Each tree votes for one label, and the label with the majority of votes is assigned to the corresponding cell. This allows us to assign to each grid cell  $i$  the probability

$p(y_i | x_i)$  that this cell has label  $y_i$  given the features  $x_i$ , as the proportion of the trees that have voted for  $y_i$ . For grid cells where data of both sensors are present, the probabilities are combined:  $p = kp_{lidar} + (1-k)p_{image}$ , with a weighting factor  $k$ .

### B. Context-Sensitive Classification

Fig. 2(d) shows a typical result of the sensor-fusion process and although quite good, many grid cells are still misclassified. As mentioned above, classifying each grid cell individually ignores the context-sensitive nature of the terrain that occurs in contiguous areas.

Let  $\mathbf{y}$  be a configuration of labels for the entire terrain grid and  $\mathbf{x}$  the corresponding features. Then the classification problem can be stated as finding the label configuration  $\mathbf{y}^*$  that maximizes the probability  $p(\mathbf{y} | \mathbf{x})$  of the configuration given the observed features  $\mathbf{x}$ . So far we have implicitly assumed this probability to be:

$$p(\mathbf{y} | \mathbf{x}) = \prod_{i=1}^M p(y_i | x_i) \quad (1)$$

where  $M$  is the number of grid cells and the probabilities  $p(y_i | x_i)$  are obtained by the Random Forest classifiers independently for each cell.

In [HALP11] they successfully use a Markov Random Field (MRF) to account for spatial dependencies between grid cells. We will now look at MRFs and Conditional Random Fields (CRF) in more detail. We compare the two models and look at the differences in order to understand why the MRF does not work for our problem.

An MRF defines a family of joint probability distributions by means of an undirected graph. It explicitly attempts to model the joint probability distribution  $p(\mathbf{y}, \mathbf{x})$  and factorizes as follows:

$$p(\mathbf{y} | \mathbf{x}) \propto p(\mathbf{y}, \mathbf{x}) = p(\mathbf{x} | \mathbf{y})p(\mathbf{y}) \quad (2)$$

$$p(\mathbf{x} | \mathbf{y}) = \prod_{i=1}^M p(x_i | y_i) \quad (3)$$

$$\begin{aligned} &= \prod_{i=1}^M \prod_{k=1}^K \frac{1}{\sqrt{2\pi}\sigma_{i_k}} \exp\left(-\frac{(x_{i_k} - \mu_{i_k})^2}{2\sigma_{i_k}^2}\right) \\ &= \exp\left(-\sum_{i=1}^M \sum_{k=1}^K \frac{(x_{i_k} - \mu_{i_k})^2}{2\sigma_{i_k}^2} + \log\left(\sqrt{2\pi}\sigma_{i_k}\right)\right) \\ p(\mathbf{y}) &= \frac{1}{Z} \prod_{(i,j) \in \mathcal{N}} \psi(y_i, y_j) \\ &= \frac{1}{Z} \prod_{(i,j) \in \mathcal{N}} \exp(-\beta_{i,j} \delta(y_i, y_j)) \\ &= \frac{1}{Z} \exp\left(-\sum_{(i,j) \in \mathcal{N}} \beta_{i,j} \delta(y_i, y_j)\right) \end{aligned} \quad (4)$$

Eq. 2 – 4 point out the two components of an MRF:

- A conditional probability distribution  $p(\mathbf{x} | \mathbf{y})$  that models how the labels generate the features
- A probability distribution  $p(\mathbf{y})$  that models the a-priori probability of a label configuration

The conditional probabilities  $p(x_i | y_i)$  are calculated for each grid cell by assuming a Gaussian distribution (see Eq. 3), where  $x_{i_k}$  is the  $k$ th component of the  $K$ -dimensional feature vector of cell  $i$ .  $\mu_{i_k}$  and  $\sigma_{i_k}$  are the corresponding means and standard deviations, which are learned from training data for every class and every feature component. The probability distribution  $p(\mathbf{y})$  does not depend on the features and is only tractable by assuming certain neighborhood relationships. That is, the probability that the  $i$ th cell has label  $y_i$  given all the other cells is equal to the probability given only the cells of the neighborhood  $\mathcal{N}$ .

$$p(y_i | Y \setminus \{y_i\}) = p(y_i | \{y_j : (i, j) \in \mathcal{N}\}) \quad (5)$$

This Markovian property of the Random Field makes it, according to the Hammersley-Clifford theorem [GG84], equivalent to a Gibbs field that has the form of Eq. 4.  $\beta_{i,j}$  is a factor that weights the neighbors impact and  $\delta(y_i, y_j)$  is  $-1$  if  $y_i = y_j$  and  $+1$  otherwise.  $Z$  is the normalization factor and sums over all labels  $y \in Y$ .

The MRF model has some shortcomings. E.g., we have to implicitly model the probability distribution  $p(\mathbf{x})$  of the features. This distribution can be very complex, and making simplifying assumptions, like modeling it as a Gaussian distribution, can be a too strong restriction. The pairwise factor  $\psi$  encourages agreement, but the way in which it does so is inflexible. The probability that two neighboring cells have the same label should be higher when the corresponding features are similar, and vice versa, but  $\psi$  in this model is independent of the features  $\mathbf{x}$ .

An alternative is to model the terrain grid as a CRF. A CRF is a discriminative model and it models the conditional distribution  $p(\mathbf{y} | \mathbf{x})$  directly, which is all that is needed for classification. It does not include a model of the probability distribution  $p(\mathbf{x})$ , which is difficult to model and not required anyway. A CRF is better suited to including rich, overlapping features.

$$p(\mathbf{y} | \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(-\lambda \sum_{i=1}^M \phi_i(y_i, x_i) - \sum_{(i,j) \in \mathcal{N}} \psi_{i,j}(y_i, y_j, x_i, x_j)\right) \quad (6)$$

$$\phi_i(y_i, x_i) = -\log(p(y_i | x_i)) \quad (7)$$

$$\psi_{i,j}(y_i, y_j, x_i, x_j) = \mathbf{1}_{\{y_i \neq y_j\}} \exp(-\beta(x_i - x_j)^2) \quad (8)$$

This CRF model [LEN10] can be seen in analogy to Eq. 2 – 4 defining the MRF model. In the CRF the conditional

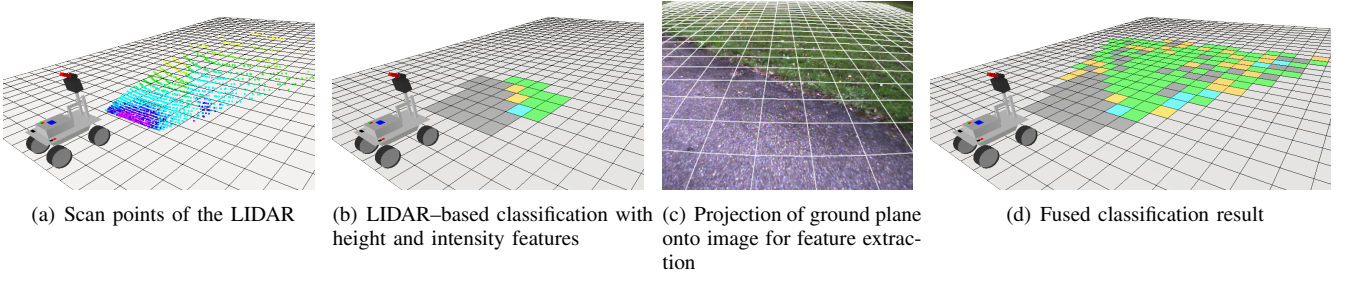


Fig. 2. Fusion of LIDAR- and image-based terrain classification. Each cell of the terrain grid is classified based on height and intensity features of the corresponding scan points of the LIDAR. To integrate the image data, the terrain grid is projected onto the image and then, for every projected cell, features based on Local ternary patterns are extracted. (Gray: asphalt, blue: cobblestones, green: grass, yellow: gravel)

probability distribution  $p(\mathbf{y} | \mathbf{x})$  is modeled directly as a Gibbs distribution. The factor  $\phi_i(y_i, x_i)$  models the feature-dependent component and  $\psi_{i,j}(y_i, y_j, x_i, x_j)$  models the label distribution. The influence of each of the two components is controlled by the parameter  $\lambda$ . It is now no longer necessary to model the feature distribution  $p(\mathbf{x})$  and we get the conditional distribution  $p(y_i | x_i)$  as output of the Random Forest classifier, which as we shall see, provides significantly better results. While in Eq. 4 the same dissimilarity penalty is imposed regardless of  $\mathbf{x}$ , we choose the factor  $\psi$  as proposed in [BJ01] to account for this. The idea behind Eq. 8 is that the probability that two neighboring grid cells belong to the same terrain class is high, but if they belong to different classes, their appearance (measured by feature vector  $x$ ) must also differ. It turned out that using LTPs here takes too much computation time, so we used simpler features, namely the average RGB color values. Thus, we are using LTPs in Eq. 7 as global features, which describe the overall appearance of a terrain patch, and in Eq. 8 we are using the average color to describe the relative change between neighboring patches.

### C. Inference with Gibbs-Sampler

After observing the features  $\mathbf{x}$ , we are interested in the labeling  $\mathbf{y}^*$  which is the single most likely labeling given the new input  $\mathbf{x}$ . This can be stated as an energy minimization problem:

$$\begin{aligned} \mathbf{y}^* &= \arg \max_{\mathbf{y}} p(\mathbf{y} | \mathbf{x}) \\ &= \arg \max_{\mathbf{y}} \frac{1}{Z(\mathbf{x})} \exp \left( - \sum_F E_F(\mathbf{y}_F, \mathbf{x}) \right) \\ &= \arg \min_{\mathbf{y}} \sum_F E_F(\mathbf{y}_F, \mathbf{x}) \end{aligned} \quad (9)$$

So maximizing the probability  $p(\mathbf{y} | \mathbf{x})$  is the same as minimizing the sum of the energies  $E_F$  of all factors  $F$ . The same is true for the MRF. The energy equivalents of the MRF factors are:

$$E_1(\mathbf{y}, \mathbf{x}) = \sum_{i=1}^M \sum_{k=1}^K \frac{(x_{i_k} - \mu_{i_k})^2}{2\sigma_{i_k}^2} + \log \left( \sqrt{2\pi} \sigma_{i_k} \right) \quad (10)$$

$$E_2(\mathbf{y}) = \sum_{(i,j) \in \mathcal{N}} \beta_{i,j} \delta(y_i, y_j) \quad (11)$$

In the same way, the energy equivalents of the CRF factors are:

$$E_1(\mathbf{y}, \mathbf{x}) = -\lambda \sum_{i=1}^M \log(p(y_i | x_i)) \quad (12)$$

$$E_2(\mathbf{y}, \mathbf{x}) = \sum_{(i,j) \in \mathcal{N}} \mathbf{1}_{\{y_i \neq y_j\}} \exp(-\beta(x_i - x_j)^2) \quad (13)$$

Note that the energy term in Eq. 13 additionally depends on the features  $\mathbf{x}$ .

Unfortunately, to get the most likely labeling  $\mathbf{y}^*$ , we would have to compute the energies of all possible label configurations  $\mathbf{y} \in Y^M$ , which is an exponentially large number, and so not tractable. We therefore use an approximate inference method, namely a Gibbs sampler with a simulated-annealing scheme [GG84], [BKYZ96], like in [HALP11]. In this scheme, the label configuration is changed iteratively until a convergence criterion is reached. In each iteration, for every cell a new label is sampled from the conditional probability distribution that describes the probability that this cell has a certain label given all the other cells.

## V. EXPERIMENTS AND RESULTS

In our experiments we consider four types of terrain that are often encountered, namely asphalt, cobblestones, grass, and gravel. There is great variations within the terrain classes. For example, the grass has varying height and density, some spots are covered with moss. The pattern of the cobblestones varies and gravel and asphalt show different textures at different spots.

The ground in front of the robot is divided into a grid with a cell size of 20 cm by 20 cm, which is a high enough resolution for subsequent tasks such as path planning. For the LIDAR-based classification only cells with at least ten scan points are considered. Therefore, because of the low resolution of the FX6, only cells that are closer than about two meters to the front of the robot can be classified with this sensor. The projection of the terrain grid determines the size of the individual image patches. Only patches with at least 200 pixels are considered, so cells that are closer than about three to four meters are classified with the image data.

To evaluate the different classifiers we need a set of frames (scans and corresponding images) with ground-truth

data for the terrain grid. We therefore hand-labeled 135 images, which build the basis for the labeling of the terrain grid. This labeling was refined using the scan points of the LIDAR. Then a 10-fold cross validation was performed on this data set.

It turns out that the Gaussian model yields very poor results in our case. Only regarding the LIDAR data, the true positive rate is 49.5%. Using the simulated-annealing scheme only moderately improves the classification result to 54.9%. For the image data the results using the Gaussian model are even worse with 33.8%. With such a bad feature-dependent classification, it makes no sense to try to improve the classification by considering the neighborhood of the cells. The poor results show that the assumption of a Gaussian distribution for the features used in our work does not hold.

TABLE I  
CLASSIFICATION RESULTS

Classification method	True positive rate in %
Image-based	80.4
Fusion of LIDAR and image data	81.5
Conditional Random Field	94.2

True positive rates after 10-fold cross validation for the image-based classification with Local Ternary Patterns (LTP), the fused classification, and the classification with a Conditional Random Field.

For training the Random Forests, we used 100 trees for each. Using only the LIDAR data and the Random Forest classifier we get a true positive rate of 93.1%. This result seems very good, but it also has to be considered that only a small area in front of the robot can be classified hereby. Therefore, it can not be directly compared with the other results. Using Random Forest with the LTP features of the image (where a threshold value of 2 gave the best results) yields a rate of 80.4%. By fusing both sensor data this can be improved to 81.5%, whereby each sensor was equally weighted. Again, the reason that this improvement is so low is due to the small area which is covered by the laser.

For the CRF we set  $\lambda = 0.5$  and  $\beta = 2.0$ . With this setting the CRF achieves a true positive rate of 94.2%, which is a huge improvement. A few typical classification results can be seen in Fig. 3, where transitions between terrain types are shown, as these are the most interesting cases.

Since we want to use the classification on the robot in real-time, we are interested in the runtimes of the algorithms. The following table shows the average runtimes of the main parts, namely the feature extraction of the LIDAR and image data, the initialization of the CRF, which corresponds to the fused classification, and finally the simulated-annealing scheme (using a CPU with 3.20GHz).

The features of the LIDAR data are very fast and simple to compute and the total computation takes only 5.0 ms. The initialization of the CRF takes 8.2 ms in average. The Gibbs sampler with the simulated-annealing scheme also does not

TABLE II  
AVERAGE RUNTIMES OF THE MAIN PARTS OF THE ALGORITHM

	Average time [ms]	Std. dev. [ms]
LIDAR features	5.0	1.1
Image features	123.4	4.4
Initialization of CRF	8.2	1.1
Simulated annealing	11.4	11.1
Total	148.0	17.7

take much time, but it has a higher standard deviation as it stops when the error falls below a certain threshold, which can vary greatly. Since the feature-dependent classification alone is already good, we initialize the annealing process with this classification and start with a low temperature, which accelerates the convergence. The real bottleneck is the calculation of the LTPs, so our focus in future work lies here. We can speed up things by parallizing, or we have to use a different local image descriptor, which achieves similar results, but which is faster to compute.

## VI. CONCLUSIONS

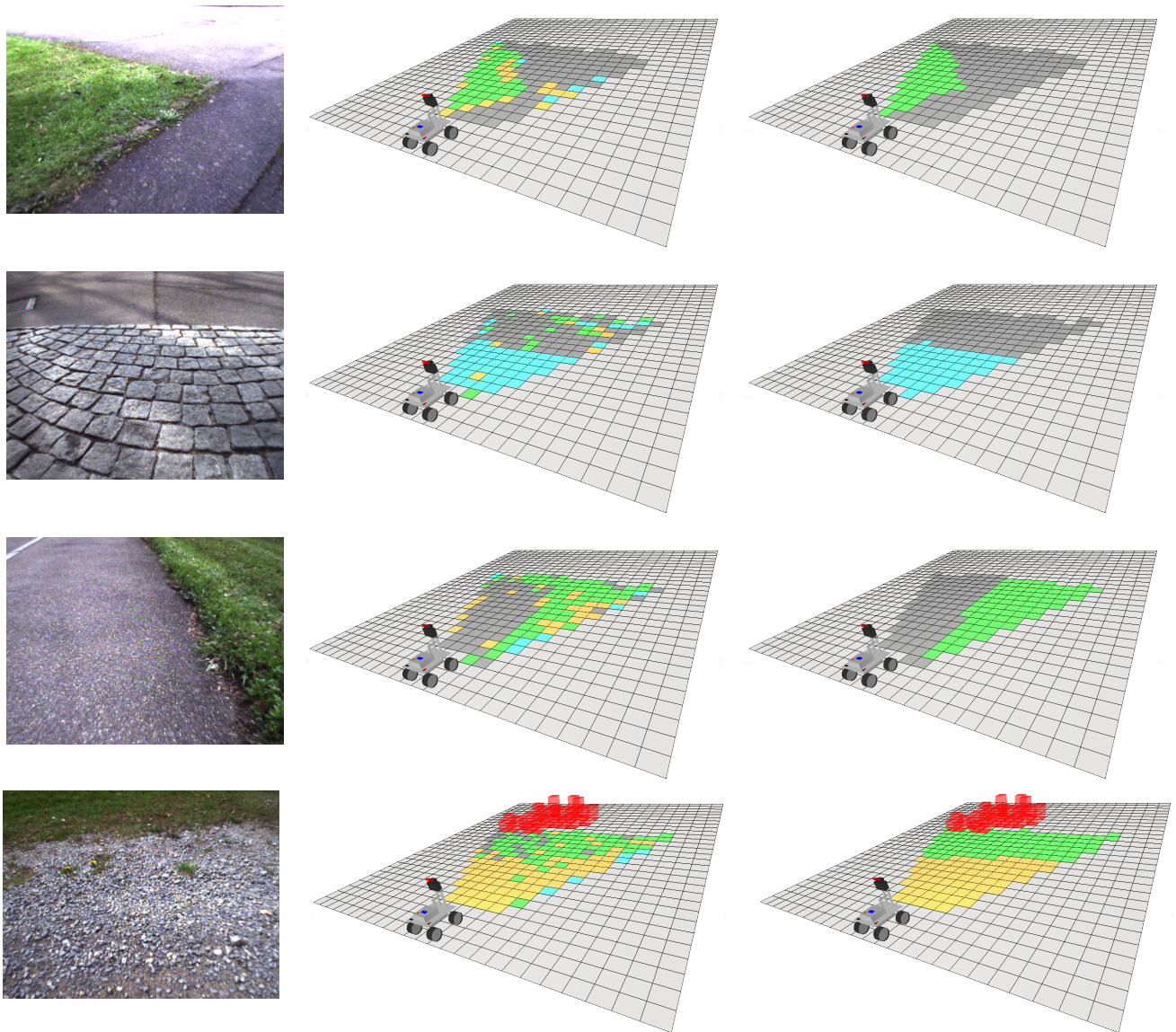
We presented a method for context-sensitive terrain classification based on 3D LIDAR and camera data. We discussed in greater detail MRFs and CRFs, and showed the advantages of the latter. In a classification setting with four terrain classes the classification with a CRF got the best results with a true positive rate of 94.2%. The reason that the MRF does not work for our problem lies in the type of features that we use. Since we not only differentiate between passable and non-passable terrain but also classify the type of terrain, we need a model that is suited to more complex dependencies between the features. For simple features, that can be modeled as Gaussians, however, the MRF can provide great results as shown in [HALP11].

In order to build a local map of the environment we have to take into account several consecutive frames. For that purpose we plan using an Occupancy Grid Map where at each time step, as a preprocess, the current classification result is obtained using the method presented here. In this way, we obtain a classification that takes into account spatial and temporal dependencies.

## REFERENCES

- [BJ01] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV*, pages 105–112, 2001.
- [BKYZ96] M. Berthod, Z. Kato, S. Yu, and J. Zerubia. Bayesian image classification using markov random fields. *Image and Vision Computing*, 14:285–295, 1996.
- [Bou08] J. Y. Bouguet. Camera calibration toolbox for Matlab, 2008.
- [Bre01] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [FVS09] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009.
- [GG84] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.





(a) Images of transitions between two terrain classes (b) Fusion of LIDAR- and image-based classification (c) Final classification with a Conditional Random Field

Fig. 3. Terrain classification with Conditional Random Fields on fused LIDAR and image data. The left column shows images of terrain transitions captured by the robot. The middle column shows the results of the classification with Random Forests, which contain many wrongly classified cells. In the right column the results of the classification using a CRF are shown, which are significantly better. This image is best viewed in color. (Gray: asphalt, blue: cobblestones, green: grass, yellow: gravel, red: cells with high elevations ( $> 0.15\text{m}$ ))

- [HALP11] M. Häselich, M. Arends, D. Lang, and D. Paulus. Terrain classification with markov random fields on fused camera and 3D laser range data. In *Proceedings of the 5th European Conference on Mobile Robotics (ECMR)*, pages 153–158, 2011.
- [HZCP04] X. He, R. S. Zemel, and M. Á. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *CVPR (2)*, pages 695–702, 2004.
- [KH03] S. Kumar and M. Hebert. Discriminative fields for modeling spatial dependencies in natural images. In *NIPS*, 2003.
- [KKZ11] Y. N. Khan, P. Komma, and A. Zell. High resolution visual terrain classification for outdoor robots. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1014–1021, Barcelona, Spain, nov. 2011.
- [LEN10] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Detecting ground shadows in outdoor consumer photographs. In *European Conference on Computer Vision*, 2010.
- [LKBZ12] S. Laible, Y. N. Khan, K. Bohlmann, and A. Zell. 3D lidar- and camera-based terrain classification under different lighting conditions. In *Autonomous Mobile Systems (AMS), 2012 22nd Conference on*, Stuttgart, Germany, 2012.
- [NL11] S. Nowozin and C. H. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3-4):185–365, 2011.
- [OPH96] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, January 1996.
- [SM12] C. A. Sutton and A. McCallum. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2012.
- [TT10] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *Trans. Img. Proc.*, 19(6):1635–1650, June 2010.
- [UH05] R. Unnikrishnan and M. Hebert. Fast extrinsic calibration of a laser rangefinder to a camera. Technical Report CMU-RI-TR-05-09, Robotics Institute, Pittsburgh, PA, July 2005.