Probabilistic Terrain Classification in Unstructured Environments

Marcel Häselich, Marc Arends, Nicolai Wojke, Frank Neuhaus, Dietrich Paulus

Active Vision Group, AGAS, Department of Computer Sciences,
University of Koblenz-Landau, Universitätsstr.1, Koblenz, Germany

robots.uni-koblenz.de

Abstract

Autonomous navigation in unstructured environments is a complex task and an active area of research in mobile robotics. Unlike urban areas with lanes, road signs, and maps, the environment around our robot is unknown and unstructured. Such an environment requires careful examination as it is random, continuous, and the number of perceptions and possible actions are infinite. We describe a terrain classification approach for our autonomous robot based on Markov Random Fields (MRFs) on fused 3D laser and camera image data. Our primary data structure is a 2D grid whose cells carry information extracted from sensor readings. All cells within the grid are classified and their surface is analyzed in regard to traversability for wheeled robots.

Knowledge of our robot's egomotion allows fusion of previous classification results with current sensor data in order to fill data gaps and regions outside the visibility of the sensors. We estimate egomotion by integrating information of an IMU, GPS measurements, and wheel odometry in an extended Kalman filter. In our experiments we achieve a recall ratio of about 90 % for detecting streets and obstacles. We show that our approach is fast enough to be used on autonomous mobile robots in real time.

Keywords: Markov random fields, terrain classification, sensor fusion

1. Introduction

Autonomous navigation in unstructured environments is a current and challenging task in robotics. Mobile systems need a detailed interpretation of the surrounding terrain to avoid obstacles and to regard the traversability of the surface. Modern 3D laser range finders (LRFs) provide a rich and thorough picture of the environment in the form of 3D distance measurements. However, the amount of data acquired by 3D LRFs is often too large for direct use in path planning algorithms. Therefore, as a first step, a reduction of the large point cloud and an efficient data structure is necessary. Our work was motivated by the terrain analysis performed by Neuhaus et al. (1), where a two dimensional grid structure was introduced to provide fast access to traversability estimates.

Differentiation between different surfaces from laser range measurements alone is a difficult task. A second type of sensor can provide more information about surface structures. By calibrating three cameras to our LRF we are able to access the fused data in one coordinate system. This allows us to determine color and texture information of the 3D points in the field of view of each camera. In unstructured environments, classification of the terrain can be challenging due to sensor noise, varying density of the data, egomotion or percussions on rough terrain. For that reason we apply a Markov random field (MRF) to add context-sensitive information to the terrain classification, which models the relationships in our data structure.

In order to interpolate data gaps and regions with sparse sensor data, our MRF accesses previous classification results. An extended Kalman filter (EKF) works on fused inertial measurement unit (IMU), Global Positioning System (GPS), and wheel odometry data to

Email addresses: mhaeselich@uni-koblenz.de (Marcel Häselich), marends@uni-koblenz.de (Marc Arends), nwojke@uni-koblenz.de (Nicolai Wojke), fneuhaus@uni-koblenz.de (Frank Neuhaus), paulus@uni-koblenz.de (Dietrich Paulus) estimate the egomotion of our robot. The result is refined using the 2D Iterative Closest Point (ICP) algorithm based on virtual 2D scans extracted from the data of our 3D LRF. Based on our assumption that the virtual 2D scans computed from the 3D distance measurements yield a sufficient representation of the local area, the 2D ICP algorithm provides an adequate refinement of the estimated pose while consuming less runtime than its 3D equivalent.

In this work, we extend our former publication (2) by an egomotion estimation which enables us to access terrain classification results from previous time steps. Further, we extend and restructure the related work section and give detailed descriptions and illustrations of the various steps of our algorithm.

Our goal is to determine the traversability of the surrounding terrain with a MRF in real time based on the sensors described in Section 2. Therefore, we discuss related work in Section 3 before describing our terrain classification approach with MRFs in detail in Section 4. Experiments and results are depicted in Section 5. A conclusion is given in Section 6.

2. Hardware

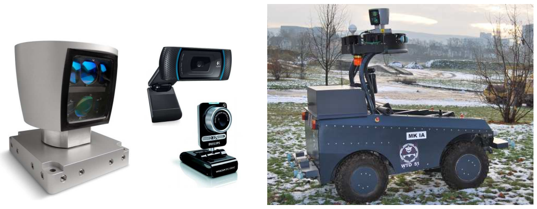

Figure 1: Deployed sensors and robot: A 3D laser range finder and two different commercially available cameras are used to perceive the environment. Sensors are mounted on top of a 500 kg outdoor robot.

As shown in Figure 1 we use a Velodyne HDL-64E S2 (3) and two different camera types. The head of the Velodyne consists of 64 lasers which permanently gather data of the environment as the head spins at a frequency of up to 15 Hz around the upright axis. The sensor thereby produces a rich dataset of 1.8 million distance measurements per second. The data points of one full rotation are accumulated into one point cloud. A Logitech HD Pro Webcam C910 (4) is installed to the front and two Philips SPC1300NC (5) cameras are fixed on the left and the right side of the construction. The sensors are either mounted on top of a 500 kg robot (autonomous driving), the Mustang MK I (cf. Figure 1), or on a car (recording of sensor data). On each platform we use a single laptop, an Intel(R) Core(TM) i7 QM with 1.73 GHz and 8 GB RAM for all tasks. The Mustang MK I possesses an all-wheel steering and drive, can be directed by steering angle and velocity and achieves a top speed of 14 km/h. We further use a Navilock NL-302U GPS receiver, an xSens MTi IMU, and a Speed Wedge SW01 radar-based speed sensor.

3. Related Work

This work was motivated by results from image processing, terrain classification, and probabilistic robotics. The following presentation of the state of the art is therefore separated into image processing (Section 3.1), terrain classification (Section 3.2), and egomotion estimation (Section 3.3).

3.1. Image Processing

Image segmentation is a fundamental task in many computer vision applications. MRFs have been frequently used in this field, especially for segmentation tasks.

Szirányi et al. (6) address the problem of the large amount of computing power required for Markovian approaches. They introduce a fully-parallel architecture for MRF segmentation of images and show that the Markovian labeling approach can be implemented in fully parallel cellular network architectures.

Meas-Yedid et al. (7) use a MRF clustering approach for color segmentation based upon color and spatial information. Their approach proves robust against noise, color changes, illumination changes, and blurring during the performed experiments.

A MRF image segmentation model that combines color and texture features is presented by Kato and Pong (8). Segmentation is obtained by classifying pixels into different pixel classes, which are represented by multi-variate Gaussian distributions either computed from training data or estimated from the input image.

Qazi et al. (9) present a segmentation methodology with robust parametric approximations proposed for multichannel linear prediction error distribution. They use a region-size-based energy term with the conventional Potts energy model and present improved results in terms of percentage errors of color texture segmentation in the case of high-resolution multispectral satellite images.

D‘Angelo et al. (10) provide a MRF description of an unsupervised color image segmentation algorithm. Their system is based on a color quantization of the image in the Lab color space and uses a fuzzy k-nearest neighbors algorithm.

Besides color images, RGB-D sensors such as the Microsoft Kinect emerged and granted the opportunity of color, texture and depth data in one dataset. Herbst et al. (11) use an RGB-D camera and apply a multi-scene MRF model to detect objects that moved between multiple visits to the same scene. By combining shape, visibility and color cues they were able to detect objects even without texture within the scenes.

A MRF that integrated high-res image data into low-res range data was presented by Diebel and Thrun (12). Their MRF exploits the fact that discontinuities in range and coloring tend to co-align and recovers the range data at the same resolution as the image data.

Camera-based classifications using MRFs have various applications and have proven to be a valuable tool in the image processing domain over the past years. The 3D distance measurements gathered by our LRF are more difficult to handle as regions of sparse data, dense data and even regions without any measurements occur and need to be handled efficiently.

3.2. Terrain Classification

There exist various approaches to classify the terrain surrounding an autonomous mobile robot platform. Especially image- or laser-based strategies are widely spread when terrain traversability information is needed.

Image-based strategies either use a single, stereo or combined setup of digital and infrared cameras. Konolige et al. (13) and Alberts et al. (14) both use stereo vision approaches to maneuver a vehicle through unstructured environments. Stereo vision allows them to extract traversable regions from the camera video streams. Furthermore, Vernaza et al. (15) present a camera-based terrain classification approach for the DARPA LAGR program. Their approach uses a MRF that classifies image data of a stereo system into obstacles or ground regions for an autonomous robot. Khan et al. (16) present a comparison of multiple approaches to color-camera-based terrain classification for outdoor mobile robots based on local features. Their approach uses random forests for classification and is able to perform in various weather conditions.

Negative obstacles (non-traversable regions underneath the ground level) present a difficult challenge in non-urban environments. Thermal infrared images have the characteristic that negative obstacles remain warmer than the surrounding terrain in the night. Rankin et al. (17) therefore combine thermal signatures and stereo range data to determine the terrain traversability. Morton and Olson (18) describe a terrain classifier for detecting both positive and negative obstacles. Their classifier utilizes the change in height and the distance between observations from adjacent surfaces as well as the point density in discretized cells.

Laser-based approaches either work with a 2D, a 2D sensor on stepper motors or a 3D LRF. Wurm et al. (19) 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 can detect grass-like vegetation and prefer paved routes with their robot. Another approach for terrain classification is presented by Wolf et al. (20). Their robot uses a 2D LRF 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. In comparison to the approaches of Wurm et al. (19) and Wolf et al. (20), our sensor produces a substantial larger point cloud that needs a fast processing.

Vandapel et al. (21) segment 3D distance measurements and classify the segments into three different classes for terrain surface, clutter or wires. Their approach worked with a detailed stationary 3D sensor as well as on a mobile platform with a rotating 2D scanning mount.

Ye and Borenstein (22; 23) 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. (24) to detect obstacles. The authors present a color-based classification system and an algorithm that analyses the laser data to discriminate between grass and obstacles.

Schenk and Csatho (25) fuse aerial images with 3D point clouds to construct surfaces of urban scenes. The 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 mounted on scanning mounts are used by Wellington et al. (26). The authors apply multiple MRFs that interact through a hidden semi-Markov model that enforces a prior on vertical structures. Their results showed that including the neighborhood structure significantly improved obstacle classification and supplied us with the idea to fuse our cameras with our LRF to improve our terrain classification.

Another approach to terrain classification is based on judging surface conditions from the vibrations caused by driving on different surfaces. Brooks et al. (27) present a method to classify terrain based on vibrations measured using an accelerometer. The presented algorithm is able to distinguish sand, gravel, and clay.

A Bayes filter is employed by Komma et al. (28) to predict terrain classes from a history of vibration signals. The authors observe that only an adaptive approach which automatically adjusts its parameters is reactive enough to detect both, high-frequent and low-frequent terrain class changes.

Coyle et al. (29) present another approach to reaction-based terrain classification to detect 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 data sets for algorithm training.

In contrast to vibration-based approaches, our method is able to perceive the surface conditions *before* the robot actually traverses the terrain. Since we want to be able to detect regions like acres or grasslands in advance without already driving on them, we consider this as a major disadvantage of vibration-based approaches.

Another research topic covers the simultaneous localization and mapping (SLAM) problem, where LRFs are used to process the 2D and 3D distance measurements to build maps (30; 31) of the environment. In contrast to mapping, we want to process the sensor data as fast as possible online on the robot for a path planning algorithm to directly access data and perform tasks autonomously in real time.

3.3. Egomotion

Many approaches related to autonomous navigation need good pose prediction in order to produce accurate and robust results. Cameras, LRFs, GPS receivers and IMUs are, among others, commonly used for egomotion estimation.

Lu and Milios (32) present two iterative algorithms to register a 2D range scan to a previous scan to compute relative robot positions in unknown environments. While their first algorithm matches data points with tangent directions in two scans and minimizes a distance function, their second algorithm establishes correspondences between points in the two scans and then solves the point-to-point least-squares problem to compute the relative transformation between the two scans.

A comparative analysis of four algorithms which compute the 3D rigid body transformation that aligns two sets of 3D points for which correspondence is known is presented by Eggert et al. (33). The four algorithms are evaluated in terms of accuracy and robustness, stability with respect to degenerate data sets and relative computation time under different conditions.

Droeschel et al. (34) present an approach to estimate the egomotion of a robot while moving. Their approach uses the coherence of depth and reflectance data of a Time-of-Flight camera and fuses the resulting motion estimate with data from an IMU to improve accuracy and robustness against distorted measurements.

Ohno et al. (35) are concerned with map-based outdoor navigation near buildings and use an EKF to fuse odometry and DGPS measurement data. The authors present two correction methods to correct DGPS position and odometry errors.

Kim et al. (36) present a localization method using different sensors with bias data. An extended Kalman filter integrates odometry and DGPS data of their robot in a global coordinate frame.

An adaptive Kalman filter for GPS data processing based on the observation of residuals in realtime outdoor applications is presented by Reina et al. (37). Their approach is based on a fuzzy indicator to define a scale factor for the predicted covariance matrix by observing the size of the residuals.

A sigma-point Kalman filter is used for integrated navigation by van der Merwe and Wan (38) and fuses GPS and IMU data.

Lamon and Siegwart (39) present a method for combining dead reckoning sensor information in order to provide an initial estimate of the six degrees of freedom of a rough terrain rover. The rover's wheel encoders and an inertial navigation system are fused in an extended information filter.

The EKF presented by Voigt et al. (40) fuses binocular visual measurements and inertial cues for egomotion estimation of an aerial vehicle. Their approach relies on inertial data if visual feature constellation is degenerate and enables pose estimation at high frame rates.

3.4. Summary

All presented approaches from the fields of terrain classification and outdoor robotics have in common, that unstructured environments are described as challenging by the authors. Sensor noise, manifold and complex vegetation, rough terrain and concussions while driving require a solid approach to perform an adequate classification of the terrain. Under the circumstances described, we want to achieve the best classification possible with our sensor data. A statistical approach can handle sensor noise and uncertainties quite well and for this reason we chose a MRF, which will be described in the following Section 4.

4. Terrain Classification with MRFs

This section describes our terrain classification approach with MRFs on fused camera and laser data. At first, we introduce our data structure in Section 4.1. A description of the neighborhoods in our MRF follows in Section 4.2. Features used for classification are described in Section 4.3 and the Markov model of the terrain is presented in Section 4.4. Our method for camera and laser fusion is defined briefly in Section 4.5 and a description on how we learn the terrain properties for our classes is given in Section 4.6. The section closes with the estimation of the egomotion of our robot, which is explained in Section 4.7

4.1. MRF Terrain Data Structure

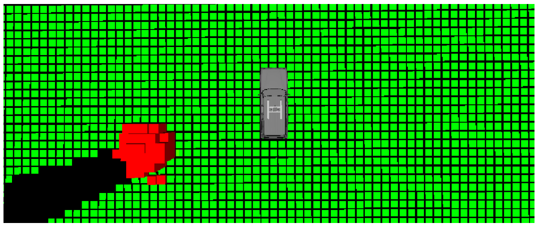

Figure 2: Markov random field data structure. The 2D grid is centered around the robot and divides the point cloud in to equidistant cells. On the lower left is an obstacle that occludes a small region behind itself.

As a first step, the large point cloud delivered by our sensor is subdivided into smaller chunks. We use a 2D equidistant grid, centered around the origin of the sensor, to subdivide the point cloud. Our grid has a size of $40 \times 40\,\mathrm{m}^2$ and each of the cells has a size of approximately $50 \times 50\,\mathrm{cm}^2$, which yields a grid resolution of 80×80 cells. We found this resolution to be sufficient to approximate obstacle borders without blocking narrow passages. The result of this step is a pre-classified grid, where each cell can either be *flat* or *obstacle* depending on the distribution of the contained 3D distance measurements or *unknown* if a cell is empty. We examine the distribution with respect to the surface normal and apply a threshold to the resulting value

$$h = \lambda_k \quad \text{with} \quad k = \underset{i \in \{0,1,2\}}{\operatorname{argmax}} |\mathbf{e}_i^{\mathrm{T}} \mathbf{z}| \tag{1}$$

where $\mathbf{z}$ is a vector pointing up, $\mathbf{e}_i$ are the eigenvectors, and λ_i the eigenvalues of the point cloud's covariance matrix in the current cell. Figure 2 shows an example of the data structure where green cells represent *flat* and red cells represent *obstacles* (cluster on the lower left). Cells without sensor readings are marked *unknown* and are illustrated in black (region behind the cluster on the lower left). Since the eigenvalue λ_k from Equation 1 corresponding to the surface normal is used to determine whether a cell is considered an obstacle or not, slopes will automatically be handled correctly, as λ_k quantifies the variance in the direction of the normal, which will be low even in sloped areas. This is based on the assumption that drivable terrain cells are locally planar. The only case where this assumption is violated occurs when the surface abruptly changes the slope. In this case, the described methodology can incorrectly classify the curvature as an obstacle. Examples are shown in Figure 3 where the blue arrows represent the plane normals indicating the orientation of the surface. The right side of the image shows the problematic case, where the surface abruptly changes its slope. The cell marked with the red circle is incorrectly regarded as an obstacle because the point distribution in regions with sharp edges is misleading. In urban scenarios, this might become an issue, however these kind of structures rarely occur in natural environments.

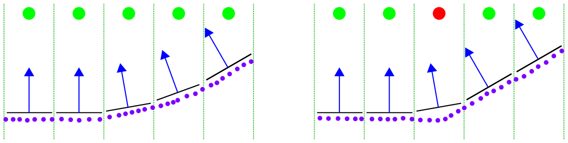

Figure 3: Lateral view on terrain cells along a slope and a sharp edge. The green dashed lines indicate the cell borders, the purple points represent the 3D distance measurements, and the black lines approximate the surface planes. The blue arrows represent the normals of the terrain surface within each cell and are computed from the contained 3D points. On the image on the left all cells are pre-classified as *flat* while one the right image the cell at the sharp edge is incorrectly regarded as an *obstacle*.

Besides obvious effects such as overestimation of obstacle boundaries if the cell size is too large or increased computational load if the cell size is too small, there is another interesting issue concerning the terrain itself. In rough terrain with slopes or ramps, the terrain cell size determines whether a ramp is traversable or not because the height is specified by the distribution of the contained 3D points. Hence, our cell size is chosen with respect to the total height our robot is able to handle.

4.2. MRF Neighborhood

MRFs can be understood as 2D expansion of Markov chains. This implies that the sites $\mathcal{S}$ are related to one another via a neighborhood system that is defined as

$$\mathcal{N} = \{\mathcal{N}_i \,|\, \forall i \in \mathcal{S}, \mathcal{N}_i \subset \mathcal{S}\} \tag{2}$$

where $\mathcal{N}_i$ is the set of sites neighboring i. Each site is not neighboring itself and the relationship is mutual:

$$i \notin \mathcal{N}_i, \tag{3}$$

$$i \in \mathcal{N}_{i'} \Leftrightarrow i' \in \mathcal{N}_i. \tag{4}$$

Neighboring states can be located in different distances forming different cliques of neighbors. In our work, we examine the 8 nearest neighbors surrounding a random state in a 2D data representation. This yields, with exception of border cells, to the neighborhood

$$\begin{aligned}\mathcal{N}_{i,j} = \{&(i-1, j-1), (i-1, j), (i-1, j+1), (i, j-1),\\ &(i, j+1), (i+1, j-1), (i+1, j), (i+1, j+1)\},\end{aligned} \tag{5}$$

for sites $\mathcal{S} = \{(i, j) \,|\, 1 \leq i, j \leq n\}$ corresponding to the cells of an $n \times n$ terrain grid.

Assuming that each state takes a value results in a configuration ω of a random field, a random field can be called MRF if the probability for a configuration takes a positive value

$$P(\omega) > 0 \tag{6}$$

and if a random state depends only on its neighbor states

$$P(\omega_{i,j} \,|\, \omega_{-\omega_{i,j}}) = P(\omega_{i,j} \,|\, \omega_\lambda), \tag{7}$$

where $\omega_{-\omega_{i,j}}$ is the set ω without $\omega_{i,j}$ and ω_λ is the set of states of the neighbors. For easier computation and modeling MRFs are often used as a Gibbs random field. MRFs and Gibbs random fields can be seen equal, which is described by the *Hammersley-Clifford Theorem*. In a Gibbs random field the distribution of the random states corresponds to the Gibbs distribution, which is defined as

$$P(\omega) = \frac{1}{Z} \exp\left(-\frac{1}{T} U(\omega)\right) \tag{8}$$

where T is a temperature parameter and $U(\cdot)$ is a function which calculates the energy E of a configuration. Z is a normalizing constant with

$$Z = \sum_{\omega} \exp\left(-\frac{1}{T} U(\omega)\right) \tag{9}$$

that calculates a sum over all possible configurations. The modeling of $U(\cdot)$ depends on the problem, which needs to be solved. Maximizing $P(\omega)$ can be achieved by finding a configuration ω that minimizes $U(\omega)$. For this purpose several sampling techniques exist, as it is too time consuming to test every possible configuration. MRFs for classification tasks are presented by Theodoridis and Koutroumbas (41) and several possibilities for modeling with MRFs are discussed by Li (42).

4.3. Features

Features are essential for classification tasks. We define a set of features which describe different terrain classes we use in our approach. This set can be divided into two categories. The first type of features is acquired from the data gained by the 3D LRF mounted on our robot. The second type consists of features computed from the camera images of the terrain.

The first laser-based feature we use is the *roughness* f_r of a terrain cell. According to Neuhaus et al. (1), the roughness is obtained by iterating over all circles and all measurements of our LRF. Within each circle of the Velodyne HDL-64E S2, the distance measurements are high-pass filtered. The intensity of the remaining signal is assigned to terrain grid cells and is interpreted as a measure for the local terrain roughness. This information is helpful for distinguishing roads from grass and farmland, for example.

The second laser-based feature, the *height difference* f_h, is capable of differentiating between obstacles and flat terrain. This feature simply displays the distance between the height of the lowest and the highest laser point of a terrain cell and was presented by Happold et al. (43). The usage of this feature is based on our assumption that the main characteristic of an obstacle is its height.

Under the assumption that texture differs for diverse types of terrain, we apply features acquired from image data. We exert the *second angular moment* f_{sam}, *variance* f_v and *inverse difference moment* f_{idm} texture feature. These are 3 easy-to-compute features from the 14 texture calculations proposed by Haralick et al. (44). We use another fast calculating feature, expressing the homogeneity f_{fh} of a texture, proposed by Knauer et al. (45). This is based on the assumption that rough terrain has an inhomogeneous texture in contrast to the texture of even terrain. In addition to texture features, the color f_c of a terrain cell is also used, which is fast and easy to compute.

Combination of all features yields the feature vector $\boldsymbol{f} = (f_r, f_h, f_{\text{sam}}, f_v, f_{\text{idm}}, f_{\text{fh}}, f_c)$ that integrates in the terrain classification process.

4.4. Markov Model of the Terrain

Flat terrain cells can either be easy or hard to pass for a robot, therefore we introduce the two classes *road* and *rough* to distinguish between the two. This leads to a total of four possible terrain-classes $\{$*unknown*, *road*, *rough*, *obstacle*$\}$. A Markov model of the terrain should respect the context-sensitivity of neighboring cells and the acquired features. For this reason we chose a MRF model used for image segmentation tasks, as described by Deng et al. (46). This model is able to fulfill both requirements and can easily be adapted to the terrain classification problem by regarding each terrain cell as random state in the MRF. Here, the energy E of a Gibbs random field is calculated piecewise by splitting it up into a component $E_{\mathcal{N}}$ that models the relationship of the neighboring cells and another component E_f which models the computed features of a cell.

The neighborhood component $E_{\mathcal{N}_{i,j}}$ of a cell $C_{i,j}$ at position (i, j) in a grid is defined as

$$E_{\mathcal{N}_{i,j}} = \sum_{\lambda \in \mathcal{N}_{i,j}} \beta \cdot \delta(c_{i,j}, c_\lambda) \qquad (10)$$

where $c_{i,j}$ is the terrain class assigned to the cell $C_{i,j}$ and c_λ is the terrain class assigned to a cell C_λ, which is part of the neighborhood of $C_{i,j}$. The function $\delta(\cdot)$ returns a value of -1 for $c_{i,j} \neq c_\lambda$ and $+1$ for $c_{i,j} = c_\lambda$. β is used to weight a neighbor's impact according to its distance to the cell. In our approach, we observe the 8 closest neighbors to a cell where the value of β is fixed and adjusted to our environment.

The corresponding feature component $E_{f_{i,j}}$ of a cell $C_{i,j}$ is based on the assumption that the features of a cell follow a Gaussian distribution. As an energy its computation is defined as

$$E_{f_{i,j}} = \sum_k \left(\frac{(f_{i,j_k} - \mu_{i,j_k})^2}{2\sigma^2_{i,j_k}} + \log\left(\sqrt{2\pi}\sigma_{i,j_k}\right) \right) \qquad (11)$$

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 respectively of the k-th feature of the class $c_{i,j}$ assigned to a cell $C_{i,j}$.

Combining the two calculations, the complete energy $E_{i,j}$ of a cell $C_{i,j}$ can be calculated as

$$E_{i,j} = E_{\mathcal{N}_{i,j}} + \alpha \cdot E_{f_{i,j}} \qquad (12)$$

where α is a weighting constant used to control the influence of the different energy types.

For classification the sum of all computed energies $E_{i,j}$ needs to be minimized, which leads to a maximization of the a posterior probability of the labeling of the terrain. These energies can be minimized by finding a label for each cell which fits best for the computed features and the labels of the neighbor cells. We apply the Gibbs Sampler described by Geman and Geman (47) to solve this task.

4.5. Camera and Laser Fusion

Within our framework we are able to fuse laser points and image pixels. This is achieved by a scenegraph representation of the used sensors, which provides transformation matrices between our sensors and their corresponding intrinsic parameters. The extrinsic parameters encoded in the scenegraph are acquired by previously calibrating each camera to our LRF by hand (48), as described by Unnikrishnan and Hebert (49). Thus we can build a lookup table for each laser point, which returns the corresponding pixel in the image. Further information on how to calibrate a camera to a LRF automatically are available in (50).

4.6. Acquisition of Terrain Properties

Application of the described MRF model requires knowledge of the feature-class relationships. In order to train our model in a supervised manner, we developed an annotation mode in our software that enables manual assignment of terrain classes for each cell. Using this annotation mode, we created our training set by manually annotating previously gathered data. From the resulting annotated terrain it is possible to calculate the required parameters of the features. The laser feature parameters can be applied in different environments in contrast to the image feature parameters, which need to be estimated separately for each scenario because of the different appearances of the terrain classes. We annotated the data from various scenarios and separated the annotations into training and test data. This has the advantage that the estimated parameters are based on real data acquired from the environment which are carefully classified by a human observer.

4.7. Egomotion Estimation

In order to be able to integrate the classification results over time, we require a local estimate of the robot's motion between the different time steps. Our robot framework already uses a simple EKF to determine the dynamic state of the robot in a global coordinate frame. Our filter fuses the information from our IMU with the measurements from our GPS receiver or, if GPS information is unavailable, with speed measurements acquired directly from the robot, using our radar-based speed sensor.

We use the following Kalman state in order to encode the current state of the robot:

$$\boldsymbol{\mu}_{t-1} = \begin{pmatrix} x & y & z & v & \phi & \vartheta & \psi \end{pmatrix}^{\mathrm{T}} \qquad (13)$$

Here, (x, y, z) and v describe the current position and velocity of the robot. The angles ϕ, ϑ and ψ encode the robot's current orientation in roll, pitch, and yaw angles. All of these quantities are expressed in the NWU (north west up) coordinate frame. In the Kalman prediction phase, we make the commonly used assumption of a simple linear motion model with additive Gaussian noise ζ

$$\boldsymbol{\mu}_{t+1} = \begin{pmatrix} x_t + v_t \Delta t \cos \vartheta_t \cos \psi_t \\ y_t + v_t \Delta t \cos \vartheta_t \sin \psi_t \\ z_t + v_t \Delta t \sin \vartheta_t \\ v_{\text{odo}} \\ \phi_t + \phi_{\text{odo}} \Delta t \\ \vartheta_t + \vartheta_{\text{odo}} \Delta t \\ \psi_t + \psi_{\text{odo}} \Delta t \end{pmatrix} + \zeta \qquad (14)$$

The values $\phi_{\text{odo}}, \vartheta_{\text{odo}}, \psi_{\text{odo}}$ are the angular velocities in roll, pitch, and yaw direction as measured by our IMU. The scalar v_{odo} describes the current velocity of the robot, as measured by our robot's built-in speed sensors.

The next step would be to perform the Kalman update using the measurements obtained by our GPS receiver. For this application however, it suffices to have an estimate of the robot's motion between two subsequent time steps. In particular, we do not require globally correct positioning information. While the Kalman update step is performed subsequently in our application, it is not directly used for the local registration, which is why we omit its explanation at this point.

We found that while the previous methodology to estimate the relative transformation between subsequent robot poses yields very good guesses, yet it was generally not precise enough to be used exclusively. Therefore, we opted to further refine these estimates using a variant of the ICP algorithm on a 2D virtual scan.

Petrovskaya and Thrun (51) describe the extraction of a two-dimensional scan from data of a Velodyne HDL-64E S2 using a piece-wise planar ground model. Therefore, all readings are projected into a 3D grid in polar coordinates around the vehicle center (cf. Figure 4). For filtering noise, a representative reading is chosen for each cell as the median of distances. A model of ground elevation is established by comparing neighboring cells with the same horizontal bearing θ. If the slope is smaller than a threshold, the point is classified as ground reading. The distance to the closest obstacle in a target height of 50 cm to 200 cm is used as entry in the virtual scan. With only mid-height objects apparent in the scan, overhanging foliage and small objects, such as curbs or small stones, are not considered for egomotion estimation. While this method was developed in the context of vehicle detection and tracking, it works equally well for egomotion estimation.

In order to refine the results of our EKF prediction, we register point sets $P = \{\boldsymbol{p}_i\}$ and $Q = \{\boldsymbol{q}_i\}$ that correspond to readings within virtual scans acquired at the prior and the current time step. Therefore, we use a 2D variant of the ICP algorithm (52) that minimizes the error function

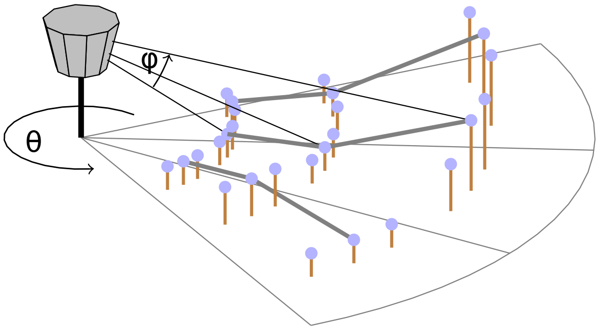

Figure 4: Projection of Velodyne HDL-64E S2 readings into a 3D grid in spherical coordinates as described in (51). Readings are illustrated as dots. Their height is indicated by a line that projects readings into the horizontal plane.

$$\rho(\boldsymbol{R}, \boldsymbol{t}) = \sum_{(i,j)\in C} \left\| \boldsymbol{p}_i - (\boldsymbol{R}\boldsymbol{q}_j + \boldsymbol{t}) \right\|^2 \qquad (15)$$

with rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$. Here, C is a set of corresponding point index pairs determined by nearest neighbor search. The transformation $(\boldsymbol{R}, \boldsymbol{t})$ that minimizes Equation 15 is computed using the direct method based on Singular Value Decomposition (SVD). Therefore, we compute correlation matrix

$$\boldsymbol{H} = \sum_{(i,j)\in C} (\boldsymbol{p}_i - \bar{\boldsymbol{p}})(\boldsymbol{q}_j - \bar{\boldsymbol{q}})^{\mathrm{T}},$$

where $\bar{\boldsymbol{p}}$ and $\bar{\boldsymbol{q}}$ are the means of points in the correspondence set:

$$\bar{\boldsymbol{p}} = \frac{1}{|C|} \sum_{(i,j)\in C} \boldsymbol{p}_i \qquad \bar{\boldsymbol{q}} = \frac{1}{|C|} \sum_{(i,j)\in C} \boldsymbol{q}_j.$$

Next, rotation and translation are calculated separately. First, the rotation is determined as

$$\boldsymbol{R} = \boldsymbol{V}\boldsymbol{U}^{\mathrm{T}},$$

where matrices $\boldsymbol{V}$ and $\boldsymbol{U}$ are obtained using SVD $\boldsymbol{H} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}}$. Then, the translation is computed as

$$\boldsymbol{t} = \boldsymbol{\mu}_p - \boldsymbol{R}\boldsymbol{\mu}_q.$$

Using egomotion estimates we are able to transform the current grid structure into a preceding coordinate frame by rotating and translating the center of each cell. Therefore, our MRF is able to access previous classification results to obtain an estimate of traversability

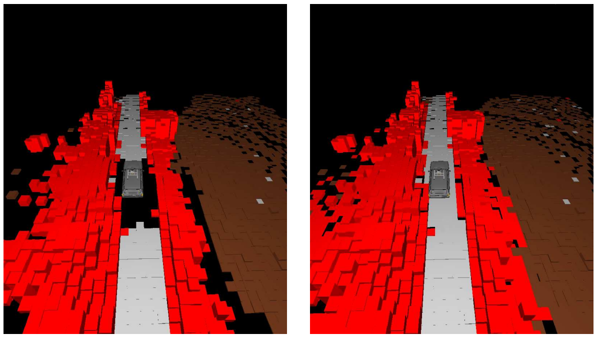

Figure 5: MRF classification with and without Egomotion estimation. The image on the left shows the MRF result on a single 3D scan while the image on the right shows an interpolation of data gaps in the MRF by accessing the previous terrain classification. Terrain cells in the direct neighborhood of the robot are concerned as well as previously visible regions on the right side of the robot behind some obstacles.

where insufficient sensor data is available. The results of this method are visualized in Figure 5. Due to self-occlusion and a minimum perceivable distance of approximately 1 m, the area close to the vehicle is imperceptible for the robot. Using egomotion, classification results are carried over from the previous time step and data is available for continued navigation. This allows an interpolation in regions with few or without any sensor measurements over small time gaps while larger regions, e. g. like holes or descents, are preserved.

5. Evaluation

In our experiments we compared our terrain classification results with ground truth which is acquired by a human observer, who annotated the terrain cells manually. The annotated terrain cells forming our ground-truth do not contain the annotations we used to acquire the terrain properties (cf. Section 4.6). By comparing the results with their respective ground truth we calculated the true positive rate and the false positive rate for each terrain class. We evaluated the classification for different scenarios where the relative appearances of the terrain classes differed. The system used for all computations is a single laptop, an Intel(R) Core(TM) i7 QM with 1.73 GHz and 8 GB RAM. An example scene is shown in Figure 7, where the 3D data, the fused data and the MRF result are visualized. Table 1 shows the result for a scenario where the robot drives on a road through fields and small to medium sized vegetation. Table 2 shows the results for a forest scenario where the class *rough* appears only at single terrain cells and not at larger regions. In both tables the usage of laser-based features is denoted by L, selected Haralick features by H, the homogeneity feature by FH and color by C. Furthermore, TPR is the true positive rate, or recall rate, and FPR the corresponding false positive rate. In all tested scenarios the results for detecting streets and obstacles are equally high. In case of Table 2, the results show that the detection of the less appearing class (rough) does not work properly. The reason is the modeling of the terrain, in which we assume that terrain cells of one class show a tendency to group and not to exist as single cells. Furthermore, we observed that the usage of image features does not improve the quality of the results in our approach. A reason for this may be that texture and color varies more than we assumed, which leads to high values for the standard deviations of these features. Thus a high deviation from the mean value of such a feature does not change the computed energy value sufficiently to have impact on the classification. For both scenarios, the corresponding receiver operating characteristics (ROC curves) are shown in Figure 6.

The evaluation of the runtime, shown in Table 3, shows that the usage of image features multiplies the required calculation time. Using laser features exclusively leads to a fast runtime and good classification re-

Class/value	**L**	**L + H**	**L + FH**	**L + C**
Road/TPR	91.0 %	91.9 %	90.4 %	91.0 %
Road/FPR	1.5 %	1.2 %	1.2 %	1.4 %
Rough/TPR	74.6 %	73.8 %	75.2 %	75.3 %
Rough/FPR	1.2 %	1.1 %	1.3 %	1.4 %
Obstacle/TPR	92.8 %	91.8 %	92.9 %	92.6 %
Obstacle/FPR	3.6 %	4.4 %	3.2 %	3.1 %

Table 1: Performance of the algorithm in a rural environment. True positive rate (TPR) and false positive rate (FPR) are presented for the three classes *Road*, *Rough* and *Obstacle*. Columns are separated by the used features: laser-based (L), Haralick features (H), the homogeneity feature (FH) and the color information (C).

Class/value	**L**	**L + H**	**L + FH**	**L + C**
Road/TPR	95.2 %	91.7 %	95.0 %	95.0 %
Road/FPR	4.3 %	5.6 %	4.3 %	4.5 %
Rough/TPR	5.7 %	0.4 %	2.8 %	3.9 %
Rough/FPR	0.3 %	0.1 %	0.3 %	0.3 %
Obstacle/TPR	96.2 %	95.4 %	96.1 %	96.3 %
Obstacle/FPR	6.9 %	6.7 %	7.3 %	6.8 %

Table 2: Performance of the algorithm in a forest environment. Classes and features are used analogous to Table 1 and true positive and false positive rates are shown. This scenario is specified by very few cells of class *Rough*, which are further suppressed by the neighborhood relations within the MRF.

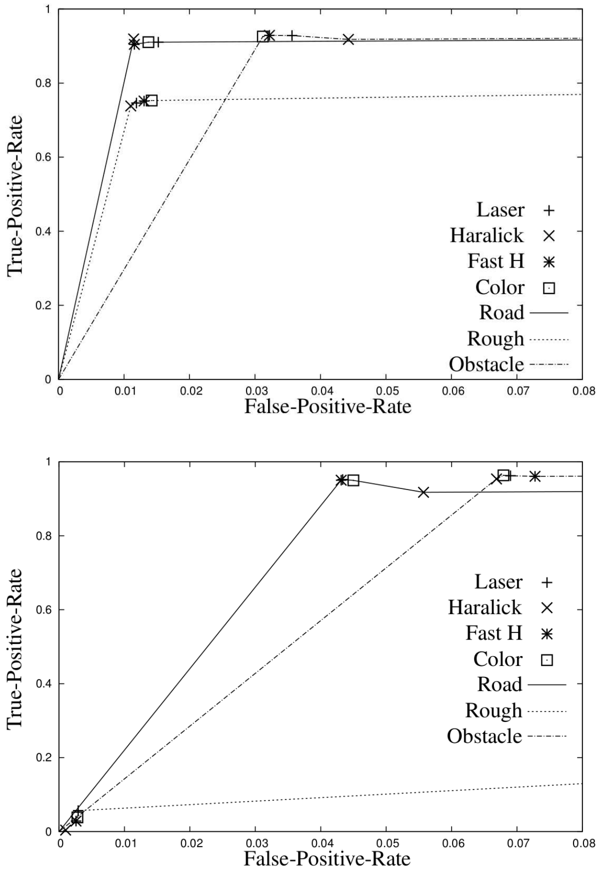

Figure 6: ROC curves corresponding to the values of Table 1 (upper figure) and Table 2 (lower figure). The three curves in both figures show the true positive rate in relation to the false positive rate for the three classes *Road*, *Rough* and *Obstacle*.

sults. In all experiments the fixed value β is set to 0.8 (cf. Equation 10).

Mode	**Mean**	**Std. dev.**	**Max**	**Min**
L	62.7 ms	7.4 ms	82.2 ms	48.8 ms
L+H	744.9 ms	131.5 ms	994.0 ms	421.5 ms
L+FH	141.4 ms	11.2 ms	188.4 ms	121.7 ms
L+C	144.4 ms	12.6 ms	187.0 ms	117.3 ms

Table 3: Runtime results for the MRF application with different features from single sensor and fused sensor data. All values represent long-term measurements and result from the direct application to real sensor data, without preprocessing or any form of data reduction. The terms L, H, FH and C are equivalent to Table 1 and Table 2.

6. Conclusion

In this work, we presented an approach to process fused data of a modern 3D LRF and three cameras using a MRF. We thereby integrated context sensitivity and different features from both laser and image data. Application of the MRF results in a classified 2D grid with terrain traversability information for each cell. We annotated terrain cells in different scenarios to learn the class parameters of the features. The resulting grid forms an efficient data structure for autonomous systems in unstructured environments. An EKF was used to fuse IMU, GPS and odometry data and the results were further refined with an ICP algorithm. Our ICP algorithm is based upon virtual 2D scans which are acquired from the 3D LRF data to perform a precise egomotion estimation in unstructured environments. This egomotion estimation enables the terrain classification to access previous classifications to interpolate data gaps, hidden regions and sparse data. We achieved a recall ratio of about 90% for detecting streets and obstacles.

Limitations of our approach revealed at higher speed. The robot platform we use has a maximum speed of 14 km/h, but the sensor data were also recorded while driving around with a car. For the usage of our approach on our autonomous system, the presented fusion is sufficient. In order to deal with notably faster egomotion, the sensor calibration needs to take into account the instant of time where the data was recorded. A system of precise time stamps should be able to solve this issue. Computation time of the image features is another difficult aspect. Image features need to be chosen wisely or a solution for speeding up this cost intensive part is needed. Another issue of our laser data interpretation arises from the fact that it is based upon the end-points of the sensor measurements only and thus does not take visibility information along the rays from our sensor into account. Hence, it is not possible to deal with a few uncertainties, e. g. caused by entirely black cars as they might absorb the measurements or by negative obstacles where no end-points indicate a threat for the vehicle. For the navigation in unstructured natural environments, we chose not to address this problem but for urban scenarios or regions where such artificial surfaces might occur it is recommended to extend our approach, e. g. by applying ray-casting techniques.

Considering future work, we plan to integrate a self-supervised learning of the feature parameters. In doing so, our approach can adapt to various situations and is able to improve during runtime if the features are relearned. In addition, we will investigate whether the application of graph cuts (53) to the MRF data structure yields an acceleration of the computation time consumed by the energy minimization step. The MRF and the fused data allow extracting semantic information from the sensor readings in real time. This can be used to enrich 3D maps (54) with further information.

Acknowledgement

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

References

1 F. Neuhaus, D. Dillenberger, J. Pellenz, 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, Catalonia, Spain, 2009, pp. 1686–1689.

2 M. Hselich, M. Arends, D. Lang, 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, rebro, Schweden, 2011, pp. 153–158.

3 Velodyne Lidar Inc., Velodyne Lidar HDL-64E, `http://www.velodyne.com/lidar` (January 2012).

4 Logitech Europe S.A., Logitech HD Pro C910, `http://www.logitech.com/de-de/webcam-communications/webcams/devices/6816` (January 2012).

5 Philips Electronics N.V., Philips SPC1300NC, `http://www.p4c.philips.com/cgi-bin/dcbint/cpindex.pl?ctn=SPC1300NC/00` (January 2012).

6 T. Szirányi, J. Zerubia, L. Czúni, D. Geldreicch, Z. Kato, Image Segmentation Using Markov Random Field Model in Fully Parallel Cellular Network Architectures, Real-Time Imaging 6 (3) (2000) 195–211.

7 V. Meas-Yedid, S. Tilie, J.-C. Olivo-Marin, Color Image Segmentation Based on Markov Random Field Clustering for Histological Image Analysis, in: Proceedings of the 16th International Conference on Pattern Recognition, Quebec City, Canada, 2002, pp. 796–799.

8 Z. Kato, T. Pong, A Markov Random Field Image Segmentation Model for Color Textured Images, Image and Vision Computing 24 (10) (2006) 1103–1114.

9 I.-U.-H. Qazi, O. Alata, J.-C. Burie, M. Abadi, A. Moussa, C. Fernandez-Maloigne, Parametric Models of Linear Prediction Error Distribution for Color Texture and Satellite Image Segmentation, Computer Vision and Image Understanding 115 (2011) 1245–1262.

10 A. d'Angelo, J.-L. Dugelay, A Markov Random Field Description of Fuzzy Color Segmentation, in: Proceedings of the 2nd International Conference on Image Processing Theory, Tools and Applications, Paris, France, 2010, pp. 270–275.

11 E. Herbst, X. Ren, D. Fox, RGB-D Object Discovery via Multi-Scene Analysis., in: Proceedings of the 2011 IEEE/RSJ international Conference on Intelligent Robots and Systems, San Francisco, USA, 2011, pp. 4850–4856.

12 J. Diebel, S. Thrun, An Application of Markov Random Fields to Range Sensing, in: Advances in Neural Information Processing Systems, Vancouver, Canada, 2005, pp. 291–298.

13 K. Konolige, M. Agrawal, R. C. Bolles, C. Cowan, M. Fischler, B. Gerkey, Outdoor Mapping and Navigation Using Stereo Vision, in: Proceedings of the International Symposium on Experimental Robotics, Rio de Janeiro, Brazil, 2006, pp. 179–190.

14 J. Alberts, D. Edwards, T. Soule, M. Anderson, 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, Edinburgh, Scotland, 2008, pp. 103–108.

15 P. Vernaza, B. Taskar, D. Lee, Online, Self-supervised Terrain Classification via Discriminatively trained Submodular Markov Random Fields, in: IEEE International Conference on Robotics and Automation, Pasadena, USA, 2008, pp. 2750–2757.

16 Y. Khan, P. Komma, K. Bohlmann, A. Zell, Grid-based visual terrain classification for outdoor robots using local feature, in: Proceedings of the 2011 IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems, Paris, France, 2011, pp. 16–22.

17 A. Rankin, A. Huertas, L. Matthies, Negative Obstacle Detection by Thermal Signature, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, USA, 2003, pp. 906–913.

18 R. D. Morton, E. Olson, Positive and Negative Obstacle Detection using the HLD Classifier, in: Proceedings of the 2011 IEEE/RSJ international Conference on Intelligent Robots and Systems, San Francisco, USA, 2011, pp. 1579–1584.

19 K. Wurm, R. Kümmerle, C. Stachniss, 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, St. Louis, USA, 2009, pp. 1217–1222.

20 D. F. Wolf, G. Sukhatme, D. Fox, W. Burgard, Autonomous Terrain Mapping and Classification Using Hidden Markov Models, in: Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005, pp. 2026–2031.

21 N. Vandapel, D. Huber, A. Kapuria, M. Hebert, Natural Terrain Classification Using 3-D Ladar Data, in: Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, USA, 2004, pp. 5117–5122.

22 C. Ye, J. Borenstein, A New Terrain Mapping Method for Mobile Robots Obstacle Negotiation, in: Proceedings of the UGV Technology Conference at the SPIE AeroSense Symposium, Orlando, Florida, 2003, pp. 21–25.

23 C. Ye, J. Borenstein, A Method for Mobile Robot Navigation on Rough Terrain, in: Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, USA, 2004, pp. 3863–3869.

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

25 T. Schenk, 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, Paris, France, 2007, pp. 11–13.

26 C. Wellington, A. Courville, A. Stentz, Interacting Markov Random Fields for Simultaneous Terrain Modeling and Obstacle Detection, in: Proceedings of Robotics Science and Systems, Cambridge, USA, 2005, pp. 1–8.

27 C. A. Brooks, K. Iagnemma, S. Dubowsky, Vibration-based Terrain Analysis for Mobile Robots, in: Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005, pp. 3415–3420.

28 P. Komma, C. Weiss, A. Zell, Adaptive bayesian filtering for vibration-based terrain classification, in: Proceedings of the IEEE

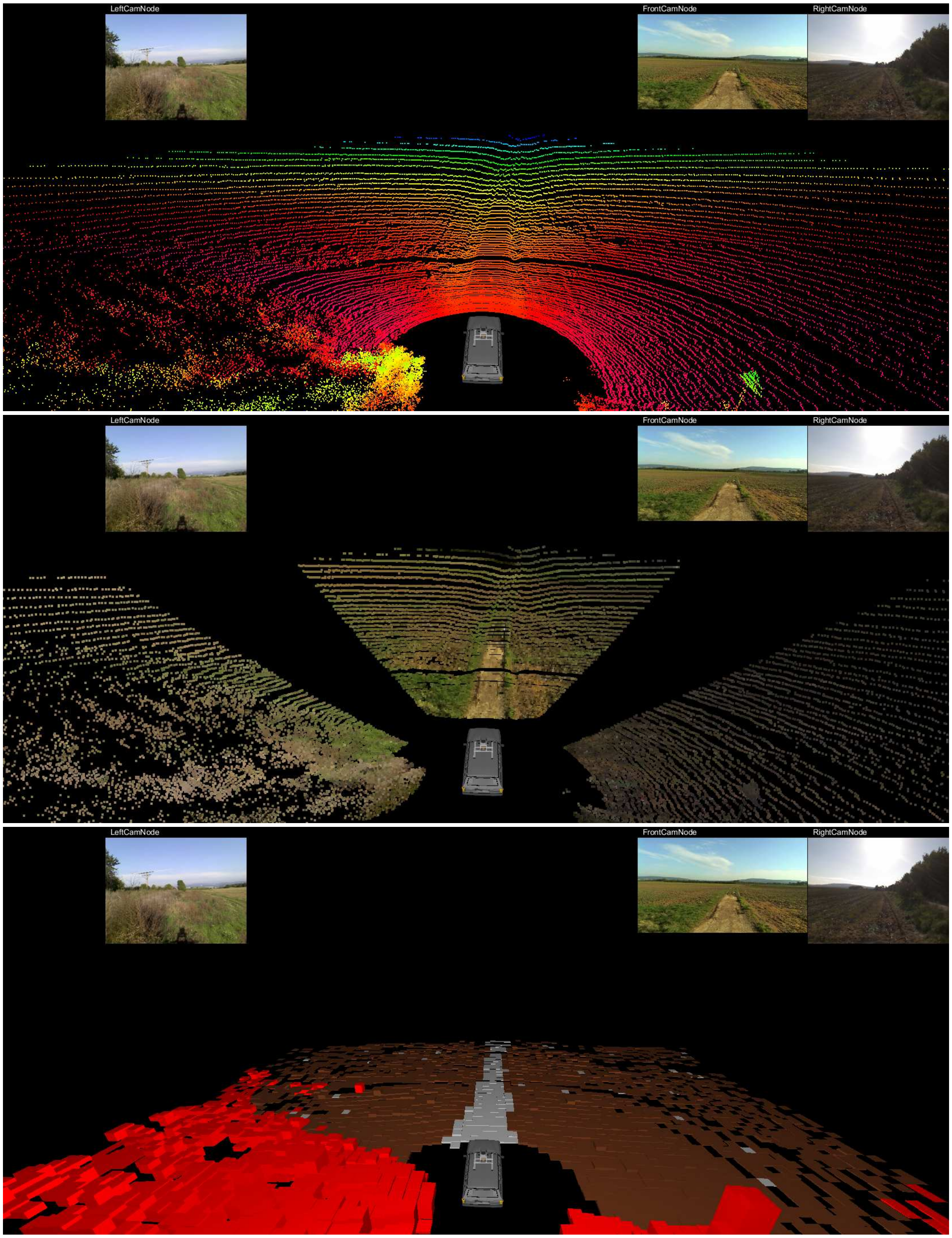

Figure 7: Example scene of an unstructured environment with three different representations (best viewed in color). In each representation the images from the left, front and right camera are shown in the upper part. The upper image shows the 3D point cloud provided by the Velodyne HDL-64E S2. The color of a point is determined by its height. On the image in the middle the LRF data are fused with all three cameras. This way color, texture and 3D position are available in one coordinate system. The lower image shows the result of the MRF application: classified terrain cells with traversability information. The classes *unknown, road, rough* and *obstacle* are respectively visualized in black, gray, brown and red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

International Conference on Robotics and Automation, Kobe, Japan, 2009, pp. 3307–3313.

29 E. Coyle, E. Collins, R. Roberts, Speed independent terrain classification using Singular Value Decomposition Interpolation, in: Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 2011, pp. 4014 – 4019.

30 J. Guivant, E. Nebot, S. Baiker, Autonomous Navigation and Map Building Using Laser Range Sensors in Outdoor Applications, Journal of Robotic Systems 17 (10) (2000) 3817–3822.

31 M. Montemerlo, S. Thrun, Large-Scale Robotic 3-D Mapping of Urban Structures, in: Proceedings of the 9th International Symposium on Experimental Robotics, Singapore, Republic of Singapore, 2006, pp. 141–150.

32 F. Lu, E. Milios, Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans, Journal of Intelligent and Robotic Systems 18 (3) (1997) 249–275.

33 D. W. Eggert, A. Lorusso, R. B. Fisher, Estimating 3-D Rigid Body Transformations: a Comparison of Four Major Algorithms, Machine Vision Applications 9 (1997) 272–290.

34 D. Droeschel, S. May, D. Holz, P. G. Ploeger, S. Behnke, Robust Ego-Motion Estimation with ToF Cameras, in: Proceedings of the 4th European Conference on Mobile Robotics, Dubrovnik, Croatia, 2009, pp. 187–192.

35 K. Ohno, T. Tsubouchi, B. Shigematsu, S. Maeyama, S. Yuta, Outdoor Navigation of a Mobile Robot between Buildings Based on DGPS and Odometry Data Fusion, in: Proceedings of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 2003, pp. 1978–1984.

36 Y.-S. Kim, B. K. Kim, K. Ohba, A. Ohya, Localization of Outdoor Mobile Robot with Multi-Path Bias Detection, in: Proceedings of the IEEE International Conference on Mechatronics and Automation, Harbin, China, 2007, pp. 705–710.

37 G. Reina, A. Vargas, K. Nagatani, K. Yoshida, Adaptive Kalman Filtering for GPS-based Mobile Robot Localization, in: Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics, Rome, Italy, 2007, pp. 1–6.

38 R. van der Merwe, E. A. Wan, Sigma-Point Kalman Filters for Integrated Navigation, in: Proceedings of the 60th Annual Meeting of The Institute of Navigation, Dayton, USA, 2004.

39 P. Lamon, R. Siegwart, Inertial and 3D-odometry Fusion in Rough Terrain - Towards Real 3D Navigation, in: Proceedings of the 2011 IEEE/RSJ international Conference on Intelligent Robots and Systems), Sendai, Japan, 2004, pp. 1716–1721.

40 R. Voigt, J. Nikolic, C. Huerzeler, S. Weiss, L. Kneip, R. Siegwart, Robust Embedded Egomotion Estimation, in: Proceedings of the 2011 IEEE/RSJ international Conference on Intelligent Robots and Systems, San Francisco, USA, 2011, pp. 2694–2699.

41 S. Theodoridis, K. Koutroumbas, Pattern Recognition, Academic Press, 2009.

42 S. Li, Markov Random Field Modeling in Computer Vision, Springer, 2009.

43 M. Happold, M. Ollis, N. Johnson, Enhancing Supervised Terrain Classification with Predictive Unsupervised Learning, in: Proceedings of Robotics: Science and Systems, Philadelphia, USA, 2006, p. Online proceedings: http://www.roboticsproceedings.org/rss02/p06.html.

44 R. Haralick, K. Shanmugam, I. Dinstein, Textural Features for Image Classification, IEEE Transactions on Systems, Man, and Cybernetics 3 (6) (1973) 610–621.

45 U. Knauer, B. Meffert, Fast Computation of Region Homogeneity with Application in a Surveillance Task, in: Proceedings of ISPRS Commission V Mid-Term Symposium Close Range Image Measurement Techniques, Newcastle, UK, 2010, pp. 337–342.

46 H. Deng, D. A. Clausi, Unsupervised Image Segmentation Using a Simple MRF Model with a New Implementation Scheme, Pattern Recognition 37 (12) (2004) 2323–2335.

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

48 M. Häselich, R. Bing, D. Paulus, Calibration of Multiple Cameras to a 3D Laser Range Finder, in: Proceedings of the 2012 IEEE International Conference on Emerging Signal Processing Applications, Las Vegas, USA, 2012, pp. 25–28.

49 R. Unnikrishnan, M. Hebert, Fast Extrinsic Calibration of a Laser Rangefinder to a Camera, Tech. rep., Robotics Institute, Pittsburgh, USA (2005).

50 D. Scaramuzza, A. Harati, R. Siegwart, Extrinsic Self Calibration of a Camera and a 3D Laser Range Finder from Natural Scenes, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, California, 2007, pp. 4164–4169.

51 A. Petrovskaya, S. Thrun, Model Based Vehicle Detection and Tracking for Autonomous Urban Driving, Autonomous Robots, Special Issue: Selected papers from Robotics: Science and Systems 2008 26 (2–3) (2009) 123–139.

52 A. Nüchter, 3D Robotic Mapping: The Simultaneous Localization and Mapping Problem with Six Degrees of Freedom, Vol. 52 of Springer Tracts in Advanced Robotics, Springer Press, Berlin, Heidelberg, Germany, 2009.

53 D. M. Greig, B. T. Porteous, A. H. Seheult, Exact Maximum A Posteriori Estimation for Binary Images, Journal of the Royal Statistical Society. Series B (Methodological) 51 (2) (1989) 271–279.

54 J. Pellenz, D. Lang, F. Neuhaus, 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, pp. 1–6.