

Normal Distributions Transform Traversability Maps: LIDAR-Only Approach for Traversability Mapping in Outdoor Environments



Juhana Ahtiainen

Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland
e-mail: juhana.ahtiainen@aalto.fi

Todor Stoyanov

Center of Applied Autonomous Sensor Systems (AASS), Örebro University, Örebro, Sweden
e-mail: todor.stoyanov@oru.se

Jari Saarinen

GIM Ltd. Espoo, Finland
e-mail: jari.saarinen@gimltd.fi

Received 26 June 2015; accepted 10 April 2016

Safe and reliable autonomous navigation in unstructured environments remains a challenge for field robots. In particular, operating on vegetated terrain is problematic, because simple purely geometric traversability analysis methods typically classify dense foliage as nontraversable. As traversing through vegetated terrain is often possible and even preferable in some cases (e.g., to avoid executing longer paths), more complex multimodal traversability analysis methods are necessary. In this article, we propose a three-dimensional (3D) traversability mapping algorithm for outdoor environments, able to classify sparsely vegetated areas as traversable, without compromising accuracy on other terrain types. The proposed normal distributions transform traversability mapping (NDT-TM) representation exploits 3D LIDAR sensor data to incrementally expand normal distributions transform occupancy (NDT-OM) maps. In addition to geometrical information, we propose to augment the NDT-OM representation with statistical data of the permeability and reflectivity of each cell. Using these additional features, we train a support-vector machine classifier to discriminate between traversable and non-drivable areas of the NDT-TM maps. We evaluate classifier performance on a set of challenging outdoor environments and note improvements over previous purely geometrical traversability analysis approaches.

© 2016 Wiley Periodicals, Inc.

1. INTRODUCTION

In recent years, there has been increasing interest in advancing robot technology in outdoor, off-road, and natural environments. For example, search and rescue robots, as well as autonomous machines in forestry and mining applications, need to cope with unstructured outdoor environments. Terrain traversability analysis is essential for unmanned ground vehicles (UGV) operating in unstructured environments, because the ability to detect and classify the surrounding obstacles is a necessity for safe navigation performance. Most existing obstacle detection systems rely on geometric representations of the environment, most commonly constructed using either a vision system, or a LIDAR. However, for a UGV operating in vegetated environments,

a naive geometric representation is usually not sufficient for efficient navigation, as vegetation can often be mistakenly interpreted as an obstacle by the perception system. In practice, traversing through sparse vegetation is often possible and may be preferable in order to avoid executing longer paths. In cases of densely vegetated environments, it can even be the only acceptable option for the UGV. For example, the grass shown in Figure 1 could be driven through, but perception systems relying only on geometric features tend to interpret it as nontraversable. To cope with limitations of purely geometrical traversability analysis in vegetated natural environments, a substantial research effort has focused on appearance-based terrain classification. However, distinguishing vegetation from other obstacles is not sufficient for safe navigation because solid obstacles (e.g., large rocks, tree trunks) hidden behind the vegetation would pose a great risk to the robot, if it were to base traversability decisions solely on vegetation classification.

Direct correspondence to: Juhana Ahtiainen, e-mail: juhana.ahtiainen@aalto.fi



Figure 1. A UGV within tall grass. Densely vegetated terrain is often mistakenly interpreted as obstacles by current state-of-the-art perception systems.

In this article, we propose a novel LIDAR-only 3D traversability mapping approach, which goes beyond the traditional statistical interpretation of local point clouds. Our algorithm is designed for unstructured outdoor environments and is able to classify sparsely vegetated areas as traversable, while still detecting partially visible objects, such as large rocks and tree stumps. Our work builds on the normal distributions transform occupancy map (NDT-OM) (Saarinen et al., 2013b) spatial representation — a voxel grid structure that models local surfaces in each cell as Gaussian probability distributions. We propose an extension of the NDT-OM representation and subsequently use the additional information in our traversability analysis algorithm. In particular, we augment the NDT-OM representation with two additional statistics. First, for each cell, we represent the intensity of the reflected laser beams as a one-dimensional Gaussian distribution. Second, we model the permeability of the geometrical structure in each cell as a Bernoulli process. In this manner, we extract three additional features to aid us in traversability classification — the mean of the intensity distribution, its variance, and the expected value of the permeability.

The contributions of this paper are twofold. First, we augment the NDT-OM representation with two additional sources of information — intensity and permeability — and perform a detailed analysis of the effects of vegetation on the augmented NDT-OM cells. Second, we propose two classification algorithms learned from experience, which outperform previous work on traversability analysis within the NDT framework. The first algorithm — *C*-support vector classification (*C*-SVC), performs classification for each cell of the NDT-TM using the roughness, inclination, permeability, and intensity distribution. The second algorithm — augmented constant threshold classification (ACTC), first performs a purely geometric CTC as per (Stoyanov et al., 2010) for each cell in the NDT-TM, after which the *C*-SVC is performed only for the cells classified as nontraversable by the CTC method. Both of the methods perform classification

directly using the proposed NDT-TM representation, which permits using significantly lower grid resolutions than reported in previous works on LIDAR-based traversability estimation in vegetated environments. Using a lower resolution model in turn has positive implications on both the memory requirements (as fewer cells need to be stored), as well as the runtime performance of our algorithms (a lower number of traversability queries need to be performed). For both of the proposed algorithms, we avoid the burdensome hand-labeling of data by collecting the training examples autonomously from semicontrolled environments.

In the rest of this article, we first review the relevant contributions in the area of spatial modeling and terrain traversability analysis, especially in vegetated environments. The foundation and details of the proposed traversability classification methods are presented in Section 3. Section 4 introduces the measurement platforms and discusses implementation details, whereas Section 5 provides the results of the feature analysis experiments and the full-scale field trials. Finally, Section 6 concludes with a summary of the key findings and a discussion of limitations and future work.

2. RELATED WORK

Terrain traversability analysis is used as a mean for generating traversability maps of the environment that quantify the difficulty a UGV would encounter in passing through a particular region. Typically, traversability maps are platform dependent because the locomotion capabilities of different-sized platforms may differ significantly (Molino et al., 2007). The essence of traversability analysis is deducing whether an area is traversable, given the platform constraints and sensor data. Nonetheless, in order to generate traversability maps, a representative spatial model of the surrounding environment is needed. Therefore, this section reviews the relevant contributions in the area of spatial modeling, as well as terrain traversability analysis. Because traversing in vegetated environments is particularly challenging for field robots, special attention is given to vegetation classification and traversability analysis within vegetated areas.

2.1. Spatial Modeling

Several different spatial modeling approaches have been successfully applied in robotic mapping systems. One of the most predominant spatial modeling technique in robotics applications is occupancy grid mapping, originally introduced by Moravec & Elfes (1985). Occupancy grid maps represent the environment as a regular grid, in which each grid cell models the probability of it being occupied. Originally, occupancy grids were used as a 2D modeling tool, but they are easily scalable to three dimensions. However, 3D occupancy grids require a lot of memory, thus rendering them infeasible for large-scale mapping applications. A popular

implementation of 3D occupancy maps is Octomap (Hornung et al., 2013), which is based on an octree grid structure. Octrees have natural multiresolution support, which provides an efficient way to maintain unobserved portions of the map, thus reducing the memory requirements.

Elevation mapping is a grid-based 2.5D spatial representation. Each cell in the 2D grid stores the height of the surface in the corresponding area (Siciliano & Khatib, 2008). Elevation maps have been popular mapping tools for outdoor applications since the early years of robotics. For example, Bares et al. (1989) use an elevation map for footprint placement selection of a legged planetary rover. However, the elevation map representation reduces the dimensionality from 3D to 2.5D; that is, it can only model a single surface per cell. Therefore, elevation maps cannot model overhanging structures (e.g., bridges, tunnels, tree branches) correctly. Another drawback of elevation mapping is that the height of areas with little or no data cannot be reliably expressed.

To take the missing data problem into consideration, Lang et al. (2007) propose the use of adaptive nonstationary kernel regression in Gaussian processes (GPs) to deal with varying data densities in terrain models. The central idea of GP terrain modeling is to represent the height value of each point as a function of its 2D space coordinates, and to subsequently approximate the value using a set of Gaussian distributions in function space. The available sensor data are used to learn the hyperparameters of a GP, which can then be used to perform regression for any point in 2D space and obtain an interpolated height value, resulting in a continuous spatial model. For example, Vasudevan et al. (2009) propose an approach for modeling large-scale and complex terrain using single neural network-based GP, which they demonstrate to preserve many of the spatial features in the terrain. Hadsell et al. (2010) extended the traditional kernel-based learning approaches for estimating continuous surfaces by providing upper and lower bounds on the surface. This was done by exploiting visibility constraints of the sensor w.r.t. the terrain surface and subsequently applying kernel-based regression techniques to improve the precision of the terrain geometry estimate.

To relax the functional constraint of a single height value per location of GPs and elevation mapping, Triebel et al. (2006) proposed multilevel surface (MLS) maps: an extension of elevation mapping, in which each cell can store multiple height values. The proposed model was used to perform localization and navigation in an outdoor environment with several overhanging obstacles, where the robot successfully traversed across and under a bridge at the site.

Polygonal meshes are another spatial representation method that is particularly popular within the computer graphics community. A polygonal mesh is a graph of interconnected vertices in which each polygon represents a facet in the mesh. To generate the best reconstruction result from noisy point cloud data, special care has to be taken

in filtering and handling of uncertainty. For example, Wiemann et al. (2010) present a method to automatically generate triangle meshes from noisy registered point cloud data. However, the resulting triangle mesh is generated as a post-processing step, and it is not maintained online. Rusu et al. (2009) propose a complete pipeline from sensor data to localization, mapping, and path planning, where the maintained map is a polygonal mesh generated from point cloud data. Similarly, Garrido et al. (2013) exploit triangle meshes representing 3D surfaces to perform path planning for robots operating outdoors.

The normal distributions transform (NDT) is a compact spatial representation, originally introduced by Biber & Strasser (2003) in the context of 2D scan matching. NDT is a grid-based representation, much like occupancy grid maps but capable of obtaining similar accuracy while using much larger cell size (Stoyanov et al., 2013). The key idea of NDT is that the observed range points are represented as a set of Gaussian probability distributions computed for each cell. That is, each distribution describes the probability of a point being measured at a particular physical location. NDT Occupancy Map (NDT-OM) is an extended NDT map, which enables recursive updates of sequential measurements and models the occupancy probability of the cell (Saarinen et al., 2013b). As a result, NDT-OM is an efficient representation for long-term, large-scale mapping, which maintains its consistency even in dynamic environments. In addition, Stoyanov et al. (2013) demonstrate that NDT spatial models allow significantly lower resolution to be used compared to other state-of-the-art 3D spatial modeling techniques, without compromising the model accuracy. An illustration of an NDT map is shown in Figure 2, along with a photograph of the observed environment and measurement platform.

2.2. Traversability Analysis

In a recent survey by Papadakis (2013), different traversability analysis methods were classified into three major constituents — namely, proprioceptive, appearance-based, and geometry-based approaches; the latter two composed the domain of exteroceptive approaches. Furthermore, Papadakis (2013) notes that hybrid approaches do exist, which may further imply the use of additional sensor modalities other than LIDARs, cameras, or proprioceptive sensors.

Traversability analysis methods based on proprioceptive sensing are useful in learning models that captures the difficulty encountered while a vehicle is traversing a given terrain. For example, Martin et al. (2013) construct large-scale traversability maps for vehicles performing repeated activity in a bounded environment, based on the vehicle power consumption, longitudinal slip, lateral slip, and vehicle orientation. However, proprioceptive sensors cannot predict the traversability of terrain that is about to be visited by the robot, and thus cannot be used to assess the drivability of the robot surroundings. Therefore, proprioceptive

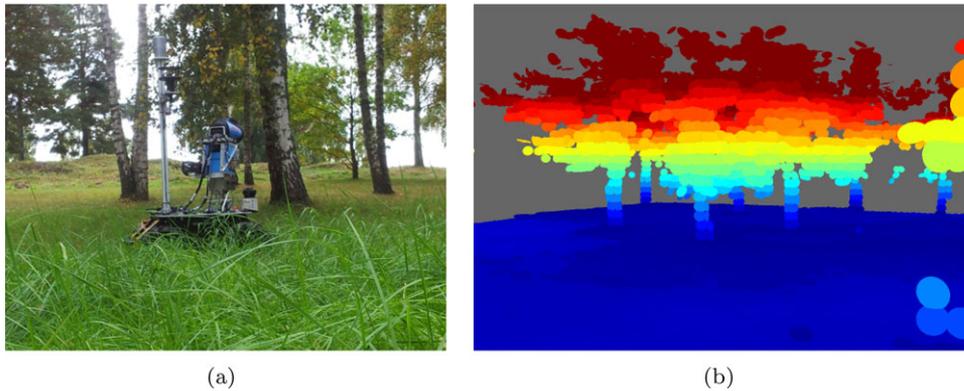


Figure 2. (a) A photograph from a forest environment featuring an UGV. (b) An illustration of an NDT map generated from forest data.

sensing should be further combined with long-range sensing modalities in order to avoid possible collisions with obstacles. Howard et al. (2006), for example, propose a learning method that associates proprioceptive sensor data underfoot with previously acquired visual information of the same terrain in order to train a model for predicting terrain properties from visual appearance only. A similar approach is presented in (Bajracharya et al., 2009).

Appearance-based approaches to traversability analysis reformulate the problem as an image-processing and classification task, and thus they usually choose between a discrete set of terrain classes, rather than regressing on traversability. Angelova et al. (2007) propose a method for learning a hierarchical classifier for color images to differentiate between sand, soil, asphalt, grass, wood chip, and gravel. Kim et al. (2007) perform natural terrain classification between traversable and nontraversable regions by using super-pixels extracted from an oversegmentation of an image. It was shown that these regions of homogeneous visual content are superior to rectangular image patches that are typically sensitive to the tessellation resolution and occlusions. However, an evident complementarity exists between LIDAR and vision sensors, which has been exploited in several works in order to increase the overall robustness or to extend the range of operations. For example, Zhou et al. (2012) introduce a self-supervised sensing approach that attempts to robustly identify a drivable terrain surface for UGVs operating in forested terrain, in which both LIDAR and vision sensors were employed. The LIDAR data are exploited to train a visual classifier to discriminate between the ground and nonground regions in the image, but the final terrain class prediction is performed solely from visual data.

Nonetheless, the majority of terrain traversability analysis methodologies are based on geometric processing, although the geometric information is often fused with other sensor modalities in order to improve the robustness of clas-

sification. Typically, a terrain model is built from 3D data and used to extract a set of features. On top of such a model, more complex and higher-level processing could be pursued by further taking into account a robot model, as well as stability and kinematic constraints.

One of the early approaches for geometric traversability analysis is to compute gradients for each cell in an elevation map, which are then compared against platform-specific thresholds (Chang et al., 1999). Another similar approach is to compute a traversability index for each cell in an elevation map using the slope and roughness of the terrain (Ye & Borenstein, 2004). Thrun et al. (2006) propose a more probabilistic grid-based approach, which forms the base for the terrain analysis module of the vehicle *Stanley* that won the DARPA Grand Challenge. This approach labels the cells as free, occupied, or unknown, based on vertical distance between nearby 3D LIDAR points. A probabilistic model was developed to take into account the errors in the robot's pose estimation.

Vandapel et al. (2004) present a method that uses local 3D point statistics to segment LIDAR data into three classes: clutter to capture grass and tree canopy, linear to capture thin objects like wires or tree branches, and surface to capture solid objects like ground terrain surface, rocks, or tree trunks. Similarly Lalonde et al. (2006) classify 3D LIDAR data online, based on their salient features (i.e., scatterness, surfaceness, and linearness). These features were computed using principal component analysis (PCA) of the neighboring 3D points. Gaussian Mixture Models (GMMs) for these classes were learned by employing expectation maximization on these features. The classification results were filtered to account for outliers in the results, after which the ground was discriminated from other surfaces. This approach has a similar idea to ours in that the classification model for complex 3D environments is learned from labeled point-cloud data. However, our approach considers the estimated permeability and the shape of the intensity distribution in

addition to the local geometry statistical analysis, thus improving classification accuracy in vegetated environments. In addition, the underlying NDT representation of our approach allows the use of significantly larger cell size. Furthermore, training the model in Lalonde et al. (2006) requires tedious hand-labeling, whereas our approach relies on automatic teaching example collection.

Santamaria-Navarro et al. (2015) present a high-level offline classification mechanism that learns traversable regions from large 3D point clouds gathered with a 3D LIDAR. The traversability is modeled as a GP and trained with automatically labeled data set. To perform the classification, slope and roughness are computed for all the points, using PCA. Two different classification approaches were proposed. The first employs GP regression and relies only on positive teaching samples, which are collected from the robot footprints. The second approach performs GP classification that requires teaching samples from both classes. Therefore, the remaining unlabeled points are randomly sampled to get the negative teaching examples. It was shown that the GP classification, although computationally more expensive, increases the classification accuracy. Contrary to our approach, the classification is only based on the slope and roughness, which may sometimes prove insufficient. Moreover, we perform the classification in the NDT framework instead of directly over the point clouds, which decreases the computational complexity and adds the benefit of making more informed decisions over data collected from multiple viewpoints.

Terrain traversability analysis within the NDT framework has only been addressed by Magnusson (2009) and Stoyanov et al. (2010), in which the CTC algorithm proposed compares the roughness and inclination calculated from each cell distributions against predefined thresholds in order to generate a traversability map for path planning purposes. These features are determined based on the eigenvectors and eigenvalues of the covariance matrices associated with each cell. Our method extends the CTC algorithm by adding intensity distribution and permeability as features and learning the classification model from vehicle experience. The CTC algorithm has been shown to perform well in structured environments, but underperforms in unstructured environments. Our method improves the classification accuracy in complex environments.

2.3. Vegetation Detection

Most of the current traversability analysis methods consider obstacles as rigid and static, which fails to deal with vegetation-like obstacles. This problem has been approached by classifying vegetation to distinguish it from other types of obstacles. One often exploited property of vegetation is the reflectance of chlorophyll, which is found in living plants, that strongly reflects near-infrared (NIR) light but absorbs blue and red visible light (Myneni & Hall,

1995). For example, Bradley et al. (2004) present a multi-spectral camera-based solution for detecting chlorophyll-rich vegetation, based on NIR light reflectance properties. Vegetation was detected by subtracting each pixel in the red channel of the visible-light image from the corresponding pixel in the near-infrared image and thresholding the result. The reflectivity of chlorophyll was also exploited by Wurm et al. (2014), wherein an SVM-based classifier was trained to classify grass in semistructured environments, such as parks. The classifier was trained in a self-supervised way by employing a vibration-based classifier to detect the surface type currently traversed by the robot.

Nguyen et al. (2012b) propose to detect vegetation in unstructured environments using a spreading algorithm for identifying color and texture dissimilarities between the neighboring pixels in multispectral images. The seed pixels, from which the spreading search is started, are selected by thresholding chlorophyll-rich vegetation pixels. In parallel, another spreading algorithm is carried out based on spectral reflectance. The results from the parallel-spreading searches are combined to form the final classification. The approach is reported to result in vegetation detection robust to illumination effects. In previous work, the authors also presented a method to double-check the passable vegetation by mounting an air compressor device in front of the vehicle and using motion detection techniques to confirm the vegetation classification (Nguyen et al., 2012a).

Another property of vegetation is that range measurements often penetrate sparse vegetation, contrary to solid obstacles. Lacaze et al. (2002) were first to exploit this property by counting the hits and misses of laser beams in cells in a voxel grid and using the obtained density value to classify solid obstacles. Macedo et al. (2000) present a statistical analysis of the range data produced by 2D LIDAR, with the goal of determining whether an obstacle is a rock (non-traversable) or a patch of grass (traversable). The proposed classifier compares local estimates of the variance and the skewness of the range distribution to predefined thresholds, resulting in robust classification even if the obstacles are partially occluded by vegetation. Castano & Matthies (2003) propose a related approach, applied to real-time foliage detection from 2D LIDAR measurements. In contrast to the work of Macedo et al. (2000), they use the expected localities and continuities of an obstacle, both in space and in time. The method first identifies those returns likely to be obstacles and then prunes them to eliminate false positives.

Wellington & Stentz (2004) propose a method for predicting the load-bearing surface within vegetation. They apply an online adaptive method to learn from experience the mapping between the real ground height and the LIDAR measurements. The environment is modeled as a set of voxels, and the number of LIDAR pass-throughs (number of LIDAR rays passing through the voxel) is recorded, along with the LIDAR rays that hit the voxel. Voxels that contain a mixture of hits and misses are then assumed to

contain vegetation. In addition, the proposed approach exploits the maximum laser remission values, simple statistics on height, and the salient features as in Lalonde et al. (2006) using a similar resolution. Their selection of features is most similar to ours. However, instead of the maximum intensity values, we take advantage of the intensity distributions that better captures the true underlying intensities of cells and provides us the intensity variance as an extra feature to use in the classification. Furthermore, we calculate the roughness and slope, whereas their approach relies on salient features. Nonetheless, their method is designed to estimate the ground-bearing surface within dense vegetation, whereas our goal is to classify solid nontraversable obstacles within sparse vegetation. That is, our approach classifies dense vegetation as an obstacle, while their method returns the estimated height of the load-bearing surface regardless of the vegetation density. This behavior is undesirable in our application scenario, because we need to avoid hitting solid obstacles that may be hidden behind the vegetation.

In subsequent work, Wellington (2005) and Wellington et al. (2006) propose a terrain model that includes spatial constraints. The work introduced Markov random field (MRF) models and Hidden semi-Markov model (HSMM) to model 3D structure of the terrain. The MRFs encode the assumptions that ground heights vary smoothly and terrain classes tend to cluster. In this work, they also incorporated infrared temperature and color information as additional features.

Unfortunately, it is not always enough for safe navigation to be able to distinguish vegetation from other obstacles, because there might be an obstacle hidden behind the vegetation. This problem was addressed in a recent study, in which UWB radar was used in parallel with LIDAR to augment traversability maps in vegetated environments (Ahtiainen et al., 2013, 2015). It was shown that because the utilized UWB radars are capable of penetrating around 40 cm of vegetation, it is possible to generate accurate traversability maps of densely vegetated environments such that solid obstacles hidden behind dense vegetation can still be detected. As in this work, we concentrate on traversability classification based on LIDAR measurements alone; we will treat very dense vegetation as nontraversable to avoid problems with hidden obstacles. This constraint may be relaxed in future works by fusing our traversability results with classifications based on the algorithms proposed in Ahtiainen et al. (2013) and Ahtiainen et al. (2015).

3. NORMAL DISTRIBUTIONS TRANSFORM TRAVERSABILITY MAPS (NDT-TM)

This section introduces the NDT-TM representation for traversability evaluation. The NDT-TM extends the NDT-OM (Saarinen et al., 2013b) for spatial modeling by adding permeability and intensity distribution to the representation. Two novel SVM-based traversability classification

methods are presented that exploit the NDT-TM framework.

3.1. Previous Work

The NDT-OM representation is a 3D spatial model based on a regular grid that concurrently estimates both the occupancy and the shape distribution in each cell. Formally, a cell c_i in NDT-OM is represented with parameters $c_i = \{\mu_i, P_i, N_i, p(m_i|z_{i=1:t})\}$, where μ_i and P_i are the parameters of the estimated Gaussian component, N_i is the number of points used in the estimation of normal distribution parameters so far, and $p(m_i|z_{i=1:t})$ is the probability of the cell being occupied. The minimum amount of parameters required to maintain an NDT-OM cell is 11 (mean, upper diagonal of covariance, number of points, and occupancy probability).

The only previous work on NDT-based traversability mapping was introduced in Stoyanov et al. (2010), in which the cell distributions $N(\mu_i, P_i)$ were exploited in estimating the terrain traversability. This CTC method calculates the roughness R and inclination θ of a cell c_i , which are then used for classifying the cells.

θ and R are calculated as in Magnusson (2009). First, we calculate the eigenvectors ($\vec{e}_0, \vec{e}_1, \vec{e}_2$) and corresponding eigenvalues ($\lambda_0, \lambda_1, \lambda_2$) of the covariance matrix of the Gaussian in c_i , where $\lambda_0 \leq \lambda_1 \leq \lambda_2$. The eigenvalues indicate the variance of the Gaussian along the corresponding eigenvectors and therefore can be used to determine the shape of the Gaussian. The roughness R depends on the smallest eigenvalue λ_0 : If λ_0 is significantly smaller than the other two eigenvalues, the Gaussian is approximately planar. Conversely, if λ_0 is large, this indicates that there is a lot of uncertainty along all principal directions and that no planar structure can be found. In other words, scattered points suggest that the Gaussian is rough and needs to be considered nontraversable. Thus, we evaluate the roughness of a cell based on the metric value of the smallest eigenvalue, that is, $R = \lambda_0$.

The inclination θ is calculated by computing the angle between \vec{e}_0 and the vertical surface normal $\vec{n}_v = (0, 0, 1)$ using the dot product:

$$\cos(\theta) = \vec{e}_0 \cdot \vec{n}_v. \quad (1)$$

For rough cells the direction of \vec{e}_0 is arbitrary as there are multiple possible planar fits to the data, and thus θ might not correspond to the inclination angle. However, with planar distributions, θ yields a good estimate of the true inclination.

The CTC algorithm is presented in Algorithm 1. Cells are classified as rough, horizontal, vertical, or inclined planar cells. All horizontal cells as well as the inclined cells whose inclination angle does not exceed the maximum pitch angle of the vehicle θ_{max} are considered traversable. The remaining inclined cells, vertical cells, and rough cells are

considered nontraversable. First, we check whether R is greater than predefined threshold R_{th} . In this case, the cell is labeled as rough (i.e., nontraversable). In case $R \leq R_{th}$, we proceed to check the inclination. In this step, the cell is first classified as inclined. However, if θ is greater than predefined threshold θ_v , the label is changed to vertical. On the other hand, if θ is less than another threshold θ_h , the label is changed to horizontal.

Algorithm 1. Constant threshold classification (CTC) algorithm

```

1:  if  $R > R_{th}$  then
2:     $Class \leftarrow ROUGH$ 
3:  else
4:     $Class \leftarrow INCLINED$ 
5:    if  $\theta > \theta_v$  then
6:       $Class \leftarrow VERTICAL$ 
7:    if  $\theta < \theta_h$  then
8:       $Class \leftarrow HORIZONTAL$ 

```

3.2. NDT-TM Representation

The CTC algorithm was shown to perform well in structured environments, but it runs into trouble in more complex areas, for example, among vegetation. Because of the diversity of unstructured environments, the NDT-TM representation incorporates additional features to enhance the classification accuracy. These additional features are expected permeability, which captures the probability of a LIDAR ray passing through a cell distribution $N(\mu_i, P_i)$ rather than reflecting back from it, and intensity distribution (mean and variance), which gives information of the reflectivity of a target. Permeability was selected because sparse vegetation reflects some of the laser beams back to the LIDAR, but the rest pass through and may be reflected by background objects. Permeability is denoted with ρ . Intensity was chosen since the NIR light emitted by most LIDARs is strongly reflected from chlorophyll-rich vegetation. The intensity of a $N(\mu_i, P_i)$ is modeled with a one-dimensional normal distribution of all the measurements used to calculate the $N(\mu_i, P_i)$. Therefore, intensity is represented by the intensity mean μ_I and the intensity variance Σ_I .

3.2.1. Intensity Distribution Mapping

An example of vegetation reflectivity can be seen in Figure 3, in which the individual intensity measurements of laser beams are plotted as a function of measured distance. The blue crosses represent measurement reflecting back from vegetated surface, while the red circles denote measurements from asphalt. Clearly, the intensities of the measurements backscattered from the vegetation are systematically greater than those from asphalt. Note also that

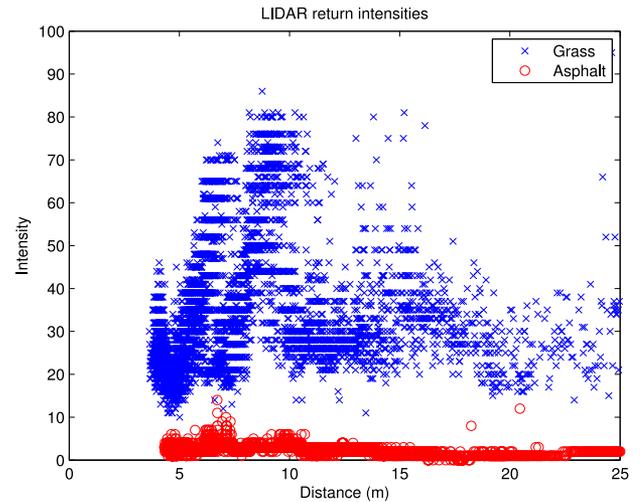


Figure 3. The red circles and the blue crosses represent the intensity of LIDAR measurement hitting asphalt and grass, respectively, as a function of distance. The measurements originate from multiple cells.

the intensity of reflected light does not only depend on the material of the measured surface but also varies slightly as a function of distance. In addition, the incidence angle also affects intensity measurements (Baribeau et al., 1992).

Although the intensity varies as a function of distance and incidence angle, we have opted to model the intensity of a cell with a one-dimensional normal distribution of all the measurements hitting the cell. However, we have truncated the maximum observation distance to 20 m, because the intensity measurements tend to become more inaccurate as the range increases. There are four reasons for this simplified modeling decision. First, the individual cells are hit by LIDAR measurements from all distances as the platform moves, which evens out the effects caused by observation distance variation. Second, there might be several different materials on the cell distribution area, especially in unstructured environments, that influence the data. Thirdly, the data of individual cells varies significantly from one environment to another, providing little justification for more complex models. Fourthly, the normal distribution provides a reasonable estimate of the intensity for real data from different environments (see Figure 4).

The intensity distribution requires storing of two additional values, the intensity mean μ_I and the intensity variance Σ_I . These values are calculated with the recursive covariance method (RCS) (Saarinen et al., 2013b), which allows combination of two arbitrary-sized sample sets.

3.2.2. Permeability Mapping

The laser rays passing through a cell distribution $N(\mu_i, P_i)$ can be modeled as Bernoulli process; that is, the

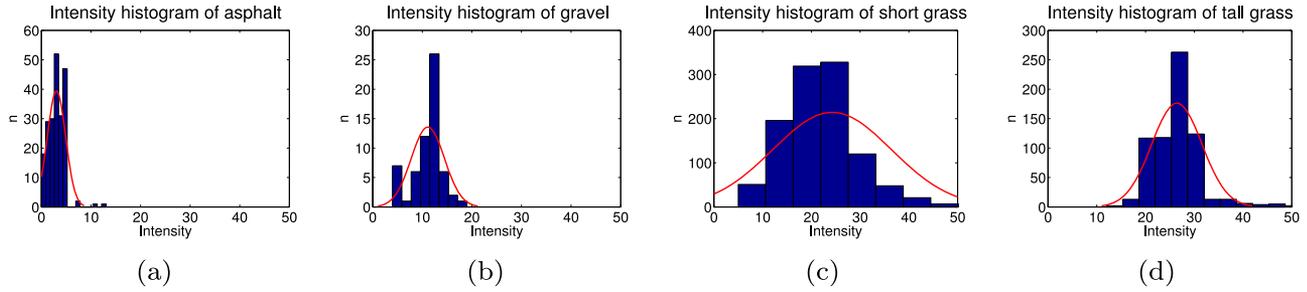


Figure 4. Histograms of the intensity measurements of a cell from different environments. The red line is a normal distribution calculated from the data. (a) Asphalt. (b) Gravel. (c) Short grass. (d) Tall grass.

probability of each ray passing through $N(\mu_i, P_i)$ has a Bernoulli distribution. The probability mass function of a Bernoulli distribution is

$$P_i(k) = p_i^k (1 - p_i)^{1-k}, \quad k \in \{0, 1\}, \quad (2)$$

where i refers to a particular cell, p_i is the success probability of that cell, and k is the outcome parameter, that is, $k = 0$ when a ray hits the distribution and $k = 1$ when a ray passes through the distribution. We learn p_i parameters using Bayesian estimation with beta distribution prior, which is the conjugate prior of a Bernoulli distribution. The hyperparameters α and β are updated for each cell with

$$\alpha_n = \begin{cases} \alpha_{n-1} + 1, & \text{if } k_n = 1 \\ \alpha_{n-1}, & \text{if } k_n = 0 \end{cases} \quad (3)$$

and

$$\beta_n = \begin{cases} \beta_{n-1}, & \text{if } k_n = 1 \\ \beta_{n-1} + 1, & \text{if } k_n = 0, \end{cases} \quad (4)$$

where n is the number of laser rays hitting or passing through the distribution. α and β correspond to the number of misses N_m and the number of hits N_h , respectively. For the estimation of the success probability p_i , the expectation of the beta distribution is used:

$$\hat{p}_i = E[p_i] = \frac{\alpha}{\alpha + \beta}. \quad (5)$$

Finally, we get the Bernoulli distributed probability of a laser ray hitting $N(\mu_i, P_i)$ by setting \hat{p} into (2) with $k = 1$. This probability is referred to as permeability, and it is denoted with ρ for the rest of this paper. That is,

$$\rho = P(k = 1) = \hat{p}^1 (1 - \hat{p})^0 = \hat{p} = \frac{\alpha}{\alpha + \beta} = \frac{N_m}{N_m + N_h}. \quad (6)$$

Equation (6) gives us a way to estimate the permeability by counting the number of hits N_h and misses N_m for a particular Gaussian $N(\mu_i, P_i)$. Figure 5 illustrates how the hits and misses are calculated by pointing out the different cases of a laser ray hitting or missing $N(\mu_i, P_i)$, namely:

- A-a) Miss: the ray passes through a Gaussian and hits a different cell

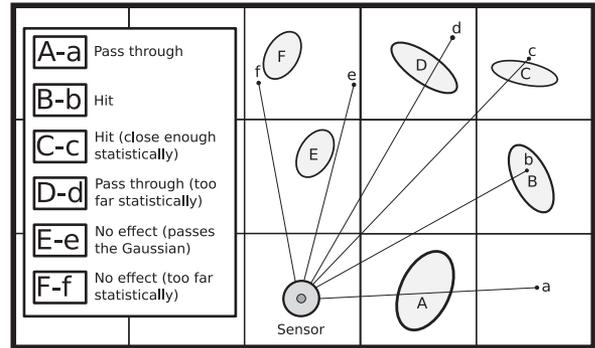


Figure 5. An illustration of the different cases of a laser ray passing through or hitting a Gaussian distribution.

- B-b) Hit: the measured point hits the Gaussian in the same cell
- C-c) Hit: the measured point is statistically close enough to the Gaussian in the same cell
- D-d) Miss: the ray passes through a Gaussian in the same cell, but the measured point is not statistically close enough
- E-e) No effect: the ray passes a Gaussian in a different cell
- F-f) No effect: the ray does not pass through the Gaussian in the same cell and the measured point is not statistically close enough

Counting the hits and misses is based on calculating the maximum likelihood along a line given a Gaussian distribution and the likelihood of the maximum likelihood point x_{ml} being measured given the observation z_i . Figure 6 illustrates how the maximum likelihood point along a line is calculated. x_{ml} is the point along the line that maximizes the likelihood of the point belonging to a particular Gaussian distribution. Details on how to calculate this point can be found from Saarinen et al. (2013b).

The candidate point x_{ml} is calculated for each cell that has a Gaussian along the laser ray. These cells are found by performing ray tracing from the sensor origin x_s to the

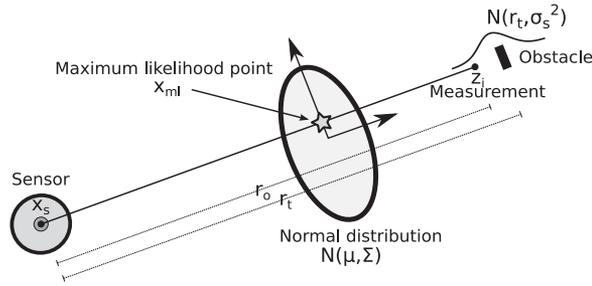


Figure 6. The maximum likelihood along a line from the sensor to the measured point.

measured point z_i . The likelihood of x_{ml} given the cell distribution is calculated as:

$$p(x_{ml}|N(\mu, P)) \sim \exp\left(-\frac{1}{2}(x_{ml} - \mu)^T P^{-1}(x_{ml} - \mu)\right). \quad (7)$$

In addition to (7), we need to consider the sensor noise. We assume that the range of the sensor $r_o = \|z_i - x_s\|$ can be represented with a normal distribution along the line from x_s to z_i such that $r_o = N(r_t, \sigma_s^2)$, where r_t is the true distance to an obstacle and σ_s represents the sensor noise. Thus, the likelihood of x_{ml} being measured given the observation z_i is:

$$p(x_{ml}|z_i) \sim \exp\left(-\frac{1}{2} \frac{\|x_{ml} - z_i\|^2}{\sigma_s^2}\right). \quad (8)$$

Now, if the tested cell distribution is in a different cell than z_i (i.e., case A-a or E-e), we update N_m with:

$$N_m = \begin{cases} N_m + 1, & \text{if } p(x_{ml}|N(\mu, P)) * (1 - p(x_{ml}|z_i)) \geq \eta \\ N_m, & \text{otherwise,} \end{cases} \quad (9)$$

where η is a threshold parameter that can be used to adjust the cutoff likelihood. In other words, we update N_m when the (7) is large (i.e., it is likely that the ray passes through the cell distribution) and the (8) is small (i.e., the observation is not near the cell distribution). Because (9) depends on both the (7) and (8), it is not straightforward to analytically select optimal value for η . Therefore, we heuristically use a value of 0.3, which is roughly equivalent to 1- σ error level of the likelihood function, throughout our experiments.

When we are examining a cell distribution within the same cell as the observation, we do the following. First, we want to update the cell distribution with the new measurement before performing the check because the shape of the distribution might change significantly, especially if the cell was just observed. This is performed with the RCS method. Second, we need more careful analysis of the likelihoods to cover all the cases in Figure 5.

Determining a hit (i.e., cases B-b and C-c) is similar to (9), but in this case, we want the observation to be near the cell distribution. That is, we update N_h with:

$$N_h = \begin{cases} N_h + 1, & \text{if } p(x_{ml}|N(\mu, P)) * (p(x_{ml}|z_i)) \geq \eta \\ N_h, & \text{otherwise.} \end{cases} \quad (10)$$

To update the misses (i.e., case D-d), we first need to check that the ray passes through the cell distribution. This is performed by first checking that the distance to x_{ml} is not greater than the measured range, because in this case the measurement cannot pass through the cell distribution. In addition, we need to check that x_{ml} is likely to belong to the cell distribution but is not within the sensor noise from the measured point. In all other cases, the observation does not hit nor travel through the cell distribution (i.e., case F-f). N_m is updated with:

$$N_m = \begin{cases} N_m + 1, & \text{if } p(x_{ml}|N(\mu, P)) * (p(x_{ml}|z_i)) < \eta \\ & \wedge p(x_{ml}|N(\mu, P)) \geq \eta \\ N_m, & \text{otherwise.} \end{cases} \quad (11)$$

3.3. Traversability Classification

Terrain traversability depends on the aforementioned features, but it is not straightforward to determine the relationship between these features and the traversability class. Therefore, we exploit a well-known SVM framework (Cortes & Vapnik, 1995) to learn the model behind the data. Two different classification techniques for classifying each cell are presented in this section. The first method exploits the C-support vector classification (C-SVC) to estimate the distribution behind the data, based on training samples from both classes. The second method is a hybrid method called ACTC and it classifies the cells first by using CTC algorithm and performs the C-SVC classification only on the cells classified as nontraversable by the CTC method. The SVM framework was chosen for traversability classification because it has been shown to work well in similar nonlinear, two-class classification problems (Gestel et al., 2004).

Nonlinear classification is performed with the so-called kernel trick, in which inputs are mapped into a higher-dimensional space with a kernel function. In our case, a Gaussian radial basis function (RBF) was selected for the kernel function because it can handle nonlinear relations and tends to perform well in cases in which there are only few features (Hsu et al., 2010).

3.3.1 C-Support Vector Classifier (C-SVC)

To avoid tedious hand-labeling of the teaching data, we have chosen a similar approach as in Thrun et al.

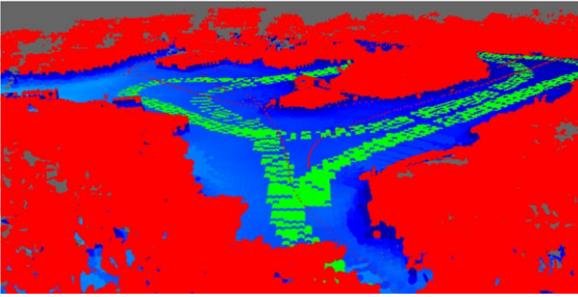


Figure 7. An example of an NDT-OM map in which positive training samples are marked with green and negative with red. The positive samples correspond to the UGV footprint cells, whereas the negative samples correspond to the cells above or below the ground plane.

(2006) to automatically collect positive teaching samples by recording the measurement platform footprint cells. Collecting the negative samples is more problematic. Our insight is that when we are operating on relatively flat terrain, all the cells above or below the ground plane by some margin can be labeled as negative samples. Clearly, not all of those cells are nontraversable; however, even training against an approximate labeling is enough to improve the overall performance of the classifier. Naturally, applying this method requires that the environment is mainly level and semicontrolled, such that areas above the ground plane are mostly nontraversable. An example of such an environment can be seen in Figure 7, in which the cells labeled using this method capture the forbidden areas with acceptable accuracy. To train a robust classifier for versatile environments, the training samples should also be collected on various terrain types.

Learning the classification model from the training samples is based on SVM that are kernel-based learning methods for data classification and regression problems. SVMs learn a hyperplane in a higher-dimensional feature space, which separates the two classes of data points, maximizing the margin between training points and the hyperplane. However, the C-SVC formulation of the SVM introduces a soft margin that adjusts the trade-off between maximizing the margin and minimizing the training error. Given training vectors $\mathbf{x}_i \in \mathbb{R}^l$, $i = 1, \dots, l$, in two classes and class label vector $\mathbf{y} \in \mathbb{R}^l$ such that $y_i \in [1, -1]$, C-SVC solves the following dual problem:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} - \mathbf{e}^T \boldsymbol{\alpha} \\ \text{subject to} \quad & \mathbf{y}^T \boldsymbol{\alpha} = 0, \\ & 0 \leq \alpha_i \leq C, \quad i = 1, \dots, l, \end{aligned} \quad (12)$$

where C is a regularization parameter, $\mathbf{e} = [1, \dots, 1]^T$ is a vector of ones, \mathbf{Q} is an l by l positive semidefinite matrix, and $Q_{ij} \equiv y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$, where $K(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel function. We use the RBF kernel function, which is defined as $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$, $\gamma > 0$, where $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, \dots, l$, is training vector and γ is kernel parameter. The following decision function is used to predict the class labels:

$$\mathbf{y} = \text{sgn} \left(\sum_{i=1}^l y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \right), \quad (13)$$

where b is the bias term (Hsu et al., 2010).

The free parameters in training the C-SVC with RBF kernel are C , which trades off misclassification of training samples against simplicity of the decision surface, and γ , which defines how far the influence of a single training sample reaches. When C is low, the margins of the support vectors are small, which might cause overfitting. With small γ , the influence of the training samples ranges far. Parameter selection can be done by cross validation, which prevents the overfitting problem. A grid-search method can be used for picking the parameters. First, a coarse grid-search should be performed, because model training with large data sets can be time consuming. After identifying the region of parameters in which the classification results seem reasonable, a finer grid-search is conducted for that region. However, with noisy training and testing set, the prediction accuracy is not always the only thing to take into account. Therefore, the best model should be selected based on how the trained model performs in real-world situations.

3.3.2 Augmented Constant Threshold Classifier (ACTC)

The CTC algorithm is shown to work well in structured and semistructured environments (Stoyanov et al., 2010). Therefore, the overall prediction accuracy could be improved by first classifying the cells with the CTC algorithm and using the C-SVC classification only for cells that have been classified as nontraversable. This prevents the potential misclassification of traversable cells. Moreover, using this hybrid method also decreases the computational complexity of the classification compared to the C-SVC method. However, because we are only interested in classifying the cells as traversable or nontraversable (as opposed to rough, horizontal, vertical, or inclined), we do not need to perform all the steps of the CTC method. First, we check whether a cell is rough and perform C-SVC classification for the cell in case this condition is met. If the first check is passed, we proceed to check whether the inclination angle θ exceeds the maximum allowed pitch angle θ_{max} of the vehicle and perform C-SVC classification in case the inclination exceeds the threshold. The rest of the cells are classified as traversable because they are determined to be smooth (\neg rough) and the

inclination is within acceptable limits. The ACTC algorithm is described in Algorithm 2.

Algorithm 2. Augmented constant threshold classifier (ACTC)

```

1: if  $R > R_{th}$  then
2:    $Class \leftarrow C-SVC(R, \theta, \rho, \mu_I, \Sigma_I)$ 
3: else if  $\theta > \theta_{max}$  then
4:    $Class \leftarrow C-SVC(R, \theta, \rho, \mu_I, \Sigma_I)$ 
5: else
6:    $Class \leftarrow TRAVERSABLE$ 

```

4. SYSTEM DESCRIPTION

This section presents the experimental systems used for validation of the proposed approach and discusses some implementation details.

4.1. Measurement Platforms

Two different measurement platforms were used in our experiments. The first one is a fully electric all-terrain vehicle (ATV) Polaris Ranger Ev 4x4 equipped with a Velodyne HDL-32E high-definition LIDAR, which can be seen on the roof of the measurement platform in Figure 1. The sensor has 32 laser/detector pairs aligned vertically such that the vertical FOV is 41.3° ($+10.67^\circ$ to -30.67°). The sensor spins at 10 Hz giving it horizontal FOV of 360° , generating approximately 700 000 data points per second. The sensor is mounted on the top of the vehicle such that the origin of the LIDAR is at the height of 2.38 m .

The other measurement platform is an all-terrain Husky A-200 robot equipped with different sensing modalities for robot localization and environmental monitoring (see Figure 2(a)) (Bennetts et al., 2014). However, in this work, only the 3D LIDAR (Velodyne HDL-32E) mounted on a pole on top of the vehicle at the height of 0.9 m was exploited.

4.2. Implementation

The localization data of the platform was estimated using the iterative closest point (ICP) algorithm. We used the *ethzasl_icp_mapper* implementation that is available in ROS (Pomerleau et al., 2013). Only LIDAR data were used as an input for the algorithm. The NDT-OM maps are generated using the ROS packages of Applied Autonomous Sensor Systems (AASS) laboratory at Örebro University in Sweden (Saarinen et al., 2013b). However, we made some modification in the source code, in order to compute the intensity distributions and the permeability of each Gaussian, as well as to perform SVM prediction with the pretrained

model. A widely used library *libSVM* (Chang & Lin, 2011) was exploited in the learning and prediction steps.

It is very important to scale the parameters before training the SVM. Scaling prevents the attributes in greater numeric ranges from dominating the smaller ones and helps in avoiding numeric instabilities. We scaled each attribute linearly to the range $[0, 1]$. Exactly the same scaling parameters need to be used to scale both the training and the testing data, which means that the scaled testing data may be slightly out of the range $[0, 1]$. The scaling parameters are calculated before training the model and used with the trained model in the prediction phase.

5. EXPERIMENTS

This section presents results of the experiments performed to validate the proposed approach. Subsection 5.1 illustrates the effect of vegetation on the selected features in the NDT-TM framework and analyses the effects of the map resolution. Subsection 5.2 demonstrates the performance of the proposed traversability classification methods in five different field trials, compared against previous work within the NDT framework.

5.1. Vegetation and the Features

First, we tested how the selected features behave in the presence of vegetation. Four different types of vegetation were placed progressively in front of the measurement platform on a gravel surface, and the platform was driven approximately 10 m toward the target. The movement enabled the generation of NDT-TM maps from the data. The vegetation types are presented in Figure 8 and referred to from now on as *Vege1* to *Vege4*. *Vege1* has fairly large waxy leaves and a woody stem. *Vege2* has a woody stem as well, but there are more branches, and the leaves are smaller and thinner. *Vege3* is herbaceous with seed heads and long tapering leaves. *Vege4* is herbaceous with small, dark green pinnate leaves.

The vegetation was placed on the field layer by layer, such that one layer consisted of two pieces of a particular vegetation placed side by side with 20 cm between them. A new layer was always placed 10 cm in front of the old one. Figures 8(a)–8(d) illustrate the placing of vegetation. Each layer of vegetation added more than 30 g of foliage. The stem weights of the woody stemmed vegetation (i.e., *Vege1* and *Vege2*) were around 20 g per layer.

The following bar diagrams in Figures 9, 10, and 11 illustrate how the selected features behave in the presence of vegetation and obstacles. All the values in the figures are mean values of the cells covering the obstacle area. Obviously, when the cell size increases, a lower number of cells are affected by the obstacles. Furthermore, it should be noted that, depending on the resolution, the cell



Figure 8. (a) Two layers of *Vege1* with a fairly large rock behind the vegetation. (b) One layer of *Vege2*. (c) Four layers of *Vege3*. (d) Three layers of *Vege4*.

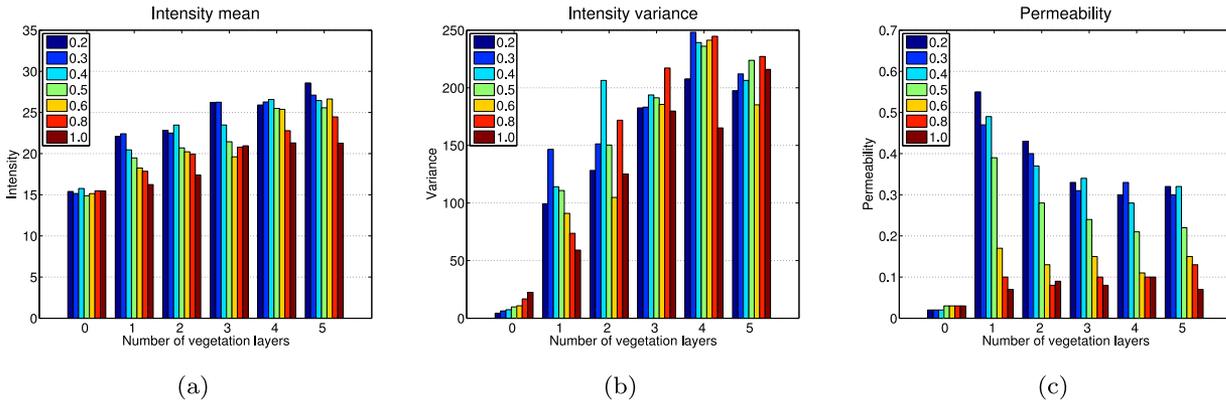


Figure 9. Bar diagrams of (a) intensity mean, (b) intensity variance, and (c) expected permeability using different cell resolutions with variable amount of vegetation (*Vege1*).

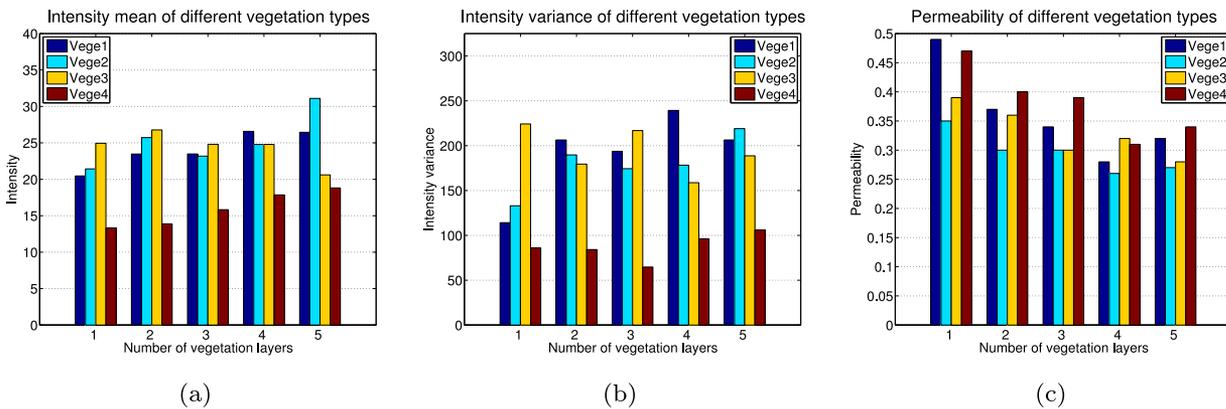


Figure 10. Bar diagrams of (a) intensity mean, (b) intensity variance, and (c) permeability of different vegetation types (*Vege1-Vege4*) with variable amount of vegetation.

distributions may capture only partially the vegetation and partially the ground plane.

The most important parameter of the NDT-OM algorithm is the resolution of the grid used for mapping. As demonstrated in Saarinen et al. (2013a), even with very large resolution it is possible to obtain reasonable localization results using NDT maps. However, when analyzing

traversability, it is beneficial that the distributions capture the geometry of the obstacles accurately, which sets a lower limit on the resolution. The bar diagrams in Figure 9 illustrate the effect of the resolution for the intensity distribution and permeability of vegetation using *Vege1*. The bars in the figures correspond to the intensity mean μ_I (Figure 9(a)), intensity variance Σ_I (Figure 9(b)), and permeability

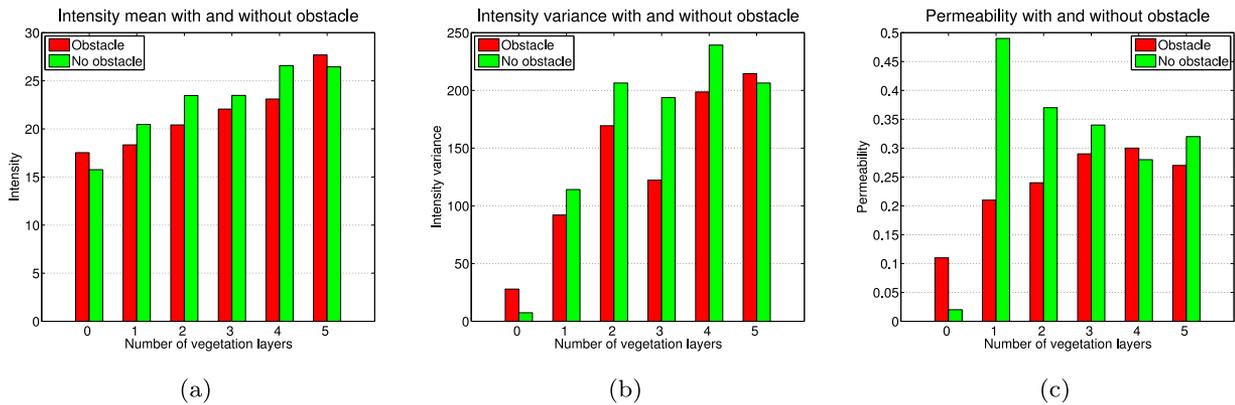


Figure 11. Bar diagrams illustrating (a) intensity mean, (b) intensity variance, and (c) permeability with and without an obstacle behind a variable amount of vegetation (Vege1).

ρ (Figure 9(c)), respectively. Bars of different colors indicate different map resolutions.

Generally, μ_I decreases as the cell size increases, because large cells cover more of the ground plane rather than just vegetation. Furthermore, the intensity increases along with the amount of vegetation present. Σ_I does not follow such a clear pattern. However, from Figure 9(b), it is evident that the variance of the ground plane is small, which decreases Σ_I in the cases for which the distributions also partially capture the ground plane. Interestingly, the difference between the mean values of vegetation and the ground plane is relatively small, but the vegetation seems to increase the intensity variance significantly. This results mainly from the LIDAR rays hitting different parts of the vegetation that have different reflectance properties. That is, the LIDAR rays backscatter strongly from the leaves, whereas the round stems tend to attenuate the measurements. In addition, some of the measurements originate from the ground plane, which further adds to the intensity variance.

The permeability parameter ρ decreases as a function of the resolution, because the distributions capture more of the ground plane (i.e., become flatter) when the cell size increases and the laser beams no longer pass through them that easily. The increase in the amount of vegetation also decreases ρ , because thick vegetation is no longer penetrable for the laser beams.

On the basis of the Figure 9, we chose the resolution to be 0.4 m , which still captures nicely the geometry of even small objects while still allowing real-time update rates of the NDT-OM map (Saarinen et al., 2013b). Especially, the permeability is quickly affected with larger resolutions, which affects the classification of sparse vegetation significantly. For the experiments reported in the rest of this article, the resolution of all the NDT maps was set to 0.4 m .

Figure 10 illustrates the effect of different vegetation types on the selected features. Notably, due to its dark green

and small leaves *Vege4* results in the lowest μ_I and generally smaller values of Σ_I . On the other hand, due to the small leaves, ρ is notably larger than for *Vege2* and *Vege3*. Nonetheless, the large but relatively sparse leaves of *Vege1* result in similar ρ as for *Vege4*.

Unfortunately, it is not enough for safe navigation to be able to distinguish vegetation from other obstacles, because there might be an obstacle hidden behind the vegetation. Figure 11 depicts how the selected features behave if there is a large rock behind the vegetation. The vegetation used in this experiment was *Vege1* and the rock can be seen in Figure 8(a) behind the vegetation.

The behavior of ρ parameter in this figure is especially interesting. There are clear differences between the values with and without the rock, up until three layers of vegetation. After this point, the values became too similar to make accurate distinctions between the two cases. Furthermore, there are clear differences also in the intensity distributions up until three layers of vegetation. Overall, as expected, the evidence supports that detecting an obstacle behind vegetation based only on LIDAR data can be done as long as parts of the obstacle are still visible. In this experiment, if the obstacle is hidden by more than three layers of vegetation, it is no longer clearly visible and cannot be detected using only the LIDAR data.

5.2. Application: Field Trials

To validate the classification performance of the proposed algorithms, four field trials were conducted in controlled environments on different surfaces using the Polaris Ranger as a measurement platform. Trial 1 was performed on asphalt, Trial 2 on gravel, Trial 3 on a field with sparse grass, and Trial 4 on a densely vegetated field with obstacles. Photographs of the environments are shown in Figure 12, while the corresponding NDT-OM maps and vehicle trajectories are shown in Figure 13. Moreover, we performed an



Figure 12. Photographs of the different field trial environments: (a) asphalt, (b) gravel, (c) sparse vegetation, and (d) dense vegetation.

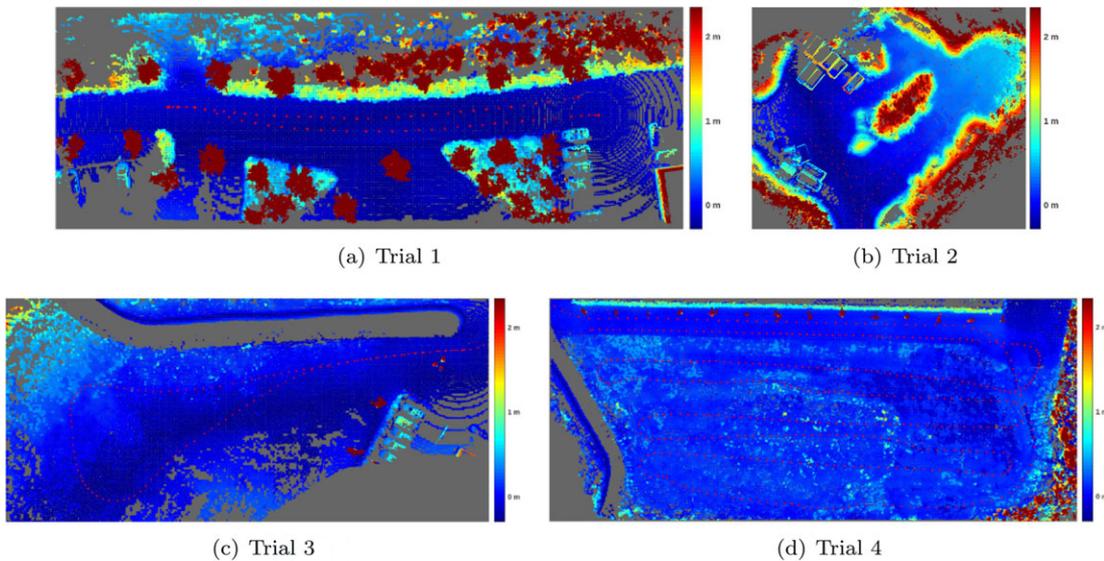


Figure 13. NDT-OM maps of the field trial areas colored by height. The red dotted lines on the maps represent the sensor trajectories.

additional experiment (Trial 5) in a forest environment without ground truth information using the Husky-A200 measurement platform. A photograph from the forest environment is shown in Figure 2(a). Table I presents illus-

trative information about the performed trials. For all the experiments, the classification was performed offline on collected data and using the free parameters presented in Table II.

Table I. Field trials in numbers.

	Trajectory length	Duration	Average speed	Area size	Number of cells
Trial 1: Asphalt	127 <i>m</i>	121 <i>s</i>	1.0 <i>m/s</i>	100 × 30 <i>m</i> ²	29 176
Trial 2: Gravel	280 <i>m</i>	230 <i>s</i>	1.2 <i>m/s</i>	100 × 100 <i>m</i> ²	48 898
Trial 3: Sparse grass	186 <i>m</i>	126 <i>s</i>	1.5 <i>m/s</i>	90 × 60 <i>m</i> ²	19 737
Trial 4: Dense grass	1274 <i>m</i>	918 <i>s</i>	1.4 <i>m/s</i>	165 × 70 <i>m</i> ²	100 887
Trial 5: Forest	172 <i>m</i>	299 <i>s</i>	0.6 <i>m/s</i>	60 × 50 <i>m</i> ²	44 916

Table II. The free parameters and their values used in the field trials.

Resolution	R_{th}	θ_h	θ_{max}	θ_v	γ	C	η
0.4	0.005	10°	30°	80°	0.0625	0.125	0.3

We compared the two proposed SVM-based classification methods, that is, the C-support vector classification (C-SVC) and the ACTC, against the CTC algorithm. The classification performance is measured in the following way. First, we run the classification algorithms on the robot footprint points, and we use this as the ground truth to compute *recall* ratios shown in Table IV. The *recall* is defined as the ratio of *true positives* over the sum of *true positive* and *false negative* samples. *True positive* and *false negative* samples are the cells on the footprint that were classified, respectively, as traversable/nontraversable.

However, this evaluation does not take the obstacles in the area into account. Therefore, we have hand-labeled obstacles from the data, which enables also the computation of *precision* and *f-score* statistics. By *precision*, we mean the ratio of *true positives* over the sum of *true positives* and *false positives*. The *f-score* is computed as twice the product of *precision* and *recall* over the sum of *precision* and *recall*.

Unfortunately, it is not feasible to label each cell individually due to the vast number of cells. Moreover, it would be extremely challenging to determine the true class of some of the cells even for a human expert. Thus, we have hand-labeled the areas that we know should not be

traversed. This, however, means that the labeled areas include also some traversable cells, which has a significant effect on the classification measures. To illustrate the problem of evaluating traversability directly on the classification results for the NDT cells in 3D, imagine two areas of flat traversable ground, separated by a nontraversable ditch. Though each area is traversable in isolation, only one of them is reachable for the robot at any given time. In a similar manner, there may be locally traversable cells in the midst of nontraversable areas. Therefore, in addition to the values calculated based on the classified NDT-TM map, we also compute the *recall*, *precision* and *f-score* for connectivity maps extracted from the NDT-TM.

We define a connectivity map as the set of all reachable traversable cells from a given known robot position. As the test environments used in this article do not feature any drivable overpasses, connectivity maps can be efficiently represented as a 2D projection of the drivable 3D terrain. We compute connectivity maps from the NDT-TM by wavefront propagation in 3D, similar to prior work (Stoyanov et al., 2010). For each cell in the wavefront, we find the traversable neighbors and add them if no overhanging obstacles would cause collisions and if the mean-to-mean slope is below a vehicle-dependent maximum step threshold (see Figure 14(b) for an example). For example, in Figure 14, a deciduous tree is highlighted with a black rectangle in the NDT-TM and connectivity map. Because the maps are shown from above, the tree seems to be a large obstacle in the NDT-TM, but in the connectivity map, only the tree trunk is classified as nontraversable because the leaves are above the vehicle height.

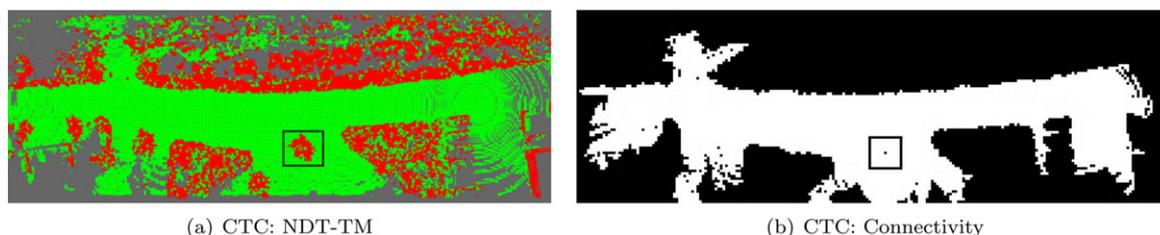


Figure 14. (a) NDT-TM map from Trial 1 classified with CTC method (red means nontraversable). (b) The corresponding connectivity map (white region is traversable). The black rectangles highlight a deciduous tree from the maps. On the connectivity map, only the tree trunk is classified as nontraversable because the leaves are above the vehicle height. The size of the test area is 100 × 30 *m*². The corresponding maps classified with the C-SVC and ACTC are nearly identical.

Table III. Classification times.

	Total (s)/per cell (10^{-3} s)		
	CTC	C-SVC	ACTC
Trial 1: Asphalt	0.0131/0.0005	21.42/0.7342	11.65/0.3993 (54.34%)
Trial 2: Gravel	0.0388/0.0008	35.92/0.7346	20.07/0.4105 (55.78%)
Trial 3: Sparse grass	0.0206/0.0010	14.52/0.7357	3.52/0.1784 (24.18%)
Trial 4: Dense Grass	0.0479/0.0005	74.09/0.7344	22.23/0.2204 (29.90%)
Forest:	0.0212/0.0005	32.77/0.7296	19.93/0.4437 (61.17%)

Table IV. Traversability classification results.

	Cell map			Connectivity map		
	Precision	Recall	f-score	Precision	Recall	f-score
Trial 1: Asphalt						
CTC	0.8609	0.9994	0.9250	0.9703	0.9983	0.9841
C-SVC	0.8187	0.9994	0.9001	0.9691	0.9983	0.9835
ACTC	0.8170	0.9994	0.8990	0.9691	0.9983	0.9835
Trial 2: Gravel						
CTC	0.8248	0.9733	0.8929	0.8709	0.9784	0.9215
C-SVC	0.8170	0.9938	0.8968	0.8525	0.9953	0.9184
ACTC	0.8076	0.9949	0.8915	0.8476	0.9979	0.9166
Trial 3: Sparse grass						
CTC	0.9427	0.9373	0.9400	0.9810	0.9017	0.9397
C-SVC	0.9481	0.9696	0.9587	0.9787	0.9490	0.9636
ACTC	0.9422	0.9702	0.9560	0.9735	0.9502	0.9617
Trial 4: Dense grass						
CTC	0.9486	0.7961	0.8657	0.9755	0.6175	0.7563
C-SVC	0.9405	0.8417	0.8884	0.9855	0.6977	0.8170
ACTC	0.9383	0.8516	0.8929	0.9822	0.7113	0.8250

are above the vehicle height. Connectivity maps can be directly used as a costmap input to classical path-planning algorithms, because they already reduce the dimensionality from 3D to 2D by taking the vehicle constraints into consideration. Finally, connectivity maps are ideally suited for evaluating the performance of the algorithms proposed in this paper, as they capture better the concept of drivability used by human operators and filter out any unreachable locally traversable patches. The *recall*, *precision*, and *f-score* values for all the data sets are shown in Table IV. In the table, the best results for each experiment (i.e., highest *precision*, *recall*, and *f-score*) are shown in bold font. The same numbers cannot be computed for Trial 5 because obtaining labeled ground truth in this trial was not feasible.

The classification times for the experiments are reported in Table III, specifying both the total time of classifying the whole map as well as the average time of classifying individual cells. The percentages inside parenthesis in the ACTC column are the percentages of cells classified using

the C-SVC method. For the reported classification times, we run the experiments on an Intel Xeon E3-1230 V2 processor at 3.30 GHz, with 16 GB of RAM, running Ubuntu 12.04 32 bits.

Table IV shows that the CTC algorithm is marginally better than the two proposed classification approaches in Trial 1, but the differences are insignificant. That is, all the approaches perform robustly on asphalt. The lower *precisions* of C-SVC and ACTC classification are mainly due to inaccuracies in the hand-labeling of obstacles. This is as expected, because we hand-labeled areas that should not be traversed, not individual cells. Therefore, there are some traversable cells within these areas and the different methods classify them differently. However, the connectivity maps mitigate this problem and the *precisions* are around the same order of magnitude. The NDT-TM map and the corresponding connectivity map from Trial 1 classified with the CTC algorithm are shown in Figure 14.

In Trial 2, the *precision* is again higher with CTC than with C-SVC or ACTC (see Table IV). However, the

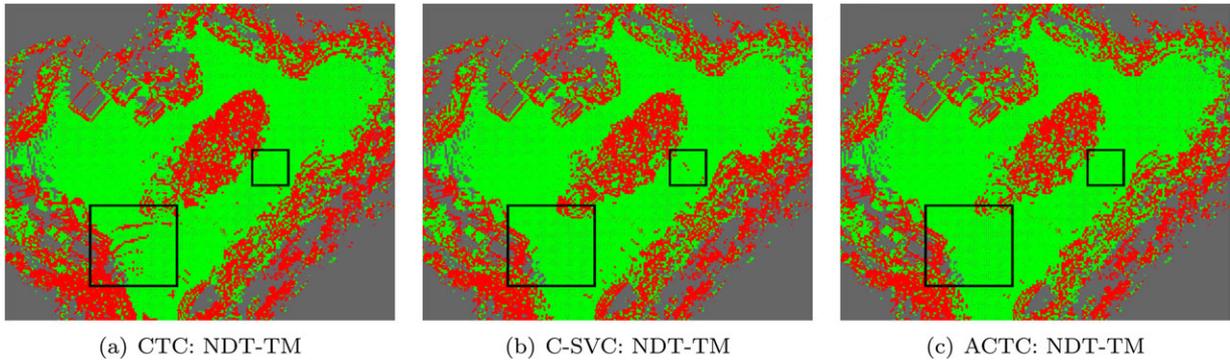


Figure 15. The NDT-TM maps from Trial 2 (red means nontraversable). The black rectangles in the figures highlight the areas where the classification methods perform differently due to minor irregularities on the terrain. The size of the test area is $100 \times 100 m^2$.

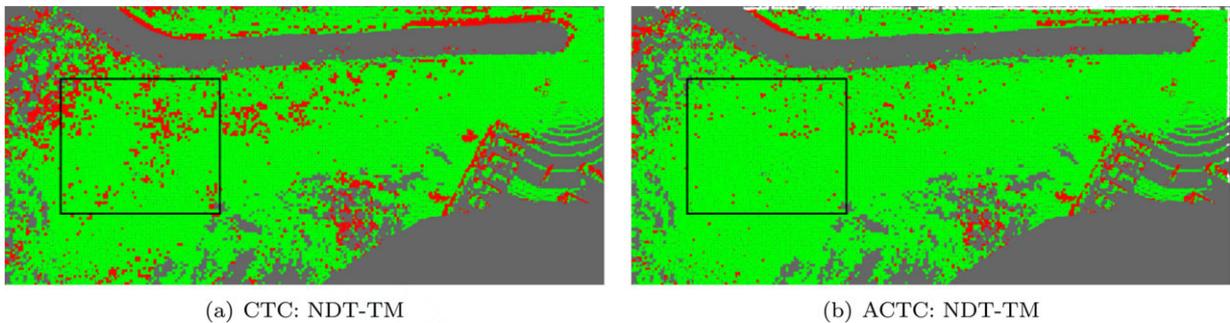


Figure 16. The NDT-TM maps from Trial 3 classified with CTC and ACTC method (red means nontraversable). The black rectangles highlight an area of tall vegetation with significant differences in the classification results. The size of the test area is $90 \times 60 m^2$.

differences remain minor and are mainly due to the approximate labeling of the obstacles. Nonetheless, the advantage of the SVM-based methods is already clear when looking at the *recall* values (i.e., correctly classified traversable cells), which are significantly better than with CTC, even though the ground plane is relatively smooth. Note that there are no hand-labeling-related inaccuracies present in the *recall* values because the labeling was done based on the robot footprint.

The advantage of SVM-based classification methods is further illustrated in Figure 15 in which differences in the NDT-TM maps are highlighted in the figures. The CTC method misclassifies cells in the area of the larger black rectangle due to small irregularities on the ground, whereas the SVM-based methods are able to classify that area correctly. However, the smaller black rectangle highlights the advantage of the ACTC method because the C-SVC method misclassifies few cells on that area, whereas the CTC and ACTC classify them correctly. Overall, the ACTC performs best in Trial 2, which can be seen as the highest *recall* value in Table IV.

Trial 3 was performed in a sparsely vegetated environment where the SVM-based methods clearly outperform the

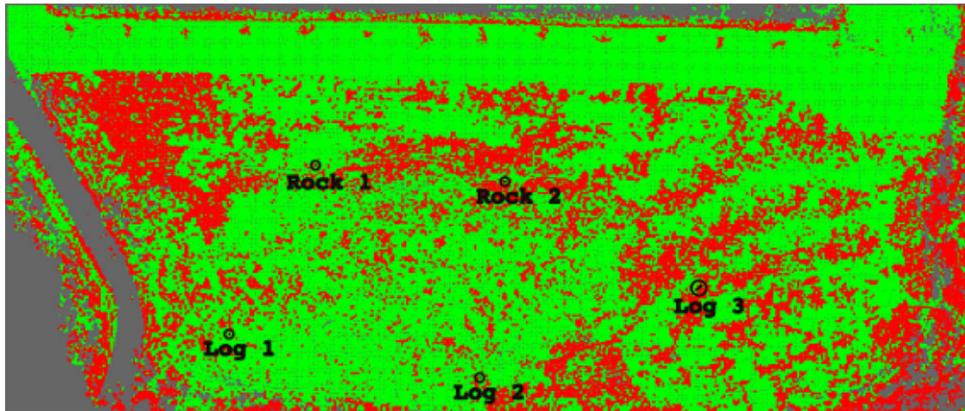
CTC method as expected (see Table IV). The *precisions* are again similar with all the methods, but the *recall* values are significantly higher than the SVM-based methods, which yields also superior *f-scores*. In this environment, there are no significant differences between the C-SVC and ACTC methods. Figure 16 illustrates the differences between classification results of CTC and ACTC method in Trial 3. The black rectangle highlights an area of tall vegetation shown on the right-hand side of Figure 12(c) with significant differences between the classification results.

Trial 4 was performed in a densely vegetated environment. Table IV illustrates that the *precision* values are similar for all the methods, but the *recall* values of the SVM-based methods show substantial improvement in classification accuracy. Hence, also the *f-scores* are significantly higher for the SVM-based methods. The ACTC method performs best also in this experiment.

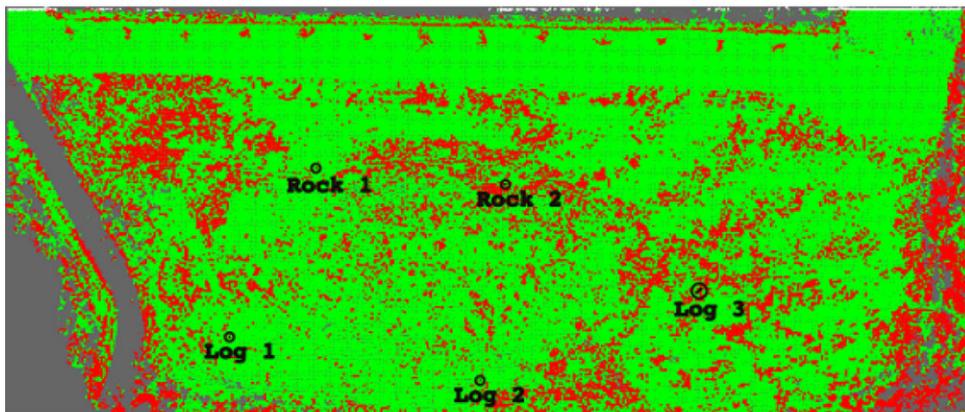
To further evaluate the capability of detecting true obstacles within the vegetated field, we placed five different obstacles in known locations on the field in Trial 4, that is, two large rocks and three logs of wood. Images of these obstacles can be seen in Figure 17. The locations of these obstacles are marked in the NDT-TM maps in Figure 18. All the



Figure 17. Photographs of the obstacles placed on the field in Trial 4.



(a) CTC: NDT-TM



(b) ACTC: NDT-TM

Figure 18. NDT-TM maps from Trial 4 classified with CTC and ACTC. The locations of the obstacles are marked into the maps. The size of the test area is around $165 \times 70 m^2$.

obstacles were correctly classified as nontraversable with all the classification methods. When selecting the model parameters, detecting obstacles needs to be the priority, because the obstacles might damage the vehicle. Unfortunately, because we rely on permeability in classification, only penetrable vegetation can be correctly classified as traversable. Therefore, in Trial 4, several parts of the environment are classified as nontraversable due to impenetrable vegetation even though in reality traversing through

those areas would be possible without damaging the vehicle. Nonetheless, it was shown that the proposed methods perform robustly in different types of environments and improve the classification accuracy significantly in vegetated environments.

The last field trial was performed in a forest environment with sparse grass as well as multiple coniferous and deciduous trees spread across the environment. Because Trial 5 was performed with a different measurement

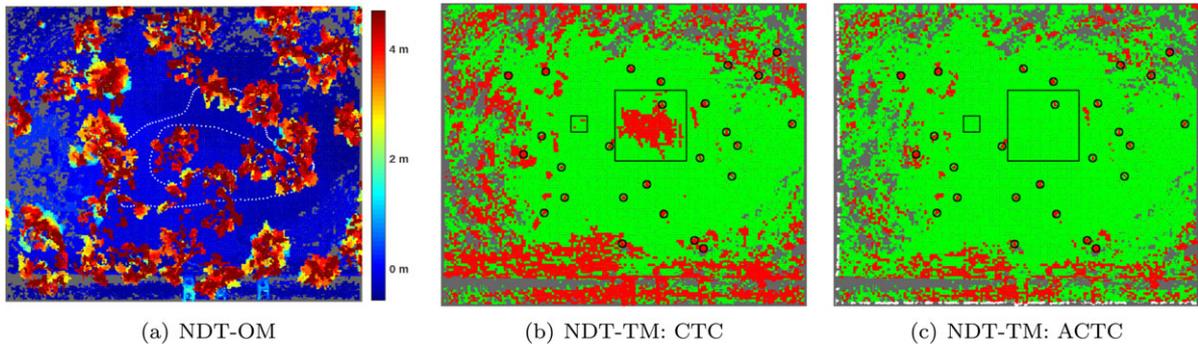


Figure 19. Top-down view of the resulting maps from the forest experiment. The size of the test area is $60 \times 50 m^2$. (a) NDT-OM map of the forest environment color coded based on height. The white dotted line illustrates the trajectory of the measurement platform. (b) NDT-TM of the forest environment classified with the CTC method. (c) NDT-TM map classified with the ACTC method. The black circles mark locations of tree trunks along the measurement platform trajectory. The black rectangles highlight areas where sparse vegetation is misclassified with the CTC method, whereas the ACTC methods is able to correctly classify the area as traversable. The NDT-TM maps are truncated to present only cells lower than the sensor height for clarity.

platform and no data from forest environments were used to train our model, this experiment illustrates that the trained model generalizes well to different environments. Figure 19 shows top-down views of maps generated from the environment. From left to right, the maps are NDT-OM colorized by height, NDT-TM classified with CTC, and NDT-TM classified with ACTC. To make the visualization of the NDT-TMs more clear, no cells above the sensor level are visualized. The white dotted line in the NDT-OM illustrates the vehicle trajectory during the data gathering, and the black circles in the NDT-TM maps mark locations of tree trunks along the trajectory. The black rectangles highlight areas with tall grass wherein the classification methods perform differently.

Both the CTC and ACTC methods correctly classified all tree trunks as nontraversable. However, some areas with tall grass are misclassified with the CTC, whereas the ACTC was able to correctly classify the whole area. That is, using the ACTC method for classification enables safe path planning in the forest environment with tall grass. Most of the nontraversable areas in the maps near tree trunks originate from low-hanging tree branches that should not be considered traversable because the branches might hit and damage the sensor.

6. CONCLUSIONS AND DISCUSSION

This article presented the NDT-TM representation for traversability evaluation. Two novel 3D terrain classification methods based on the NDT-TM representation in outdoor environments were proposed and demonstrated to outperform previous work on traversability analysis using NDT representation. The NDT-TM extends NDT-OM by adding an intensity distribution and expected permeability as additional features for each cell in the representation.

Classification is performed that exploits an SVM framework, using the roughness and slope of each cell distribution in addition to the intensity mean, intensity variance, and permeability as features. The first algorithm uses C-SVC, wherein positive teaching samples are gathered from the robot footprints and the negative samples automatically from semicontrolled environments. The second approach is a hybrid method combining the advantages of both CTC and C-SVC by applying the more complex C-SVC classification only to cells that are at first classified as nontraversable with CTC.

To avoid the approximative automatic collection of negative samples, we performed preliminary experiments also with one-class SVM algorithm (Schölkopf et al., 2001) in which only positive samples are used for training the model. However, tuning one-class SVM is difficult if no labeled data are available from both classes. Therefore, we trained a one-class SVM using an RBF kernel function, selecting the parameters similarly as with the C-SVC approach, that is, by running cross validation on the same validation data. Nonetheless, even the models tuned using validation data with labeled samples from both classes performed systematically significantly worse than the two-class models. Further investigation into one-class classifiers is left as one direction of future work.

We presented a detailed study of the features of LI-DAR data and NDT cells in the presence of vegetation. One key finding is that even though NDT-OM allows low resolution without compromising model accuracy, it is beneficial to keep the voxel size relatively small to capture the real dimensions and properties of the environment in order to classify vegetated environments correctly. However, significantly lower resolution than in previous work can still be used. Nonetheless, adapting a multiresolution grid as the frame of the NDT-TM approach as in Stoyanov et al.

(2010) would further decrease the computational complexity. Another important conclusion is that by exploiting the permeability and intensity distribution, it is possible to distinguish vegetation from solid obstacles. However, it was found that with LIDAR-only approach at least parts of the obstacle need to be clearly visible in order to do the distinction. Therefore, the proposed classification approaches were trained such that only sparse vegetation is classified as traversable.

Furthermore, we demonstrated that adding the intensity distribution and the permeability as features increases the accuracy of classification in complex vegetated environments. Although the densely vegetated environment in Trial 4 proved to be challenging for the proposed classification methods, the results demonstrate that the methods significantly improve the classification accuracy even though safe navigation on this environment based only on LIDAR data is not feasible. In our experiments, there were no significant differences in classification accuracy between the two proposed classification methods. However, the ACTC method decreases the classification times significantly and adds robustness to the classification by performing CTC-based classification in structured environments.

The computational efficiency of our approach could be further improved by utilizing a multiresolution framework as a basis of the NDT-TM mapping, which would allow reasonably small resolution for classifying sparse vegetation in places of interest and large grid size in areas with little or no data. However, this is left as future work. Moreover, the NDT-TM provides a solid foundation for traversability classification that could be further improved by adding other sensor modalities. For example, UWB radar measurements — that have been shown to penetrate vegetation — could be incorporated into the representation, which could enable safe navigation performance even in densely vegetated environments.

ACKNOWLEDGMENTS

This research was supported in part by the Finnish Society of Automation as well as the Finnish Funding Agency for Technology and Innovation (TEKES), Forum for Intelligent Machines (FIMA), research institutes, and companies under the Energy and Life Cycle Cost Efficient Machines (EFFIMA) research program, managed by the Finnish Metals and Engineering Competence Cluster (FIMECC). Their support is gratefully acknowledged. Furthermore, we thank Dr. V. Hernandez Bennetts and M. A. Arain for providing the Örebro university forest data set.

REFERENCES

- Ahtiainen, J., Peynot, T., Saarinen, J., & Scheduling, S. (2013). Augmenting traversability maps with ultra-wideband radar to enhance obstacle detection in vegetated environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5148–5155, Tokyo, Japan.
- Ahtiainen, J., Peynot, T., Saarinen, J., Scheduling, S., & Visala, A. (2015). Learned ultra-wideband RADAR sensor model for augmented LIDAR-based traversability mapping in vegetated environments. In The International Conference on Information Fusion, pages 953–960, Washington, D.C., USA.
- Angelova, A., Matthies, L., Helmick, D., & Perona, P. (2007). Fast terrain classification using variable-length representation for autonomous navigation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–8, Minneapolis, MN, USA.
- Bajracharya, M., Howard, A., Matthies, L., Tang, B., & Turmon, M. (2009). Autonomous off-road navigation with end-to-end learning for the LAGR program. *Journal of Field Robotics*, 26(1), 3–25.
- Bares, J., Hebert, M., Kanade, T., Krotkov, E., Mitchell, T., Simmons, R., & Whittaker, W. (1989). Ambler: An autonomous rover for planetary exploration. *Computer*, 22(6), 18–26.
- Baribeau, R., Rioux, M., & Godin, G. (1992). Color reflectance modeling using a polychromatic laser range sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 263–269.
- Bennetts, V., Schaffernicht, E., Stoyanov, T., Lilienthal, A., & Trincavelli, M. (2014). Robot assisted gas tomography—localizing methane leaks in outdoor environments. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 6362–6367, Hong Kong, China.
- Biber, P., & Strasser, W. (2003). The normal distributions transform: A new approach to laser scan matching. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), volume 3, pages 2743–2748, Las Vegas, NV, USA.
- Bradley, D., Thayer, S., Stentz, A., & Rander, P. (2004). Vegetation detection for mobile robot navigation. Technical Report CMU-RI-TR-04-12, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.
- Castano, A., & Matthies, L. (2003). Foliage discrimination using a rotating lidar. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), volume 1, pages 1–6, Taipei, Taiwan.
- Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Chang, T., Legowik, S., & Abrams, M. (1999). Concealment and obstacle detection for autonomous driving. In Proceedings of the IASTED Conference on Robotics and Applications, pages 28–30, Santa Barbara, CA, USA.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.
- Garrido, S., Malfaz, M., & Blanco, D. (2013). Application of the fast marching method for outdoor motion planning in robotics. *Robotics and Autonomous System*, 61(2), 106–114.

- Gestel, T. V., Suykens, J., Baesens, B., Viaene, S., Vanthienen, J., Dedene, G., Moor, B. D., & Vandewalle, J. (2004). Benchmarking least squares support vector machine classifiers. *Machine Learning*, 54(1), 5–32.
- Hadsell, R., Bagnell, J., Huber, D., & Hebert, M. (2010). Spacecarving kernels for accurate rough terrain estimation. *International Journal of Robotics Research*, 29(8), 981–996.
- Hornung, A., Wurm, K., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3), 189–206. Software available at <http://octomap.github.com>.
- Howard, A., Turmon, M., Matthies, L., Tang, B., Angelova, A., & Mjolsness, E. (2006). Towards learned traversability for robot navigation: From underfoot to the far field. *Journal of Field Robotics*, 23(11-12), 1005–1017.
- Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2010). A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University.
- Kim, D., Oh, S., & Rehg, J. (2007). Traversability classification for UGV navigation: A comparison of patch and superpixel representations. In *Proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems (IROS)*, pages 3166–3173, San Diego, CA, USA.
- Lacaze, A., Murphy, K., & DelGiorno, M. (2002). Autonomous mobility for the Demo III experimental unmanned vehicles. In *Proceedings of the International Conference on Unmanned Systems (AUVSI)*, Lake Buena Vista, FL, USA.
- Lalonde, J., Vandapel, N., Huber, D. F., & Hebert, M. (2006). Natural terrain classification using three-dimensional lidar data for ground robot mobility. *Journal of Field Robotics*, 23(10), 839–861.
- Lang, T., Plagemann, C., & Burgard, W. (2007). Adaptive non-stationary kernel regression for terrain modelling. In *Proceedings of the Robotics: Science and Systems Conference (RSS)*, Seattle, WA, USA.
- Macedo, J., Manduchi, R., & Matthies, L. (2000). Laser-based discrimination of grass from obstacles for autonomous navigation. In *Proceedings of the International Symposium on Experimental Robotics*, Honolulu, HI, USA.
- Magnusson, M. (2009). The three-dimensional normal-distributions transform: An efficient representation for registration, surface analysis, and loop detection. PhD thesis, Laboratory of Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden.
- Martin, S., Murphy, L., & Corke, P. (2013). Building large scale traversability maps using vehicle experience. In J. Desai, G. Dudek, O. Khatib, & V. Kumar, editors, *Experimental Robotics*, volume 88 of Springer Tracts in Advanced Robotics, pages 891–905. Heidelberg: Springer International Publishing.
- Molino, V., Madhavan, R., Messina, E., Downs, A., Balakirsky, S., & Jacoff, A. (2007). Traversability metrics for rough terrain applied to repeatable test methods. In *Proceedings IEEE/RSJ International Conference on Robotics and Automation (IROS)*, pages 1787–1794.
- Moravec, H., & Elfes, A. (1985). High resolution maps from wide angle sonar. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 116–121, Nagoya, Japan.
- Myneni, R., & Hall, F. (1995). The interpretation of spectral vegetation indexes. *IEEE Transactions on Geoscience and Remote Sensing*, 33(2), 481–486.
- Nguyen, D.-V., Kuhnert, L., & Kuhnert, K. (2012a). Spreading algorithm for efficient vegetation detection in cluttered outdoor environments. *Robotics and Autonomous Systems*, 60(12), 1498–1507.
- Nguyen, D.-V., Kuhnert, L., Thamke, S., Schlemper, J., & Kuhnert, K.-D. (2012b). A novel approach for a double-check of passable vegetation detection in autonomous ground vehicles. In *Proceedings of the International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 230–236, Anchorage, AK, USA.
- Papadakis, P. (2013). Terrain traversability analysis methods for unmanned ground vehicles: A survey. *Engineering Applications of Artificial Intelligence*, 26(4), 1373–1385.
- Pomerleau, F., Colas, F., Siegwart, R., & Magnenat, S. (2013). Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34(3), 133–148.
- Rusu, R. B., Sundaesan, A., Morisset, B., Hauser, K., Agrawal, M., Latombe, J.-C., & Beetz, M. (2009). Leaving flatland: Efficient real-time three-dimensional perception and motion planning. *Journal of Field Robotics*, 26(10), 841–862.
- Saarinen, J., Andreasson, H., Stoyanov, T., & Lilienthal, A. (2013a). Normal distributions transform Monte-Carlo localization (NDT-MCL). In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 382–389, Tokyo, Japan.
- Saarinen, J., Andreasson, H., Stoyanov, T., & Lilienthal, A. (2013b). 3D normal distributions transform occupancy maps: An efficient representation for mapping in dynamic environments. *International Journal of Robotic Research*, 32(14), 1627–1644.
- Santamaria-Navarro, A., Teniente, E., Morta, M., & Andrade-Cetto, J. (2015). Terrain classification in complex three-dimensional outdoor environments. *Journal of Field Robotics*, 32(1), 42–60.
- Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., & Williamson, R. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7), 1443–1471.
- Siciliano, B., & Khatib, O., editors (2008). *Springer Handbook of Robotics*. Berlin: Springer.
- Stoyanov, T., Magnusson, M., Andreasson, H., & Lilienthal, A. (2010). Path planning in 3D environments using normal distributions transform. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3263–3268, Taipei, Taiwan.
- Stoyanov, T., Magnusson, M., & Lilienthal, A. (2013). Comparative evaluation of the consistency of three-dimensional spatial representation used in autonomous robot navigation. *Journal of Field Robotics*, 30(2), 216–236.

- Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niek-erk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., & Mahoney, P. (2006). Stanley: The robot that won the DARPA grand challenge. *Journal of Field Robotics*, 23(9), 661–692.
- Triebel, R., Pfaff, P., & Burgard, W. (2006). Multi-level surface maps for outdoor terrain mapping and loop closing. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2276–2282, Beijing, China.
- Vandapel, N., Huber, D., Kapuria, A., & Hebert, M. (2004). Natural terrain classification using 3-D ladar data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 5, pages 5117–5122, New Orleans, LA, USA.
- Vasudevan, S., Ramos, F., Nettleton, E., & Durrant-Whyte, H. (2009). Gaussian process modeling of large-scale terrain. *Journal of Field Robotics*, 26(10), 812–840.
- Wellington, C. (2005). Learning a terrain model for autonomous navigation in rough terrain. PhD thesis, Robotics Institute, Carnegie Mellon University, PA, USA.
- Wellington, C., Courville, A., & Stentz, A. (2006). A generative model of terrain for autonomous navigation in vegetation. *The International Journal of Robotics Research*, 25(12), 1287–1304.
- Wellington, C., & Stentz, A. (2004). Online adaptive rough-terrain navigation vegetation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 96–101, New Orleans, LA, USA.
- Wiemann, T., Nuchter, A., Lingemann, K., Stiene, S., & Hertzberg, J. (2010). Automatic construction of polygonal maps from point cloud data. In *Proceedings of the IEEE International Workshop on Safety Security and Rescue Robotics (SSRR)*, pages 1–6, Bremen, Germany.
- Wurm, K., Kretschmar, H., Kümmerle, R., Stachniss, C., & Burgard, W. (2014). Identifying vegetation from laser data in structured outdoor environments. *Robotics and Autonomous Systems*, 62(5), 675–684.
- Ye, C., & Borenstein, J. (2004). T-transformation: A new traversability analysis method for terrain navigation. In G. Gerhart, D. Gage, & C. Shoemaker, editors, *Proceedings of the SPIE: Unmanned Ground Vehicle Technology VI*, volume 5422, pages 473–483, Orlando, FL, USA.
- Zhou, S., Xi, J., McDaniel, M., Nishihata, T., Salesses, P., & Iagnemma, K. (2012). Self-supervised learning to visually detect terrain surfaces for autonomous robots operating in forested terrain. *Journal of Field Robotics*, 29(2), 277–297.