Contactless obstacle penetrability classification for mobile outdoor robotics

Lars Kuhnert
University of Siegen
Institute for Realtime Learning Systems
Siegen, Germany
Email: lars.kuhnert@fb12.uni-siegen.de

Duong-Van Nguyen
University of Siegen
Institute for Realtime Learning Systems
Siegen, Germany
Email: nguyenyan@fb12.uni-siegen.de

Klaus-Dieter Kuhnert
University of Siegen
Institute for Realtime Learning Systems
Siegen, Germany
Email: kuhnert@fb12.uni-siegen.de

***Abstract*—Recent research in the field of autonomous outdoor navigation has made great progress allowing mobile robots to robustly evade obstacles of different appearances. Still, the discrimination of impenetrable and penetrable obstacles remains a substantial challenge in this context. Especially a mobile robot trying to navigate efficiently in a vegetated forest environment is frequently faced with this specific classification task as vegetation is often intruding into the driving corridor of the robot. This paper addresses this classification problem by systematically designing and analysing numerous different per-sample features (and their combinations) which are derived from a low-resolution textured triangle mesh generated using an actuated laser range sensor and a color camera. Subsequently a support vector machine-based classifier is trained to distinguish between the two obstacle classes. Experimental results are presented to demonstrate the accuracy of the proposed classifier on real sensor data gathered in a realistic outdoor environment being over 90%.**

I. INTRODUCTION

Mobile robot navigation in vegetated outdoor terrain is a complex task even if sensor data from various complementary sources is available (like laser range sensors and (stereo) color cameras). Therefore, many previously published algorithms simplify this task due to the inability of discriminating impenetrable from penetrable ”obstacles”. Obstacles are simply identified by analysing the height of a measurement point over a local ground surface patch of a certain scale [6]. As a consequence an algorithm based on this assumption fails to overcome ”obstacles” that may only be the size of a single thin branch of tree or even a single blade of grass. To improve this concept we present a classification algorithm that is able to distinguish between obstacles that can be safely overrun by the robot and obstacles that would cause severe damage if hit by the robot. To ease explanations we are going to call penetrable obstacles *soft obstacles* and impenetrable obstacles *hard obstacles* in the further course of this paper.

II. RELATED WORK

In the past much work has been invested in the detection and classification of obstacles for mobile outdoor robots operating in unstructured environments.

A very common approach is the description of the environment using one or more planes fitted to the environment's geometry to generate an estimated description of the ground surface. Every part of the environment, that is elevated higher than a certain threshold (and lower than the height of the robot) above the estimated ground plane, is classified as an obstacle. The RANSAC algorithm [4] is a very popular technique for finding planar structures in noisy point cloud data. [7] is just one of many examples successfully demonstrating results of RANSAC ground plane estimation in an outdoor environment. Further approaches to solving the ground plane estimation problem include algorithms based on Markov random fields [19] and Octree data representation [17]. In our work, too, a ground plane estimation is used as initial classification step to get rid of a large number of non-obstacle points before classifying the remaining points as soft or hard obstacles.

Unfortunately deciding on which part of the environment belongs to the ground surface and which part does not, is a very conservative way of determining traversable parts of the robot's environment. Although this may keep the robot from colliding with misclassified obstacles, it heavily limits the robot's choice of actions in vegetated outdoor terrain. Due to this fact several optimized methods for outdoor terrain classification have been developed in the recent years. Howard et al. propose a vision-based approach using terrain traversability characteristics (roughness, slope, discontinuity and hardness) to create a single Fuzzy Traversability Index describing the difficulty of traversal for a mobile robot [5]. In Vibration-based terrain classification is another popular method in the context of terrain traversability analysis. [16] is only one of several examples for this approach. Unfortunately this technique has the backdraw that the robot first has to drive on the terrain which is to be classified or at least has to predict the terrain traversability based on the terrain it is currently driving on which induces a substantial amount of insecurity as terrain characteristics can be subject to rapid local changes.

Among the various different obstacles classes that can be distinguished in an outdoor environment, vegetation (e.g. grass, bushes, tree trunks or branches) has been uncovered as a very important obstacle class by many researchers when it comes to improving autonomous outdoor navigation. Numerous publications can be found on the specific topic of vegetation detection and classification in this context. Macedo et al. describe a simple classifier which is inspired by a statistical analysis of range data from a laser range sensor.

The classifier is able to identify non-traversable obstacles in the presence of grass even if the obstacles are partially occluded by grass [11]. In [2] similar results are presented, but the discrimination between different obstacle classes is achieved by exploiting the expected localities and continuities of obstacles. [12] describes the extension of this work by the additional integration of color information. McDaniel et al. classify points of a laser-scanned scene using the following classes: *ground*, *bushes/shrubs*, *trunks* and *canopy*. Various geometric features describing the local environment of each measured point and a SVM-based classifier are utilized to produce these results [13]. Lalonde et al. present a classification method separating the classes *clutter*, *linear* and *surface* based on LADAR measurements. The features used in the proposed method are based on a statistical analysis of the local environment of each measurement and thus are very similar to the PCA-based features used in our work [10]. Wurm et al. describe a classification method based on infrared remission values of the environment to be able to avoid low grass-like structures which previously could not be detected by using the environment's geometry alone [20].

III. ENVIRONMENT MODEL

The environment model which was used throughout all experiments in this work is generated by fusion of several sensors mounted on our autonomous mobile robot outdoor robot (AMOR) [8].

A custom-made actuated laser range sensor [14] is used to generate range measurements of the robot's environment. In contrast to many other actuated planar laser range sensors not the scanner itself but a mirror mounted in front of the scanner is rotated. In this way it is possible to generate a scan of a scene consisting of several scanlines recorded at different scanning pitch angles at an update rate of up to 10 Hz. This is important as we want our robot to be able to travel at reasonable speeds (approx. 3 m/s) and want to be able to cope with moving obstacles. Unfortunately the used laser range sensor (SICK LMS 221) generates individual scanlines at a rate of 75 Hz which leaves us with only around 8 scanlines per complete scene scan, but our proposed classification method is able to cope with the low vertical resolution of the resulting scene description. Scanlines are recorded while the robot is moving, so data of wheel encoder sensors, digital compass and inertial measurement unit are used to generate a motion-corrected point cloud. Finally, texture information is added to the point cloud using a registered color camera. A detailed description of these steps can be found in our previous work [9]. The resulting environment model forms the basis of the subsequent steps.

Furthermore, to reduce the number of points to be classified, an initial classification of all points is performed by separating points that lie in an estimated ground plane (and therefore are not to be considered as obstacles) from points not lying in the ground plane. Similar to Yuan's approach to road detection in urban areas [21], we exploit the known scanline structure of the environment model to select promising ground plane inlier candidates before applying a RANSAC-based plane fitting method to estimate the pose of the ground plane. This technique boosts the stability of the ground plane estimation significantly compared to a standard RANSAC-based plane fitting solution.

IV. FEATURE SELECTION

Feature selection is an essential part of the design process of a classifier. Thus, we invested a considerable amount of time in the uncovering and systematic analysis of features that may make a contribution to solving this classification task. Many of the studied features have been used in other approaches, but have never been analysed on such a comprehensive scale which makes this work unique in the context of object penetrability classification. Additionally, novel features e.g. based on the known adjacency relationship of the geometrical environment model are added to the pool of evaluated features. In this section we will describe the different feature types that were used throughout this work.

Generally it is important to state that all discussed features are "per-range-measurement" features. So every range measurement generated by the laser range sensor gets an attached feature vector f consisting of an instance of each presented feature f_n where n is a unique index of the feature. Also, in contrast to many other approaches, we gather a large number of potential features without ruling out features based on manual data analysis or other heuristically determined factors at an early stage of the classifier design process (see section V-C for results of our evaluation of the presented features).

At this point we would like to state that we are aware of the fact that using higher-level features (e.g. detection of individual shrubs, trunks or rocks) might boost the performance of the proposed classifier. Nevertheless the presented results V-D in this work show that our approach is already able to solve the classification problem posed in a forest track scenario and is therefore sufficient at this point.

A. *Visual features*

1) Color: Utilizing a homography-based registration technique a 4x4 matrix M can be determined defining the relative pose between a laser range sensor and a camera [9]. This allows us to project 3-D points originating from range measurements onto the image plane of a camera:

$$\begin{pmatrix} x_c & y_c & z_c & 1 \end{pmatrix}^T = M \begin{pmatrix} x_w & y_w & z_w & 1 \end{pmatrix}^T \tag{1}$$

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} f_x \frac{x_c}{z_c} + c_x \\ f_y \frac{y_c}{z_c} + c_y \end{pmatrix} \tag{2}$$

$\begin{pmatrix} x_w & y_w & z_w \end{pmatrix}^T$ and $\begin{pmatrix} x_c & y_c & z_c \end{pmatrix}^T$ denote 3-D points in the laser range sensor's and camera's coordinate system respectively. $\begin{pmatrix} f_x & f_y \end{pmatrix}^T$ and $\begin{pmatrix} c_x & c_y \end{pmatrix}^T$ define focal length and principal point of the camera. The image information at the projected position of the 3-D point defines the color information features ***red*** *(f_1)*, ***green*** *(f_2)* and ***blue*** *(f_3)* in RGB-color space. To reduce the influence of illumination changes the RGB-color information was additionally converted into the HSI-color space generating three additional features: ***hue** (f_4)*, ***saturation** (f_5)* and ***intensity** (f_6)*.

2) Remission value: Several laser range sensors (including the model we used throughout this work) return a surface reflectivity value of an object which correspond to the amount of light reflected by the measured object's surface. Baribeau et al. already pointed out that remission values depend on object material, object distance to the sensor and angle of incidence [1]. As proposed in [20] we thus added distance and incidence angle to the list of features (see Sections IV-C3 and IV-D) and use the feature ***remission value** (f_7)* as an indirect descriptor of the object's material.

B. Bounding box features

We define a local neighbourhood S_{bb} as a set of points around a central point $P_0 = \begin{pmatrix} x_0 & y_0 & z_0 \end{pmatrix}^T$ by

$$S_{\text{bb}} = \{P_i = \begin{pmatrix} x_i & y_i & z_i \end{pmatrix}^T \in \mathbb{R}^3 \mid q(P_0, P_i)\} \tag{3}$$

$$q(P_a, P_b) = |x_a - x_b| < \frac{l}{2} \wedge |y_a - y_b| < \frac{l}{2} \wedge |z_a - z_b| < \frac{l}{2} \tag{4}$$

where l denotes the side length of the bounding box defining the local neighbourhood S_{bb}. Based on this local neighbourhood the following features can be defined. Please note that each feature is additionally normalized using the volume $V = l^3$ of the bounding box.

1) Density: The ***density** (f_8)* of all points in S_{bb} is defined by

$$|S_{\text{bb}}|\,. \tag{5}$$

2) Compactness: The ***compactness** (f_9)* of all points in S_{bb} is defined by

$$\sum_{i=1}^{|S_{\text{bb}}|} (P_i - \overline{P}). \tag{6}$$

3) PCA-based shape descriptors: The eigenvalues $\lambda_{\text{max}} \geq \lambda_{\text{med}} \geq \lambda_{\text{min}}$ determined by a spectral decomposition of the covariance matrix

$$C = \sum_{i=1}^{|S_{\text{bb}}|} (P_i - \overline{P})(P_i - \overline{P})^T \tag{7}$$

of all points in S_{bb} can be used as shape descriptors of the local neighbourhood [18]. Therefore, the features ***maximal eigenvalue** (f_{10})*, ***medium eigenvalue** (f_{11})* and ***minimal eigenvalue** (f_{12})* were added to the list of features.

C. Scan matrix adjacency features

In this section we introduce novel features that make use of the matrix structure of environment models that are recorded using scanline based range sensors. The scan matrix A (not to be confused with a depth image) of a scene is composed of all recorded scanlines as rows of the scan matrix

$$A^{m \times n} = (a_{i,j}) \tag{8}$$

where m is the number of scanlines of the complete scan and n is the number elements each scanline consists of. The elements $a_{i,j}$ can be defined in various ways. We define the scan matrix A^{obst} by defining the matrix elements by

$$a_{i,j}^{obst} = \begin{cases} 1, & \text{if } e(i,j) \text{ is valid ground plane outlier} \\ 0, & \text{else} \end{cases} \tag{9}$$

The function $e(i,j)$ returns the j-th scanline element of the i-th scanline of the whole scan. Subsequently connected object components were grouped together to segments. This compact data description allows us to conveniently exploit the adjacency relationship between measurements and define the following features describe in the course of this subsection.

1) Porosity: Porosity is used to describe how porous the surface of an object in a local neighbourhood of the scan matrix is. Simply put this is achieved by detecting holes in the geometric structure of the environment. Holes in this context are defined as measurement points that are ground plane inliers which are surrounded by obstacle points that define valid hole boundaries in a certain search window. As the distances of adjacent scan matrix elements grow with the increasing distance to the range sensor a variable search window size has to be implemented. The search window widths (in number matrix cell elements) for each scanline with scanline index i can be estimated by

$$w_i = \lfloor \frac{2 \arcsin \frac{d}{2\overline{l_i}}}{180 \pi r} + 2 \rfloor \tag{10}$$

where d is the maximal diameter of a hole, $\overline{l_i}$ is the average distance of all ground plane outliers to the range sensor and r is the angular resolution of the range sensor (in the respective direction). As the search is separately executed in horizontal as well as in vertical direction, it gives us the horizontal and vertical search window widths w_i^{hor} and w_i^{vert} by inserting the corresponding angular resolutions in horizontal and vertical direction. Whenever a ground plane inlier is surrounded by valid ground plane outliers within the limits of the adapted search window in horizontal or vertical direction respectively, the corresponding scan matrix element is marked as a hole producing the scan matrices A^{hor} and A^{vert} with matrix elements defined by

$$a_{i,j}^{hor} = \begin{cases} 1, & \text{if } e(i,j) \text{ is marked as horizontal hole} \\ 0, & \text{else} \end{cases} \tag{11}$$

$$a_{i,j}^{vert} = \begin{cases} 1, & \text{if } e(i,j) \text{ is marked as vertical hole} \\ 0, & \text{else} \end{cases} \tag{12}$$

A $3w_i^{vert} \times 3w_i^{hor}$-sized box filter is subsequently applied to both matrices to realize an integration of the local scan matrix neighbourhood resulting in the filtered matrices $A^{hor\prime}$ and $A^{vert\prime}$. A point-wise multiplication by the ground plane outlier scan matrix A^{obst} finally results in the porosity scan matrices

$$H^{hor} = A^{obst} \circ A^{hor\prime} \tag{13}$$

$$H^{vert} = A^{obst} \circ A^{vert\prime} \tag{14}$$

The elements of these matrices (which are mapped directly to the original measurements) define the features ***horizontal porosity** (f_{13})* and ***vertical porosity** (f_{14})* for each point.

2) Width/height of connected object structures: As thin geometric structures are often likely to be penetrable by a mobile robot the width and height of an object are interesting features. Beginning at each non-zero element of A^{obst} a search for the next zero elements in same row (column respectively) is performed. This search yields the boundary points p_{left}, p_{right} (p_{top}, p_{bottom}) of the connected object component. Based on these boundary points the features ***connected width** (f_{15})* and ***connected height** (f_{16})* are defined by

$$\left|p_{left} - p_{right}\right| \text{ and } \left|p_{top} - p_{bottom}\right|. \tag{15}$$

Additionally, the connected width is projected on the horizontal axis and the connected height is projected on the vertical axis of the robot's coordinate system defining the features ***projected connected width** (f_{17})* and ***projected connected height** (f_{18})*. These projected features serve the purpose of describing an object from the viewpoint of the robot which proved to be a good complement to the non-projected versions of these features (see Section V-C for further details).

3) Incidence angle: The ***incidence angle** (f_{19})* describes at what angle a laser beam emitted from the laser range sensor hits the surface of the scanned object. It is defined by

$$\arccos\left(-\frac{np_0}{|p_0|}\right) \tag{16}$$

where n is the normalized surface normal which is derived from the adjacent measurement points around the central point p_0.

D. Point coordinate features

We did not solely want to model the local environment of a point but also the position of the point in relation to the position of the robot as this might also affect the final classification result. The easiest way to achieve this is to add the components of each point's Cartesian coordinates $\begin{pmatrix} x & y & z \end{pmatrix}^T$ as features ***point x** (f_{20})*, ***point y** (f_{21})* and ***point z** (f_{22})*. Additionally, the derived feature ***distance to sensor** (f_{23})* which is the length of the vector defined by the measured 3-D point. Please note that all points were transformed into the coordinate system defined by the estimated ground plane first to maintain the comparability of the point coordinate-based features even in presence of tilted terrain in the environment of the robot.

V. CLASSIFICATION

A. Support vector machines

A Support Vector Machine (SVM) is a classification method that constructs a N-dimensional hyperplane that separates a set of input feature vectors into two classes [15]. The hyperplane is constructed so that the distance between feature vectors and hyperplane is maximized. To be able to separate classes that cannot be separated by a linear decision function (which is usually the case) the kernel-trick was introduced. Basically this means that the input feature vectors are first transformed into a feature space which is of higher dimensionality by applying a kernel function. The transformation into the higher-dimensional feature space allows the definition of a linear decision function in the transformed feature space which corresponds to a non-linear decision function in the original feature space that is able to separate the two classes. Our implementation of the classifier (SVM-based classifier with a Gaussian radial basis function) is based on the widely used SVM-implementation LIBSVM [3].

B. Training data

Numerous scenes containing training objects were manually labelled to train the classifier. Training scenes were chosen to be as significant as possible to represent a large percentage of objects encountered in a realistic forest track scenario. Obstacle points were either labelled as soft or hard obstacle points in 362 training scenes producing a total of 6938 soft obstacle points and 8963 hard obstacle points. All labelled obstacle points were used to create a training feature matrix containing a feature vector for each labelled obstacle point. The training feature vector components were scaled to the interval $[0,1]$ using the minimal and maximal values of each component.

C. Feature evaluation

This section will present all results of the evaluation of the previously defined list of potential features with respect to the question of whether each individual feature is able to make a contribution to classifying soft and hard obstacles in a vegetated forest environment. We systematically analysed the features (and combinations of them) by generating SVM-models using labelled training data and testing the accuracy of the model with the complete training dataset described in previous section. To prevent overfitting of the model cross-validation using a test dataset (see V-D2) was performed during feature evaluation.

As a first step all features were analysed individually. Although for example the maximal eigenvalue feature (f_{10}) produces promising results, no single feature leads to a satisfying accuracy. Thus, a systematic analysis of all dual and triple feature combinations was performed as the second feature evaluation step (see Table I for triple feature results).

The analysis of the tested dual and triple feature combinations leads to several interesting conclusions:

- Different orientation components (e.g. horizontal and vertical) of features should be combined.
- Density and compactness have not proven to be as effective as the PCA-based shape-descriptor features.
- Combining features describing the local neighbourhood of a point with features describing the position of a point in reference to the position of the range sensor often yields improved results.

TABLE I
ACCURACIES OF TRIPLE FEATURE COMBINATIONS SORTED BY ACCURACY

$(f_{15}, f_{10}, f_{18}) = 95.41\%$	$(f_{21}, f_{10}, f_{20}) = 93.60\%$
$(f_{15}, f_{22}, f_{10}) = 94.97\%$	$(f_{2}, f_{10}, f_{22}) = 93.40\%$
$(f_{18}, f_{17}, f_{10}) = 94.67\%$	$(f_{10}, f_{23}, f_{20}) = 93.35\%$
$(f_{18}, f_{10}, f_{20}) = 94.50\%$	$(f_{12}, f_{15}, f_{10}) = 93.27\%$
$(f_{21}, f_{10}, f_{22}) = 94.47\%$	$(f_{10}, f_{15}, f_{8}) = 93.25\%$
$(f_{22}, f_{10}, f_{23}) = 94.43\%$	$(f_{15}, f_{18}, f_{21}) = 93.16\%$
$(f_{21}, f_{10}, f_{18}) = 94.31\%$	$(f_{10}, f_{7}, f_{22}) = 93.06\%$
$(f_{18}, f_{23}, f_{10}) = 94.26\%$	$(f_{10}, f_{17}, f_{20}) = 93.02\%$
$(f_{10}, f_{18}, f_{22}) = 94.09\%$	$(f_{10}, f_{17}, f_{21}) = 93.01\%$
$(f_{22}, f_{10}, f_{20}) = 94.01\%$	$(f_{18}, f_{15}, f_{8}) = 93.01\%$
$(f_{15}, f_{10}, f_{21}) = 93.83\%$	$(f_{10}, f_{17}, f_{23}) = 92.99\%$
$(f_{20}, f_{15}, f_{10}) = 93.81\%$	$(f_{18}, f_{15}, f_{20}) = 92.98\%$
$(f_{8}, f_{10}, f_{22}) = 93.77\%$	$(f_{20}, f_{18}, f_{17}) = 92.94\%$
$(f_{2}, f_{18}, f_{10}) = 93.67\%$	$(f_{18}, f_{7}, f_{10}) = 92.94\%$

- The different types of geometry-describing features often complement one another resulting in improved results.
- The PCA-based shape-descriptor features are very descriptive and usually improve the performance of others feature they are combined with.

As the number of feature combinations grows exponentially with the dimension of the feature vector a further "brute-force" analysis of all feature combinations was not found to be practicable at this point. By iteratively adding promising features and testing the resulting feature combinations based on the findings of the previous analysis of dual and triple feature combinations we could uncover a best-performing feature vector

$$f^{best} = (f_{10} f_{11} f_{12} f_{13} f_{14} f_{15} f_{17} f_{18} f_{22} f_{23})^T. \qquad (17)$$

In summary the feature vector is composed of the shape-describing eigenvalues (f_{10}, f_{11}, f_{12}), the surface porosity (f_{13}, f_{14}), the size of locally connected structures $(f_{15}, f_{16}, f_{17}, f_{18})$ and the position of the classified measurement in reference to the position of the laser range sensor (f_{22}, f_{23}). Further additions of features lead to over-fitting effects when testing the classifier on realistic datasets. For that reason no further features were added to the final feature vector f^{best}.

D. Experimental results

The accuracy of the classifier using the best feature vector f^{best} described in Section V-C was tested experimentally to prove the quality of the classification result.

1) Training data accuracy: In the first experiment the complete training feature matrix was used for training as well as for testing of the classifier. It yielded an accuracy of **95.54%**. In the second experiment several subsets of the training feature matrix were generated to train the classifier. The remaining feature vectors, not belonging to the corresponding randomized training subset, were used to test the classifier's accuracy. This experiment yielded an average accuracy of **94.66%**.anteil

2) Realistic test data accuracy: The third and most important experiment was conducted to evaluate the quality of the previously trained SVM-model originating from the training dataset in a realistic vegetated forest environment. During the test run the robot travelled a total distance of 450 m (see Figure 1 for an aerial map of the travelled route[1]). Several natural as well as man-made obstacles of both obstacles classes (soft and hard) were encountered during the test run. Afterwards the measurements were labelled analogously to the labelling process of the training dataset. The labelled objects were chosen to be approximately uniformly distributed along the path of the robot during the test run producing a total of 19842 labelled points. The complete test sequence consisted of 5002 scenes of which 519 were manually labelled. Approximately 80 different objects were labelled in total. Trees, trunks, rocks, garbage cans, and cardboard boxes are examples for labelled hard objects types. Grass, small bushes, and tree branches (with and without leave) are examples for labelled soft objects types. Our proposed classifier achieved a classification accuracy of **91.57%** when classifying all labelled points gathered during the test run which proves the effectiveness of the proposed method in a realistic vegetated forest environment. Figure 2 shows a small selection of test run scenes including classification result.

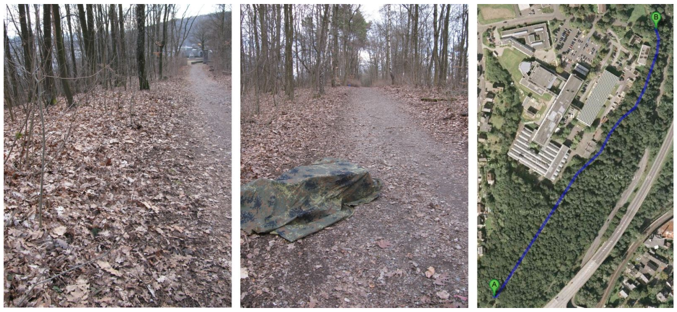

Fig. 1. Scenes of the vegetated forest environment used for test runs with some soft obstacles (left) and a hard obstacle on the ground (middle). Aerial map of route travelled during test runs (right).

VI. CONCLUSION

We presented a classification method that is able to distinguish between penetrable and impenetrable objects in a vegetated forest environment. This allows a differentiated obstacle classification for use that largely improves the navigational capabilities of a mobile robot acting in this environment. The classifier design process was supported by the systematic evaluation of a list of potential features which was never done before on such a comprehensive scale in the context of this classification problem. We proved the effectiveness of our approach by evaluating the proposed classifier on realistic data which was recorded on a mobile outdoor robot during test runs in the concrete scenario of forest track following. The classification accuracy was over 90% on all classified data. These results were achieved although the vertical resolution of the used laser range sensor is extremely low.

[1] satellite imagery property of Google Inc.

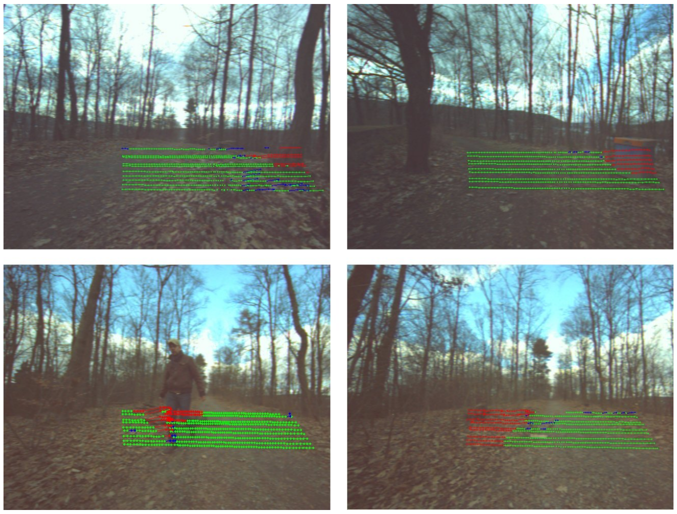

Fig. 2. Classification results on test run scenes. Ground plane inliers (green), soft obstacles (blue) and hard obstacles (red).

References

[1] R. Baribeau, M. Rioux, and G. Godin. Color reflectance modeling using a polychromatic laser range sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):263–269, 1992.

[2] A. Castano and L. Matthies. Foliage discrimination using a rotating ladar. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages 1–6. IEEE, 2003.

[3] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[4] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[5] A. Howard and H. Seraji. Vision-based terrain characterization and traversability assessment. *Journal of Robotic Systems*, 18(10):577–587, 2001.

[6] A. Kelly and A. Stentz. Rough terrain autonomous mobilitypart 2: An active vision, predictive control approach. *Autonomous Robots*, 5(2):163–198, 1998.

[7] K. Konolige, M. Agrawal, M.R. Blas, R.C. Bolles, B. Gerkey, J. Solà, and A. Sundaresan. Mapping, navigation, and learning for off-road traversal. *Journal of Field Robotics*, 26(1):88–113, 2009.

[8] K.-D. Kuhnert and W. Seemann. Design and realisation of the highly modular and robust autonomous mobile outdoor robot AMOR. In *Proceedings of the 13th IASTED International Conference on Robotics and Applications*, pages 464–469. ACTA Press, 2007.

[9] L. Kuhnert and K.-D. Kuhnert. Sensor-Fusion Based Real-Time 3D Outdoor Scene Reconstruction and Analysis on a Moving Mobile Outdoor Robot. *KI-Künstliche Intelligenz, Special Issue Offroad Robotics*, pages 1–7, 2011. ISSN 0933-1875.

[10] J.F. Lalonde, N. Vandapel, D.F. Huber, and M. Hebert. Natural terrain classification using three-dimensional ladar data for ground robot mobility. *Journal of Field Robotics*, 23(10):839–861, 2006.

[11] J. Macedo, R. Manduchi, and L. Matthies. Ladar-based discrimination of grass from obstacles for autonomous navigation. *Experimental Robotics VII*, pages 111–120, 2001.

[12] R. Manduchi, A. Castano, A. Talukder, and L. Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous Robots*, 18 (1):81–102, 2005.

[13] M.W. McDaniel, T. Nishihata, C.A. Brooks, and K. Iagnemma. Ground plane identification using lidar in forested environments. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3831–3836. IEEE, 2010.

[14] J. Schlemper, L. Kuhnert, M. Ax, and K.D. Kuhnert. Development of a high speed 3d laser measurement system for outdoor robotics. *Research and Education in Robotics-EUROBOT 2011*, pages 277–287, 2011.

[15] A.J. Smola and B. Schölkopf. *Learning with kernels*. Citeseer, 1998.

[16] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al. Stanley: The robot that won the darpa grand challenge. *The 2005 DARPA Grand Challenge*, pages 1–43, 2007.

[17] Y.H. Tseng and M. Wang. Automatic plane extraction from lidar data based on octree splitting and merging segmentation. In *Geoscience and Remote Sensing Symposium, 2005. IGARSS'05. Proceedings. 2005 IEEE International*, volume 5, pages 3281–3284. IEEE, 2005.

[18] N. Vandapel, D.F. Huber, A. Kapuria, and M. Hebert. Natural terrain classification using 3-d ladar data. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5, pages 5117–5122. IEEE, 2004.

[19] C. Wellington, A. Courville, and A. Stentz. Interacting markov random fields for simultaneous terrain modeling and obstacle detection. In *proceedings of Robotics: Science and Systems*, 2005.

[20] K.M. Wurm, R. Kummerle, C. Stachniss, and W. Burgard. Improving robot navigation in structured outdoor environments by identifying vegetation from laser data. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1217–1222. IEEE, 2009.

[21] Xia Yuan, Chun-Xia Zhao, and Hao-Feng Zhang. Road detection and corner extraction using high definition lidar. *Information Technology Journal*, 9:1022–1030, 2010.