A Stereo Vision Based Obstacle Detection System for Agricultural Applications

Patrick Fleischmann and Karsten Berns

Abstract In this paper, an obstacle detection system for field applications is presented which relies on the output of a stereo vision camera. In a first step, it splits the point cloud into cells which are analyzed in parallel. Here, features like density and distribution of the points and the normal of a fitted plane are taken into account. Finally, a neighborhood analysis clusters the obstacles and identifies additional ones based on the terrain slope. Furthermore, additional properties can be easily derived from the grid structure like a terrain traversability estimation or a dominant ground plane. The experimental validation has been done on a modified tractor on the field, with a test vehicle on the campus and within the forest.

1 Introduction

According to [10], the agricultural guidance research exploring the capabilities of image sensors started in the mid-1980s in North America. With the full availability of the NAVSTAR Global Positioning System (GPS) one decade later, researchers also started to explore this new technology including its application for the agricultural sector. This research on GPS-based guidance solutions led to successful commercial products which are nowadays offered by almost all big manufacturers of agricultural products or can be bought from component suppliers. The success of this technology can be probably explained by its universal applicability. In contrast to early camera-based and specialized solutions such as crop row guidance, the GPS guidance is not restricted to individual field work or a special machine. The already mentioned, commercial products for example, offer functions such as creating a linear trajectory defined by a waypoint and a direction. Furthermore, a complete track can be recorded and by specification of the implement's width, the system can cal-

Patrick Fleischmann · Karsten Berns
Robotics Research Lab, Department of Computer Science, University of Kaiserslautern, 67663 Kaiserslautern, Germany, e-mail: {fleischmann, berns}@cs.uni-kl.de

culate parallels to cover the whole field.
Here, a systemic disadvantage of GNSS-based (Global Navigation Satellite System) guidance systems is visible, which alone is not solvable with the GNSS technology: the calculated trajectories are not necessarily free of obstacles, which can lead to serious accidents. Accidents are caused by fatigue or inattention of the driver who has to monitor the Advanced Driver Assistance System (ADAS), where the two main reasons can be identified. On one hand, the use of an automated steering system can increase the monotony of work and thus cause fatigue—especially with large acreage. On the other hand, agricultural manufacturers are constantly increasing the working width of their machines and implements for economic reasons. For modern sprayers of 40 m width, it is difficult for the driver to estimate if the boom of the implement can be safely moved past an obstacle, especially at higher speeds.
While GNSS-based products are already very successful on the agricultural market, solutions using cameras or time-of-flight sensors are still a niche product for very specialized tasks and still in the focus of research. In the research domain stereo vision based obstacle detection for off-road and on-road is a large area, a recent survey [2] summarizes the contributions of the last decade. A very popular method is presented in [7] where obstacles are detected by analyzing the so called compatibility of the 3D points. To speed up the process, the evaluation is performed in the Disparity Space Image (DSI) where the truncated cones that have to be examined to get the compatibility turn into triangles. The well cited method has been extended and refined several times, e.g in [13] or in [4], where the DSI was splitted into different stripes with different resolutions to allow for parallelization and to reduce the number of comparisons.

Another group of approaches can be identified which rely on a 2D grid or use a Digital Elevation Map (DEM). One recent example [5] uses the grid representation to fit B-spline surfaces into the reduced data to estimate the traversability of the ground and presence of obstacles. For road application [9] demonstrates an approach where a DEM and a density measurement of the points within a cell are used to separate the road surface from obstacles.

The QUAD-AV project [12] addresses the obstacle detection problem for agricultural vehicles by the investigation of different sensor technologies like stereo vision, thermography, ladar and a microwave radar. Along with this project, several interesting publications were made, e.g. a self-learning framework which uses geometric 3D data and color information of a trinocular camera [11] to classify the ground. Both classifiers are updated during runtime to adapt the approach to changing environments. [8] describes the same framework for a multi-baseline camera but only relies on the geometric classifier. Additionally, a so called Unevenness Point Descriptor has been proposed [1] by the same research group which uses the normal vector distribution of small surfaces which are fitted using PCA.

In this paper, an obstacle detection system for the field is presented which can prevent collisions with obstacles while using a guidance system. For different reasons which are explained in detail in Sec. 2 a colored stereo camera was chosen to approach this task. As a first step after the data acquisition and pre-processing, the 3D points are sampled into a 2D grid. Initially, each grid cell is analyzed indepen-

dently of its neighbors, which enables a strong parallelization of the method. Afterwards, the relations to the neighborhood are examined, the obstacles are grouped and a terrain abstraction is generated.

2 System and Scenario Description

As described in the introduction the motivation to start the research on a field obstacle detection system was driven by reported accidents with automated guidance systems. Possible and probable obstacles in this scenario can be divided into 3 categories: natural, artificial or man-made and dynamic obstacles. The first class includes any kind of vegetation which is not traversable like bushes or trees and additionally impassable terrain like ground with high slope or negative obstacles like ditches and trenches. For the field scenario, the second category includes any kind of poles (transmission, power), buildings, bridges and fences. The most difficult class contains dynamic obstacles like persons, other agricultural equipment and animals.

The system described in this paper uses a Bumblebee2 stereo vision camera by Point Grey. It has a focal length of 2.5 mm which results in a wide horizontal field of view of 97 deg. Furthermore, the stereo setup has a fixed 12 cm baseline and includes two Sony ICX204 1/3" color CCD sensors providing a maximum usable resolution of 1024×768 pixels at 20 FPS. The decision to use a stereo camera instead of more precise sensor like a 3D laser scanner was influenced by the following properties. Firstly, the stereo vision system provides a very dense point cloud together with additional color information. Additional advantages like the low price of camera systems in mass production, the low energy consumption and the light weight makes the technology interesting for commercial applications. Furthermore, it could be pointed out during the tests, that the dust influence is lower than for a laser sensor which makes the device interesting for agricultural purposes.

The camera system has been mounted at a height of 2.8 m above the ground in front of the driver's cabin of a modified John Deere 6R series tractor. It was tilted downwards by about 10.5 deg to reduce the amount of sunlight falling into the camera. For better understanding, the mounting position together with the used Cartesian coordinate systems is shown in Fig. 1.

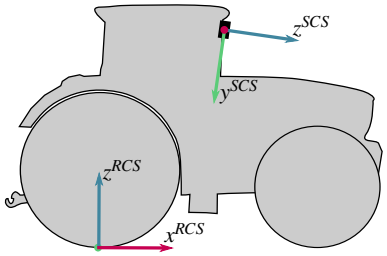


Fig. 1 Overview on the used coordinate systems: *SCS*: sensor, *RCS*: robot/tractor and the position of the camera which was mounted in front of the cabin below the roof.

3 Implementation and Algorithms

3.1 Data Acquisition and Pre-Processing

To grab the images from the camera, the libdc1394 library is employed. For the undistortion and rectification step, the calibration offered by Point Grey is used. Therefore, functions of the Triclops SDK[1] together with the calibration parameters stored on the sensor were used to generate lookup-tables for each camera in an offline process which map the pixels of the original image to the target. By applying these pre-computed tables, undistortion, rectification, cropping and scaling to a desired resolution can be done in one step. In addition, this enables the usage of the calibration together with other libraries, e.g. OpenCV[3] remap functionality which is applied in this case. The rectified images are then processed by a block matching algorithm which uses the sum of absolute differences as a metric to compute the disparity map. Neither the matching algorithm nor the metric are known to produce the best possible results. But its simplicity and its efficiency makes the algorithm suitable for embedded or GPU implementations. Knowing the disparity map, 3D points according to the camera reference frame $(x^{CCS}, y^{SCS}, z^{SCS})$ (see Fig. 1) can be calculated. In this step, the properties of the stereo vision system like the principal point, the focal length measured in pixels and the baseline is needed. All parameters are stored on the camera and can be scaled to the selected resolution. As a last step of the point cloud generation, all points are projected into the robot coordinate system $(x^{RCS}, y^{RCS}, z^{RCS})$. This system is originated on the ground below the kinematic center, with the x^{RCS}-axis pointing in the driving direction, z^{RCS} directed into the sky. The transformation requires the knowledge of the camera position in relation to this coordinate system. Please note that neither a statistical filtering nor a density reduction, e.g. a voxel grid filter, has been applied as it is often done in other approaches after this step.

3.2 Grid Generation and Pre-Processing

The point cloud $P = \{p_1, \ldots, p_n\}$ in the robot coordinate system is splitted into a 2D grid lying in the horizontal (x^{RCS}, y^{RCS}) plane. Each cell C_j has a parametrized dimension of $w \times h$. Due to the characteristics of the matching algorithm which produces a more dense cloud in the y-direction than in the x-direction, the width w and the height h could be set to different values. Additionally, the extend of the grid is limited in two directions $[0, x_{max}] \times [-\frac{y_{max}}{2}, \frac{y_{max}}{2}]$ as the output of a stereo vision system is only useful in a certain range, which depends on the baseline. The target cell index (c_x, c_y) of a point $p_i \in P$, $p_i = (p_{i_x}, p_{i_y}, p_{i_z})$ can be calculated as:

[1] http://www.ptgrey.com/triclops

$$c_x = \left\lfloor \frac{p_{i_x}}{w} \right\rfloor \tag{1}$$

$$c_y = \left\lfloor \frac{p_{i_y}}{h} + \frac{y_{max}}{2h} \right\rfloor \tag{2}$$

To avoid errors in addressing the cells, the maximum grid dimensions should be defined as $x_{max} = a \cdot w$ and $y_{max} = 2 \cdot b \cdot h$ where $a, b \in \mathbb{N}^*$. After this step, each cell C_j contains a subset P_j of the original point cloud P. Due to the defined boundaries of the grid, the following relation applies

$$P = \left(\bigcup_{j=1}^{\frac{x_{max}}{w} \cdot \frac{y_{max}}{h}} P_j \right) \cup Q \tag{3}$$

where Q contains all points which do not belong to the grid and are not further analyzed. In a first parallelized step, the points $p_i^{(j)} \in P_j$ of each cell are sorted ascending according their $p_{i_z}^{(j)}$-coordinate to prepare the further steps which results into $P_j = \{p_i^{(j)}\} \mid i = 1, \ldots, n^{(j)};\ p_{i_z}^{(j)} \leq p_{(i+1)_z}^{(j)}$.
This is followed by a sequential extraction of the z-coordinate of the lowest point $p_1^{(j)}$ of each non-empty cell. Combined with the cell's 2D center $(m_x^{(j)}, m_y^{(j)})$, this set of lowest points is used to define an initial dominant ground plane by applying a least-squares fitting algorithm (see Sec. 3.3 and Eq. 17 for the details). Afterwards, the shortest distance between the plane and the points $(m_x^{(j)}, m_y^{(j)}, p_{1_z}^{(j)})$ is tested. If the distance is larger than a threshold t_g or in the case of empty cells, the z-value is extracted from the fitted plane. Furthermore, all these height values—either originated from $p_1^{(j)}$ or determined using the plane—are saved in a matrix whose number of rows and columns is equal to the grid. This matrix is then smoothed using a Gaussian blurring (kernel size 5×5) to reduce the influences of cells which do not provide ground points as they are containing large obstacles. Thereafter, the determined height values are stored in the corresponding grid cells as a ground guess value g_j. These height values are used to be able to separate overhangs even if no ground points are available, for instance in "obstacle shadows".

3.3 Cell Evaluation

One advantage of the presented approach is—as already mentioned—the ability to parallelize the following steps, as each subset P_j is firstly evaluated individually without incorporating the neighborhood. As a first step, the number of points assigned to a cell C_j is calculated, as it has to be above a defined threshold $|P_j| \geq \rho$ to get meaningful results. If this density of points is too low, the cell is marked as `non-evaluable`. Based on the vehicle's properties shown in Table 1, the following derived quantities can be calculated:

$$d = \sqrt{(c_x w + \frac{w}{2})^2 + (c_y h - \frac{y_{max}}{2} + \frac{h}{2})^2} \tag{4}$$

$$\alpha = \max(v_\beta, d \cdot v_\alpha) \tag{5}$$

$$z_{max} = \max(v_g, d \cdot \sin(\alpha)) \tag{6}$$

Afterwards, a decision tree is used to test if a cell contains an obstacle. If one of the rules 7, 9 or 14-16 applies, the label `obstacle` is assigned to the cell and the evaluation is terminated. In the other case, the next test is executed. The first rule (Eq. 7) checks if the lowest measured sample p_{1_z} (from this point on, the superscript (j) is omitted to improve the readability) is above the position which could be reached with the given maximum slope v_α. Similarly, the highest measured point p_{n_z} has to be higher than the lowest reachable position.

$$p_{1_z} > z_{max} \vee p_{n_z} < -z_{max} \tag{7}$$

In forestry scenarios it often happens that overhanging parts are detected. In combination with missing ground points, Eq. 7 would lead to many false classifications. Thus, the distance between p_{1_z} and the ground guess g_j is evaluated and the cell label is fixed to `non-evaluable` if $p_{1_z} - g_j > v_h$.

For cells which include ground as well as overhanging objects, the space between these clusters has to be examined to see if the robot can safely pass this cell. To handle this situation, the range of the z-coordinates is tested:

$$(p_{n_z} - p_{1_z}) > v_h \tag{8}$$

In the case that the range is larger than v_h, a k-means clustering algorithm is applied to the point cloud subset P_j to see if the points can be separated into ground and overhang. The number of clusters k is set to 2. Additionally, it is ensured that the center of first cluster $P_g = \{p_i\} \mid i = 1,\ldots,k;\ p_{i_z} \leq p_{(i+1)_z}$ has a lower z-value than the center of the second cluster $P_o = \{p_i\} \mid i = (k+1),\ldots,n;\ p_{i_z} \leq p_{(i+1)_z}$ which is expected to contain the points of the overhang. If both clusters P_g and P_o fulfill a density criterion, the distance between the highest point of the ground cluster P_g and the lowest of the overhang cluster is evaluated:

$$p_{(k+1)_z} - p_{k_z} < v_h \tag{9}$$

Here, the ground guess value g_j calculated during the pre-processing is used instead of p_{k_z}, if the density of P_g is too low. Furthermore, the cell label is set to `non-evaluable` if both densities are below a threshold or g_j has been applied to Eq. 9 which was evaluated to false. If the space is insufficient (Eq. 9 is true), the cell is rated as an `obstacle` in all other cases.

At this point, the remaining point cloud is either still the original one (P_j) or the overhangs have been successfully separated and only the portion P_g has to be further analyzed. To improve readability, the next steps are just explained for P_j, nevertheless the same tests will be executed on P_g if the overhang separation was conducted.

To get rid of outliers and matching errors of the stereo correspondence module, a smoothing filter as well as statistical outlier filter is applied to a copy of P_j to not lose the original measurements.

$$p_{i_z} := \frac{p_{i_z} + \mu_z}{2}, \;\; \mu_z = \frac{1}{n}\sum_{i=1}^{n} p_{i_z} \tag{10}$$

The output of the smoothing filter shown in Eq. 10 is used as the input for the statistical filter shown in 13. Therefore, the indexes $q1$ of the first quartile Q_1 (also known as 25th percentile) and $q3$ of the third quartile Q_3 (75th percentile) are calculated. Using these indexes, the following boundaries are defined, where $(p_{q3_z} - p_{q1_z})$ is known as the interquartile range which contains 50% of the data.

$$f_{min} = p_{q1_z} - 1.5 \cdot (p_{q3_z} - p_{q1_z}) \tag{11}$$

$$f_{max} = p_{q3_z} + 1.5 \cdot (p_{q3_z} - p_{q1_z}) \tag{12}$$

$$P_f = \{p_i\} \mid i = 0,\ldots,n;\; l \leq i \leq m;\; f_{min} \leq p_{l_z};\; p_{m_z} \leq f_{max};\; p_{i_z} \leq p_{(i+1)_z} \tag{13}$$

The resulting filtered point cloud P_f is tested according the following criteria. If one of the conditions apply, the `obstacle`-label is assigned to the cell.

$$|P_f| < \rho \tag{14}$$

$$p_{m_z} - p_{l_z} > v_g \tag{15}$$

$$p_{m_z} > z_{max} \tag{16}$$

Here, the first rule is again a density check, the second rule tests if the cell range is acceptable and the third rule if the cell contains points which are above a reachable height.

The last and maybe strongest criterion evaluates properties of a plane fitted to the point cloud P_f. Therefore, a least squares fitting algorithm which minimizes the distance between the plane and the z-components of the points is used to find a plane defined as $z = ax + by + c$. For determination, the error $E(a,b,c) = \sum_{i=l}^{m} [(ap_{i_x} + bp_{i_y} + c) - p_{i_z}]^2$ needs to be minimized. According to [6], the following equation system 17 solves the problem, as the function $E(a,b,c)$ has its vertex when the gradient is zero.

$$\begin{pmatrix} \sum_{i=l}^{m} p_{i_x}^2 & \sum_{i=l}^{m} p_{i_x} p_{i_y} & \sum_{i=l}^{m} p_{i_x} \\ \sum_{i=l}^{m} p_{i_x} p_{i_y} & \sum_{i=l}^{m} p_{i_y}^2 & \sum_{i=l}^{m} p_{i_y} \\ \sum_{i=l}^{m} p_{i_x} & \sum_{i=l}^{m} p_{i_y} & \sum_{i=l}^{m} 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \sum_{i=l}^{m} p_{i_x} p_{i_z} \\ \sum_{i=l}^{m} p_{i_y} p_{i_z} \\ \sum_{i=l}^{m} p_{i_z} \end{pmatrix} \tag{17}$$

After the plane is determined, the slope γ is calculated as the enclosed angle between the normal of the plane and the z-axis: If the slope is above the maximum slope α (see Eq. 5) or above the desired attitude v_β the cell is interpreted as an obstacle. This decision can be overwritten and the cell is marked as `non-evaluable`, if the range $p_{m_z} - g_j$ is below the ground clearance value v_g.

Table 1: Vehicle properties used to evaluate a grid cell

Symbol	Property
v_h	Height of the vehicle
v_α	Maximum slope which the vehicle can handle between two cells
v_β	Maximum desired attitude (roll and pitch)
v_g	Ground clearance of the vehicle

3.4 Neighborhood Evaluation

At this stage, only the cells have been evaluated without taking their neighborhood into account. This could lead to some misclassifications and has to be corrected in the following steps. As the methods are working on the grid structure, the neighboring cells need to be known for each cell C_j. Figure 2 shows the naming convention which is used to describe the evaluation. The full neighborhood contains 8 cells $N_{(8)}(C_j) = \{N_i(C_j)\} \mid i = 1,\ldots,8$ while a reduced neighborhood $N_{(4)}(C_j)$—shown in red—only contains the neighboring cells with even indexes. Some extra attention is required at the borders of the grid as these cells do not have the full number of neighbors.

First, a function iterates over the whole grid and does the following analysis for each grid cell C_j which has not been labeled as `obstacle` or `non-evaluable` as described in Sec. 3.3. For each neighbor $N_i(C_j) \in N_{(8)}(C_j)$ which has been marked as potentially drivable in the cell analysis as well as for the center C_j, the mean height above the horizontal plane $\mu_z(C_j)$ is calculated based on the distance of the cell's center $(c_x w + \frac{w}{2}) + (c_y h - \frac{y_{max}}{2} + \frac{h}{2})$ to the fitted plane. Afterwards, the slope between the center and each adjoining cell N_i is determined as follows

$$\gamma_i(C_j, N_i(C_j)) = \text{atan2}\left[|\mu_z(C_j) - \mu_z(N_i(C_j))|, \text{dist}(C_j, N_i(C_j))\right] \tag{18}$$

where $\text{dist}(C_j, N_i(C_j))$ returns the spatial distance between two cell centers in the 2D $x-y$-plane. In addition, a counter is incremented for each slope measurement $\gamma_i(C_j, N_i(C_j))$ which is above the threshold v_α. If this counter is smaller than 4 after the evaluation, the cell is labeled as `obstacle` otherwise the cell is labeled as `drivable`. Some special cases have to be handled at the borders of the grid, in areas where no data points are available or if the label `non-evaluable` was assigned.

Finally, a post-processing step is executed to remove scattered `drivable` cells which are surrounded by obstacles. Therefore, a `drivable` cell close to the origin of the *RCS* is determined and used as a seed S. Furthermore, the cell is added to a list of non-isolated cells and its $N_{(4)}(S)$ neighbors are identified. For each of the neighbors $N_i(S) \in N_{(4)}(S)$ the assigned label is inspected. If it is not on the list of non-isolated cells and has been marked as `drivable` or `non-evaluable` it is used as a new seed and the method is recursively called.

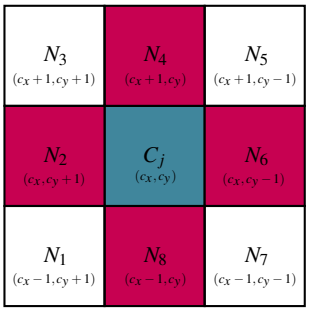


Fig. 2 Naming convention and traversing scheme of the grid cell neighborhood

3.5 Derived Properties

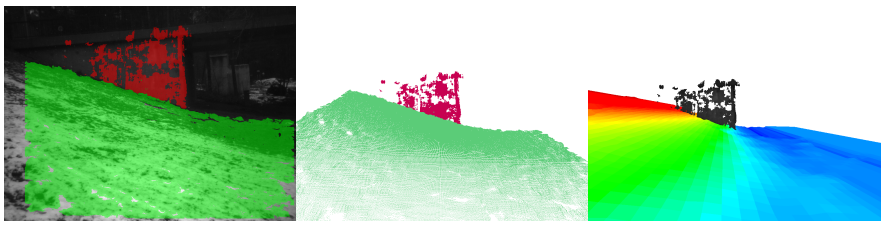
Based on the cell and the neighborhood evaluation different properties and views can be derived. For the presented application, the segmented obstacle view is the most important information. To generate this information, all cells tagged as `obstacle` are collected and added to an obstacle list. As long as this list contains elements, the following steps are repeated. The first element of the list generates a new obstacle cluster and is added to an auxiliary stack. Until the stack is empty, the $N_{(4)}(C_j)$ neighborhood of the top of the stack is analyzed and if it contains cells which are also on the obstacle list, they are added to the cluster as well as to an auxiliary stack and removed from the obstacle list. The process generates a collection of clusters which are enriched with some attributes like the maximum and minimum sample height within the cluster and the total number of 3D samples of the cluster. Furthermore, a polygon is calculated which describes the outer hull of the obstacle.

For the purpose of classification, the 3D points and the RGB-data of all obstacle clusters can be combined and accessed. This is possible since each cell of the grid still contains the original piece of the point cloud which was assigned to the cell. Besides the obstacle clusters, the evaluation results can also be used to divide the original point cloud into 3 separate clouds. The first one contains all points which belong to the traversable ground. The second one represents the measurements labeled as obstacles and the last one the overhanging objects which are higher than the vehicle. An example of a partitioned point cloud is shown in Fig. 3b. Here, the green points are showing all 3D points which belong to the ground. The red points are classified as an obstacle. In Figure 3a the point cloud is shown as an overlay on the left image of a grayscale stereo camera.

For some applications, like an inverse perspective mapping which can be used to extract waysides or road markings, a dominant ground plane is a valuable information. Using the presented grid based structure, such a plane can be easily extracted. In a first step, the 2D cell center points of all `drivable` cells are collected. Afterwards, the medium cell heights of the same cells are determined using the approach depicted in Sec. 3.4 incorporating the distance to the planes fitted to the cells. Finally, the generated 3D points are used as an input for a least-squares plane fitting as described at the end of Sec. 3.3 or by applying a RANSAC plane fitting.

In addition to a binary obstacle or not-obstacle view, also the shape of the terrain is interesting as some areas might be traversable but with an increased effort or unwanted side effects like reduced wheel grip. Here, an abstract terrain model can be helpful for path planning or implement guidance. Using the grid representation, this shape of the ground can again be extracted using the small planes fitted to each cell. The algorithm to create a reduced version of a Digital Elevation Map (DEM) starts at a cell close to the *RCS*'s origin which has not been labeled as `obstacle` or `non-evaluable` (it has a reasonable height information). This starting cell is added to an auxiliary stack. In this case, a stack is required as the used grid size blasts the maximum recursion depth. While the stack is not empty, the following steps are repeated: 1. The top element is removed. 2. The $N_{(8)}(C_j)$-neighbors are calculated. 3. If the cell can provide a height value, the value is added to the DEM together with the 2D coordinates of the cell center. If it has been labeled as `obstacle` or `non-evaluable` the height is averaged using the neighboring cells. If they cannot provide valuable data, the last height value is used. 4. All neighbors which are not yet represented in the DEM are added to the stack. The final elevation map is then generated by triangulating the determined points.

Figure 3c shows the result of the traversability analysis. The hilly ground on the left side of the image has been classified as drivable ground as shown in green on picture 3a based of the capabilities of the vehicle. Nevertheless, the height map in 3c shows that the slope is quite high and if it is not required to go there, this area should be avoided.

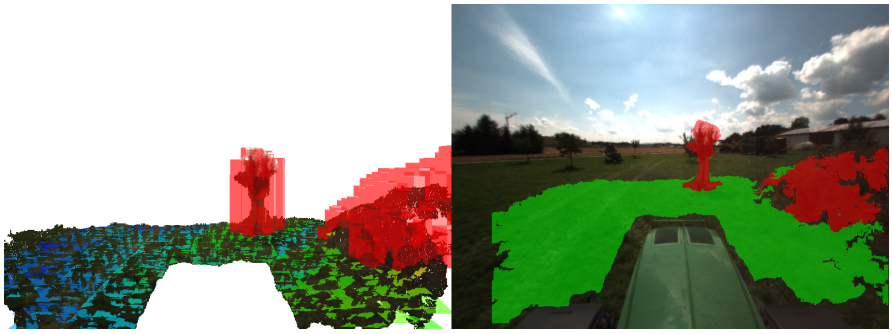


(a) (b) (c)

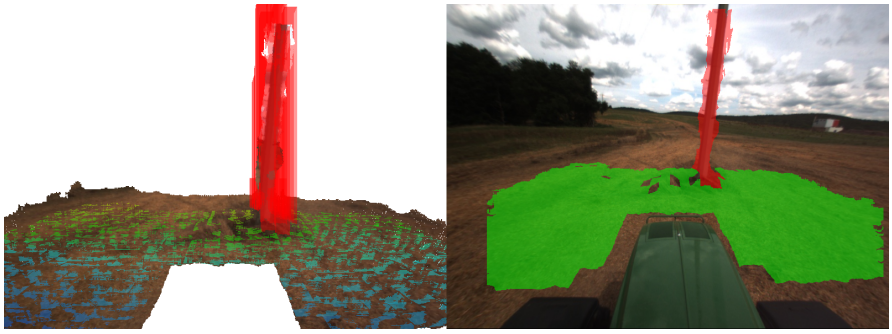
Fig. 3: Properties derived from the grid based evaluation: (a) image of the left camera showing a winter scenario of a hill and the wall of a bridge. Additionally, the classification results are overlayed (green: drivable, red: obstacle). (b) clustered point cloud: points belonging to the ground are shown in green, obstacle points are red, Terrain classification: the triangulated surface is color coded based on height above the x-y-plane.

4 Experiments and Results

To test and evaluate the presented obstacle detection approach, different scenarios have been recorded on the field, the campus of the University of Kaiserslautern, and the forest connected to it. For the field scenarios a John Deere 6R tractor was equipped with the stereo camera system described in Sec. 2, a differential GPS-system, an inertial measurement unit and other time-of-flight sensors to evaluate the data quality of the stereo camera. The first collection of datasets was recorded on grassland (see Fig. 4a) and on fields with different kinds and sizes of grass and weeds in summer 2014. With varying speeds from 1-15 $^{\mathrm{km}}/_{\mathrm{h}}$ different obstacle types and open field scenarios have been captured during different daytimes. In a second test, data has been recorded while using a stubble cultivator on a harvested grain



(a) Trees and a mound on grassland



(b) A power pole on a harvested grain field

Fig. 4: Two typical obstacle situations captured on the field. The left images depict the results of the grid based evaluation showing the fitted planes—the color scale depends on the height above the x-y-plane—and the identified obstacles as red boxes together with the colored 3D points. Right: resulting classification projected back to the *SCS* and visualized as an overlay on the left camera image. Green pixels show the drivable ground, obstacles are shown with red pixels.

field. This field contained several obstacles like a forest on one side, some houses at the opposite border, 2 power poles within the field and a ditch to a street nearby. Both datasets with 96,372 stereo image pairs in total were used to evaluate the obstacle detection system offline before deploying it to a real machine.

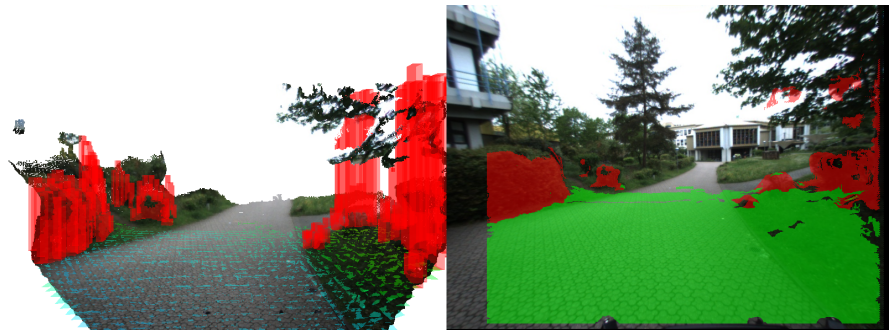
Both Figures 4a and 4b show some typical classification results which were created in the offline analysis. In all of these scenarios a cell size of $0.5\,\mathrm{m} \times 0.5\,\mathrm{m}$ was used to get rid of some small weeds sticking out of the ground. The grid dimensions where limited to $16\,\mathrm{m} \times 16\,\mathrm{m}$, as the point cloud density was too low for higher distances and the noise dramatically increased in farther regions. The 3D points belonging to the ground are summarized by the fitted planes which are shown in different colors depending on their distance to the horizontal plane. Cells which are classified as obstacles are highlighted with a red transparent box. Additionally, the individual 3D points are shown along with their color. For better understanding, the right part shows the left image of the stereo camera. The obstacle points have been back-projected to the image coordinates and are overlayed as a red mask for visualization. The hole in the center of the grid arises from the engine hood which was removed from the point cloud before handing the data to the obstacle detection module. Images 4a and 4b depict scenarios where the tractor was manually driven. The first figure shows the system's response to an apple tree and a small mound, the second visualization demonstrates the detection results of a large power pole which is blocking the path calculated by the GPS guidance system.

To quantitatively evaluate the detection performance, 100 randomly selected stereo pairs have been extracted from the recorded dataset described above. The ground-truth (obstacle or drivable ground) was manually set for each grid cell for all items of the selection. Using the parameters $v_h = 3.2\,\mathrm{m}$, $v_\alpha = 10\,\mathrm{deg}$ and $v_g = 0.5\,\mathrm{m}$ suitable for the tractor, it resulted in an average precision of 81.76%, a recall of 93.16% and an accuracy of 99.41%. The determined false positive rate is 0.45%. It should be mentioned, that the example images contained much more drivable cells (53941) than obstacles (1169) as the data was collected on real fields. Furthermore, it could be seen that most of the false positive detections were caused by weeds sticking out of the ground or by cells connected to real obstacles which appear larger in the stereo cloud. Additionally, most of the false positives which were identified by the slope estimation γ were positioned at the border of the camera's field of view were the grid cells are only partially filled with 3D points.

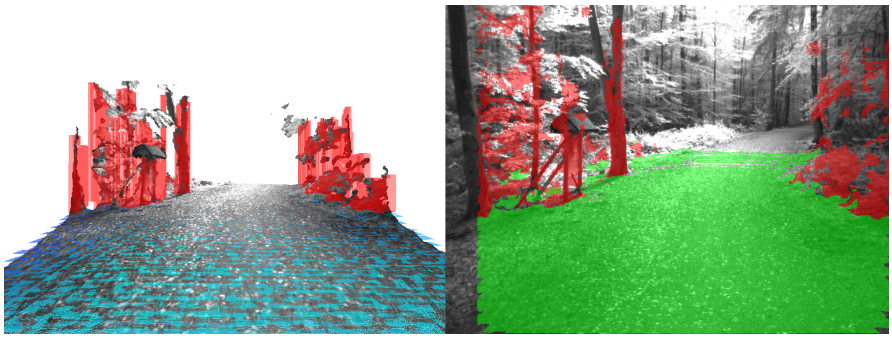
To demonstrate the applicability of the approach, the obstacle detection was integrated into a guidance system and implemented on a modified tractor with electronically controllable steering and velocity. Here, the output of the detection was passed to a map where the results of multiple frames were combined to increase the robustness and to get rid of single misclassified cells. If an obstacle intersected the space requirements along the calculated GPS-path, the tractor was stopped by decreasing the speed to zero. The prototype was used to demonstrate an emergency stop in front of a person, a tree and a small pole while driving tracks on the field.

In addition, the methodology has also been tested at the campus and inside the Palatinate Forest as the number of obstacles and overhangs is higher than in the field scenarios and the detection has to be more precise. Two examples are given in Fig.

5a and 5b. For both sites the grid cell size was reduced to 0.25 m $\times$ 0.25 m for a better segmentation of obstacles. Additionally, the ground clearance was reduced to 0.2 m to fit to the capabilities of the testing vehicle, a John Deere Gator XUV 855D. It can been seen in both examples that the drivable ground is correctly classified by the system as well as the obstacles within the detection range of the camera. For the forest capture, a Bumblebee XB3 grayscale camera was used instead of the Bumblebee2. Due to the larger baseline of 24 cm also the x dimension of the grid could be increased to 35 m. For both examples the same visualization scheme as described for the field scenarios applies.



(a) Campus scenario



(b) Forest scenario

Fig. 5: Obstacle detection applied to a campus and a forest scenario using a cell size of 0.25 m $\times$ 0.25 m. For the tests, the camera was mounted on a John Deere Gator XUV 855D at a height of 2 m above the ground. For the forest scenario, a grayscale Bumblebee XB3 camera was used instead of the Bumblebee2 model employed for the other experiments. Both visualizations use the same color coding as described in the previous figure.

5 Summary and Future Work

The obstacle detection system presented in this paper was successfully used to detect severe obstacles on the field, the campus and inside the forest. Splitting the detection into a grid cell and a neighborhood based part allows for parallelization of the detection process which makes the approach real-time capable. Furthermore, the results are more robust than a point-wise analysis as small outliers have a reduced influence on the evaluation.
The collected data showed that further research is needed to distinguish between soft weeds sticking out of the ground and dangerous solid objects which is challenging based on the geometric data extracted by the stereo system. Thus, the system is currently extended to extract the image patches representing the obstacles found by the geometric evaluation. Afterwards, the obstacle is analyzed in the image domain to further classify the obstruction and neglect it in case of weeds.

References

1. Bellone, M., Reina, G., Giannoccaro, N.I., Spedicato, L.: 3d traversability awareness for rough terrain mobile robots. Sensor Review **34**(2), 220–232 (2014)
2. Bernini, N., Bertozzi, M., Castangia, L., Patander, M., Sabbatelli, M.: Real-time obstacle detection using stereo vision for autonomous ground vehicles: A survey. In: ITSC, 2014 IEEE 17th Int. Conf. on, pp. 873–878. IEEE (2014)
3. Bradski, G., Kaehler, A.: Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media Inc. (2008)
4. Broggi, A., Buzzoni, M., Felisa, M., Zani, P.: Stereo obstacle detection in challenging environments: the viac experience. In: IROS, 2011 IEEE/RSJ Int. Conf. on, pp. 1599–1604. IEEE (2011)
5. Broggi, A., Cardarelli, E., Cattani, S., Sabbatelli, M.: Terrain mapping for off-road autonomous ground vehicles using rational b-spline surfaces and stereo vision. In: Intelligent Vehicles Symposium (IV), 2013 IEEE, pp. 648–653. IEEE (2013)
6. Eberly, D.: Least squares fitting of data. Chapel Hill, NC: Magic Software (2015)
7. Manduchi, R., Castano, A., Talukder, A., Matthies, L.: Obstacle detection and terrain classification for autonomous off-road navigation. Autonomous robots **18**(1), 81–102 (2005)
8. Milella, A., Reina, G.: 3d reconstruction and classification of natural environments by an autonomous vehicle using multi-baseline stereo. Intelligent Service Robotics **7**(2), 79–92 (2014)
9. Oniga, F., Nedevschi, S.: Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection. Vehicular Technology, IEEE Transactions on **59**(3), 1172–1182 (2010)
10. Reid, J.F., Zhang, Q., Noguchi, N., Dickson, M.: Agricultural automatic guidance research in north america. Computers and electronics in agriculture **25**(1), 155–167 (2000)
11. Reina, G., Milella, A.: Towards autonomous agriculture: Automatic ground detection using trinocular stereovision. Sensors **12**(9), 12,405–12,423 (2012)
12. Rouveure, R., Nielsen, M., Petersen, A., Reina, G., Foglia, M., Worst, R., Seyed-Sadri, S., Blas, M.R., Faure, P., Milella, A., et al.: The quad-av project: multi-sensory approach for obstacle detection in agricultural autonomous robotics. In: Int. Conf. of Agricultural Engineering CIGR-Ageng, pp. 8–12 (2012)
13. Santana, P., Guedes, M., Correia, L., Barata, J.: A saliency-based solution for robust off-road obstacle detection. In: ICRA, 2010 IEEE Int. Conf. on, pp. 3096–3101. IEEE (2010)