An Efficient Extension of Elevation Maps for Outdoor Terrain Mapping

Patrick Pfaff and Wolfram Burgard

Department of Computer Science, University of Freiburg, Germany,
{pfaff,burgard}@informatik.uni-freiburg.de

Summary. Elevation maps are a popular data structure for representing the environment of a mobile robot operating outdoors or on not-flat surfaces. Elevation maps store in each cell of a discrete grid the height of the surface the corresponding place in the environment. The use of this $2\frac{1}{2}$-dimensional representation, however, is disadvantageous when it is used for mapping with mobile robots operating on the ground, since vertical or overhanging objects cannot be represented appropriately. Such objects furthermore can lead to registration errors when two elevation maps have to be matched. In this paper we propose an approach that allows a mobile robot to deal with vertical and overhanging objects in elevation maps. We classify the points in the environment according to whether they correspond to such objects or not. We also describe a variant of the ICP algorithm that utilizes the classification of cells during the data association. Experiments carried out with a real robot in an outdoor environment demonstrate that the scan matching process becomes significantly more reliable and accurate when our classification is used.

1 Introduction

The problem of learning maps with mobile robots has been intensively studied in the past. In the literature, different techniques for representing the environment of a mobile robot prevail. Topological maps aim at representing environments by graph-like structures, where edges correspond to places, and arcs to paths between them. Geometric models, in contrast, use geometric primitives for representing the environment. Whereas topological maps have the advantage to better scale to large environments, they lack the ability to represent the geometric structure of the environment. The latter, however, is essential in situations, in which robots are deployed in potentially unstructured outdoor environments where the ability to traverse specific areas of interest needs to be known accurately. However, full three-dimensional models typically have too high computational demands for a direct application on a mobile robot.

Elevation maps have been introduced as a more compact $2\frac{1}{2}$-dimensional representation. An elevation map consists of a two-dimensional grid in which each cell stores the height of the territory. This approach, however, can be problematic when a

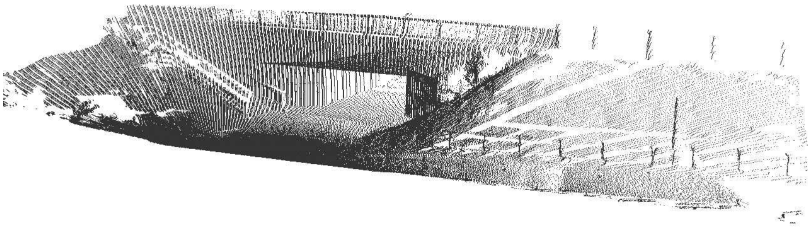

Fig. 1. Scan (point set) of a bridge recorded with a mobile robot carrying a SICK LMS laser range finder mounted on a pan/tilt unit.

robot has to utilize these maps for navigation or when it has to register two different maps in order to integrate them. For example, consider the three-dimensional data points shown in Figure 1. They have been acquired with a mobile robot standing in front of a bridge. The resulting elevation map, which is computed from averaging over all scan points that fall into a cell of a horizontal grid (given a vertical projection), is depicted in Figure 2. As can be seen from the figure, the underpass has completely disappeared and the elevation map shows a non-traversable object. Additionally, when the environment contains vertical structures, we typically obtain varying average height values depending on how much of this vertical structure is contained in a scan. Accordingly, if one registers two such elevation maps, one obtains incorrect alignments.

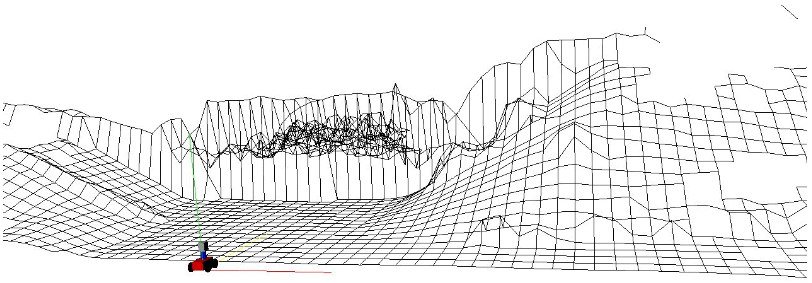

Fig. 2. Standard elevation map computed for the outdoor environment depicted in Figure 1. The passage under the bridge has been converted into a large un-traversable object.

In this paper we present a system for mapping outdoor environments with elevation maps. Our algorithm transforms range scans into local elevation maps and combines these local elevation maps using a variant of the ICP algorithm [3]. In our elevation maps, we classify locations in the environment into four classes, namely locations sensed from above, vertical structures, vertical gaps, and traversable cells. The advantage of this classification is twofold. First, the robot can represent obstacles corresponding to vertical structures like walls of buildings. It also can deal with overhanging structures like branches of trees or bridges. Furthermore, the classification can be utilized in the ICP algorithm to more robustly match local elevation maps. We present experimental results illustrating the advantages of our approach regarding the representation aspect as well as regarding the robust matching.

This paper is organized as follows. After discussing related work in the following section, we will describe our extension to the elevation maps in Section 3. In Section 4 we then describe how to incorporate our classification into the ICP algorithm used for matching elevation maps. Finally, we present experimental results in Section 5.

2 Related Work

The problem of learning three-dimensional representations has been studied intensively in the past. One of the most popular representations are raw data points or triangle meshes [1, 7, 12, 15]. Whereas these models are highly accurate and can easily be textured, their disadvantage lies in the huge memory requirement, which grows linearly in the number of scans taken. An alternative is to use three-dimensional grids [9] or tree-based representations [13], which only grow linearly in the size of the environment. Still, the memory requirements for such maps in outdoor environments are high.

In order to avoid the complexity of full three-dimensional maps, several researchers have considered elevation maps as an attractive alternative. The key idea underlying elevation maps is to store the $2\frac{1}{2}$-dimensional height information of the terrain in a two-dimensional grid. Bares et al. [2] as well as Hebert et al. [4] use elevation maps to represent the environment of a legged robot. They extract points with high surface curvatures and match these features to align maps constructed from consecutive range scans. Parra et al. [11] represent the ground floor by elevation maps and use stereo vision to detect and track objects on the floor. Singh and Kelly [14] extract elevation maps from laser range data and use these maps for navigating an all-terrain vehicle. Ye and Borenstein [16] propose an algorithm to acquire elevation maps with a moving vehicle equipped with a tilted laser range scanner. They propose special filtering algorithms to eliminate measurement errors or noise resulting from the scanner and the motions of the vehicle. Lacroix et al. [6] extract elevation maps from stereo images. They use a two-dimensional grid and store in each cell of this grid the average height. Hygounenc et al. [5] construct elevation maps in an autonomous blimp using 3d stereo vision. They propose an algorithm to track landmarks and to match local elevation maps using these landmarks. Olson [10] describes a probabilistic localization algorithm for a planetary rover that uses elevation maps for terrain modeling.

Compared to these techniques the contribution of this paper lies in two aspects. First, we classify the points in the elevation map into horizontal points seen from above, vertical points, and gaps. This classification is important especially when a rover is deployed in an urban environments. In such environments, typical structures like the walls of buildings cannot be represented in standard elevation maps. Second, we describe how this classification can be used to enhance the matching of different elevation maps.

3 Extended Elevation Maps

As already mentioned above, elevation maps are $2\frac{1}{2}$-dimensional representation of the environment. The maintain a two-dimensional grid and maintain in every cell of this grid an estimate about the height of the terrain at the corresponding point of the environment. To correctly reflect the actual steepness of the terrain, a common assumption is that the initial tilt and the roll of the vehicle is known.

When updating a cell based on sensory input, we have to take into account, that the uncertainty in a measurement increases with the distance measured due to errors in the tilting angle. In our current system, we a apply a Kalman filter to estimate the parameters $\mu_{1:t}$ and $\sigma_{1:t}$ about the elevation in a cell and its standard deviation. We apply the following equations to incorporate a new measurement z_t with standard deviation σ_t at time t [8]:

$$\mu_{1:t} = \frac{\sigma_t^2 \mu_{1:t-1} + \sigma_{1:t-1}^2 z_t}{\sigma_{1:t-1}^2 + \sigma_t^2} \tag{1}$$

$$\sigma_{1:t}^2 = \frac{\sigma_{1:t-1}^2 \sigma_t^2}{\sigma_{1:t-1}^2 + \sigma_t^2} \tag{2}$$

Note that the application of the Kalman filter allows us to take into account the uncertainty of the measurement. In our current system, we apply a sensor model, in which the variance of the height of a measurement increases linearly with the distance of the corresponding beam. This process is indicated in Figure 3.

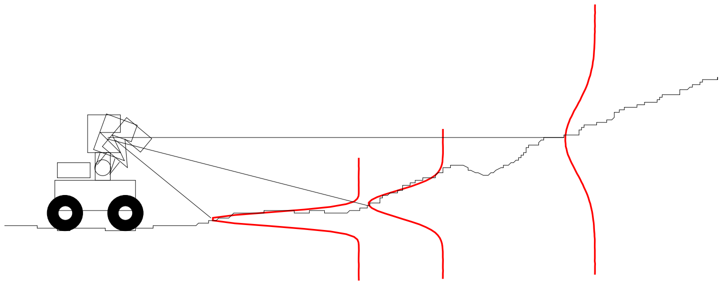

Fig. 3. Variance of a height measurements depending on the distance of the beam.

In addition we need to identify which of the cells of the elevation map correspond to vertical structures and which ones contain gaps. In order to determine the class of a cell, we first consider the variance of the height of all measurements falling into this cell. If this value exceeds a certain threshold, we identify it as a point that has not been observed from above. We then check, whether the point set corresponding to a cell contains gaps exceeding the height of the robot. When a gap has been identified, we determine the minimum traversable elevation in this point set.

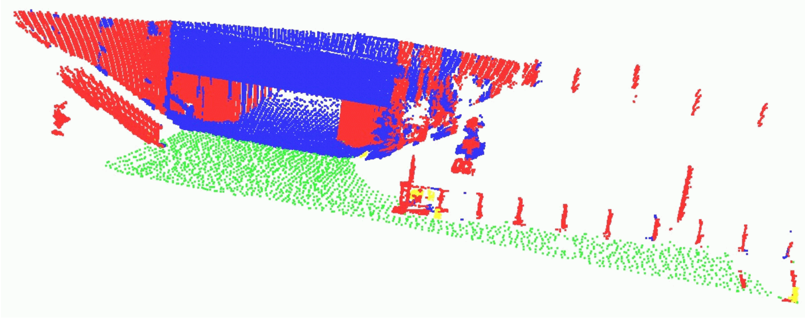

Fig. 4. Labeling of the data points depicted in Figure 2 according to their classification. The different colors/grey levels indicate the individual classes.

Figure 4 shows the same data points already depicted in Figure 2. The classes of the individual cells in the elevation map are indicated by the different colors/grey levels. The blue/dark points indicate the data points above a gap. The red/medium grey values indicate cells that are classified as vertical. The green/light grey values, however, indicate traversable terrain. Note that the not traversable cells are not shown in this figure.

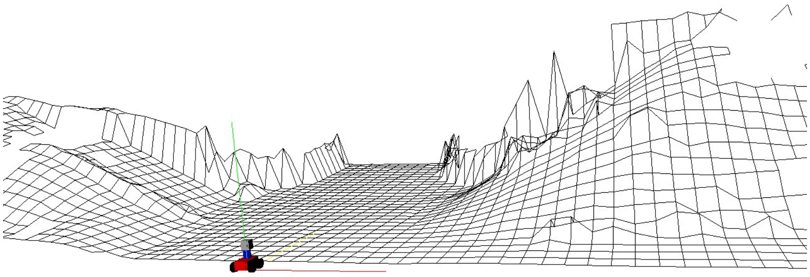

Fig. 5. Extended elevation map for the scene depicted in Figure 1.

A major part of the resulting elevation map extracted from this data set is shown in Figure 5. As can be seen from the figure, the area under the bridge can now be represented appropriately by ignoring data points above the lowest surface. This in turn enables the robot to plan a path through the passage under the bridge.

4 Efficient Matching of Elevation Maps in 6 Dimensions

To integrate several local elevation maps into a single global elevation map we need to be able register two maps relative to each other. In our current system, we apply the ICP algorithm for this purpose. The goal of the matching process is to minimize an error function defined over two point sets $X = \{x_1, \ldots, x_L\}$ and $Y = \{y_1, \ldots, y_L\}$, where each pair x_i and y_i is assumed to be the points that corresponding to each other. We are interested in the rotation R and the translation t that minimizes the following cost function:

$$E(R, t) = \frac{1}{n} \sum_{l=1}^{L} ||x_l - Ry_l - t||^2, \tag{3}$$

where $||\cdot||$ is a distance function that takes into account the variance of the Gaussians corresponding to each pair x_i and y_i.

In principle, one could define this function to directly operate on the height values and their variance when aligning two different elevation maps. The disadvantage of this approach, however, is that in the case of vertical objects, the resulting height seriously depends on the view point. The same vertical structure may lead to varying heights in the elevation map when sensed from different points. In practical experiments we observed that this introduces serious errors and often prevents the ICP algorithm from convergence. To overcome this problem, we separate Equation (3) into four components each minimizing the error over the individual classes of points. The first two classes consist of the cells corresponding to vertical objects and gaps. The latter two classes contain only cells whose points have been sensed from above. To increase the efficiency of the matching process, we only consider a subset of these cells. In practical experiments we found out that traversable cells and edge cells yield the best registration results. The traversable cells are those cells for which the elevation of the surface normal obtained from a plane fitted to the local neighborhood exceeds 83 degrees. Additionally, we consider edge cells, i.e., cells which lie more than 20cm above their neighboring points.

Let us assume that $\alpha_1, \ldots, \alpha_{N_\alpha}$ and $\alpha'_1, \ldots, \alpha'_{N_\alpha}$ are the corresponding vertical points, $\beta_1, \ldots, \beta_{N_\beta}$ and $\beta'_1, \ldots, \beta'_{N_\beta}$ are the vertical gaps, $\gamma_1, \ldots, \gamma_{N_\gamma}$ and $\gamma'_1, \ldots, \gamma'_{N_\gamma}$ are the edge points, and $\delta_1, \ldots, \delta_{N_\delta}$ and $\delta'_1, \ldots, \delta'_{N_\delta}$ are the traversable cells. The resulting error function then is

$$E(R, t) = \underbrace{\sum_{n=1}^{N_\alpha} d(\alpha_n, \alpha'_n)}_{\text{vertical objects}} + \underbrace{\sum_{n=1}^{N_\beta} d(\beta_n, \beta'_n)}_{\text{vertical gaps}} + \underbrace{\sum_{n=1}^{N_\gamma} d(\gamma_n, \gamma'_n)}_{\text{edge cells}} + \underbrace{\sum_{n=1}^{N_\delta} d(\delta_n, \delta'_n)}_{\text{traversable cells}}, \tag{4}$$

where $d(x, y) = ||x - Ry - t||$.

Figure 6 illustrates how two elevation maps are aligned over several iterations of the minimization process. Whereas the left column shows the point clouds the right column shows the cells in the elevation map used for minimizing Equation (4). In our current implementation, each iteration of the ICP algorithm usually takes between one and five seconds on a 2.8GHz Pentium 4. The time necessary to acquire a scan by tilting the laser is 5 seconds.

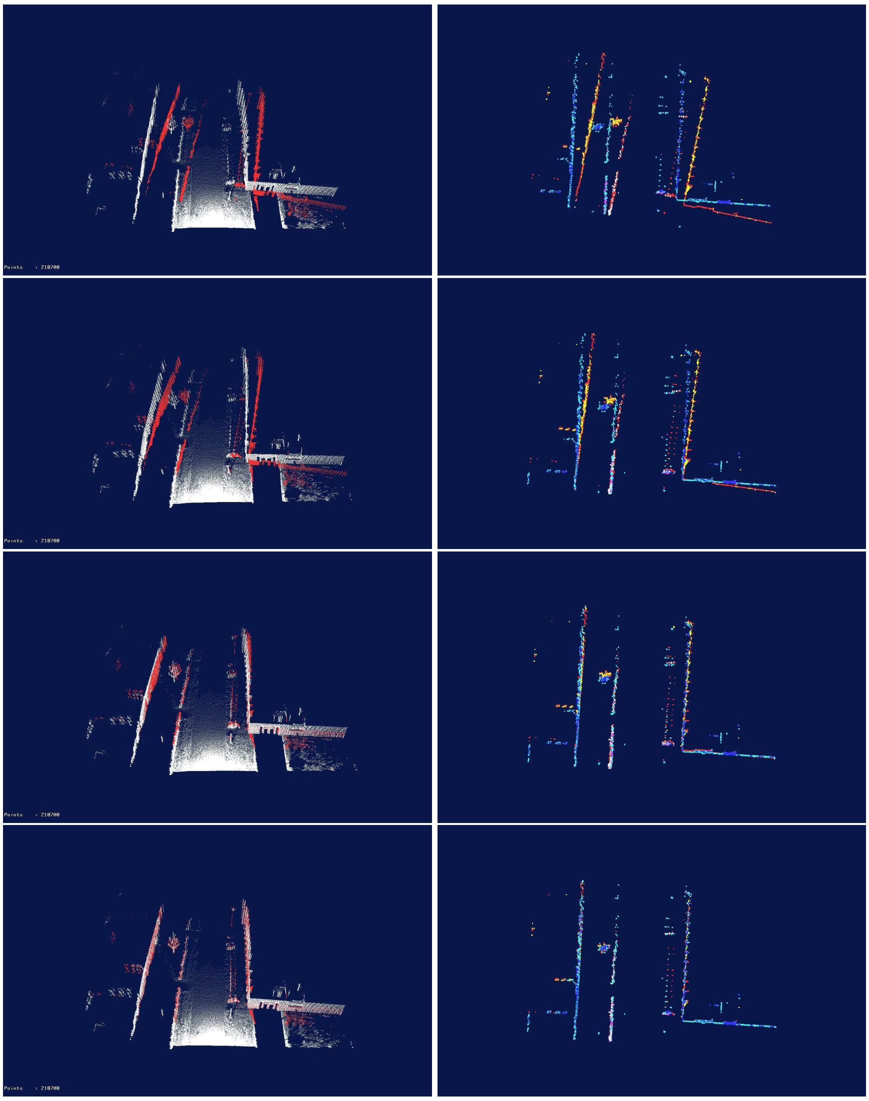

Fig. 6. Incremental registration of two elevation maps. The left column depicts the original point clouds. The right column shows the vertical and edge cells of the elevation maps used by the ICP algorithm. The individual rows correspond to the initial relative pose (top row), alignment after 5 iterations (second row), after 10 iterations (third row) and the final alignment after 30 iterations (fourth row).

In addition to the position and orientation of the vehicle we also have to estimate the tilt and roll of the vehicle when integrating two elevation maps. In practical experiments we found that an iterative scheme, in which we repeatedly estimate the tilt and roll of the robot and then determine the relative position and orientation of the involved elevation maps, improves the registration accuracy. In most cases, two iterations are sufficient to achieve precise matchings and to obtain highly accurate maps from multiple local maps generated from different viewpoints.

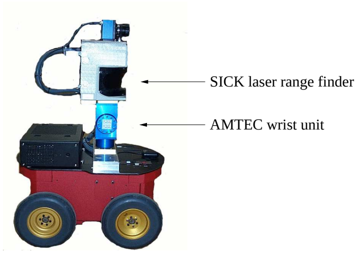

Fig. 7. Robot Herbert used for the experiments.

5 Experimental Results

The approach described above has been implemented and tested on a real robot system and in simulation runs with real data. The robot used to acquire the data is our outdoor robot Herbert, which is depicted in Figure 7. The robot is a Pioneer II AT system equipped with a SICK LMS range scanner and an AMTEC wrist unit, which is used as a pan/tilt device for the laser.

5.1 Learning Accurate Elevation Maps from Multiple Scans

To evaluate our approach we steered our robot Herbert through different areas of our university campus and visually inspected the maps obtained with our technique. In all cases, we obtained highly accurate maps. Figure 8 shows a typical example, in which the robot traveled under the bridge depicted in Figure 1 and then continued driving up a ramp. Along its path the robot generated local elevation maps from 36 scans. The overall number of data points recorded was 9,500,000. The size of each cell in the elevation map is 20 by 20cm. The whole map spans approximately 70 by 30 meters. As can be seen from the figure, the map clearly reflects the details of the environment. Additionally, the matching of the elevation maps is quite accurate.

Figure 9 shows a typical example in which our algorithm yields more accurate maps than the standard approach. In this situation the robot traveled along a paved way and scanned a tree located in front of the scene. Whereas the left image shows the map obtained with the standard elevation map approach, the right image shows

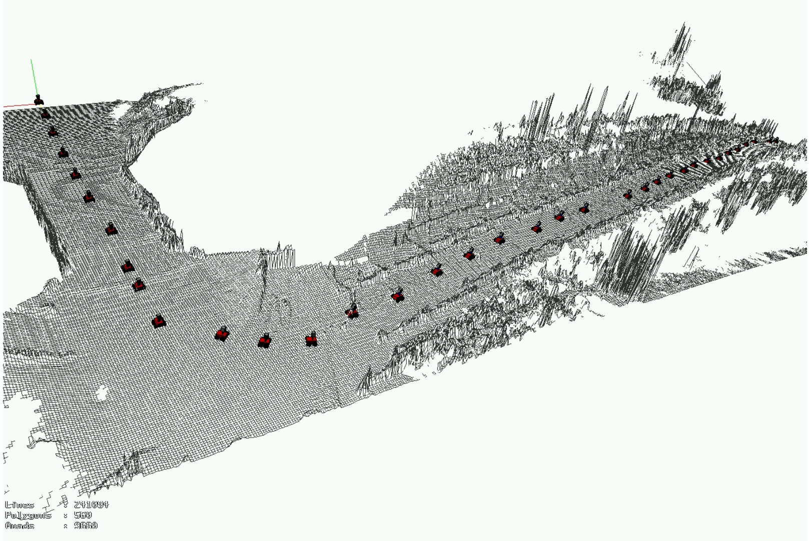

Fig. 8. Elevation map generated from 36 local elevation maps. The size of the map is approximately 70 by 30 meters.

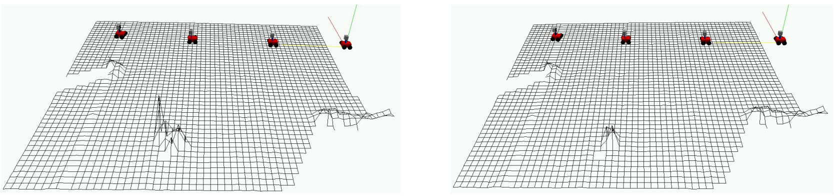

Fig. 9. Maps generated from four local elevation maps acquired with Herbert. The left image shows a standard elevation map. The right image depicts the map obtained with our approach. The peak in front of the scene corresponds to a tree, which is modeled more accurately with our approach.

the map obtained with our method. The individual positions of the robot where the scans were taken are also shown in the images. As can be seen from the figures, our method results in more free space around the stem of the tree.

5.2 Statistical Evaluation of the Accuracy

Additionally, we performed a series of experiments to get a statistical assessment as to whether the classification of the data points into normal, vertical and gap points combined with the sub-sampling of the normal points leads to better registration results. To perform these experiments we considered two different elevation maps for which we computed the optimal relative pose using several runs of the ICP algorithm. We then randomly added noise to the pose of the second map and applied the ICP algorithm to register both maps. We performed two sets of experiments to compare the registration results for the unclassified and the classified point sets. Table 1 shows the individual classes of noise that we added to the true relative pose of the two maps before we started the ICP algorithm. In this experiment described here, we only varied the pose error of the maps and kept the error in the rotations constant. In particular, we randomly chose rotational displacements from ±5 degrees around the real relative angle and also varying random displacements in the *x* and *y* direction.

displacement class	max. rot. displ.	max. displ. in *x* and *y*
1	±5 degrees	±0.5*m*
2	±5 degrees	±1.0*m*
3	±5 degrees	±1.5*m*
4	±5 degrees	±2.0*m*
5	±5 degrees	±2.5*m*

Table 1. Displacement classes used to evaluate the performance of the ICP algorithm on the classified and unclassified points extracted from the elevation maps.

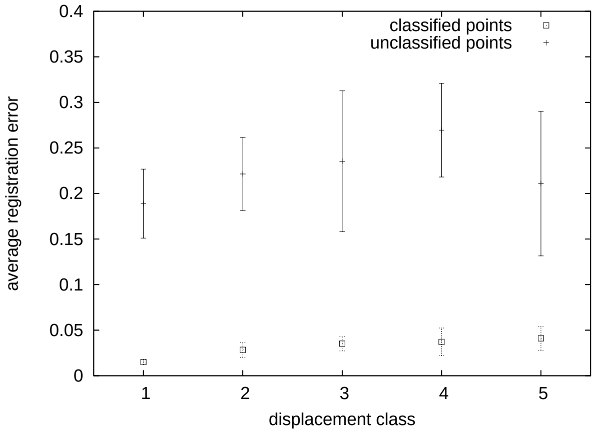

Fig. 10. Average registration error for the individual types of initial displacement.

The resulting average displacement errors after convergence of the ICP algorithm are depicted in Figure 10. As can be seen from the figure, the ICP algorithm performed significantly better on the classified point sets. In this figure, the error bars indicate the $\alpha = 0.05$ confidence level.

Additionally, we evaluated how often the ICP algorithm failed to accurately register the two maps. Figure 11 depicts the normalized divergence frequencies in percent for the individual displacement classes. As this plot illustrates, the utilization of the individual classes in the ICP algorithm leads to a seriously better convergence

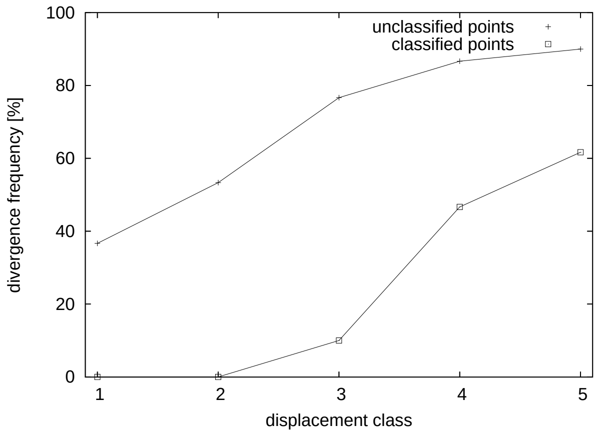

Fig. 11. Number of times the ICP algorithm diverges for the individual initial displacements.

rate. In additional experiments not reported here we obtained similar results for the different orientational errors.

6 Conclusions

In this paper we presented an approach to generate elevation maps from three-dimensional range data acquired with a mobile robot. Our approach especially addresses the problem of acquiring such maps with a ground-based vehicle. On such a system one often encounters situations, in which certain objects, such as walls or trees, are not seen from above. Accordingly, the resulting elevation maps contain incorrect information. The approach in this paper classifies the individual cells of elevation maps into four classes representing parts of the terrain seen from above, vertical objects, overhanging objects such as branches of trees or bridges, and traversable areas. We also presented an extension of the ICP algorithm that takes this classification into account when computing the registration.

Our algorithm has been implemented and tested on a real robot and using outdoor terrain data. Experimental results show that our classification yields more accurate elevation maps, especially in the cases of vertical objects and overhanging objects. Additionally, our extension of the ICP algorithm, which utilizes our classification, produces more accurate alignments and additionally converges more often.

Acknowledgment

This work has partly been supported by the German Research Foundation (DFG) within the Research Training Group 1103 and under contract number SFB/TR-8.

References

1. P. Allen, I. Stamos, A. Gueorguiev, E. Gold, and P. Blaer. Avenue: Automated site modeling in urban environments. In *Proc. of the 3rd Conference on Digital Imaging and Modeling*, pages 357–364, 2001.
2. J. Bares, M. Hebert, T. Kanade, E. Krotkov, T. Mitchell, R. Simmons, and W. R. L. Whittaker. Ambler: An autonomous rover for planetary exploration. *IEEE Computer Society Press*, 22(6):18–22, 1989.
3. P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.
4. M. Hebert, C. Caillas, E. Krotkov, I.S. Kweon, and T. Kanade. Terrain mapping for a roving planetary explorer. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 997–1002, 1989.
5. E. Hygounenc, I.-K. Jung, P. Souères, and S. Lacroix. The autonomous blimp project of laas-cnrs: Achievements in flight control and terrain mapping. *International Journal of Robotics Research*, 23(4-5):473–511, 2004.
6. S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury and; M. Herrb, and R. Chatila. Autonomous rover navigation on unknown terrains: Functions and integration. *International Journal of Robotics Research*, 21(10-11):917–942, 2002.
7. M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project: 3D scanning of large statues. In *Proc. SIGGRAPH*, pages 131–144, 2000.
8. P.S. Maybeck. The Kalman filter: An introduction to concepts. In *Autonomous Robot Vehicles*. Springer Verlag, 1990.
9. H.P. Moravec. Robot spatial perception by stereoscopic vision and 3d evidence grids. Technical Report CMU-RI-TR-96-34, Carnegie Mellon University, Robotics Institute, 1996.
10. C.F. Olson. Probabilistic self-localization for mobile robots. *IEEE Transactions on Robotics and Automation*, 16(1):55–66, 2000.
11. C. Parra, R. Murrieta-Cid, M. Devy, and M. Briot. 3-d modelling and robot localization from visual and range data in natural scenes. In *1st International Conference on Computer Vision Systems (ICVS)*, number 1542 in LNCS, pages 450–468, 1999.
12. K. Pervölz, A. Nüchter, H. Surmann, and J. Hertzberg. Automatic reconstruction of colored 3d models. In *Proc. Robotik 2004*, 2004.
13. Hanan Samet. *Applications of Spatial Data Structures*. Addison-Wesley Publishing Inc., 1989.
14. S. Singh and A. Kelly. Robot planning in the space of feasible actions: Two examples. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1996.
15. S. Thrun, C. Martin, Y. Liu, D. Hähnel, R. Emery Montemerlo, C. Deepayan, and W. Burgard. A real-time expectation maximization algorithm for acquiring multi-planar maps of indoor environments with mobile robots. *IEEE Transactions on Robotics and Automation*, 20(3):433–442, 2003.
16. C. Ye and J. Borenstein. A new terrain mapping method for mobile robot obstacle negotiation. In *Proc. of the UGV Technology Conference at the 2002 SPIE AeroSense Symposium*, 1994.