Autonomous Exploration for 3D Map Learning

Dominik Joho, Cyrill Stachniss, Patrick Pfaff, and Wolfram Burgard

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

Abstract. Autonomous exploration is a frequently addressed problem in the robotics community. This paper presents an approach to mobile robot exploration that takes into account that the robot acts in the three-dimensional space. Our approach can build compact three-dimensional models autonomously and is able to deal with negative obstacles such as abysms. It applies a decision-theoretic framework which considers the uncertainty in the map to evaluate potential actions. Thereby, it trades off the cost of executing an action with the expected information gain taking into account possible sensor measurements. We present experimental results obtained with a real robot and in simulation.

1 Introduction

Robots that are able to acquire an accurate model of their environment are regarded as fulfilling a major precondition of truly autonomous mobile vehicles. So far, most approaches to mobile robot exploration assume that the robot lives in a plane. They typically focus on generating motion commands that minimize the time needed to cover the whole terrain [1,2]. A frequently used technique is to build an occupancy grid map since it can model unknown locations efficiently. The robot seeks to reduces the number of unobserved cells or the uncertainty in the grid map. In the three-dimensional space, however, such approaches are not directly applicable. The size of occupancy grid maps in 3D, for example, prevents the robot from exploring an environment larger than a few hundred square meters.

Whaite and Ferrie [3] presented an exploration approach in 3D that uses the entropy to measure the uncertainty in the geometric structure of objects that are scanned with a laser range sensor. In contrast to the work described here, they use a fully parametric representation of the objects and the size of the object to model is bounded by the range of the manipulator. Surmann et al. [4] extract horizontal planes from a 3D point cloud and construct a polygon with detected lines (obstacles) and unseen lines (free space connecting detected lines). They sample candidate viewpoints within this polygon and use 2D ray-casting to estimate the expected information gain. In contrast to this, our approach uses an extension of 3D elevation maps and 3D ray-casting to select the next viewpoint. González-Baños and Latombe [5] also build a polygonal map by merging safe regions. Similar to our approach, they sample candidate poses in the visibility range of frontiers to unknown area. But unlike in our approach, they build 2D maps and do not consider the uncertainty reduction in the known parts of the map.

The contribution of this paper is an exploration technique that extents known techniques from 2D into the three-dimensional space. Our approach selects actions that reduce the uncertainty of the robot about the world and constructs a full three-dimensional

model using so-called multi-level surface maps. It reasons about the potential measurements when selecting an action. Our approach is able to deal with negative obstacles like, for example, abysms, which is a problem of robots exploring a three-dimensional world. Experiments carried out in simulation and on a real robot show the effectiveness of our technique.

2 3D Model of the Environment

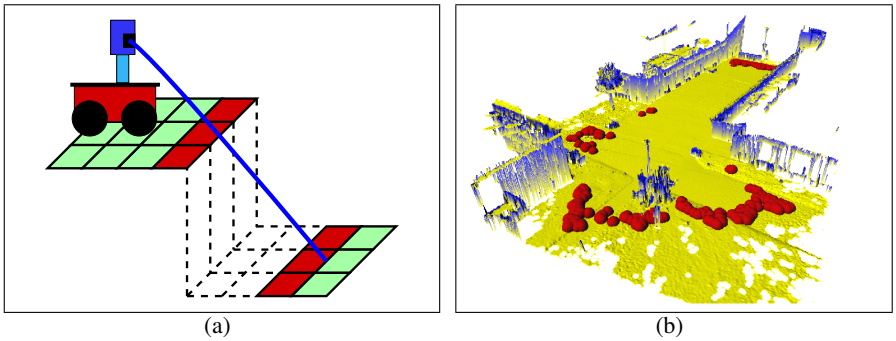
Our exploration system uses multi-level surface maps (MLS maps) as proposed by Triebel et al. [6]. MLS maps use a two-dimensional grid structure that stores different elevation values. In particular, they store in each cell of a discrete grid the height of the surface in the corresponding area. In contrast to elevation maps, MLS maps allow us to store multiple surfaces in each cell. Each surface is represented by a Gaussian with the mean elevation and its uncertainty σ. In the remainder of this paper, these surfaces are referred to as patches. This representation enables a mobile robot to model environments with structures like bridges, underpasses, buildings, or mines. They also enable the robot to represent vertical structures by storing a vertical depth value for each patch.

2.1 Traversability Analysis

A grid based 2D traversability analysis usually only takes into account the occupancy probability of a grid cell – implicitly assuming an even environment with only positive obstacles. In the 3D case, especially in outdoor environments, we additionally have to take into account the slope and the roughness of the terrain, as well as negative obstacles such as abysms which are usually ignored in 2D representations.

Each patch p will be assigned a traversability value $\tau(p) \in [0,1]$. A value of zero corresponds to a non-traversable patch, a value greater zero to a traversable patch, and a value of one to a perfectly traversable patch. In order to determine $\tau(p)$, we fit a plane into its local 8-patch neighborhood by minimizing the z-distance of the plane to the elevation values of the neighboring patches. We then compute the slope and the roughness of the local terrain and detect obstacles. The slope is defined as the angle between the fitted plane and a horizontal plane and the roughness is computed as the average squared z-distances of the height values of the neighboring patch to the fitted plane. The slope and the roughness are turned into traversability values $\tau_s(p)$ and $\tau_r(p)$ by linear interpolation between zero and a maximum slope and roughness value respectively. In order to detect obstacles we set $\tau_o(p) \in \{0,1\}$ to zero, if the squared z-distance of a neighboring patch exceeds a threshold, thereby accounting for positive and negative obstacles, or if the patch has less than eight neighbors. The latter is important for avoiding abysms in the early stage of an exploration process, as some neighboring patches are below the edge of the abysm and therefore are not visible yet (see Fig. 1 (a)).

The combined traversability value is defined as $\tau(p) = \tau_s(p) \cdot \tau_r(p) \cdot \tau_o(p)$. Next, we iteratively propagate the values by convolving the traversability values of the patch and its eight neighboring patches with a Gaussian kernel. For non-existent neighbors, we assume a value of 0.5. The number of iterations depends on the used cell size and the robot's size. In order to enforce obstacle growing, we do not perform a convolution



(a) (b)

Fig. 1. (a) While scanning at an abysm, some of the lower patches will not be covered by a laser scan (dashed area). Since the patches at the edge of the abysm have less than eight neighbors, we can recognize them as an obstacle (red / dark gray area). **(b)** Outdoor map showing sampled candidate viewpoints as red (dark gray) spheres.

if one of the neighboring patches is non-traversable ($\tau = 0$), but rather set the patch's traversability directly to zero in this case.

3 Our Exploration Technique

An exploration strategy has to determine the next viewpoint the robot should move to in order to obtain more information about the environment. Identifying the best viewpoint is a two step procedure in our system. First, we define the set of possible viewpoints or candidate viewpoints. Second, we evaluate those candidates to find the best one.

3.1 Viewpoint Generation

One possible definition of the set of candidate viewpoints is that every reachable position in the map is a candidate viewpoint. However, this is only feasible if the evaluation of candidate viewpoints is computationally cheap. If the evaluation is costly, one has to settle for heuristics to determine a smaller set. A popular heuristic is the frontier approach [2] that defines candidate viewpoints as viewpoints that lie on the frontier between obstacle-free and unexplored areas. In our approach, a patch is considered as explored if it has eight neighbors and its uncertainty, measured by the entropy in the patch, is below a threshold. Additionally, we track the entropy as well as the number of neighbors of a patch. If the entropy or number of non-existing neighbors cannot be reduced as expected over several observations, we consider it to be explored nonetheless since further observations do not seem to change the state of the patch.

A frontier patch is defined as an unexplored patch with at least one explored neighboring patch. Most of these patches have less than eight neighbors and therefore are considered as non-traversable, since they might be at the edge of an abysm. Therefore, we cannot drive directly to a frontier patch. Instead, we use a 3D ray-casting technique to determine close-by candidate viewpoints. A patch is considered as a candidate

viewpoint, if it is reachable and there is at least one frontier patch that is likely to be observable from that viewpoint. Instead of using ray-casting to track emitted beams from the sensor at every reachable position, we use a more efficient approach. We emit virtual beams from the frontier patch instead and then select admissible sensor locations along those beams. This will reduce the number of needed ray-casting operations as the number of frontier patches is much smaller than the number of reachable patches.

In practice, we found it useful to reject candidate viewpoints, from which the unseen area is below a threshold. We also cluster the frontier patches by the neighboring relation, and prevent patches from very small frontier clusters to generate candidate viewpoints. This will lead to a more reliable termination of the exploration process. Candidate viewpoints of an example map are shown in Fig. 1 (b).

3.2 Viewpoint Evaluation

The utility $u(v)$ of a candidate viewpoint v, is computed using the expected information gain $I(v)$ and the travel costs $t(v)$. As the evaluation involves a costly 3D ray-casting operation, we reduce the set of candidate viewpoints by sampling uniformly a fixed number of viewpoints from that set.

In order to simultaneously determine the shortest paths to all candidate viewpoints, we use a deterministic variant of the value iteration [7]. The costs

$$c(p,p') = \mathrm{dist}(p,p') + w(1-\tau(p')) \tag{1}$$

from patch p to a traversable neighboring patch p' considers the distance $\mathrm{dist}(p,p')$, as well as the traversability $\tau(p')$. A constant factor w is used to weight the penalization for traversing poorly traversable patches. The travel costs $t(v)$ of a viewpoint v is defined as the accumulated step costs of the shortest path to that viewpoint.

In order to evaluate the information gain of a viewpoint candidate, we perform a ray-cast operation to determine the patches that are likely to be hit by a laser measurement similar to [8]. We therefore determine the intersection points of the cell boundaries and the 3D ray projected onto the 2D grid. Next we determine for each cell the height interval covered by the ray and check for collisions with patches contained in that cell by considering their elevation and depth values. Using a standard notebook computer, our approach requires around 25 ms to evaluate one potential viewpoint including the 3D ray-cast operation. This allows us to run our algorithm with minimal delays only for typical environments.

For each casted ray that hits a patch, we temporary add a new measurement into the patch's grid cell with a corresponding mean and variance that depends on the distance of the laser ray. The mean and variance of the patch will then be updated using the Kalman update. As a patch is represented as a Gaussian, we can compute the entropy $H(p)$ of the patch as

$$H(p) = \frac{1}{2}\left(1+\log\left(2\pi\sigma^2\right)\right). \tag{2}$$

The information gain $I(p)$ of a ray-cast is then defined as the difference between the entropy $H(p)$ of the patch before and the entropy $H(p \mid m)$ after the temporary incorporation of the simulated measurement

$$I(p) = H(p) - H(p \mid m). \tag{3}$$

Additionally, we add a constant value for each empty cell traversed by the ray. In this way, we reward viewpoints from which unseen areas are likely to be visible, while we are still accounting for the reduction of existing uncertainties in the known map. Rays that do not hit any patch and do not traverse any empty cells, will result in an information gain of zero. The information gain $I(v)$ of a viewpoint v is then defined as the sum of the information gains of all casted rays. Finally, the utility $u(v)$ of each candidate viewpoint is computed by a relative information gain and travel costs as

$$u(v) = \alpha \frac{I(v)}{\max_x I(x)} + (1-\alpha)\frac{\max_x t(x) - t(v)}{\max_x t(x)}. \quad (4)$$

By varying the constant $\alpha \in [0,1]$ one can alter the exploration behavior by trading off the travel costs and the information gain.

3.3 Localization and Termination

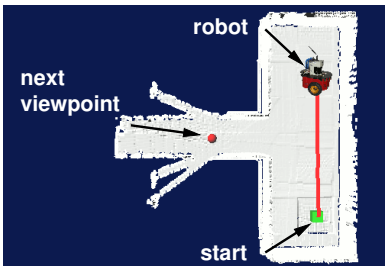
The registration of newly acquired information involves a scan matching procedure with the previous local map. We therefore cannot drive directly to the next viewpoint, as the resulting overlap with the previous local map may be to small. Hence, we perform several 3D scans along the way, which has the benefit, that it allows us to optimize the localization of the robot with the pose returned by the scan matcher. We apply a 6D Monte Carlo localization proposed by Kümmerle et al. [9]. After each 3D scan, we replan the path to the selected viewpoint. If the viewpoint is unreachable, we choose a new one. The exploration terminates if the set of candidate viewpoints is empty.

4 Experiments

The experiments described here are designed to illustrate the benefit of our exploration technique which is able to build three-dimensional models of the environment and takes into account the travel costs and the expected change in the map uncertainty to evaluate possible actions. For the real-world experiments we used an ActivMedia Pioneer2-AT robot with a SICK laser range finder mounted on a pan-tilt unit to acquire three-dimensional range data. For a 3D scan we tilt the laser in a range of 40 degrees at four equally spaced horizontal angles while acquiring the laser data.

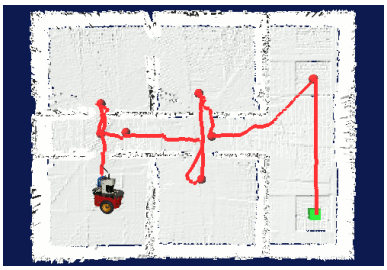
We tested our approach in simulation and in a real-world scenario. For the simulation experiments, we used a physical simulation environment that models our Pioneer robot with its pan-tilt unit. The simulated indoor environment consisted of four rooms, each connected to a corridor, and a foyer where the robot is located initially. The upper two rooms are connected directly through a door, while the lower ones are not. The robot efficiently covered the environment taking into account its constraints like travel cost, and information gain. The robot traveled 59 meters, visited eight viewpoints, and performed 15 scans (see Fig. 2 (a)-(c)). The final map, depicted in Fig. 2 (c) and (d), covers an area of $22m \times 17m$ and contains about 41,000 patches.

Real-world experiments have been carried out on the university campus. In the experiment shown in Fig. 2 (e)-(h) the robot traveled 84 meters, visited six viewpoints, and performed 23 scans. The map depicted in Fig. 2 (g) and (h) contains about 197,000

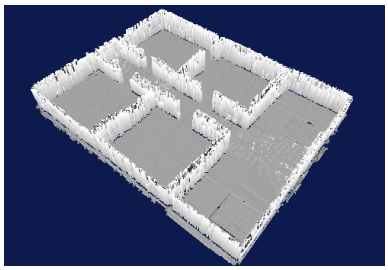


(a) Robot reached the first viewpoint.

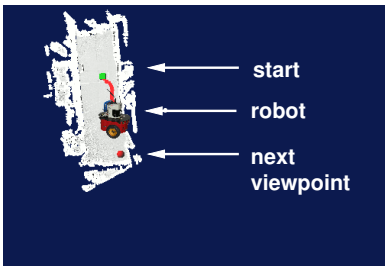
(b) Robot reached the fourth viewpoint.



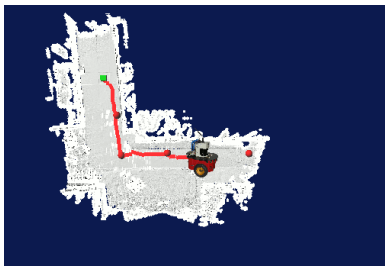
(c) Robot reached the final viewpoint.



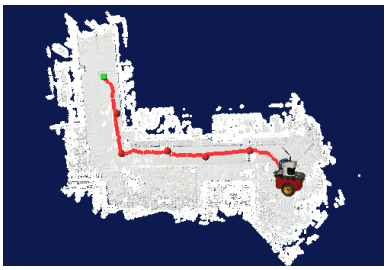
(d) Perspective view of the final map.



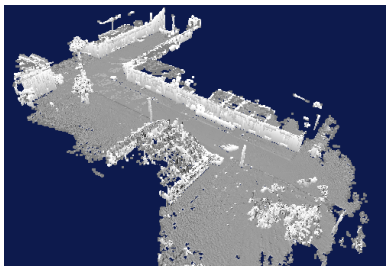
(e) Robot reached the first viewpoint.



(f) Robot reached the fourth viewpoint.



(g) Robot reached the sixth viewpoint.



(h) Perspective view of the map.

Fig. 2. (a)-(d) Exploration in a simulated indoor environment. One can see four rooms, a corridor, and the foyer where the robot started the exploration. **(e)-(h)** Real-world exploration in an outdoor scenario. One can see the walls of three buildings, the pitched roof of a green house, and several street lamps and trees.

patches and its bounding box roughly covers an area of about $70m \times 75m$. In both experiments, we set $\alpha = 0.55$ and used a cell size of $0.1m \times 0.1m$.

5 Conclusion

In this paper, we presented an approach to autonomous exploration for mobile robots that is able to acquire a three-dimensional model of the environment, which is compactly represented by a multi-level surface map. We addressed problems which are not encountered in traditional 2D representations such as negative obstacles, roughness, and slopes of non-flat environments. The viewpoint generation and evaluation procedure utilizes 3D ray-casting operations to account for the 3D structure of the environment. We applied a decision-theoretic framework which considers both the travel costs and the expected information gain to efficiently guide the exploration process. Simulation and real-world experiments showed the effectiveness of our technique.

6 Acknowledgment

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

7 References

1. Tovey C, Koenig S: Improved analysis of greedy mapping. Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2003.
2. Yamauchi B: Frontier-based exploration using multiple robots. Proc. of the Second Int. Conf. on Autonomous Agents 47–53, 1998.
3. Whaite P, Ferrie FP: Autonomous exploration: Driven by uncertainty. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(3):193–205, 1997.
4. Surmann H, Nüchter A, Hertzberg J: An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. Journal of Robotics and Autonomous Systems 45(3-4):181–198, 2003.
5. González-Baños HH, Latombe JC: Navigation strategies for exploring indoor environments. Int. Journal of Robotics Research 21(10-11):829–848, 2002.
6. Triebel R, Pfaff P, Burgard W: Multi-level surface maps for outdoor terrain mapping and loop closing. In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2006.
7. Burgard W, Moors M, Stachniss C, et al.: Coordinated multi-robot exploration. IEEE Transactions on Robotics 21(3):376–378, 2005.
8. Stachniss C, Grisetti G, Burgard W: Information gain-based exploration using rao-blackwellized particle filters. Proc. of Robotics: Science and Systems (RSS) 65–72, 2005.
9. Kümmerle R, Triebel R, Pfaff P, et al.: Monte carlo localization in outdoor terrains using multi-level surface maps. Proc. of the Int. Conf. on Field and Service Robotics (FSR), 2007.