

Foresighted Navigation Through Cluttered Environments

Peter Regier

Stefan Oßwald

Philipp Karkowski

Maren Bennewitz

Abstract—In this paper, we introduce an approach to efficient robot navigation through cluttered indoor environments. We propose to estimate local obstacle densities based on already detected objects and use them to predict traversal costs corresponding to potential obstacles in regions not yet observable by the robot’s sensors. By taking into account the predicted costs for path planning, the robot is then able to navigate in a more foresighted manner and reduces the risk of getting stuck in cluttered regions. We thoroughly evaluated our approach in simulated and real-world experiments. As the experimental results demonstrate, our method enables the robot to efficiently navigate through environments containing cluttered regions and achieves significantly shorter completion times compared to a standard approach not using any prediction.

I. INTRODUCTION

Today, mobile robots are able to localize and navigate in laboratory environments that are nicely tidied up, static, and have an accurate, up-to-date world representation such as a floor plan or a pre-recorded SLAM map. When robots step out of the laboratory, however, they face an ever-changing world filled with piles of objects and moving obstacles that are not contained in the static environment representation. In particular, service robots operating alongside humans in daily-life domestic and office environments need to navigate efficiently and robustly also through cluttered scenes, for example in children’s rooms with toys scattered on the floor, in workshops with tools laying around, or in storage rooms with piles of boxes.

Driving through clutter is challenging as it requires accurate sensing of obstacles and precise motion execution. The robot has to drive slower, turn frequently, replan a collision-free path when new obstacles come into its field of view, backtrack and take a detour when it gets stuck in a dead end, execute dangerous maneuvers such as driving backwards, and may eventually even have to give up when it is unable to find a suitable path back out of the cluttered area.

Hence, the shortest path planned on a given floor plan is not necessarily the most efficient path, as taking a detour around cluttered areas right from the beginning may be longer, but faster and safer to execute. Consider, for example, the scene depicted in Fig. 1. If the robot takes the green path, it can drive fast and steadily to the goal without encountering obstacles, which is more efficient than taking the red path and potentially getting stuck.

If the world was fully observable and the distribution of clutter was known, the robot would be able to plan a path that is optimized for efficiency by reasoning about safety

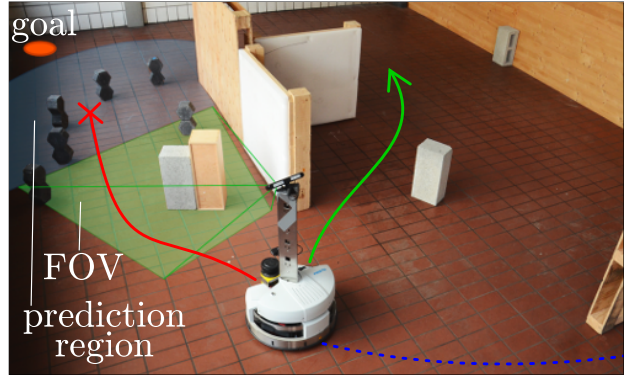


Fig. 1. A robot navigating through a cluttered environment has to choose between different paths for navigating to a goal. While the red path is shorter, it passes through a cluttered area, putting the robot at risk of getting stuck. The green path is longer, but it is safer. Our approach predicts traversal costs corresponding to potential obstacles outside the field of view (FOV) based on already detected objects. This enables the robot to choose more promising paths that are likely to lead to a shorter completion time.

distances, allowable velocities, and probabilistic sensing and motion execution errors. While solving this problem is already challenging, accurate planning is impossible when the world is only partially observable. Due to limited sensor ranges and occlusions from the robot’s point of view, the true distribution of clutter is unknown to the robot, hence it cannot directly plan and optimize the planned path for efficiency.

Traditional planners can only consider objects that are either registered in the given map of the environment, or have already been observed by the sensors, so these planners would assign equal traversal costs to all unobserved areas in the free space of the given map. However, clutter is typically not spread uniformly. In a children’s room, toys often come in piles. In a workshop, tools usually gather around the workbench. In storage rooms, boxes are usually stacked in heaps close to each other. Using knowledge about the distribution of clutter allows for predicting cluttered areas in regions that the robot has not yet observed, which enables the robot to plan its path in a more foresighted manner.

In this paper, we propose a method to predict the occurrence of obstacles in the space outside the field of view from the information about objects in already observed areas. By increasing the costs for traveling through areas where obstacles are expected, we allow a path planner to avoid cluttered areas, which leads to more foresighted navigation. Since the world is only partially observable, there is no guarantee that the predicted obstacles actually exist and that the planned path is optimal with respect to efficiency, but it seems reasonable to avoid difficult navigation challenges if easier and safer options are available.

We implemented our system in the Robot Operating System (ROS) and thoroughly evaluated it both in simulated and real-world experiments with a Robotino robot from Festo Didactics. As the experimental results demonstrate, our approach enables the robot to react to unexpected objects in a foresighted manner and to navigate efficiently. In comparative experiments with a traditional path planner, our method achieves significantly shorter completion times in various complex scenarios.

II. RELATED WORK

Lu *et al.* introduced the notion of *layered costmaps* and implemented it in ROS [1]. Instead of maintaining a single costmap, the authors proposed to split the information of the costmap into several layers with different semantics. Each layer represents a different type of obstacle or constraint, such as the static map, caution zones, and the personal space of a human that the robot should not penetrate. This concept has been widely used, such as for path planning or in order to represent the exploration progress. Marder-Eppstein *et al.* [2] suggested to exploit the costmap representation for navigating a PR2 robot through a cluttered real-world office environment containing obstacles of varying shapes and sizes. While their navigation system tries to navigate even through difficult obstacle fields, our approach predicts the scene behind initially detected obstacles and considers taking a detour right from the start if navigating through the clutter does not look promising.

Hornung *et al.* [3] considered 3D environments containing a moderate amount of cluttered objects. The focus was mainly on finding collision-free upper body configurations of the robot for traversing tight passages. For humanoid robots, cluttered environments are particularly challenging as the robots have to adhere to balancing constraints in addition to the navigation task. Hence, they have to choose their footholds carefully when moving through cluttered areas. Orthey and Stasse [4] as well as Maier *et al.* [5] proposed suitable solutions for this task given observed obstacle locations.

Joho *et al.* [6] presented a technique using nonparametric Bayesian models for learning exact geometric arrangements of objects from large data sets. The authors showed that their unsupervised approach is able to learn how to set the table. In a related approach, Sudderth *et al.* [7] also reasoned about the number of objects and their spatial relation for detecting items in visual scenes. While the authors are interested in finding reproducible patterns in the geometric arrangements, we cannot expect that there are typical structures that can be learned in our scenario. Instead, we propose an efficient approach for predicting traversal costs for unobserved parts of the environment based on cluttered objects already observed.

When navigating alongside humans in crowded environments such as pedestrian areas, robots have to adapt to the habits of humans, for example with respect to crowd flows or personal spaces of humans. Henry *et al.* [8] proposed an approach for learning to imitate human behavior in these situations. Urban environments are another example for highly cluttered environments that introduce particular challenges for navigation such as traversability estimation, traffic rules,

and interactions with passers-by [9]. While these approaches focus on predicting how the state of the observed world will evolve in the future, our work focuses on predicting traversal costs corresponding to clutter in so-far unobserved parts of the environment. It would be interesting to combine the two orthogonal prediction approaches in future work to achieve better prediction of both moving and static obstacles.

Here, we focus on domestic and office environments. Based on detected objects, we predict costs corresponding to potential obstacles in close but not yet observed areas. By making detours around regions that are likely too cluttered for the robot to easily pass through, our robot avoids getting stuck and efficiently navigates to the goal.

III. COST MAPS FOR PATH PLANNING IN CLUTTERED ENVIRONMENTS

We assume that a 2D grid map containing the static obstacles in the environment is given. As the robot moves through the environment, it updates the map continuously by marking cells as *occupied* or *free* whenever previously unknown objects appear in the robot's field of view. The robot uses this map to localize itself and to predict costs corresponding to potential cluttered objects in areas that it has not yet observed with its sensors.

For path planning, a cost value is assigned to each grid cell and an A*-based planner finds the path with the lowest costs to the given goal location of the robot. In the following, we first describe a standard cost function that considers a safety distance to all obstacles. We then introduce an extension to the cost function that also takes into account the clutter density of the local surroundings.

A. Standard Cost Function

The standard cost function sets the costs of all occupied cells to infinity, furthermore, occupied cells are inflated by a safety distance and their neighbor cells get also assigned infinite costs. In that way, the robot keeps a safety distance r to all obstacles. While this results in the computation of collision-free paths, it may lead to undesirable behavior as the shortest path from one room to another would often run close to walls and door posts when entering a room. While some situations require to squeeze through narrow passages or move close to obstacles, the robot should generally prefer to maintain a certain clearance. A commonly used approach to represent such a preference for open space is to add costs to the regions adjacent to inflated obstacles that decay exponentially with growing distance to the nearest obstacle. This approach allows the planner to trade off between path length and obstacle clearance. As this cost term quickly loses influence with increasing distance, it only has to be computed for cells near obstacles and can be neglected for cells further away.

Putting these cost terms together yields the following cost function:

$$C_{base} = \begin{cases} 0 & \text{if } n = 0 \\ \infty & \text{if } \exists i : d_i < r, \\ C_{max} \cdot \max_i e^{k(r-d_i)} & \text{otherwise} \end{cases}, \quad (1)$$

where r is the robot's safety distance, d_1, \dots, d_n are the distances to the nearby obstacles, C_{max} defines the highest traversable cost, and k is a constant scaling factor controlling the decay of the costs. The red curve in Fig. 2 illustrates this cost function in an example environment with two obstacles.

B. Cost Function for Cluttered Environments

The cost function described in Eq. (1) takes only the nearest obstacle into account and neglects all other obstacles. However, areas with multiple obstacles close to each other are more challenging for the robot, as locomotion in confined spaces is difficult and the sensor view is often obstructed. Increasing factor k would incite the robot to keep a larger distance to all obstacles irrespective of the amount of clutter, leading to unnecessarily long detours around single, free-standing objects. By contrast, we propose to introduce an additional cost term that reflects the amount of clutter in the local surroundings of each grid cell.

The new cost function should meet these requirements:

- The cost value should increase exponentially with the number of objects in the vicinity of the robot, so that the planner will avoid cluttered regions.
- The cost function should have the same value range as the original function Eq. (1).
- The cost function should approach zero as the distances go to infinity.
- If there is only one object in the vicinity of the robot, then the cost function should coincide with the original cost function from Eq. (1).

The following cost function fulfills these requirements:

$$C_{clut} = \begin{cases} 0 & \text{if } n = 0 \\ \infty & \text{if } \exists i : d_i < r \\ C_{max} \cdot \min \left\{ 1, \prod_{i=1}^n (E_i + 1) - 1 \right\} & \text{otherwise} \end{cases} \quad (2)$$

where $E_i = e^{k \cdot (r - d_i)}$ is the same exponential decay function as used above in Eq. (1). In this formula, we add 1 to E_i so that far-away objects with $E_i \approx 0$ turn into the neutral element in the multiplication. After the multiplication, we have to subtract 1 again so that the cost function approaches 0 when the distances go to infinity.

In addition to the requirements defined above, the function also has the following properties:

- The product can be computed incrementally as

$$E_{new} = (E_{old} + 1) \cdot (E_j + 1) - 1 \quad (3)$$

when a new object j is observed, which allows for efficient implementation.

- In contrast to the original function Eq. (1), it is continuously differentiable in the areas the robot can traverse, which is required by some planning algorithms. Like the original function, it is not differentiable at the borders between free space and obstacles.

Fig. 2 shows a comparison of the standard cost function (red) and our approach (blue) in a simple environment containing two obstacles.

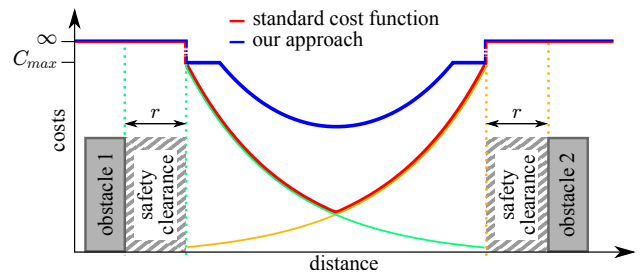


Fig. 2. Cost function for a simple environment containing two obstacles. The green and yellow curves represent the exponentially decaying costs for obstacle 1 and 2, respectively. These costs encourage the planner to keep a clearance to obstacles in addition to the safety distance. While the standard approach takes the maximum of those curves as the cost function (red), our approach combines the curves according to Eq. (2), which leads to higher costs in cluttered areas.

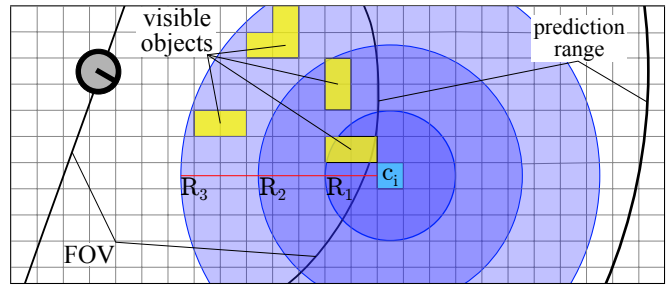


Fig. 3. Our framework predicts an average clutter density for cells c_i in close but not yet observed areas by calculating densities within the radii R_j around c_i based on occupied cells corresponding to non-static obstacles.

IV. CLUTTER DENSITY AND COST PREDICTION

The cost function defined in the section above can be applied to determine the costs in regions the robot has already observed. However, we would like to also predict costs corresponding to potential objects in adjacent regions that are not yet within the field of view or occluded by other objects. To do so, we estimate local obstacle densities depending on the already observed objects and increase the traversal costs for cells in close-by regions that are not yet visible.

To allow for efficient computation of the object density, we calculate the observed occupation density inside a circle of radius R_j for each cell c_i within the prediction region. We perform this calculation for M circles with different radii as shown in Fig. 3. The predicted density ρ_i is then found by summing over all M density estimates, weighted by an exponential factor that decreases with R ,

$$\rho_i = \frac{\sum_{j=1}^M e^{-R_j} \frac{o_{i,j}}{t_j}}{\sum_{j=1}^M e^{-R_j}}, \quad (4)$$

where t_j is the total count of cells and $o_{i,j}$ is the count of occupied cells around c_i , both inside the circle with radius R_j . The cost $C_{predict}$ for each cell c_i is then defined as

$$C_{predict} = \alpha \cdot \rho_i, \quad (5)$$

where α has to be chosen such that clutter densities that raise navigation difficulties lead to predicted costs close to C_{max} . In the future, we will extend our framework to include strategies to learn about its best value.

V. EXPERIMENTS

We implemented our approach in ROS and evaluated it both in simulation (Gazebo [10]) and in real-world experiments. The practical experiments were performed using a Robotino robot from Festo Didactics with a ASUS Xtion Pro Live fixed to the mounting tower for obstacle detection and a Sick-S300 laser scanner mounted horizontally above the ground for localization. For detecting objects, the robot uses data from the RGBD camera to build a point cloud of the environment. All points above the ground plane are considered as obstacles. Points that cannot be explained by known obstacles contained in the given environment representation are classified as corresponding to cluttered objects. In our experiments, the prediction region is located from 2.5 m up to 5 m around the front half of the robot. The radii of the circles to compute the occupation density are 0.5 m, 1.0 m, and 1.5 m.

A. Path Planning and Trajectory Execution

We use a grid-based A* planner to find a global plan using the predicted costs described in Sec. IV. The robot then executes the planned trajectory using the default local planner included in ROS. This trajectory rollout planner samples velocities in the robot’s control space, simulates and evaluates the robot’s path in a short lookahead time frame, and sends the highest-rated velocity command to the robot (see [11] for details). By sampling and evaluating in the robot’s control space, the planner is able to adapt the robot’s velocities to the constraints imposed by the environment. In particular, the robot will drive slower in cluttered areas as it has to respect acceleration limits when turning or evading obstacles.

B. Quantitative Evaluation

We evaluated the impact of the amount of clutter on the completion time in a series of simulation experiments with varying obstacle densities for both the standard technique without modifications and our approach with the modified cost function and prediction. We randomly sampled objects within a rectangular area of size $23 \times 8 \text{ m}^2$. The clutter area is surrounded by free space leaving the robot the possibility to drive around it. As clutter objects, we used boxes with lengths varying from 0.3 to 1.2 m (see Fig. 6 for an example map). For the sake of comparison, we define a parameter for the obstacle density D_c in the clutter area as the average number of objects that appear in a region of one square meter.

The task of the robot was to navigate to a goal point at a distance of 28 m at the opposite end of the arena with the cluttered area in between. In this scenario, the robot cannot get stuck in the clutter area as there is always a possible path to the goal, the robot can rotate on the spot and drive back the path that it came from, and the simulator provides error-free sensor measurements. For each D_c , we created 20 randomly cluttered environments with a variety of objects. We averaged the results over two runs with different start and goal positions for each generated environment and evaluated the average travel time for reaching the goal. Fig. 4 shows the mean and 95% confidence interval of the overall traveling time in relation to the density of the clutter region.

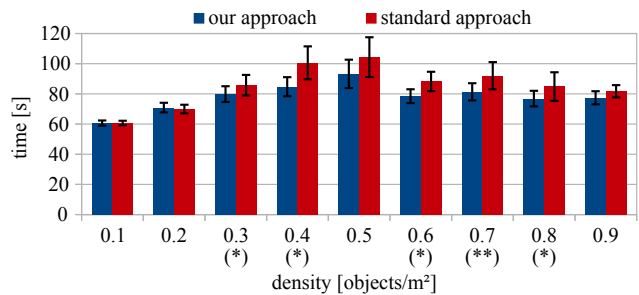


Fig. 4. Mean and 95% confidence interval of the traveling time in relation to the amount of clutter. For low and high clutter densities, both approaches behave similarly. In the middle range, our approach leads to shorter completion times. For the densities marked with (*) and (**), our approach performs significantly better than the standard approach according to a paired t-test at the 0.05 and 0.001 level, respectively.

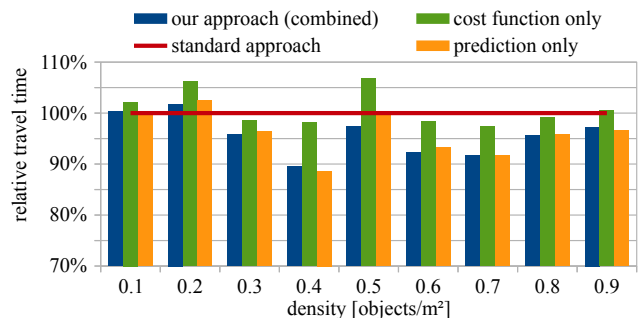


Fig. 5. Average of the traveling time in comparison to the standard approach broken down by the individual components of our system. Overall, the combination of our cost function and prediction performs best. See Sec. V-B for a detailed discussion of the results. Note that the vertical axis starts at 70% to improve readability.

For the extreme values $D_c = 0$ and $D_c \rightarrow \infty$, both approaches yield the same behavior by definition: If no clutter obstacles are present, both approaches will choose the shortest path without being impeded, thus both the mean and variance of the travel time are low. As the clutter density increases, the completion time generally increases as the robot has to take longer paths to avoid obstacles. If the clutter is too dense for the robot to fit through, then both approaches will plan a detour around the whole area, again leading to similar completion times. The approaches behave differently when the clutter is sparse enough for the robot to drive through, but dense enough to impede the robot, i.e., for values of D_c between 0.3 and 0.8. In this case, the robot has to decide whether to drive through the clutter area or to take a detour around it. Driving through the clutter leads to shorter trajectories, but the robot has to drive slower near obstacles and the risk of driving into a dead end and having to backtrack increases. As the results show, our approach finds a good trade-off and achieves shorter completion times in all cases. For the densities marked with (*) and (**), the difference is statistically significant according to a paired t-test at the 0.05 and 0.001 levels, respectively. For densities below 0.5, driving through the clutter is usually the favorable option, while for densities above 0.5 taking a detour around groups of obstacles is often faster. For densities around 0.5, the completion time strongly depends on the geometrical distribution of the clutter, as dead ends and maze-like structures occur

frequently, making it hard to predict which path will lead to the goal. Hence, both mean and variance increase for both approaches. While our approach still performs better on average, the difference is not statistically significant due to the high variance. On average, updating the cost map after new sensor readings takes 10.8 ± 4.0 ms for our approach compared to 5.9 ± 2.3 ms for the standard approach.

To assess the contribution of the individual components, we report the results in Fig. 5 separately for a planner using only the cost function from Sec. III-B (green), a planner using only the prediction method from Sec. IV (yellow), and a planner using the combination of both (blue). For better comparison, we normalized the times so that the standard approach matches 100%.

The main effect of the cost function introduced in Sec. III-B is to increase the costs for squeezing through narrow passages between obstacles. Such navigation maneuvers slow down the robot and they are risky, as sensing errors might lead to collisions or trap the robot in situations where its planner cannot find a valid path anymore. As the results from Fig. 5 show, the cost function on its own increases the completion time compared to the standard approach for low clutter densities, because the planner will keep a bigger distance to groups of obstacles. In combination with the clutter prediction, however, it decreases the completion time.

Overall, the combination of our cost function and the clutter prediction performs best. In some cases, the prediction alone performs marginally better than the combination, as the cost function increases the clearance between the robot and groups of obstacles, leading to slightly longer trajectories.

In the next section, we will discuss the influence of the individual parts in more detail based on exemplary situations.

C. Qualitative Evaluation

Fig. 6 shows a typical simulation result. The task of the robot was to navigate from the left side to the right side of the room. The center of the room is filled with clutter objects that are not contained in the robot’s map. In the depicted situation, the robot has traveled about half the way to the goal. The standard approach without the modified cost function and prediction (top) tries to follow the shortest path to the goal, which leads the robot through dense clutter. As the robot has to respect acceleration limits when turning and evading obstacles, the robot has to slow down. It arrives at the goal after 102 s. Our approach, by contrast, predicts that there are probably more objects behind the obstacles that gradually appear in its field of view while driving. The prediction and the modified cost function increase the costs towards the center of the clutter area as well as in between the objects, inciting the planner to choose the longer, but safer option of making a detour around the obstructed area. Even though the traveled distance of our approach is 11% longer, the robot arrives at the goal 29% earlier after 72 s.

As our algorithm predicts clutter in regions that the robot has not observed yet, it has to update the prediction, based on real measurements, once the robot travels through previously unknown regions and observes the actual scene. Fig. 6 also

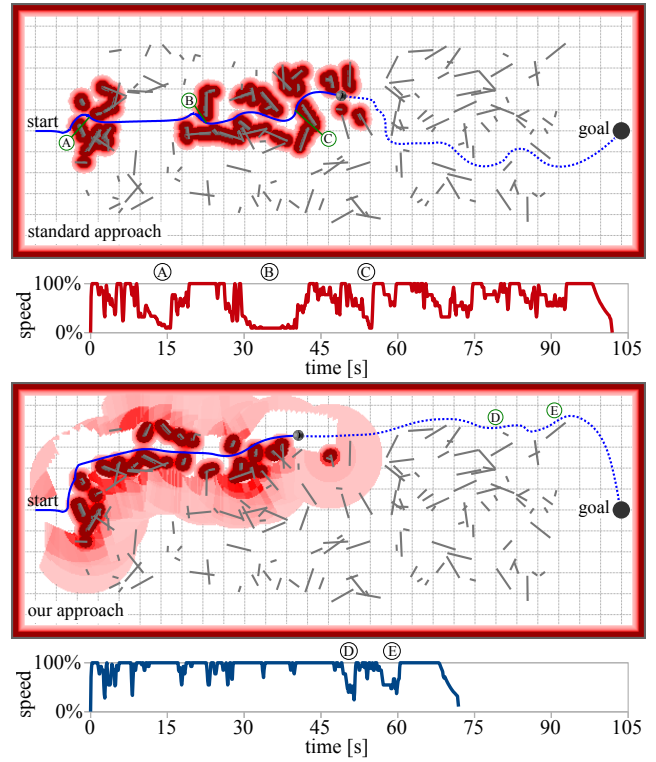


Fig. 6. Comparison of the two approaches on an example map ($D_c = 0.35$) and velocity profiles in percent of the maximum allowable speed. The standard approach (top) tries to follow the shortest path to the goal, leading the robot through dense clutter where the robot has to slow down to avoid collisions. With our approach (bottom), the costs increase in locations where clutter is predicted, driving the robot around the clutter field and leading to a shorter completion time.

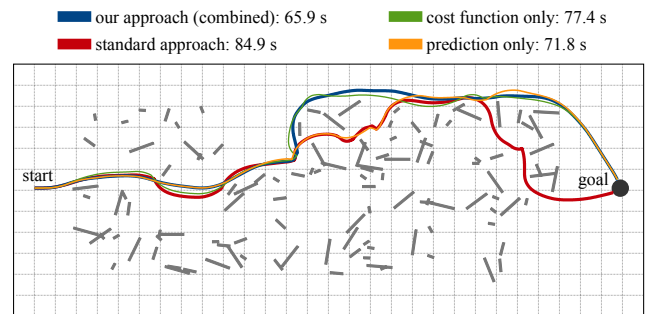


Fig. 7. Comparison of the individual components of our system on an example map ($D_c = 0.25$). The standard approach loses time by squeezing through narrow gaps. The prediction incites the planner to drive around dense clutter. The cost function prevents the robot from driving through narrow passages. Our combined approach leads to the shortest time.

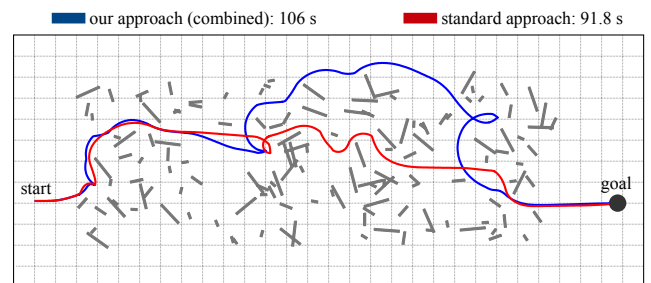


Fig. 8. Our approach chooses the conservative option of driving around the dense clutter in the center. The standard approach drives through the narrow passage and arrives at the goal earlier.

shows how the robot clears the prediction for areas that it has observed as being freespace, e.g., in the center top area.

Fig. 7 shows the effects of the individual components of our system on a sample map. The standard approach loses time as it has to slow down near obstacles. The prediction incites the planner to drive around dense clutter, but it still drives through narrow passages. In the combined approach, the cost function additionally prevents the robot from entering cluttered regions through narrow passages, hence the combined approach yields the shortest completion time.

In Fig. 8, our approach evades the dense region in the center of the map and prefers the less dense area in the top. As there is no guarantee that predicted obstacles exist, in this case the clutter avoiding strategy leads to an unnecessary detour. On average, however, the advantages of the prediction outweigh the risk of superfluous detours.

D. Real-World Experiments

We conducted real-world experiments using the Robotino robot in the environment shown in Fig. 9. The environment is surrounded by walls and contains additional walls separating a corridor and multiple rooms. Additionally, we placed several obstacles on one side of the environment. While the robot has access to a map containing the walls, it does not know the amount or distribution of the clutter objects beforehand.

The standard approach computed a full path through the cluttered region, which is displayed on the left in Fig. 9. Although the robot successfully navigated through the clutter, the robot needed to slow down on several occasions to change its orientation and avoid collisions leading to a total navigation time of 45.5 s.

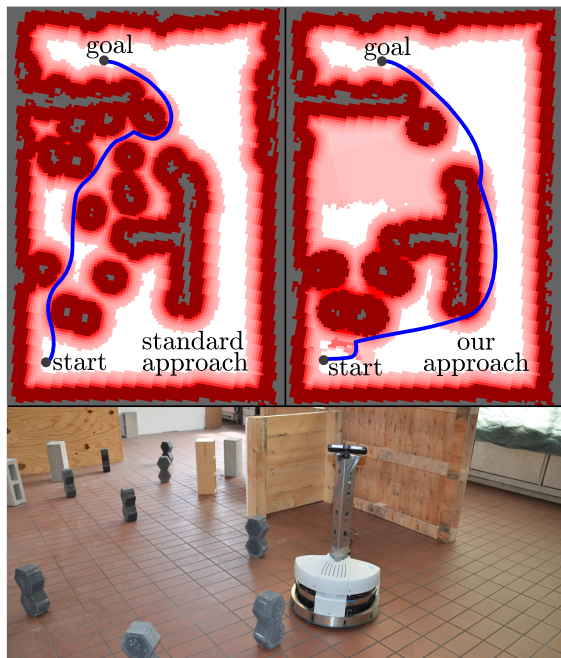


Fig. 9. Experiment with the real Robotino traveling through a cluttered scene. The standard approach computes a path through the clutter, where the navigation is slowed down. Our approach predicts higher traversal costs leading to an early replan around the cluttered region. The displayed costmaps correspond to the time after the robot has reached the corresponding goals.

Our approach started with a similar navigation plan, however, once the robot discovered the nearby clutter at the start, the increased predicted costs in the cluttered region immediately led to a replanned path around the corridor. This path is shown on the right in Fig. 9. While resulting in a longer path, the total travel time was only 34 s as the robot was able to drive faster due to the free space on the right corridor.

VI. CONCLUSIONS

In this paper, we proposed a novel solution to efficient navigation through environments containing areas with many cluttered objects. Our approach predicts traversal costs resulting from potential obstacles in regions that are not yet observable by the robot’s sensors. The prediction is based on estimated obstacle densities from already detected objects. By considering the predicted traversal costs directly for path planning, the robot navigates foresightedly and avoids regions that are likely to be too cluttered for the robot to easily pass through.

We implemented our system within ROS and represented the predicted traversal costs as a new cost map layer for path planning. As we demonstrate in the experiments with a wheeled robot equipped with a depth sensor, in several situations the resulting navigation behavior significantly outperforms the one generated by a standard path planner that only considers detected objects.

REFERENCES

- [1] D. Lu, D. Hershberger, and W. Smart, “Layered costmaps for context-sensitive navigation,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots & Systems (IROS)*, Sept 2014, pp. 709–715.
- [2] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, “The office marathon: Robust navigation in an indoor office environment,” in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2010.
- [3] A. Hornung, M. Phillips, E. G. Jones, M. Bennewitz, M. Likhachev, and S. Chitta, “Navigation in three-dimensional cluttered environments for mobile manipulation,” in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2012.
- [4] A. Orthey and O. Stasse, “Towards reactive whole-body motion planning in cluttered environments by precomputing feasible motion spaces,” in *Proc. of the IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2013.
- [5] D. Maier, C. Lutz, and M. Bennewitz, “Integrated perception, mapping, and footstep planning for humanoid navigation among 3D obstacles,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots & Systems (IROS)*, 2013.
- [6] D. Joho, G. Tipaldi, N. Engelhard, C. Stachniss, and W. Burgard, “Nonparametric Bayesian models for unsupervised scene analysis and reconstruction,” in *Proc. of Robotics: Science and Systems (RSS)*, 2012.
- [7] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky, “Describing visual scenes using transformed objects and parts,” *Int. Journal of Computer Vision*, vol. 77, no. 1–3, pp. 291–330, 2008.
- [8] P. Henry, C. Vollmer, B. Ferris, and D. Fox, “Learning to navigate through crowded environments,” in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2010.
- [9] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard, “A navigation system for robots operating in crowded urban environments,” in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2013.
- [10] N. Koenig and A. Howard, “Design and use paradigms for Gazebo, an open-source multi-robot simulator,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots & Systems (IROS)*, 2004.
- [11] B. P. Gerkey and K. Konolige, “Planning and control in unstructured terrain,” in *Proc. of the ICRA Workshop on Path Planning on Costmaps*, 2008.