Efficient Path Planning in Belief Space for Safe Navigation

Robert Schirmer[1] Peter Biber[2] Cyrill Stachniss[3]

***Abstract*— Robotic lawn-mowers are required to stay within a predefined working area, otherwise they may drive into a pond or on the street. This turns navigation and path planning into safety critical components. If we consider using SLAM techniques in that context, we must be able to provide safety guarantees in the presence of sensor/actuator noise and featureless areas in the environment. In this paper, we tackle the problem of planning a path that maximizes robot safety while navigating inside the working area and under the constraints of limited computing resources and cheap sensors. Our approach uses a map of the environment to estimate localizability at all locations, and it uses these estimates to search for a path from start to goal in belief space using an extended heuristic search algorithm. We implemented our approach using C++ and ROS and thoroughly tested it on simulation data recorded on eight different gardens, as well as on a real robot. The experiments presented in this paper show that our approach leads to short computation times and short paths while maximizing robot safety under certain assumptions.**

I. INTRODUCTION

Navigation is a critical component for robotic lawn-mowers as they are meant to function without supervision for long periods of time on a wide variety of gardens while never posing a threat to humans or destroying themselves. The safety-relevant worst-case scenario that should be avoided at all costs is leaving the working area (i.e., the customer's lawn), as erratic behaviour outside this protected space might lead to harm or injury.

Navigation in lawn border areas on most current robotic lawn-mowers relies on a physical perimeter wire laid along the borders and obstacles. The wire can be sensed with precision and robustness from a short range, which means the robot will sense it before leaving the garden. The main drawback of the perimeter wire is the inconvenience to the customer: it is tedious to set up while respecting all instructions, and eventual mistakes are difficult to correct. It is, therefore, the "Holy Grail" of lawn-mower navigation to make the product more customer friendly by getting rid of that perimeter wire while respecting the hard constraints set on service robotics: product safety and price competitiveness. One approach towards that goal is to equip the robot with a cheap 2D laser scanner and then navigate using Simultaneous Localization and Mapping (SLAM) techniques.

[1]Robert Schirmer is a PhD student at the University of Bonn and is working at Robert Bosch GmbH, Corporate Research, Germany. robert.schirmer@de.bosch.com

[2]Peter Biber is with Robert Bosch GmbH, Corporate Research, Germany. peter.biber@de.bosch.com

[3]Cyrill Stachniss is with the Institute of Geodesy and Geoinformation, University of Bonn, Germany. cyrill.stachniss@igg.uni-bonn.de

This work has partly been supported by the European Commission under the Horizon 2020 framework programme under grant agreement no. 645403.

Fig. 1: Motivating scenario: A robotic lawn-mower needs to plan paths maximizing the probability of the robot staying localized while navigating. Top: A simulated lawn-mower in a reference garden. Bottom: a projection of obstacles and taught-in working area (red) on a gridmap.

The requirement of not leaving the working area unintentionally needs to be handled at the path planning level, as featureless areas should be handled specifically or in extreme cases avoided altogether. In this paper, we concentrate on the problem of path planning under uncertainty from point A to point B, particularly focusing on maximizing the probability that the lawn-mower does not collide with obstacles or taught-in borders by planning paths considering the evolution of uncertainty. Besides coverage, maneuvering from A to B is important for robotic lawn-mowers as they often need to plot safe paths to the charging station or unmowed areas. The robot we are considering is a modified Bosch Indego lawn-mower equipped with a 2D laser scanner deployed in a garden as sketched in Fig. 1 (top). Certain obstacles such as low bushes cannot be reliably detected by the lawn-mower, so the customer has to teach the borders of every obstacle and limits of the garden manually. This results in a map of visible landmarks and invisible borders seen in Fig. 1 (bottom). While the application scenario is very specific, it should be noted that none of the methods or results are restricted to lawn-mowers.

Planning under uncertainty or in belief space differs fundamentally from planning without uncertainty where the state of the robot is always assumed to be known precisely. In uncertainty-aware planning, the robot cannot directly observe its state but can only infer it from past observations and actions. This leads to the necessity of maintaining a probability distribution over possible states of the robot called the belief and computing a control policy to select the best actions. This problem can be formalized as a partially observable Markov decision process (POMDP), which in the general case quickly becomes computationally intractable for real world problems.

The main contribution of this paper is the development of a safe path planning algorithm in belief space for laser scanner equipped mobile robots with a special focus on robotic lawnmowers. We simulate the localizability (or expected to be gained information) of the laser scanner at all locations. We introduce a heuristic search algorithm that uses the localizability map and other assumptions (e.g., speed, discretization of possible positions, dominance relations between different beliefs, etc.) to search in belief space and compute paths maximizing the probability that the robot stays localized while navigating. The algorithm to search in belief space can be instantiated with different evaluation functions / dominance relations, letting us investigate how they influence the search on realistic test sets.

Summarizing, we make the following key claims:

(i) Our approach computes similar solutions to existing techniques that incorporate the uncertainty in the path planning process in a similar way, but does so in a smaller amount of time.

(ii) Our approach investigates the effect of different evaluation functions / dominance relations on search algorithms in belief space on realistic test sets.

(iii) Our approach can be executed on a real robot.

These claims are backed up by the paper and especially in our experimental evaluation.

II. RELATED WORK

Research for motion planning under uncertainty is often concerned with making efficient assumptions to and discretizations of the belief space in order to render the resulting POMDP solvable for problem instances of the required size. Therefore, state-of-the-art approaches can be categorized by the considered uncertainty, e.g., uncertainty from movement (noise from actuators), uncertainty from localization (noise from sensors), unknown future motion of dynamic obstacles and uncertainty in the map.

We will mainly focus on approaches taking into account uncertainty from localization and movement, as these come closest to the problem we are treating. Those have in common that the belief dynamics of the POMDP are computed using a Bayesian Filter. The approach by Prentice *et al.* [15] plans in belief space by extending Probabilistic Roadmaps. The approach by Van den Berg *et al.* [18] uses Rapidly exploring Random Trees (RRT) to plan a large amount of candidate paths that are then tested using a simulated Linear-Quadratic-Gaussian controller. The approaches of Bry *et al.* [4] and Lenz *et al.* [13] aim at planning the path of a robot using Rapidly Exploring Random Belief Trees. These approaches differ from ours through the chosen motion and sensor models: The authors assume to have easily linearizable line-of-sight or beacon localization sensors while our approach focuses on laser scanners. They also use higher-dimensional state spaces better suited to model UAVs or car-like robots but slowing computation because of the curse of dimensionality. Lambert *et al.* [12] and the subsequent RRT based extension by Pepy *et al.* [14] present a Bayesian framework for planning in an extended *pose* $\times$ *covariance* space that takes sensing and motion uncertainty into account. They use the sensor model described in Lambert *et al.* [11] which simulates the reaction of sensors to the environment. Censi *et al.* [9] use the same representation but differ in using an information space approach for the description of the robot belief. They also introduce other search algorithms such as a backward search that provides reusable plans. We will compare the approach proposed in this paper to the forward-search algorithm by Censi *et al.*, as our assumptions are similar and the authors have provided the most comprehensive experiments for their results. Bopardikar *et al.* [3] further discretize the belief space from Lambert *et al.* [12] by using a bound on the maximum eigenvalue of the covariance and use it to solve multi-objective optimization problems. The approach by Agha-Mohammadi *et al.* [1] breaks the curse of history inherent to POMDPs by using local feedback controllers enabling fast belief space replanning. The strong assumptions (observability and controllability of the system) restrict the method to sensor models not applicable to laser scanners. The approach of Carrillo *et al.* [5] specifically emphasizes the problem of path planning using active SLAM concepts but limits possible paths to those taken during the mapping process.

When one considers non-linear sensor models like we require in the case of laser scanners, studying how different external sensors influence the achievable accuracy for robot localization becomes crucial. Censi [7], [8] and Bengtsson *et al.* [2] study how much localization-related information scan-matching algorithms are expected to provide in different environments and how it affects the localizability of the robot. Incorporating this research into our own approach is important as it gives precious information about where the robot might risk collisions and how to avoid them.

III. PROBLEM DESCRIPTION

We address the problem of path planning in belief space which involves finding short paths that avoid obstacles, while maximizing the probability of staying localized during navigation along the path. We consider the uncertainties from motion and from sensing, while assuming the map issued by a SLAM algorithm to be largely free of gross errors and the world to be static. The map-building process can be considered to be error free because the initialization of the product i.e., map-building and teach-in of borders, is supervised by the customer.

While our test datasets recorded on real gardens are non-planar, we project obstacles and a circular hull of the robot footprint to the ground-plane, which leads us to a two-dimensional configuration space similar to [12]. For a mobile robot moving in a static world, the belief space is defined as the set of all possible probability distributions over the configurations of the robot. We make the assumptions of Gaussianity and possible linearization of robot dynamics, which means that the robot belief is described by a bivariate Gaussian distribution with mean μ and covariance Σ. At planning time, we forward-simulate the evolution of the robot belief as it advances and gains information on its state from sensors and loses it through odometric drift recursively using an Extended Kalman Filter (EKF), as described in Thrun *et al.* [17]. We can consider the belief space to be a *poses* $\times$ *covariances* space. This space is infinite as every path leads to a different belief. To make the computations lighter, we further discretize the belief space by only considering beliefs centered at regular intervals of a grid and by discounting the orientation of the robot.

IV. OUR APPROACH

To solve our belief space motion planning problem in the lawn-mower setting, we propose a two-step method using the belief dynamics and assumptions described in Sec. III. The first step is the creation of a localization performance model for a 2D laser scanner sensor, which we describe in Sec. IV-A. The second step of our approach involves searching for a path in belief space by using a modified heuristic search algorithm, see Sec. IV-B.

A. Localizability Map

When a robot estimates its position against a given map by scan-matching, it registrates the currently acquired point cloud given by the sensor against the map and calculates a transformation returning the most probable location. Formally, the scan-matching optimizes $p(x_t|x_{t-1}, u_t, z_t, m)$ where x_t is the distribution of possible robot poses at time t, z_t are the sensor measurements, m is the map and u_t is the odometry/command given to the robot.

To estimate the effectiveness of scan-matching based localization on any point of the map, we propose to simulate a laser scanner with the help of a ground truth point cloud recorded using a Leica 3D point scanner and measure the convergence radius of a Normal Distributions Transform (NDT)-based registration algorithm using the approach by Bengtsson *et al.* [2]. To summarize the approach for a given position, we first simulate a laser scan using a ground truth point cloud and a model of the laser scanner. We then repeatedly translate/rotate the scan by a certain distance within a tested convergence radius. Finally, we register the translated simulated laser scans with a SLAM-map of the environment in order to compute the covariance of converged-to locations. If the resulting covariance is small, this means that the sensed part of the environment possesses a lot of useful features (e.g. the presence of a corner) and the algorithm was able to recover the actual location of the laser scan. If the convergence radius is big, this means that the sensed part of the environment does not include many useful features. We do this on every point of the map with a given resolution and consider the result to be an accurate estimate of an EKF update-step. An extract from the localizability map computed for the motivation scenario can be seen in Fig. 2 (top), while a semantic view is shown in Fig. 2 (bottom). We have chosen a maximum scanning range of $4\,\mathrm{m}$ to mimic a low-cost range sensors that can be used on a lawn-mower. We assume the noise in the range measurement to be Gaussian with a variance of $3\,\mathrm{cm}$, and test for a convergence diameter of $1\,\mathrm{m}$. Currently, we consider 1850 readings per scan. Fig. 2 (top) shows how the localizability near corners is good in both x and y dimensions, while the detection of only one wall leads to good localizabilty in only one dimension. Bushes (large black masses on the gridmap) are challenging for localization, as unevenness of the ground leads to different parts of the bush being scanned which in turn causes greater uncertainty.

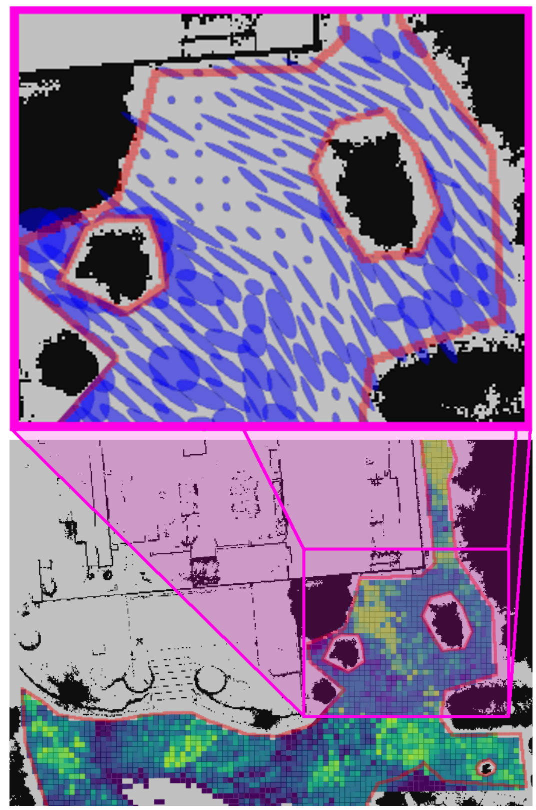

Fig. 2: Localizability map: Top: An extract of a localizability map for the motivational problem. Each blue ellipse corresponds to 99.5% ellipses of expected EKF update steps. Ellipse axes of $1\,\mathrm{m}$ are found in dimensions where no information is present. Bottom: semantic view of a localizability map for a larger area. Yellow values indicate good localizability (sum of eigenvalues $\leq 10\,\mathrm{cm}$) and dark purple tones indicate bad localizability (sum of eigenvalues $\geq 200\,\mathrm{cm}$). The grey areas are either featureless or outside the working area.

B. Heuristic Search Algorithm

The previous sections provide a method for simulating the evolution of path uncertainty in belief space: we can transition from one belief to the next using an EKF with an odometric drift prediction-step and an expected sensor update-step from the localizability map.

To complete the path-planning approach, we use a generic search algorithm extended to work in belief space to find a path to the goal that optimizes different aspects, depending on the chosen evaluation function. We adapt the notation and description of the search algorithm from Censi *et al.* [9].

Algorithm 1 SEARCH ALGORITHM IN BELIEF SPACE

VISITED: the set of visited nodes
OPEN: the set of opened nodes, ordered by $\blacktriangleleft$-relation
Put n_{start} in OPEN
while OPEN is not empty **do**
    Pop first (according to $\blacktriangleleft$) node n from OPEN
    for all s in SUCCESSORS(n) **do**
        Return s if IS_GOAL(s)
        if s is $\trianglerighteq$-dominated in VISITED **then**
            Ignore s
        else
            Put s in VISITED
            Put s in OPEN
Report failure

A node n is a tuple $\langle x, \Sigma, d\rangle$ that encodes the expected evolution of the belief on a path from the starting position to the mean of the node, x. The intuitive meaning is: "There exists a path from the start to x with length d and covariance Σ". The function SUCCESSORS(n) generates successor nodes to all neighbours (adjacent grid cells) using the belief dynamics equation of the EKF and rejects those that lead to collisions.

The generic search algorithm described in Alg. 1 aims at computing the path encoded in node $n_{return} = \langle goal, \Sigma_{return}, d_{return}\rangle$ ranked as the first element according to the $\blacktriangleleft$-relation, the evaluation function. It does so by greedily expanding the node dominating the others according to $\blacktriangleleft$ in OPEN and verifying whether it can discard any new incoming nodes using the dominance $\trianglerighteq$-relation. This leads to provably optimal solutions if $\blacktriangleleft$ is admissible, such as the sum of the euclidean distance to the goal and d when the problem is set in the configuration space.

The main part of our work focuses on investigating the influence of different evaluation functions and dominances on solution quality and computation time. The following subsections present the evaluation functions and dominance relations we use to compare two nodes: $n_1 = \langle x_1, \Sigma_1, d_1\rangle$ and $n_2 = \langle x_2, \Sigma_2, d_2\rangle$.

1) Dominance Relation: A dominance relation defined in Eq. (1) is an ordering over nodes used to decide whether one node will always lead to a better solution than another. We use it to prune parts of the search space that cannot lead to good solutions, such as longer paths with worse robot localization going to the same position.

$$(n_1 \trianglerighteq n_2) \Leftrightarrow (x_1 = x_2 \wedge \Sigma_1 \leq \Sigma_2 \wedge d_1 \leq d_2) \tag{1}$$

Comparing the state x and distance d is fairly straightforward, but the covariance Σ is more difficult because of its multi-dimensionality. The importance of using a dominance relation is shown in Fig. 3 (left). By ignoring nodes that encode paths with worse localization than the blue one, e.g. the dashed green path, we prune large parts of the search space while not losing any solution quality. Some authors such as Censi *et al.* in [9] reject nodes using a FULL SUBSUMPTION test: $\Sigma_1 \leq \Sigma_2$ *iff* $\Sigma_1 - \Sigma_2$ *is negative semidefinite*. Other authors such as Bry *et al.* [4] compare the TRACE. This leads to faster computations as it is a total ordering (it can always be decided whether $\Sigma_1 \leq \Sigma_2$) but comes at the cost of losing completeness as illustrated in Fig. 3 (right). To our knowledge, this is the first attempt to compare these two dominance relations on extensive test sets.

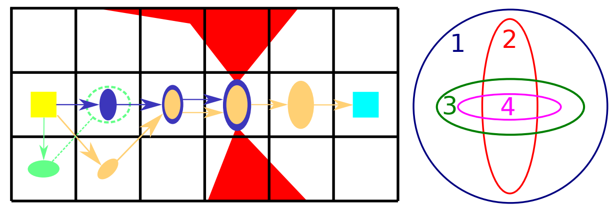

Fig. 3: Dominance relations. (Left) demonstrates how a dominance relation in belief space must consider the covariance. In order to find a solution and pass the red obstacle, the safer but longer beige path must be considered. The dashed green path has a worse localization than the blue one and can safely be pruned from the search. (Right) illustrates how the partial ordering over covariances induced by FULL SUBSUMPTION leads two minima (*4* and *2*) to be kept in the search. TRACE induces a total ordering that only keeps one (*4*).

2) Evaluation Function: An evaluation function is a function f as seen in Eq. (2) used to rank the nodes in OPEN that defines a total ordering over nodes. Evaluation functions can use different criteria to rank nodes and in the present context usually do it by covered distance, accumulated uncertainty, or a weighted combination of both.

$$(n_1 \blacktriangleleft n_2) \Leftrightarrow f(n_1) \leq f(n_2) \tag{2}$$

Approaches such as Censi *et al.* in [9] rank the nodes by DISTANCE: here, the distance traveled d is added to a heuristic value h predicting the distance to the goal: $f(n) = d(n) + h(n, goal)$. Comparable approaches in belief space set $h(n, goal) = \text{EUCLIDEAN}(n, goal)$. We propose to use Dijkstra's algorithm over the lower dimensional 2D configuration space to compute the shortest path to the goal not considering uncertainty, which we will refer to as $\text{DIJKSTRA}(n, goal)$. Both heuristics are admissible and consistent, albeit only for the distance, meaning that the computed paths will be optimal distance-wise. Using the result of lower-dimensional planners as a heuristic for use in high-dimensional motion planners is a widely-followed approach, e.g. Stachniss *et al.* [16], but this work is the first to use it for planning in belief space to the knowledge of the authors.

Another approach Carrillo *et al.* [5] ranks the nodes by the accumulated uncertainty over the path. The authors use metrics developed for statistical testing that are used in robotics for active SLAM that are solely computed over the covariance. The metric we have chosen to quantify the uncertainty of a belief is a variant of D-OPT given by Eq. (3), with $\lambda_{1,2}(\Sigma)$ the eigenvalues of Σ considering the x, y-space. Carrillo *et al.* [6] discuss D-OPT and other metrics in-depth and come to the conclusion that this version of D-OPT presents the best qualities for dead-reckoning mobile-robot scenarios. We define our UNCERTAINTY evaluation function as $f = \sum_i^n \text{D-OPT}(\Sigma_i)$, or the sum of all $\text{D-OPT}(\Sigma)$ over the path culminating at n, which we will also denote as $\text{D-OPT}(n)$.

$$\text{D-OPT}(\Sigma) = \frac{1}{2}(\exp(\log \lambda_1(\Sigma) + \log \lambda_2(\Sigma))) \tag{3}$$

Finally, weighted evaluation functions take both the path length and the accumulated uncertainty over the path into account. Costante *et al.* [10] combine an UNCERTAINTY metric $u(n)$ and a DISTANCE metric $v(n)$ using a parameter α weighing the advantages of a shorter path with the disadvantage of bigger covariances, $f(n) = \alpha\, u(n) + (1-\alpha)\, v(n)$.

We avoid the problem of setting α explicitly by comparing the current node to an ideal solution. The ideal path length L_{ideal} = DIJKSTRA*(start,goal)* is the shortest collision-free path to the goal. The ideal localizability C_{ideal} is a small covariance c_{ideal} over the whole ideal path length $C_{ideal} = L_{ideal} \times \text{D-OPT}(c_{ideal})$. We thus define the WEIGHTED evaluation function as f = DISTANCE SCORE(n) + UNCERTAINTY SCORE(n) with DISTANCE SCORE and UNCERTAINTY SCORE defined in Eq. (4). This multi-optimizing evaluation penalizes nodes that are further from the ideal path and privileges nodes that are on a short path with low covariance.

$$\begin{aligned} \text{DISTANCE SCORE} &= \text{DIJKSTRA}(n, goal) + d(n) - L_{ideal} \\ \text{UNCERTAINTY SCORE} &= \text{D-OPT}(c_{ideal}) \times \text{DIJKSTRA}(n, goal) \\ &\quad + \text{D-OPT}(n) - C_{ideal} \end{aligned} \tag{4}$$

An overview of the different evaluation functions and dominances we use can be seen in Tab. I, the weighted algorithms use $c_{ideal} = 20\,\text{cm}$.

Abbreviation	**Dominance $\trianglerighteq$**	**Evaluation Function ◄**
FSE	FULL SUBSUMPTION	EUCLIDEAN + d
FSD	FULL SUBSUMPTION	DIJKSTRA + d
FSDOPT	FULL SUBSUMPTION	D-OPT
FSW	FULL SUBSUMPTION	WEIGHTED
TE	TRACE	EUCLIDEAN + d
TD	TRACE	DIJKSTRA + d
TDOPT	TRACE	D-OPT
TW	TRACE	WEIGHTED

TABLE I: Overview of tested orderings.

V. EXPERIMENTAL EVALUATION

The goal of this work is to provide a fast and efficient algorithm for path planning under uncertainty. Our experiments are designed to show the capabilities of our method and to support the three claims we made in the introduction. Our approach (i) computes similar solutions to existing techniques that incorporate the uncertainty in the path planning process in a similar way but does so in a smaller amount of time, (ii) investigates the effect of different evaluation functions / dominance relations on search algorithms in belief space on realistic test sets and (iii) can be executed on a real robot. All computations are performed using a prototypical implementation of the algorithms in C++ running ROS on a Core i7 CPU @2.8 GHz.

A. Comparison with the State of the Art

The first set of experiments is designed to support claim (i) and shows how our approach leads to smaller computation times in the same scenario as Censi *et al.* [9]. The comparison to their approach is particularly relevant as they published precise results on a specific problem set and are using similar assumptions (e.g., two dimensional state, Gaussian assumption, comparable motion model, etc.). The problem set they provide is depicted in Fig. 4 and has a very specific structure that favours orderings quickly seeking information. The robot starts with a large state uncertainty and is equipped with a simplified four-sampled North, South, West, East (*N,S,W,E*) range-finder, meaning the path must go through the enclave on the bottom left (area 1) in order to go through the needle-hole that is further to the right (area 2). The performance measures we use to quantify the results are:

- NC (Nodes Created) measures how many nodes were created during the search and estimates complexity.
- MM (Maximum in Memory) measures the most nodes present in memory at any given time and showcases memory use.
- CT (Computation Time s) measures runtime and includes the computation of heuristic costs.
- PL (Path Length m) denotes the length of the path.
- PU (Path Uncertainty) = $\frac{1}{PL} \sum_{i \in path} \text{TRACE}(\Sigma_i)$ is the average trace of the covariance over the path.

This experiment shows how our baseline algorithm FSE (see Tab. I), using the same orderings as the forward search algorithm from Censi *et al.*, is comparable to theirs in speed and memory load while providing a similar shape of the resulting paths, thus we can use it for fair comparisons for our other experiments and algorithms. The results can be seen in Tab. II while an illustration can be found in Fig. 4. The experiments indicate that the TRACE dominance reduces the computational load by approximately 30% while returning the same path, showing that pruning the belief space more aggressively leads to shorter computation times and solutions of similar quality. Every other dominance relation produces the expected results: D-OPT leads to safer, longer paths and the weighted solution balances both aspects. The shorter computation time of D-OPT is due to the bias of this experiment favouring quick information seeking. As a conclusion, the evaluation results show how choosing the evaluation function/dominance leads to significantly less expanded nodes while returning the same solution as the approach by Censi *et al.* [9].

B. Analysis of Evaluation Functions/Dominance Relations

The second set of experiments is designed to support claim (ii) and investigates the performance of our approach in

Algorithm	NC	MM	CT	PL	PU
Censi08	5474	-	0.51	≈ 37.0	-
FSE	5134	461	0.39	33.0	4.2
FSD	3742	592	0.34	33.0	4.2
FSDOPT	3666	281	0.28	46.1	2.8
FSW	9819	990	0.63	42.7	3.2
TE	3170	283	0.28	33.0	4.2
TD	2468	397	0.22	33.0	4.2
TDOPT	1984	101	0.18	46.1	2.8
TW	9146	566	0.42	42.7	3.2

TABLE II: Results of Experiment 1. NC (Nodes Created), MM (Maximum in Memory), CT (Computation Time s), PL (Path Length m), PU (Path Uncertainty).

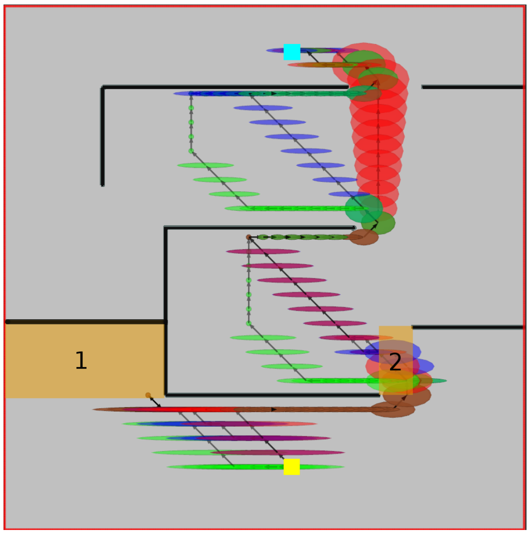

Fig. 4: Results of experiment 1: The algorithms must plot a path leading from the bottom to the top. The sensor is a simplified four-sampled *N,S,W,E* range-finder, leading to necessary relocalization detours. The path by FSE, closely resembling the forward search algorithm by Censi08, is red. The FSDOPT and FSW paths are green and blue. Ellipses are inflated for readability.

realistic settings while analyzing the influence of evaluation functions and dominance relations on search algorithms in belief space. The test data is taken from a set of reference gardens which are chosen by the Robert Bosch GmbH in order to test robotic lawn-mowers, they display a large spectrum of difficult challenges for mobile robots (e.g., slopes, large glass panes, big gardens, uneven ground, etc.). Tab. III aggregates the results of 403 computed paths per algorithm on eight reference gardens, while example trajectories over the longest possible paths are shown in Fig. 5.

There are several conclusions that can be made from this experiment: firstly, the TRACE dominance relation leads to a consistently better runtime and lesser complexity (by a factor of around 50%). The path lengths and obtained certainties are nevertheless very similar, showing that pruning large parts of the search space using a less strict dominance relation leads to faster solutions of similar quality.

Furthermore, the distance based evaluation functions in general reach the solution in less expanded nodes. This is due to the greedy expansion of the search space leading to the goal quickly. This is opposed to the other evaluation functions that either optimize over the covariance (having no bias towards the goal region) or are faced with a multi-objective optimization problem leading to a more balanced exploration of the search space and a more pronounced effect of POMDP dimensionality/history curses. Nevertheless, TDOPT computes comparably short paths i.e. barely $6\,\mathrm{m}$ longer than the optimum in the mean, or 30%. It also has better localizability than the distance based evaluation functions. The WEIGHTED heuristic combines shortness with good localization, although the search is much longer.

Distance based evaluation functions privilege going through the information poor center, only optimizing the path length, see Fig. 5. On the other hand, TDOPT and TW stick to the edges of the garden where localizability is better. In this particular example, TDOPT exceptionally leads to shorter computation time and lesser complexity because there is a ”corridor” of continuously good localizability going from start to goal. For completeness, we have also tested some non-admissible evaluation functions such as a variant of TD where $f(n) = 1.5 \times \textsc{Dijkstra}(n, goal) + d(n)$, where DIJKSTRA is used to define the heuristic. This leads to 50% shorter computation times than TD in the mean, while only making the paths slightly longer in all our experiments.

Algorithm	NC	MM	CT	PL	PU
FSE	5557±(27121)	867.6±(2599.1)	0.2±(1.3)	17.9±(9.1)	0.8±(1.0)
FSD	3874±(17530)	839.1±(2425.8)	0.2±(0.8)	17.9±(9.1)	0.8±(1.1)
FSDOPT	7247±(8956)	632.7±(604.3)	0.2±(0.3)	24.2±(16.6)	0.2±(0.1)
FSW	53565±(181230)	5579±(13929)	2.0±(7.2)	22.0±(14.2)	0.2±(0.2)
TE	2446±(6102)	494.7±(741.9)	0.1±(0.2)	17.9±(9.1)	0.8±(1.0)
TD	1917±(4344)	507.1±(789.2)	0.1±(0.2)	17.9±(9.1)	0.8±(1.0)
TDOPT	4529±(4730)	288.1±(184.4)	0.2±(0.2)	24.3±(16.5)	0.2±(0.1)
TW	37223±(118612)	2643.1±(5440)	1.3±(4.0)	22.0±(14.3)	0.2±(0.2)

TABLE III: Results of Experiment 2: resolution of the grid is $0.25\,\mathrm{m}$.

C. Real World Evaluation

The third set of experiments is designed to support claim (iii) and tests our approach on a real robot. We have done our experiments on a garden in Renningen, Germany using a Clearpath Robotics Jackal robot depicted in Fig. 6 (top). The robot has been modified by adding a Velodyne VLP-16, although the experiments only use one ray of the scanner truncated to $4\,\mathrm{m}$. The robot makes a plan using the TW algorithm, see Fig. 6 (bottom). The path goes from the bottom left to the top right, while computations took $5\,\mathrm{s}$ on the Jackal Celeron J1800. This algorithm privileges information seeking, so it leads the robot through the information rich top left corner area while staying close to the walls. The motion commands are computed using a Dynamic Window Approach that smooths the rough edges of the path. The green line represents the mean of the robot belief as it travels down the path, while the blue line is the ground truth data with a precision of $5\,\mathrm{cm}$). This experiment shows how the algorithm is able to run on a real robot and produces paths maximizing localizability that can be followed by a robot having very constrained sensors.

VI. CONCLUSION

In this paper, we presented a novel approach for motion planning under uncertainty resulting from movement and

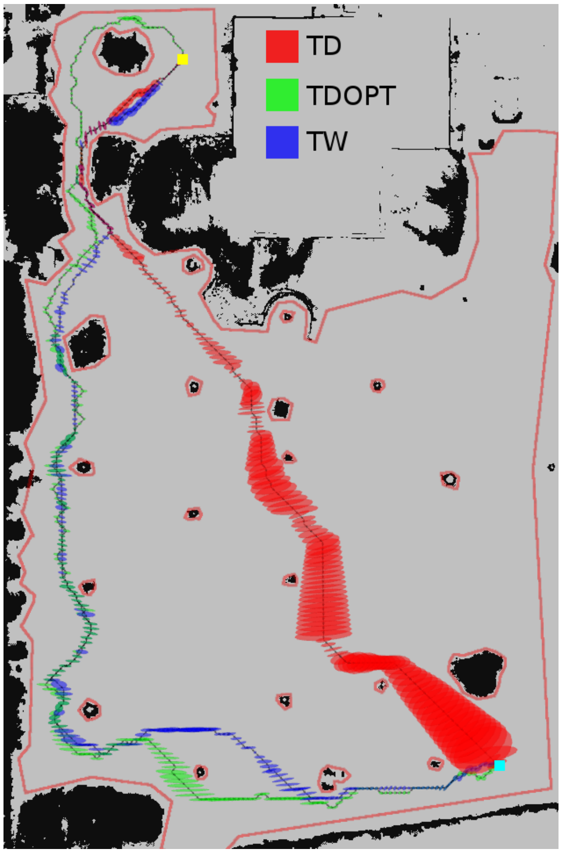

Fig. 5: Experiment 2 on the largest garden showing the divergence between possible paths clearly. The algorithms plan a path from the yellow (top left) to the cyan (bottom right) square. The red (TD) path CT: 1.3, NC: 29722, MM: 5749, PL: 51.6, PU: 1.1. The green (TDOPT) one CT: 0.7, NC: 14859, MM: 678, PL: 83.5, PU: 0.2. The blue (TW) one CT: 4.9, NC: 154804, MM: 10032, PL: 69.5, PU: 0.2. Note how the green path is always safer but longer, while the red one is the shortest.

sensing. Our approach operates in belief space and works in two steps. The first step is the computation of a localizability map used to simulate the update step of a Bayesian filter computing the robot's current belief. The second step uses a best-first search algorithm instantiated with different evaluation functions / dominance relations and the localizability map to find a path from start to goal. We implemented and evaluated our approach on different datasets and provided comparisons to other existing techniques and supported all claims made in this paper. The experiments suggest that our approach can be used to plan paths that maximize localization information in real-time on small robots such as robotic lawn-mowers.

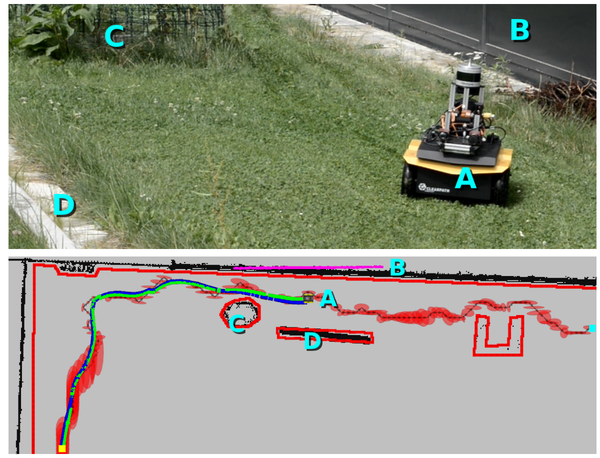

Fig. 6: Experiments on a real garden and robot. Top: A modified Clearpath Robotics Jackal (A) on a real garden used in the experiment during the execution of a path. Bottom: An example path computed in the experiment. A grid-map view of the environment (black and grey map), obstacles (red lines, B, C, D), the planned path with the predicted belief of the robot (red ellipses), the ground truth path the robot took (blue path) and the actual evolution of the robot belief as it drove on the path (green path). The magenta points show the current laser scan, which consists of a Velodyne VLP-16 ray truncated to $4\,\mathrm{m}$.

References

[1] A.A. Agha-mohammadi, S. Chakravorty, and N.M. Amato. FIRM: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements. *Int. Journal of Robotics Research*, 2014.

[2] O. Bengtsson and A.-J. Baerveldt. Robot localization based on scan-matching - estimating the covariance matrix for the IDC algorithm. *Journ. of Rob. & Aut. Systems*, 2003.

[3] S. Bopardikar, B. Englot, and A. Speranzon. Multiobjective path planning: Localization constraints and collision probability. *IEEE Transactions on Robotics*, 2015.

[4] A. Bry and N. Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.

[5] H. Carrillo and J.A. Castellanos. Navigation under uncertainty based on active SLAM concepts. In *Handling Uncertainty and Networked Structure in Robot Control*. Springer International Publishing, 2015.

[6] H. Carrillo, Y. Latif, M. L. Rodriguez-Arevalo, J. Neira, and J.A. Castellanos. On the monotonicity of optimality criteria during exploration in active SLAM. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2015.

[7] A. Censi. An accurate closed-form estimate of ICP's covariance. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2007.

[8] A. Censi. On achievable accuracy for range-finder localization. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2007.

[9] A. Censi, D. Calisi, A. De Luca, and G. Oriolo. A bayesian framework for optimal motion planning with uncertainty. *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2008.

[10] G. Costante, C. Forster, J.A. Delmerico, P. Valigi, and D. Scaramuzza. Perception-aware path planning. *Accepted at IEEE Transactions on Robotics*, 2016.

[11] A. Lambert and N. Le Fort-Piat. Safe task planning integrating uncertainties and local maps federations. *Int. Journal of Robotics Research*, 2000.

[12] A. Lambert and D. Gruyer. Safe path planning in an uncertain-configuration space. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2003.

[13] D. Lenz, M. Rickert, and A. Knoll. Heuristic search in belief space for motion planning under uncertainties. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2015.

[14] R. Pepy and A. Lambert. Safe path planning in an uncertain-configuration space using RRT. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2006.

[15] S. Prentice and N. Roy. The belief roadmap: Efficient planning in belief space by factoring the covariance. *Int. Journal of Robotics Research*, 2009.

[16] C. Stachniss and W. Burgard. An integrated approach to goal-directed obstacle avoidance under dynamic constraints for dynamic environments. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2002.

[17] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.

[18] J. van den Berg, P. Abbeel, and K. Goldberg. LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. *Int. Journal of Robotics Research*, 2011.