Coverage Path Planning in Belief Space

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

Abstract— **For safety reasons, robotic lawn mowers and similar devices are required to stay within a predefined working area. Keeping the robot within its workspace is typically achieved by special safeguards such as a wire installed in the ground. In the case of robotic lawn mowers, this causes a certain customer reluctance. It is more desirable to fulfill those safety-critical tasks by safe navigation and path planning. In this paper, we tackle the problem of planning a coverage path composed of parallel lanes that maximizes robot safety under the constraints of cheap, low range sensors and thus substantial uncertainty in the robot's belief and ability to execute actions. Our approach uses a map of the environment to estimate localizability at all locations, and it uses these estimates to search for an uncertainty-aware coverage path while avoiding collisions. We implemented our approach using C++ and ROS and thoroughly tested it on real garden data. The experiment shows that our approach leads to safer meander patterns for the lawn mower and takes expected localizability information into account.**

I. INTRODUCTION

Robotic mowers should operate autonomously over a long period of time, on various types of gardens, and without human intervention. They may mow the lawn in parallel lanes for aesthetic reasons and to make the robot path more understandable by the end-user. Furthermore, lawn mowers must not put themselves or humans in danger, which is achieved by forbidding the mower to leave the working area and by avoiding collisions with obstacles inside it. Both the working area and static obstacles are usually defined by laying a wire manually along their borders. The wire can be sensed with high precision and robustness by a short range sensor. The main drawback of this approach is the inconvenience for end-users: it is tedious to set up and eventual mistakes are difficult to correct. The next leap in robotic lawn mower navigation is to make the product more customer friendly through a wireless solution. It should be robust, safe, and provide price competitiveness. An approach towards that goal is to equip the robot with a low cost 2D laser scanner and use Simultaneous Localization and Mapping (SLAM) techniques in combination with uncertainty-aware navigation.

[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 German Research Foundation under Germany's Excellence Strategy, EXC-2070 - 390732324 (PhenoRob).

We thank Sergey Alatartsev, Narunas Vaskevicius and Andrey Rudenko for the fruitful discussions.

In this paper, we concentrate on the problem of planning a coverage path under motion and sensing uncertainty that may run on a robot operating without a perimeter wire. Uncertainties in movement and sensing lead to problems in belief space, as the robot state cannot be assumed to be precisely known at all times. We must thus maintain a probability distribution over possible states of the robot, called the belief, and compute a control policy to select the best actions given the knowledge about the environment and the current belief. Our setting implies prior knowledge about the environment, i.e., a point-cloud or occupancy grid map of the surroundings, and the emplacement of obstacles and work-area borders as polygons. This is a safety requirement in the absence of a perimeter wire, as certain borders such as ones separating the working-area from board walks cannot be sensed by on-board sensors and must be taught in.

The main contribution of this paper is a new approach for coverage path planning in belief space for mobile robots with laser scanners, and has a special focus on robotic lawn mowers. Our coverage builds upon our uncertainty aware planning system [16], where we plan point-to-point paths in belief space. Similarly, the approach starts by computing a localizability prior for the robot's sensor at all possible locations in the map. This is fused with an odometry drift to yield the expected belief dynamics, which are taken into account at planning time to compute a set of parallel lanes that avoid collisions. In sum, our approach computes coverage paths in belief space that are safer than state-of-the-art techniques achieve by taking into account the collision probability of the path.

II. RELATED WORK

Covering polygonal regions in 2D configuration space is a well studied problem. Traditional approaches work by decomposing the workspace into regions using cell decompositions, by discretizing the workspace using grids, or by using a spanning tree. If the space to be covered is not known in advance, online approaches exist that cover the integrality of the workspace while exploring it. Recent approaches have focused on extensions to non-holonomic robots [12], the inclusion of other externalities such as battery power [21], or planning coverage for 2.5D elevation maps [5]. For more details on approaches for robotic coverage path planning, see the surveys by Galceran *et al.* [6] or Bormann *et al.* [14].

Accounting for positional uncertainty adds significant challenges to motion planning, as the planning takes place in belief space instead of the configuration space. This is done by accounting for the predicted evolution of the robot belief at planning time, or belief dynamics. They model the interplay between information loss through odometry drift and information gain through exteroceptive sensors. This can be done for different sensor types and belief representations: Van den Berg *et al.* [4] use an Extended Kalman Filter (EKF) to model the belief of different robot types (car-like, differential drive, aircraft-like) that gain information on their state through GPS beacons. Censi *et al.* [2] model belief dynamics in information space and assume the robot to have a wall-detection sensor. The belief dynamics are then used in the path planning process using different methods: They can be included in many robot problems that are modeled as Markov Decision Processes using Augmented Markov Decision Processes as done by Roy *et al.* [15] or [13]. Lambert *et al.* [11] extend an A* heuristic search algorithm to incorporate the belief dynamics, while the approach by Van den Berg *et al.* [19] combines them with Rapidly exploring Random Trees.

Recently, some research has tried to combine coverage and belief space planning with the goal of accounting for sensing uncertainty. In this paper, we combine our precedent uncertainty-aware point-to-point path planning approach [16] into the boustrophedon coverage algorithm in 2D configuration space by Choset *et al.* [3]. Similarly, the approach by Kim *et al.* [9] uses techniques from active-SLAM to determine when to backtrack to previously explored areas. They consider camera equipped underwater robots that actively build a map by drawing parallel lane patterns. In contrast to that, we use a 2D laser scanner on a map that has previously been explored under supervision by the end-user. Paull *et al.* [10] present an approach to plan coverage paths in belief space for teams of underwater robots that can cooperatively localize with a focus on achieving complete coverage. From a modeling perspective, the approach closest to the one presented is the work by Galceran *et al.* [7]. They also assume to have a localization prior and compute paths along parallel lanes. However, they plan for underwater robots that float above obstacles, so do not take into account the probability of colliding with objects or leaving the working area and thus have a different objective.

III. PROBLEM DESCRIPTION

We address the problem of coverage path planning in belief space for differential-drive equipped mobile robots. In the robotic wireless lawn mower setting, this involves finding parallel lanes covering the entire working area while maximizing the probability of staying localized during navigation along the path. Parallel lanes are used to make the robot more understandable by the end-user and for aesthetically pleasing mowing patterns. We account for the uncertainties from motion and sensing, and assume the SLAM-map to be largely free of gross errors and the world to be static. We assume this large prior knowledge because the product is initialized, i.e., taught the working area borders while mapping, supervised by the end-user.

For the computation of belief dynamics at planning time, we describe the robot configuration by its position (x, y) and the robot belief by a two-dimensional Gaussian distribution $b \sim \mathcal{N}(\mu_b, \Sigma_b)$, $\Sigma_b = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{bmatrix}$. We discount the robot orientation during belief dynamics computation as this reduces the dimensionality of the path planning problem, does not have a large influence on model accuracy because of the low robot speed, and still permits the detection of collisions.

We forward-simulate the belief evolution at planning time and model how the robot gains information on its state through its exteroceptive sensors and loses it through odometry drift recursively using an EKF, as described in Thrun *et al.* [18]. The belief space is defined as the set of all probability distributions over the configurations of the robot. In our case this is a *position* $\times$ *covariances* space. To make the computations lighter, we further discretize by only considering beliefs centered at regular intervals of a grid.

IV. OUR APPROACH

To plan in belief space, we require a localization prior for the 2D laser-scanner of the robot, which we show how to compute in Sec. IV-A. We then review how coverage in *configuration space* can be formulated as a Generalized Travelling Salesman Problem (GTSP) in Sec. IV-B. We finally formulate the coverage problem in *belief space* as a GTSP using a three-step approach in Sec. IV-C.

A. Localizability Map

Given the information gained about the garden layout during the initialization of the lawn mower and its sensor model, we can estimate how measurements will affect the robot localization using the approach by Bengtsson *et al.* [1], also similar to [20]. We start by simulating the laser scanner by ray tracing the SLAM-map of the environment (we use a point-cloud, although it would also work with an occupancy grid map). We then repeatedly transform the ray traced scan by up to 50 cm and 15°. Finally, we register the transformed simulated laser scans with the SLAM-map of the environment in order to compute the covariance of converged-to locations. When the resulting covariance is small, the sensed part of the environment possesses useful features such as the presence of a corner or another geometric structure that enables an unambiguous alignment. As a result of that, the map-matching is able to converge to the actual location where the laser-scan was ray traced from.

We perform this for every point of the map with a given resolution and consider the result to be a localizability prior, or estimate of an EKF update-step for the given robot and odometry noise. An extract from the localizability map computed for the test garden is depicted in Fig. 1. We have chosen a maximum scanning range of 8 m to mimic low-cost range sensors that could be used on a lawn mower. We assume the noise in the range measurement to be Gaussian with a variance of 3 cm. Fig. 1 shows how the map can be divided in 5 areas where localizability is similar: scans taken in area 1 sense the corner of a building leading to good localizability. Scans taken in areas 2 and 3 can only detect one stretch of the wall, leading to good localizability in only one dimension. Scans in area 4 have little information about the location of the robot, while areas closer to bushes and poles towards the bottom have better localizability again.

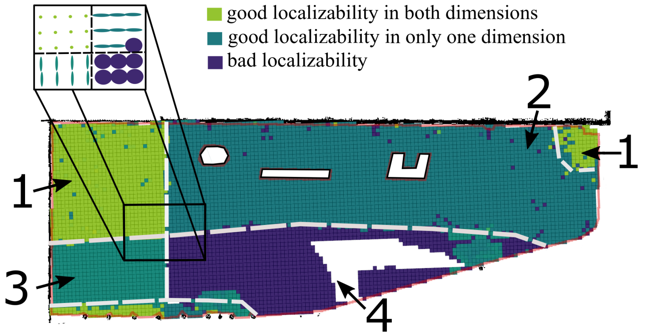

Fig. 1: Localizability map: Top: An extract of a localizability map for the test garden. Each ellipse corresponds to a 3 - σ projection of expected EKF update steps. Ellipse axes of 50 cm are found in dimensions where no information is present. Bottom: semantic view of a localizability map for a larger area. The white areas are either featureless or outside the working area.

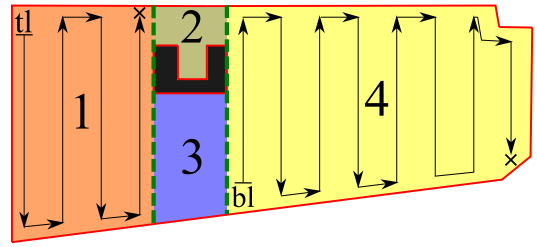

Fig. 2: View of the right part of the test garden decomposed using a boustrophedon decomposition. Two types of movement are required for covering a cell: moving across parallel lanes (long arrows) or moving along obstacles (short arrows). Every decomposition cell $i \in \{1, 2, 3, 4\}$ has 4 solutions with starting points *sp* in different corners: *sp* $\in$ {*tl, tr, bl, br*}. Covering the whole working area is the optimal way of combining decomposition cell solutions into a contiguous path of the form $(1_{tl}, 2_{tl}, 4_{tr}, 3_{tr})$.

B. Coverage in Configuration Space

In this paper, we propose a novel combination of the boustrophedon coverage path planning algorithm introduced by Choset *et al.* [3] and our recent path planning in belief space approach [16] to plan coverage paths taking the uncertainty into account. We first explain coverage as GTSP in configuration space using the boustrophedon approach and then generalize it by including the uncertainty in Sec. IV-C.

The boustrophedon coverage algorithm generates a parallel lane pattern covering the entire working area that accounts for obstacles by dividing the workspace into obstacle-free regions called decomposition cells, as shown in different colors in Fig. 2. Boustrophedon decomposed cells can be completely covered using two simple motions: either by moving in the parallel lane direction (long arrows) or by moving along a cell edge (short arrows). We note that using those two simple motions, there are four different possibilities to cover an entire decomposition area. One path covering the decomposition area starts at the top left corner and follows obstacles towards the right, (tl). Three other paths are defined symmetrically starting in the bottom left (bl), top right (tr) and bottom right (br) corners. Covering the entire working area now becomes a matter of connecting exactly one of the four solutions for each decomposition cell into a contiguous path using a GTSP formulation.

The GTSP generalizes the Travelling Salesman Problem by partitioning the vertices to be visited into subsets, and computing a shortest tour visiting each subset exactly once. Formally, it is defined over a graph $G = (V, E, w)$ where V are vertices, E edges and w edge weights. Additionally, the vertices V are partitioned into pairwise disjoint sets $V = \bigcup_i V_i$. The GTSP solution is the minimum-cost tour containing exactly one vertex from each disjoint set V_i.

To formulate the coverage in configuration space problem as a GTSP, we must design the sets V, E and w. Without loss of generality, we assume the boustrophedon decomposition to yield a set A consisting of n decomposition cells. To solve the problem for any one decomposition cell, it is enough to drive along one of the four possible paths covering it. The vertex $v_{i,sp}$ can be intuitively understood as "the solution to decomposition cell i starting from starting point *sp*". A decomposition cell thus becomes a set of possible solutions V_i of which one must appear in the working area coverage path, see Eq. (1). The set of vertices V of the GTSP is the union of all disjoint decomposition-cell solutions V_i and has size $4 \times n$:

$$V = \bigcup_{i \in A} V_i \quad \text{with } V_i = \bigcup_{sp \in \{tl,tr,bl,br\}} v_{i,sp} \tag{1}$$

The edges E and their associated weights w are used to encode the costs of travelling from v_i to v_j and performing the covering path associated with v_j. This cost is the path length between the end point of v_i and the starting point of v_j, added to the solution length of v_j. Every vertex has $(n-1) \times 4$ edges, namely one towards the starting points of all other areas, so there are $(16n^2 - 16n)$ edges and weights.

To compute the final coverage path, we implement a GTSP solver that we initialize with the insertion heuristic, i.e., iteratively adding one closest vertex from each decomposition cell until a complete tour is found. This initial solution is then refined by using three local search operators for GTSP [8] until a termination criterion is met:

- Swap starting point: For a random cell in the solution, change the starting point to another one.
- k-random swap: switch $k \leq 3$ random cells in the solution with each other.

Summarizing, the steps to plan a coverage plan in configuration space using the boustrophedon algorithm are the following:

- Decompose the work-area into the set of decomposition cells A with $|A| = n$ that can be covered by parallel patterns using the boustrophedon decomposition.
- Compute the set of vertices that encode possible solutions for all decomposition cells. There are 4 different starting points for every cell, each inducing an exit point, see Eq. (1).
- Compute the edges E and weights $w(v_i, v_j)\ \forall i, j \in V$ that correspond to the cost of travelling from vertex v_i to v_j and the cost of covering v_j.
- Solve the GTSP to obtain one contiguous minimal length path.

C. Coverage in Belief Space

We now generalize the coverage approach from configuration to belief space, also relying on the GTSP formulation. Similarly to the situation in configuration space, we use a three-step procedure. First, we decompose the work-area into decomposition cells and build the set of vertices V representing possible solutions. Second, we compute the edge set E and weights w that encode the costs of travelling between decomposition cells and covering them. In the final step, we compute a contiguous path where each decomposition cell is covered exactly once. Our approach is modular as there is freedom in implementing one-decomposition-cell policies in the first step that can privilege different aspects. We choose to maximize aesthetically pleasing parallel lane patterns for the lawn mowing application, while avoiding collisions. The GTSP formulation on the other hand generalizes to any coverage problem with freedom in choosing between one-decomposition-cell policies that differ in some aspects (e.g., localizability, collisions, length, # turns, ...).

Compared to coverage path planning in configuration space, moving the problem into belief space means we must consider all possible robot beliefs b and decide for every one how to solve a decomposition cell or compute a transition.

1) Discretization of a belief: To make the space of all possible robot beliefs tractable, we approximate any two-dimensional Gaussian belief b used as a starting point for a transition or a decomposition cell solution using Eq. (2) and Eq. (3), where $\lambda_{1,2}(b)$ are the eigenvalues of Σ_b considering the x, y-space.

$$\bar{\lambda}(b) = \max(\lambda_1(b), \lambda_2(b)) \tag{2}$$

$$\Sigma_{b_{discretized}} = \begin{bmatrix} \bar{\lambda}(b)^2 & 0 \\ 0 & \bar{\lambda}(b)^2 \end{bmatrix} \tag{3}$$

We quantize all starting beliefs for decomposition cells or transitions into one of $|U|$ uncertainty bins sized δ using Eq. (4) and Eq. (5), where λ_{max} and λ_{min} represent the largest and lowest eigenvalues of any expected robot belief during operation. We will call any $u_b \in U$ the uncertainty level of belief b and set $|U| = 10$ for the experiments.

$$|U| = \frac{\lambda_{max} - \lambda_{min}}{\delta} \tag{4}$$

$$u_b = \left\lceil \frac{\bar{\lambda}(b) - \lambda_{min}}{\delta} \right\rceil \tag{5}$$

2) Coverage of one decomposition cell in belief space: Using the GTSP formulation, we now compute the set of vertices V containing the solutions for all decomposition cells. As we are working in belief space, a vertex $v_{i,sp,b}$ encodes the solution for a decomposition cell i from starting point *sp* and belief b. We reduce the computational burden by discretizing b using Eq. (5). A vertex now becomes $v_{i,sp,u}$, intuitively understood as "the solution to decomposition cell i starting from starting point *sp* with uncertainty level u". While any policy covering the entirety of a decomposition cell could be implemented, we propose to minimize collisions while maximizing lane length.

Given the starting point and uncertainty level, we track the expected evolution of the robot belief along the path and avoid any motions that might lead to collisions by prematurely switching to the next lane, as seen in Fig. 3. Once the entire width of the cell has been covered according to this policy, uncovered areas may remain as illustrated in purple in Fig. 4. We treat the uncovered areas as independent decomposition cells and solve a local GTSP connecting the current predicted belief of the robot b (the blue ellipse) with the uncovered areas $v_{4,tl,u_3,1}$ and $v_{4,tl,u_3,2}$. We solve the local GTSP using the same method used to solve the global GTSP problem described in Sec. IV-C.4. We avoid excessive information seeking behaviour, e.g., driving a 50 m detour to cover a small area such as $v_{4,tl,u_3,2}$, by trading travelled distance with uncovered area using parameter β which we set to 5.0 for the experiments. We note that this policy is too cautions to guarantee complete coverage, as certain environments will not yield enough information to counteract odometry drift at all positions.

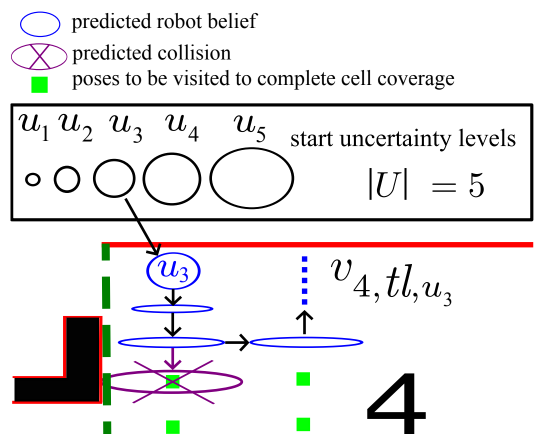

Fig. 3: Policy for solving one decomposition cell under uncertainty. We compute a solution to starting in the top-left corner with u_3. The uncertainty grows in the x dimension through odometric drift, while the uncertainty in the y dimension is bounded due to sensing the upper wall. When the robot belief touches an obstacle, this belief is recognized as a possible collision and the robot moves to the next lane prematurely and continues the parallel lane pattern.

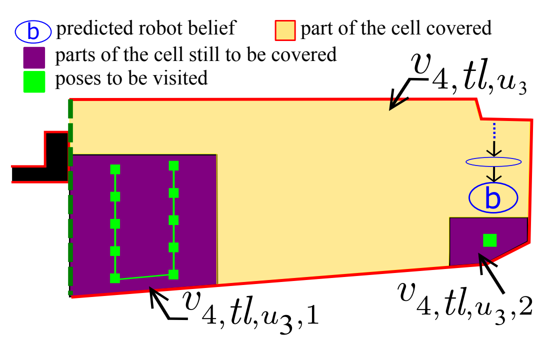

Fig. 4: Solution to one decomposition cell under uncertainty. When the robot has finished the policy of moving until detecting a collision and changing lanes, certain areas might go uncovered as shown in purple. In order to achieve complete coverage, the robot must still visit the sub-decomposition cells $v_{4,tl,u_3,1}$ and $v_{4,tl,u_3,2}$ from the predicted belief of the robot, b.

By iterating over all decomposition cells, starting points and uncertainty levels, we compute the vertex set for each decomposition cell V_i using Eq. (6).

Additionally, every vertex must also store the induced exit belief $b_{exit}(v_{i,sp,u})$, the path length $d(v_{i,sp,u})$ and left untreated area $o(v_{i,sp,u})$ in order to compute the vertex costs $c(v_{i,sp,u})$ as done in Eq. (7). Those are required for computing the edge weights w.

$$V_i = \{v_{i,sp,u}\}\ \forall i \in A,\ sp \in \{\textit{tl, tr, bl, br}\},\ u \in U \quad (6)$$

$$c(v_{i,sp,u}) = d(v_{i,sp,u}) + \beta\ o(v_{i,sp,u}) \quad (7)$$

3) Transitions between possible areas: The last step for exploiting the GTSP formulation requires us to compute the edges E and edge weights w. As those are the only elements carrying information into the optimization process, the edge weights must contain the goal vertex costs alongside the transition costs:

$$w(v_i, v_j) = c(v_i, v_j) + c(v_j) \quad \forall i, j \in V \quad (8)$$

$$v_i = v_{i,sp,u} \textit{ with } i \in A,\ sp \in \{\textit{tl, tr, bl, br}\},\ u \in U$$

As the cost $c(v_j)$ has already been computed, we only need to compute the transition cost $c(v_i, v_j)$ for all possible edges, which represents the costs of moving the robot from $b_{exit}(v_i)$ to the starting point and uncertainty level of v_j. Planning this transition in belief space is useful to regain information after meandering within an area, as the robot does not need to follow a parallel lane pattern: this leads to greater freedom in trading off path distance and uncertainty.

One of the keys to our approach is to avoid planning a transition to every uncertainty level at the goal. Optimizing path safety implies that transitions leading to better uncertainty-levels at the goal-pose are more desirable than those leading to worse uncertainty. It is thus sufficient to search for one path that fulfills an optimal trade-off between final uncertainty and path length. We do this efficiently by using a variant of the point-to-point planner from our previous work [16] and adapting the generic search algorithm to plan "point-to-set-of-points" paths in belief space. This is effective as the transitions are computed from one final belief to all other start positions, leading to faster computations by searching for them batchwise. As paths towards sets of points are searched, we cannot use heuristic-guided approaches. We opt for a simple wave-front algorithm combining the distance travelled with the accumulated uncertainty via a trade-off parameter α: $cost(path) = \alpha \times (path\ length) + 1 - \alpha\ (path\ uncertainty)$, with *path uncertainty* the sum of belief variances over a predicted path. We set $\alpha = 0.05$ to bias the search towards safer, longer paths.

Thus, we compute one transition for all (exit-belief, start-point) pairs of different decomposition cells which, depending on α and the information contained in the environment, induces the uncertainty-level at the goal vertex v_j: $(b_{exit}(v_i),\ (v_{j,sp,u_{induced}}))$ with $u_{induced} \in U$. Furthermore, we set all edge weights according to Eq. (9), meaning that the only edges carrying non-infinite costs are those where good transitions have been found.

$$w(v_{i_1,sp_1,u_{b1}}, v_{i_2,sp_2,u_{b2}}) = \begin{cases} c(v_{i_2,sp_2,u_{b2}}) & \\ + \ c(b_{exit}(v_{i_1,sp_1,u_{b1}}), (v_{i_2,sp_2,u_{b2}})) & \text{if } u_{b2} = u_{induced} \\ + \ \infty & \text{else} \end{cases} \quad (9)$$

Similarly to the case in configuration space, every vertex has $(n-1) \times 4 \times |U|$ edges. As there are $4 \times |U|$ vertices per decomposition cell and n decomposition cells, we have to compute $16 \times |U|^2 \times (n-1) \times n$ edges. On the other hand, we reduce the amount of edges flowing into the optimization by only considering one transition for any (exit-belief, starting point) pair. This reduces the amount of non-infinite edges to $16 \times |U| \times (n-1) \times n$.

4) Solving the GTSP for coverage in belief space: Now that we have computed V, E and w, we search for a contiguous shortest path covering all decomposition cells. We note that belief space planning has a specific structure that differentiates it from configuration space planning. In the configuration space GTSP, the salesman gets to choose which vertex he travels to. In belief space, the robot cannot decide the uncertainty level with which it starts a decomposition cell, as it is induced by the currently taken trajectory (i.e., the starting points and decomposition cells on the path so far). To solve the GTSP, we use the same approach as for the problem in configuration space (Sec. IV-B) by first computing a greedy solution which we iteratively improve by using local GTSP operators.

V. EXPERIMENTAL EVALUATION

The main focus of this work is to combine belief space with coverage planning in order to compute a safe coverage path for a laser-scanner equipped robotic lawn mower. Our experiment is designed to support our key claim, that our approach computes safer coverage paths than current coverage techniques achieve by taking into account the collision probability of the path.

We compare three approaches: OURS, the collision avoiding, parallel lane maximizing belief space approach we have described. MIN, a belief space approach that replaces the cost term $d(v_{i,sp,u})$ with the mean predicted uncertainty. This minimizes the uncertainty without accounting for the probability of collision and resembles the approach by Galceran *et al.* [7], but differs in finishing a cell before relocalizing. And CONFIG, a classical Boustrophedon coverage algorithm in configuration space.

The experiment consists in computing coverage paths on data from a real garden and driving along every path in simulation using Gazebo 90 times. The simulated 2D laser scanner has a range of 8 m and 1850 samples at 10 Hz. All scans are used in a point-to-plane scan-matching algorithm for localization [17] with resolved data association. The simulated odometry has zero-mean Gaussian noise with variance (0.25 m,0.25 m,15°) (longitudinal, lateral, rotational) per meter. We choose these high noise values to show how the approaches deal with difficult situations that a robotic lawnmower might encounter, such as driving on wet grass over uneven ground. All computations are performed on an Intel Core i7 CPU @2.8 GHz. The garden measures 12×36 m and the coverage pattern has 1 m spacing, while the GTSP solver terminates after 40 s, which we include in the computation time.

We show the paths covering the working area for all approaches in Fig. 5 and present other quantitative results for the path computation in Tab. I. All coverage paths start in the top right and clearly show the parallel lane patterns underlying their one-cell-coverage policies. Path-length wise, we note how CONFIG is the shortest as it optimizes the distance, while MIN is the longest, as is does not include a cost term for solution length. Computation-time wise, OURS takes five times longer than the next slowest approach. The main reason for this is the computation of many transitions between two points in belief space (shown as # edges), as we must solve all decomposition-cell-sub-GTSPs as described in Fig. 4. These paths do not change without adding new information about the environment, so they can be stored in a look-up table or computed offline. As the uncertainty-aware approaches have difficulties finding transitions towards very narrow areas such as the U-shaped obstacle in the top right, their final coverage percentage is not 100%. We also show the predicted belief evolution for OURS in Fig. 6 and draw attention to the effects of localizability: the predicted uncertainty for paths in less informative parts of the garden is large (cyan, bottom), while safe areas are recognized as such (blue, top left). The figure shows how our approach interrupts lanes before the predicted 3-σ uncertainty ellipses touch an obstacle.

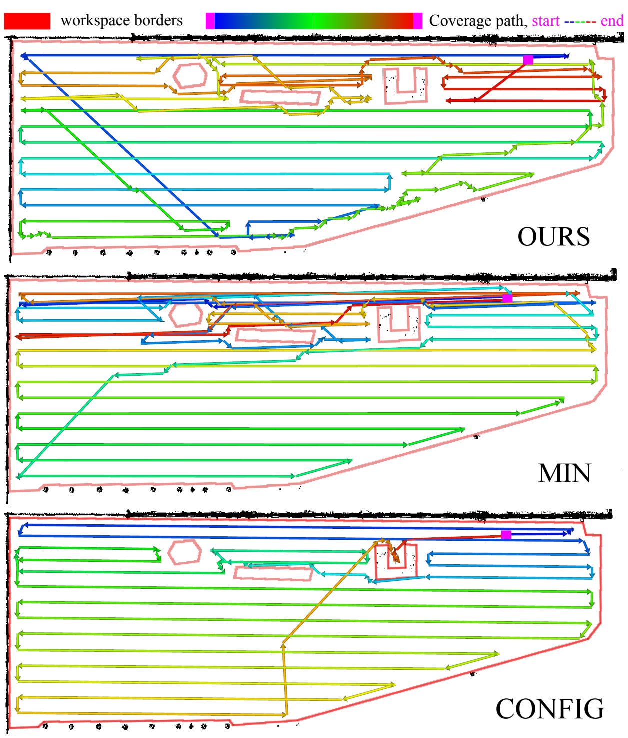

Fig. 5: The coverage paths computed by the three approaches.

We show the results of performing the coverage paths 90 times in simulation in Tab. II. The simulated robot drives

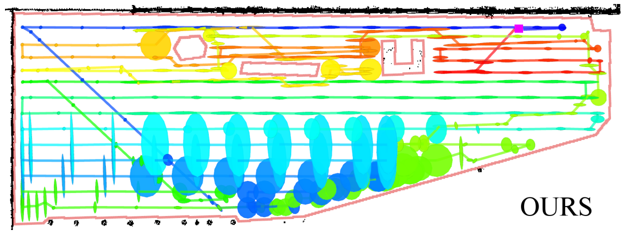

Fig. 6: Predicted robot belief evolution over the coverage path using our approach, the visualization is thinned out for readability.

Approach	Path length (m)	Computation time (s)	# Edges
OURS	486	584	72449
MIN	526	112	19372
CONFIG	**438**	**45.3**	**2998**

TABLE I: Computational aspects of the computed paths.

along every coverage path for about 45 km. We sample the ground truth position and robot belief every 25 cm, from which we compute the localization error (distance between ground truth pose and robot belief), lane error (distance between ground truth position and lane) and the number of collisions. OURS has the lowest lane errors and collisions, showing how we plan safer coverage paths than the other approaches. This is due to the ability of the approach to interrupt lanes as soon as the risk of collision is too high, which happens when the predicted lane and localization errors are large while close to a border. We note how in this experiment, MIN has the lowest localization error, while still having more collisions than OURS. This is due to this approach's smaller impact on lane error and it's indifference to the collision probability. In sum, our experiment shows how planning coverage paths in belief space leads to paths with fewer collisions and better localization.

Approach	Dist.(m)	Loc. err.(m)	Lane err.(m)	Collisions
OURS	45592	0.29 $\pm$ 0.28	**0.18** $\pm$ 0.27	**72** $\pm$ 3.7
MIN	49193	**0.28** $\pm$ 0.31	0.24 $\pm$ 0.31	421 $\pm$ 22
CONFIG	**41229**	0.38 $\pm$ 0.4	0.29 $\pm$ 0.4	1172 $\pm$ 93

TABLE II: Results of the 90 simulated runs of the three approaches.

VI. Conclusion

We present a novel approach to plan coverage paths in belief space. We take the predicted evolution of the robot belief at planning time into account to avoid static obstacles. Our method exploits prior knowledge of the map to compute a localization prior which we then use to predict the robot belief for any path. Our approach relies on formulating the coverage problem in belief space as a GTSP. This allows us to pre-compute solutions to decomposition cells and transitions in belief space, which we then quickly combine into one path covering the entire working area by solving the GTSP. We evaluate our approach on data from a real garden and provide comparisons to other existing techniques and show the additional collision avoidance capabilities gained by planning coverage paths in belief space.

REFERENCES

[1] O. Bengtsson and A.-J. Baerveldt. Robot localization based on scan-matching - estimating the covariance matrix for the IDC algorithm. *Journal on Robotics and Autonomous Systems (RAS)*, 2003.

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

[3] H. Choset and P. Pignon. Coverage Path Planning: The Boustrophedon Cellular Decomposition. In *International Conference on Field and Service Robotics*, 1998.

[4] J. Van den Berg, S. Patil, and R. Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *Intl. Journal of Robotics Research (IJRR)*, 2012.

[5] E. Galceran and M. Carreras. Planning coverage paths on bathymetric maps for in-detail inspection of the ocean floor. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2013.

[6] E. Galceran and M. Carreras. A survey on coverage path planning for robotics. *Journal on Robotics and Autonomous Systems (RAS)*, 2013.

[7] E. Galceran, S. Nagappa, M. Carreras, P. Ridao, and A. Palomer. Uncertainty-driven survey path planning for bathymetric mapping. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[8] G. Gutin and D. Karapetyan. A memetic algorithm for the generalized traveling salesman problem. *Natural Computing*, 2010.

[9] A. Kim and Ryan Eustice. Active visual SLAM for robotic area coverage: Theory and experiment. *Intl. Journal of Robotics Research (IJRR)*, 2015.

[10] J. J. Leonard L. Paull, M. Seto and H. Li. Probabilistic cooperative mobile robot area coverage and its application to autonomous seabed mapping. *Intl. Journal of Robotics Research (IJRR)*, 2018.

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

[12] J. Lewis, W. Edwards, K. Benson, I. Rekleitis, and J. O'Kane. Semi-boustrophedon coverage with a dubins vehicle. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[13] L. Nardi and C. Stachniss. Uncertainty-Aware Path Planning for Navigation on Road Networks Using Augmented MDPs. *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[14] R. Bormann, J. Florian, J. Hammp and M. Haegele. Indoor Coverage Path Planning: Survey, Implementation, Analysis. *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.

[15] N. Roy and S. Thrun. Coastal navigation with mobile robots. In *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, volume 12, pages 1043–1049, 1999.

[16] R. Schirmer, P. Biber, and C. Stachniss. Efficient path planning in belief space for safe navigation. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[17] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Proc. of Robotics: Science and Systems (RSS)*, 2009.

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

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

[20] O. Vysotska and C. Stachniss. Improving slam by exploiting building information from publicly available maps and localization priors. *Photogrammetrie – Fernerkundung – Geoinformation (PFG)*, 85(1):53–65, 2017.

[21] M. Wei and V. Isler. Coverage path planning under the energy constraint. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.