Towards Uncertainty-Aware Path Planning for Navigation on Road Networks Using Augmented MDPs

Lorenzo Nardi Cyrill Stachniss

***Abstract*— Although most robots use probabilistic algorithms to solve state estimation problems such as localization, path planning is often performed without considering the uncertainty about robot's position. Uncertainty, however, matters in planning. In this paper, we investigate the problem of path planning considering the uncertainty in the robot's belief about the world, in its perceptions and in its action execution. We propose the use of an uncertainty-augmented Markov Decision Process to approximate the underlying Partially Observable Markov Decision Process, and we employ a localization prior to estimate how the uncertainty about robot's belief propagates through the environment. This yields to a planning approach that generates navigation policies able to make decisions according to different degrees of uncertainty while being computationally tractable. We implemented our approach and thoroughly evaluated it on different navigation problems. Our experiments suggest that we are able to compute policies that are more effective than approaches that ignore the uncertainty and also to outperform policies that always take the safest actions.**

I. Introduction

Over the past decades, there has been a great progress in autonomous robot navigation and today we find lots of robots that navigate indoors and outdoors. Although most robots use probabilistic algorithms for localization or mapping, most path planning systems assume to know the position of the robot while computing a path. Ignoring position uncertainty during planning may be acceptable if the robot is precisely localized, but it can lead to suboptimal navigation decisions if the uncertainty is large. Consider for example the belief about robot's position represented in Fig. 1 by the black shaded area (the darker the more likely). The robot could be at intersection A or B, but the localization system is not able to disambiguate them. Ignoring the uncertainty, we could assume the robot to be at the most likely position B. Thus, it should turn to the right to reach the goal through the shortest path (blue). However, if the robot is at A (less likely, but possible), going right would lead it to a detour (red).

In this paper, we investigate the problem of path planning under uncertainty. Uncertainty-aware plans reduce the risk to make wrong turns when the uncertainty is large. For example, in Fig. 1, the robot could navigate towards intersection C, which has distinctive surrounding and, thus, the robot is expected to localize better. There, it can safely turn towards the goal avoiding the risk of long detours (green). A general formalization for this type of problem is the Partially Observable Markov Decision Process (POMDP). POMDPs, however, become quickly intractable for real-world applications. Our goal is to investigate an approximation that is still able to consider the localization uncertainty.

All authors are with the University of Bonn, Germany.

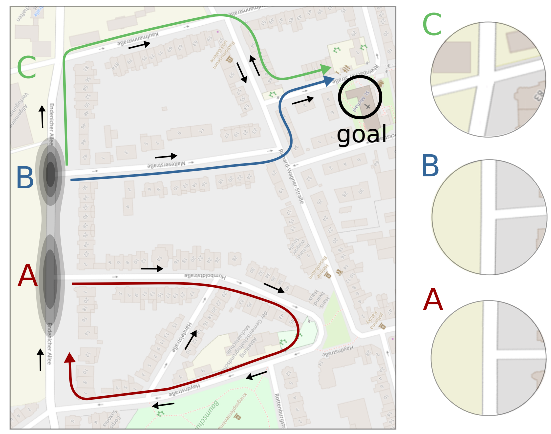


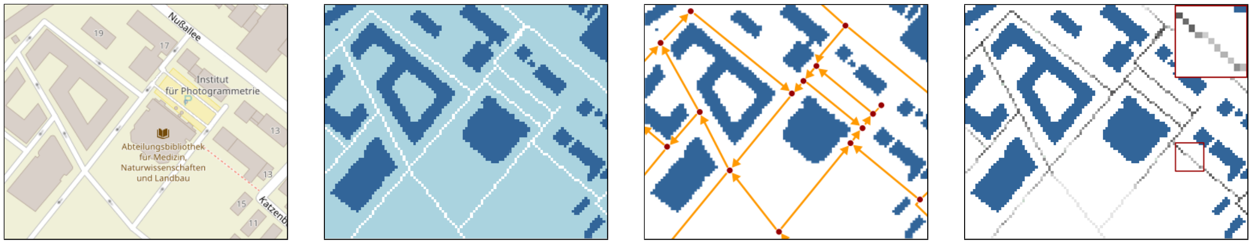
Fig. 1: Example of robot navigation with high pose uncertainty. The black shaded area is the belief about robot's position (the darker, the more likely). A, B, C are the road intersections (in detail on the right side). The paths from each intersection are colored respectively in red, orange and green. The black arrows indicate the roads' directions.

The main contribution of this paper is a novel approach that is a step forward in planning routes on road networks considering the uncertainty about robot's position and action execution. It relies on the Augmented Markov Decision Process (A-MDP) [15], which approximates a POMDP by modeling the uncertainty as part of the state. We employ a localization prior to estimate how robot's belief propagates along the road network. The resulting policy minimizes the expected travel time while reducing the mistakes that the robot makes during navigation with large position uncertainty. As a result, our planning approach, first, explicitly considers the robot's position uncertainty, and thus it is able to take different actions according to the degree of uncertainty; second, in complex situations, it leads to plans that are on average shorter than a shortest path policy operating under uncertainty but ignoring it.

II. Related Work

Although planning under uncertainty has received substantial attention, most robotic systems such as Obelix [10] or the Autonomous City Explorer [11] still use A* to navigate in urban environments. Navigation in urban environments often exploits topological or topo-metric maps [9]. These maps can be stored compactly as a graph and free maps of most cities exist, for example, through OpenStreetMap.

The Markov Decision Process (MDP) allows for optimally solving planning problems in which the actions are noisy but the state is fully observable. If the state is not fully observable, the problem turns into a Partially Observable Markov Decision Process (POMDP). However, in POMDPs,



(a) OSM map. (b) Traversability grid map $\mathcal{X}$. (c) Roads graph $\mathcal{G}$. (d) Localizability map $\mathcal{Z}$.

Fig. 2: Environment representation $\mathcal{X}, \mathcal{G}$ and localizability map $\mathcal{Z}$ extracted from OpenStreetMaps (a). In (b), the traversable roads are in white, blue refers to non-traversable areas and buildings are in dark blue. In (c), the orange arrows are the roads E and the red dots are their intersections V. In (d), the darker the pixels along the roads, the smaller the localizability covariance.

the computational complexity is often too high to provide useful results for real-world problems [12]. Roy *et al.* [15] proposed the Augmented Markov Decision Process (A-MDP) to approximate the state space of a POMDP. A-MDPs formalize POMDPs as MDPs with an augmented state representation including the uncertainty. Thus, A-MDPs can be solved using the tools of the MDP world. A-MDPs have been used by Hornung *et al.* [6] for planning while minimizing the motion blur of its camera, and by Kawano [8] to control under-actuated blimps. In this paper, we use A-MDPs to plan routes on road networks taking the uncertainty about robot's position into account.

Approaches that incorporate the robot's uncertainty into the planning process are usually referred to as planning in belief space. The belief roadmap [14] and the FIRM [1] generalize probabilistic roadmap algorithm to plan in the belief space. Platt *et al.* [13] assume maximum likelihood observations to define the belief space. The LQG-MP [2] plans using a linear-quadratic controller with Gaussian uncertainty. Most of these approaches compute a fixed path offline and execute it without considering any sensor or process noise. Our approach generates offline a policy that deals with different degrees of uncertainty, and selects online the optimal action given the current belief of the robot.

Candido *et al.* [4] and Indelman *et al.* [7] approach planning in belief space in the continuous domain. However, these approaches are computationally expensive. On the contrary, we consider a discrete space representation and use a compact representation of the robot's belief similar to Bopardikar *et al.* [3] to tackle larger environments and, thus, to take a step towards real world applications.

III. Planning and Localization in Road Networks

A. Metric-Topological Maps

Most probabilistic approaches for robot localization rely on occupancy grid maps, whereas topology graphs are an effective representation for planning. We combine these two representations and represent the environment using a metric-topological map, similar to the hierarchical maps [9].

We define our environment representation by extracting information about buildings and roads from publicly available map services such as OpenStreetMap (OSM) (see for example Fig. 2a). We store this data in a 2D grid map $\mathcal{X}$ in which each cell contains information about its traversability (Fig. 2b). We use $\mathcal{X}$ to estimate the position of the robot assuming it always moves along the roads. In addition to that, we define a topological graph $\mathcal{G} = (V, E)$ over the discretized metric space of $\mathcal{X}$ in which the vertexes $V \subset \mathcal{X}$ are the road intersections and the oriented edges E are the roads connecting them (Fig. 2c). We use $\mathcal{G}$ for planning routes. Note that an edge of $\mathcal{G}$ corresponds to the sequences of traversable cells in $\mathcal{X}$ representing the corresponding road.

B. Localization System

We consider a mobile robot equipped with a 360-degree range sensor that uses a Markov localization system [5] to localize in $\mathcal{X}$. Markov localization estimates the robot's position by considering a probability distribution over $\mathcal{X}$ in form of a histogram over all cells of the grid map, without requiring probabilities to be restricted to any particular class of distributions. As the robot moves and acquires a new scan from the laser range finder, the localization system uses the scan and the wheel odometry to estimate the new position of the robot using Bayes filter.

C. Localizability Map

Given the buildings' footprints and the sensor model of the laser range finder, we can compute an estimate of how scans fired at a location will affect the localization. We compute this prior using the method proposed by Vysotska and Stachniss [16]. It simulates at each location a virtual laser scan by ray-casting the map of the buildings. Then, it translates/rotates the virtual sensor and estimates the error between the scan and the map around its firing location. Considering these errors, it computes a covariance matrix that estimates how well the scan matches the map under position uncertainty. At locations where the surrounding environment has a distinctive structure, the resulting covariance is small, whereas it is large if the surrounding environment is not informative or ambiguous. We compute this prior for each traversable cell in $\mathcal{X}$ and we refer to this as the localizability map $\mathcal{Z}$ (see for example Fig. 2d).

D. MDP-based Planning

Given our representation of the environment $\mathcal{G}$, we can plan routes using a Markov Decision Process (MDP) in which the states are the road intersections V and the actions correspond to selecting roads E at intersections. The transition function allows for transitions between intersections if a road connecting them, and the rewards correspond to the length of the roads. Solving this MDP generates navigation policy that leads the robot to the goal through the shortest path. However, MDPs assume to always know the location of the robot, and this is often not the case in robot navigation. Thus, following a MDP policy in situations with high position uncertainty may lead the robot to take the wrong way and thus to reach the goal through a longer path.

IV. Our Approach to Planning in Road Networks Considering Localization Uncertainty

We propose to improve decision making at intersections by integrating into the planning process the uncertainty about robot's position provided by the localization system. We formulate this planning problem using Augmented MDP (A-MDP) [15]. It efficiently approximates a POMDP by augmenting the conventional MDP state with a statistic about the uncertainty, such as its entropy or covariance. Due to the augmented state representation, transition and reward functions become more complex, but, in their final formulation, A-MDPs have an analogous representation as MDPs, except for a larger number of states. Thus, they can be solved by using the same algorithms as MDPs such as policy iteration.

A. States

Even though our localization system can potentially generate any kind of probability distribution due to its non-parametric nature, we approximate the uncertainty about robot's position *during planning* by a Gaussian distribution with isotropic covariance, and we augment the MDP states with the corresponding variance. Therefore, we define an *augmented state* s as the pair $s = (v, \sigma^2)$ that corresponds to the normal distribution $\mathcal{N}(v, \Sigma)$ defined over $\mathcal{X}$ with $\Sigma = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$. This representation keeps the state space compact by augmenting it by only one dimension and, thus, avoids planning to explode in complexity. As the state s represents a distribution over the discrete space $\mathcal{X}$, we also refer to it as the probability mass function $p(x \mid \mathcal{N}(v, \Sigma))$ or, equivalently, $p(x \mid s)$.

The set of augmented states S is

$$S = \{(v, \sigma^2) \mid v \in V, \sigma^2 \in W\}, \tag{1}$$

where W is a set of variances that discretizes the possible degrees of uncertainty.

B. Actions

In our A-MDP, performing an action corresponds to take a direction at a road intersection, analogously as in MDPs. We assume that every road intersection is a junction of up to 4 roads corresponding to the four cardinal directions. Thus, the set of actions is $A = \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$.

C. Transition Function

The A-MDP transition function $T(s' \mid s, a)$ takes as input an augmented state $s \in S$ and an action $a \in A$, and maps it to a probability distribution of possible A-MDP end states $s' \in S$. As our A-MDP states represent probability distributions,

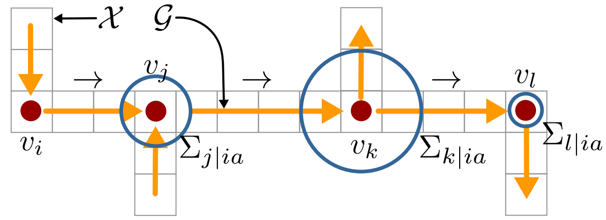


Fig. 3: Estimated uncertainty about robot's position (blue circles) at intersections v_j, v_k, v_l (red dots) when the robot takes action $a =' \rightarrow '$ in v_i represented in the grid map $\mathcal{X}$ underlying the road graph $\mathcal{G}$.

the transition function is more complex to define compared to standard MDPs. We define T in three steps:

1) We compute the posterior probability about robot's position given that it executes a from an intersection v, to which we refer as the *posterior from an intersection* $p(x \mid v, a)$.
2) We compute the *posterior from a state* $p(x \mid s, a)$ given that the belief about the input position of the robot is represented by the state s by combining the possible posteriors from intersections according to s.
3) We map the posterior from a state into our A-MDP state representation to define the *state transitions* $T(s' \mid s, a)$.

Posterior from an intersection: First, we compute the posterior probability about robot's position $p(x \mid v, a)$, $x \in \mathcal{X}$ given that it executes a at v without considering any uncertainty in its input position.

To this end, we *simulate* the robot taking action a at v and moving along the corresponding road in $\mathcal{X}$ according to

$$x_t = g(x_{t-1}, u_t) + \epsilon_t, \; \epsilon_t \sim \mathcal{N}(0, M_t), \tag{2}$$

where g is a linearizable function, u_t is the one-step control corresponding to action a and M_t is the motion noise. Assuming that the belief about robot's position can be approximated as a Gaussian distribution, we estimate the position of the robot while navigating along a road using the prediction step of the Extended Kalman Filter (EKF)

$$p(\hat{x_t} \mid x_{t-1}, u_t) = \mathcal{N}(\hat{\mu_t}, \hat{\Sigma_t}) \tag{3}$$

where $\hat{\mu_t} = g(\mu_{t-1}, u_t)$, $\hat{\Sigma}_t = G_t \Sigma_{t-1} G_t^\top + M_t$, and G_t is the Jacobian of g. As we simulate robot navigation, we do not have measurements to correct the EKF prediction. Instead, we estimate how position uncertainty propagates along the road by combining $\hat{\Sigma}_t$ with the localizability covariance $\Sigma_{\hat{\mu_t}, \mathcal{Z}}$ that estimates how much informative would be a measurement at $\hat{\mu_t}$ to localize the robot:

$$p(x_t \mid x_{t-1}, u_t, \mathcal{Z}) = \mathcal{N}(\hat{\mu_t}, (\hat{\Sigma}_t^{-1} + \Sigma_{\hat{\mu_t}, \mathcal{Z}}^{-1})^{-1}). \tag{4}$$

If intersection v_j is reachable from v_i through an action a as in Fig. 3, we estimate the posterior probability about robot's position of executing this action as the Gaussian distribution $\mathcal{N}(v_j, \Sigma_{j|ia})$ that we compute by recursively applying Eq. (4) along the cells of $\mathcal{X}$ belonging to the corresponding road.

We explicitly model the possibility that the robot might miss an intersection and end up in a successive one while navigating with high position uncertainty. For example, in Fig. 3, while navigating rightwards from v_i, the robot could fail to detect v_j and end up in v_k or in v_l. We compute the probability to detect the intersection v_j so that the smaller the uncertainty $\Sigma_{j|ia}$, the higher the probability to detect it:

$$p_{\text{detect}}(v_j \mid v_i, a) = p(x = v_j \mid \mathcal{N}(v_j, \Sigma_{j|ia})). \tag{5}$$

We compute the posterior $p(x \mid v_i, a)$ of taking action a at intersection v_i by considering the probability to end up in each of the reachable intersections taking action a:

$$\begin{aligned} p(x \mid v_i, a) = \sum\nolimits_{j=1}^{|J|} p(x \mid \mathcal{N}(v_j, \Sigma_{j|ia}))\, p_{\text{detect}}(v_j \mid v_i, a) \\ \cdot \prod\nolimits_{k=1}^{j-1} (1 - p_{\text{detect}}(v_k \mid v_i, a)), \end{aligned} \tag{6}$$

where J is the ordered set of $|J|$ subsequent intersections that the robot may reach by missing the previous intersections. The probability that the robot ends up in each of the J intersections decays according to the probability that a previous one has been detected. If no road exists for executing a at v, we set the posterior to be equal to the input intersection v.

Posterior from a state: Given the posteriors from the intersections, we compute the posterior probability of taking action a given that input belief about the position of the robot is the probability represented by A-MDP state $s \in S$. As the input is a probability distribution about the robot's position, the posterior from a state should represent all of the possible transitions that might occur by executing action a. Thus, we compute the posterior from a state as the weighted sum of the posteriors from the intersections according to the input state s:

$$p(x \mid s, a) = \eta \sum\nolimits_{i=1}^{|V|} p(x \mid v_i, a)\, p(x = v_i \mid s). \tag{7}$$

State Transitions: We define the transition probability between the A-MDP states by computing a correspondence between the posteriors from the states and the A-MDP states S using the Bhattacharyya distance. The Bhattacharyya distance $D_B(p, q)$ measures the similarity between two distributions p and q over the same domain. We define the state transition $T(s' \mid s, a)$ with $s, s' \in S$ according to the Bhattacharyya distance over the domain $\mathcal{X}$ between the posterior $p(x \mid s, a)$ and the distribution represented by s':

$$T(s' \mid s, a) = \eta\, e^{-D_B(p(x|s,a), s')}, \tag{8}$$

where η is a normalization factor and we use the softmax function to transform the distances into probabilities.

D. Reward Function

We define the A-MDP reward function such that the resulting policy makes uncertainty-aware decisions that lead the robot to the goal in average in the minimum time or, equivalently, maximum negative time. Similarly as for the transition function, we first compute the rewards without uncertainty in the input and end position, that we call *reward between intersections* r. Then, we combine the rewards between the intersections to define the A-MDP reward function R.

Assuming that the robot moves with unitary velocity, we define the reward $r(v_i, a, v_j)$ of taking action $a \in A$ from v_i to v_j with $v_i, v_j \in V$ similarly to the MDP reward:

$$r(v_i, a, v_j) = -\ell_{\text{road}}(v_i, a, v_j), \tag{9}$$

where $\ell_{\text{road}}(v_i, a, v_j)$ indicates the length of the road that connects v_i to v_j taking action a. If v_j is not reachable from v_k by taking the action a, we give a penalty as reward

$$r(v_i, a, v_k) = r_{\text{noroad}},\, r_{\text{noroad}} < 0. \tag{10}$$

For each intersection v_i that brings the robot to the goal $v_{\text{goal}} \in V$ through action a, we give a positive reward

$$r(v_i, a, v_{\text{goal}}) = r_{\text{goal}} - r(v_i, a, v_j),\, r_{\text{goal}} \geq 0. \tag{11}$$

We define the reward of taking action a from the A-MDP state s to s', with $s, s' \in S$, by combining the rewards between intersections r according to the distributions corresponding to the input and end states to reflect the uncertainty of the transitions:

$$\begin{aligned} R(s', a, s) = \sum\nolimits_{i=1}^{|V|} p(x = v_i \mid s) \\ \cdot \sum\nolimits_{j=1}^{|V|} p(x = v_j \mid s')\, r(v_i, a, v_j). \end{aligned} \tag{12}$$

E. Solving the A-MDP

In our planning problem we deal in general with non-deterministic transitions. Thus, we compute a policy that tells the robot which action to select at any intersection it might reach. As the A-MDP formulation allows for solving our planning problem as an MDP, we compute the optimal policy π^* using the policy iteration algorithm. Solving A-MDPs has the same computational complexity as MDPs but A-MDPs require a larger number of states, $|S| = |V| \cdot |W|$. POMDPs are PSPACE-complete [12], thus A-MDPs are practically and theoretically much more efficient than POMDPs.

F. Navigation Following an A-MDP Policy

At each step of the robot, the localization system computes an estimate $bel(x)$ over $\mathcal{X}$ about the robot's position as described in Sec. III-B. When the robot recognizes to be at an intersection, it has to make a decision where to navigate. In order to make decisions according to our optimal policy π^*, we transform $bel(x)$ into the A-MDP state $s \in S$ with the minimum Bhattacharyya distance:

$$s_{\text{bel}} = \underset{s \in S}{\operatorname{argmin}}\; D_B(bel(x), s). \tag{13}$$

Thus, the robot takes the action corresponding to the optimal policy $a^* = \pi^*(s_{\text{bel}})$ and keeps navigating along the selected road until it detects the next intersection.

V. Experimental Evaluation

The objective of this work is a planning approach for robot navigation on road networks that explicitly takes the uncertainty about robot's position into account. Our experiments aim at showing that our planner makes different effective navigation decisions depending on the robot's uncertainty, the environment, and the goal location to reach. We furthermore provide comparisons to two baseline approaches.

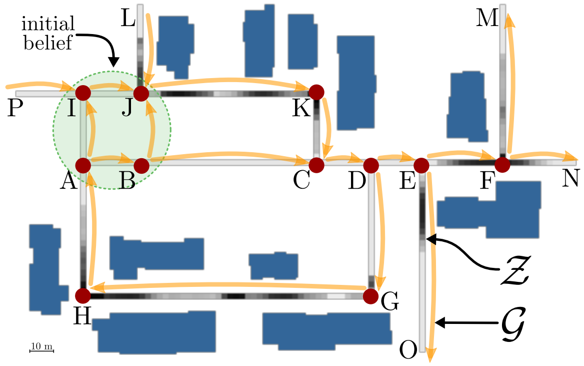


Fig. 4: Environment of Exp. 1: In the graph representation $\mathcal{G}$, intersections are the red dots denoted by letters, whereas road edges are the orange arrows. Buildings are colored in blue. Localizability information $\mathcal{Z}$ along roads is represented such that the darker, the higher the expected localization.

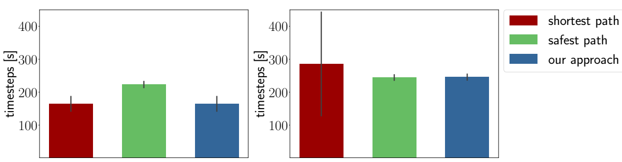


Fig. 5: Avg. travel time of Exp. 1 to the goal F (left) and G (right).

A. Simulator and Baseline

All experiments presented here are simulation experiments. The simulator uses a grid map containing buildings and road information. The robot navigates along the roads and uses building information to simulate laser range observations as well as to compute the localizability map as described in Sec. III-C. The scans and the odometry are affected by noise. The navigation decisions at the intersections are non-deterministic and the probability of missing an intersection is proportional to the variance of the robot's belief. The robot localization system implements Markov localization as described in Sec. III-B.

For comparisons, we consider a shortest path policy similar to the one described in Sec. III-D that assumes the robot to be located at the most likely position given by the localization system. We compare our approach also against a safest decision policy that considers the localizability information to reduce the expected uncertainty about robot's position and by selecting always safe actions.

B. Situation-Aware Action Selection

The first experiment (Exp. 1) is designed to show that our approach reacts appropriately to the situation given the planning problem. Fig. 4 depicts an environment together with the localizability information $\mathcal{Z}$ along the roads. According to the localizability information, the robot expects to localize well along some roads such as $\overline{\mathrm{JK}}, \overline{\mathrm{KC}}$, but finds little structure to localize in others as $\overline{\mathrm{AB}}, \overline{\mathrm{BC}}$ causing a growth in the position uncertainty. Given the initial belief that the robot is at A, B, I, or J with uniform probability (green ellipse), we sample accordingly the actual initial location, and consider two different navigation tasks to show how our approach adapts the action selection depending on the planning problem.

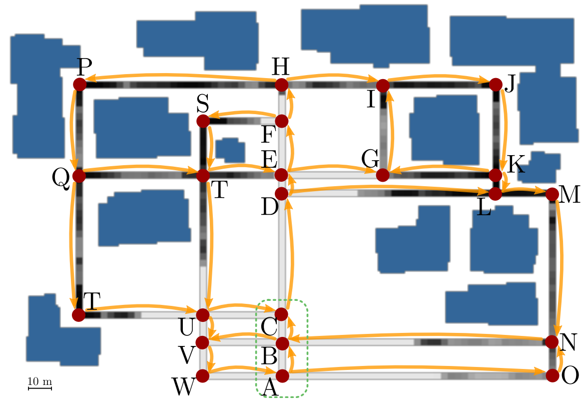


Fig. 6: Environment of Exp. 2: Same notation as in Fig. 4.

First, we set F as the goal location. The shortest path policy seeks to navigate rightwards to reach the goal fast, whereas the safest path policy seeks to go through $\overline{\mathrm{JK}}$ where the localizability is high. The policy generated by our planner performs similarly to the shortest path one. In fact, although the robot cannot localize perfectly along $\overline{\mathrm{AE}}$, it is expected to relocalize along $\overline{\mathrm{EF}}$ and thus to reach safely the goal even following a greedy plan. Fig. 5 (left) shows the average travel time of the three policies. Our policy presents the same performances as the shortest path and outperforms the safest path policy.

The situation changes if we set G as the goal and assume a time penalty corresponding to a long detour if the robot navigates towards O or N. The safest path policy seeks again to go through $\overline{\mathrm{JK}}$ to reduce the uncertainty and take the correct turn at D. Whereas, the shortest path policy leads the robot rightwards to quickly reach D and make the turn to the goal. However, navigating along $\overline{\mathrm{AD}}$, the uncertainty about robot's position grows and, thus, it increases the probability that the robot takes the wrong turn or misses the intersection D. This leads to an overall suboptimal performance, see Fig. 5 (right). As reaching D with large uncertainty may lead the robot to long detours, our planner seeks to reduce the uncertainty before making the turn and, thus, in this case, behaves similarly to the safest path policy. This shows that our planner adapts to the situation by picking the best of both the shortest and the safest path worlds.

C. Uncertainty-Aware Action Selection

The second experiment (Exp. 2) is designed to illustrate how our approach deals with different degrees of uncertainty. To do so, we consider the environment depicted in Fig. 6. The robot starts from A, B, and C with different initial levels of position uncertainty and navigates to the goal G.

Trivially, the shortest path to the goal is to navigate upwards and make a right turn to the goal at E. When the robot is accurately localized, following this path leads it fast and safely to the goal. However, as there is little structure to localize in the environment along $\overline{\mathrm{AE}}$, the uncertainty about the robot's position upon reaching E grows. Reaching E with large uncertainty increases the probability to mismatch the intersections D and E. If the robot expects to be at E

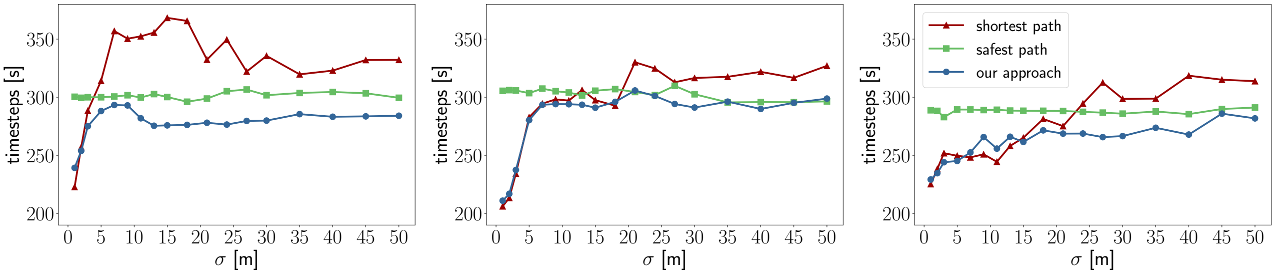


Fig. 7: Avg. travel time of Exp. 2 (Fig. 6) to reach the goal G starting from A, B, C respectively with different uncertainty levels σ.

whereas it is actually at D, the shortest path policy makes the robot turn right leading it to a long detour through L. Large uncertainty increases also the probability that the robot misses to detect E or even F leading also to detours.

The safest path policy seeks to make safe turns at intersections in which the robot is expected to localize well, for example, at the end of the roads or where the localizability is high. Therefore, to reach the goal, it leads the robot upwards to H and makes a safe right turn towards I. From I, it moves the robot rightwards to J, turns to K and, finally, to the goal G. However, the safest path policy always makes safe decisions ignoring the uncertainty about the robot's position while executing the plan. Therefore, it leads the robot through a conservative (and often longer) path also in the situations in which the position uncertainty is small.

Our approach makes decisions by explicitly considering the uncertainty about the position of the robot provided by the localization system. Thus, depending on the degree of uncertainty, it selects the action that leads the robot to the goal trading off safety and travel time.

Fig. 7 shows the performance of the three algorithms in Exp. 2. We considered 18 different levels of uncertainty with σ ranging from 1 to 50 meters and performed for each initial location and uncertainty 200 runs. The safest path policy presents in average similar travel time when varying the initial uncertainty. The shortest path policy shows short travel time when the uncertainty is small but, when the uncertainty grows, it takes in average longer than the safest path to reach the goal. Our approach follows a strategy similar to the shortest path when the uncertainty is small and thus mistakes are unlikely. However, in tricky situations when the uncertainty becomes large, our approach makes decisions similarly to the safest path, thereby avoiding long detours. Therefore, our approach is able to take the appropriate navigation action according to the degree of uncertainty, overall outperforming the shortest and safest path policies.

VI. Conclusion

In this paper, we presented a step towards efficient path planning under uncertainty on road networks. We formulate this problem as an augmented Markov Decision Process that incorporates the robot's position uncertainty into the state space but does not require solving a full POMDP. We define the transition function of the A-MDP by estimating how the robot's belief propagates along the road network through the use of a localization prior. During navigation, we transform the belief provided by the robot's localization system into our state representation to select the optimal action. Our experiments illustrate that our approach performs similarly to the shortest path policy if the uncertainty is small, but outperforms it when the uncertainty is large and the risk of making suboptimal decisions grows. Therefore, our approach is able to trade off safety and travel time by exploiting the knowledge about the robot's uncertainty.

References

[1] A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato. Firm: Feedback controller-based information-state roadmap-a framework for motion planning under uncertainty. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[2] 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*, 30(7):895–913, 2011.

[3] S. Bopardikar, B. Englot, and A. Speranzon. Robust belief roadmap: Planning under uncertain and intermittent sensing. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2014.

[4] S. Candido and S. Hutchinson. Minimum uncertainty robot navigation using information-guided pomdp planning. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2011.

[5] D. Fox, W. Burgard, and S. Thrun. Active markov localization for mobile robots. *Journal on Robotics and Autonomous Systems*, 25(3-4):195–207, 1998.

[6] A. Hornung, H. Strasdat, M. Bennewitz, and W. Burgard. Learning efficient policies for vision-based navigation. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[7] V. Indelman, L. Carlone, and F. Dellaert. Planning under uncertainty in the continuous domain: a generalized belief space approach. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2014.

[8] H. Kawano. Study of path planning method for under-actuated blimp-type uav in stochastic wind disturbance via augmented-mdp. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2011.

[9] K. Konolige, E. Marder-Eppstein, and B. Marthi. Navigation in hybrid metric-topological maps. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2011.

[10] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard. Autonomous Robot Navigation in Highly Populated Pedestrian Zones. *Journal of Field Robotics (JFR)*, 2014.

[11] G. Lidoris, F. Rohrmuller, D. Wollherr, and M. Buss. The autonomous city explorer (ace) project – mobile robot navigation in highly populated urban environments. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2009.

[12] C. Papadimitriou and J. Tsitsiklis. The complexity of markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[13] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez. Belief space planning assuming maximum likelihood observations. In *Proc. of Robotics: Science and Systems (RSS)*, 2010.

[14] S. Prentice and N. Roy. The belief roadmap: Efficient planning in belief space by factoring the covariance. *Intl. Journal of Robotics Research (IJRR)*, 28(11-12):1448–1465, 2009.

[15] N. Roy and S. Thrun. Coastal navigation with mobile robots. In *Advances in Neural Information Processing Systems (NIPS)*, 1999.

[16] 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.