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, path planning is often performed without considering the uncertainty about the robot's position. Uncertainty, however, matters in planning, but considering it often leads to computationally expensive algorithms. 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 belief about the robot's position propagates through the environment. This yields to a planning approach that generates navigation policies able to make decisions according to the degree 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 that also outperform policies that always take the safest actions.**

I. Introduction

Whereas robots often use probabilistic algorithms for localization or mapping, most planning systems compute paths assuming that the robot's position is known. Ignoring position uncertainty during planning may be acceptable if the robot is precisely localized, but it can lead to sub-optimal navigation decisions if the uncertainty is large. Consider for example the situation depicted in Fig. 1. The belief about the robot's position while following the blue path is represented by the black shaded area (the darker, the more likely). This belief indicates that the robot could be in the proximity of intersection A or B. The localization system is not able to disambiguate these intersections as they present similar structure (see right side of Fig. 1). Ignoring the position uncertainty, we would assume the robot to be at the most likely intersection, that is B. Thus, the robot should turn to the right to reach the goal (dotted orange path). However, if the true position of the robot is A (less likely, but possible), turning right would lead it to a detour (dash dotted red path).

In this paper, we investigate the problem of path planning under uncertainty. Uncertainty-aware plans reduce the risk to make wrong turns when the position uncertainty is large. For example, in Fig. 1, the robot could navigate towards intersection C, which has distinctive surrounding and, thus, where the robot is expected to localize accurately. There, it can safely turn towards the goal reducing the risk to go for

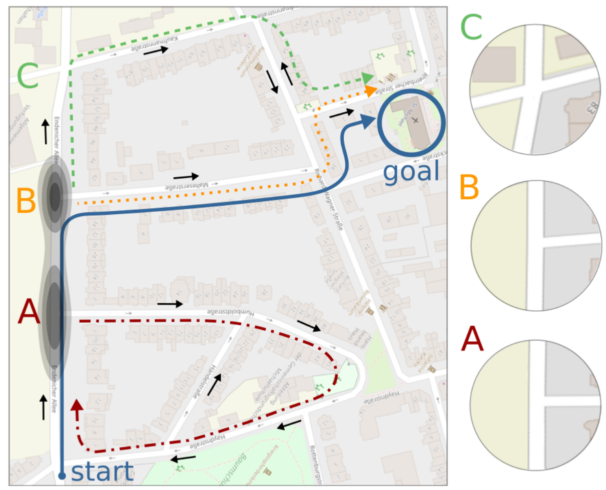


Fig. 1: Robot navigation along the blue path with large position uncertainty. The black shaded area is the belief about robot's position (the darker, the more likely). A, B, C are the road intersections (in detail on the right side). The paths from each intersection are colored respectively in red, orange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

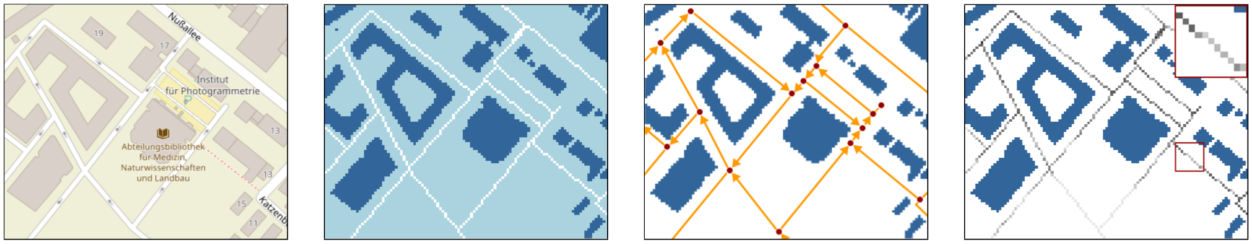
long detours (green). A general formalization for this type of problem is the Partially Observable Markov Decision Process (POMDP). POMDPs, however, become quickly intractable for real-world applications. In this work, we investigate an approximation that is able to consider the localization uncertainty, while being computationally efficient.

The main contribution of this paper is a novel approach to take a step forward towards planning routes on road networks considering the uncertainty about robot's position and action execution. Our approach relies on the Augmented Markov Decision Process (A-MDP) [24], which approximates a POMDP by modeling the uncertainty as part of the state. We employ a localization prior to estimate how the uncertainty propagates along the road network. The resulting policy minimizes the expected travel time while reducing the number of mistakes that the robot makes during navigation with large position uncertainty. Considering explicitly the robot's uncertainty, our planning approach, first, is able to select 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

Path planning in urban environments has received a substantial attention in the robotics community. Several urban navigation robots such as Obelix [18] or the Autonomous City Explorer [19] use A* on topo-metric

All authors are with the University of Bonn, Germany.
We thank Olga Vysotska for the help with the localizability maps.
This work has partly been supported by the German Research Foundation under Germany's Excellence Strategy, EXC-2070 - 390732324 (PhenoRob).



(a) OSM map. (b) Traversability grid map $\mathcal{X}$. (c) Road 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.

maps [16] to navigate. Publicly available map services, such as OpenStreetMap, offer free topo-metric maps of most cities that have been used in various robotic systems for path planning, localization, and mapping [9], [11], [27]. We also use these maps in this work.

The Markov Decision Process, or MDP, allows for optimally solving planning problems in which state is fully observable but the actions are noisy. There are many widely studied algorithms to solve MDPs such as policy iteration. If the state is not observable, the problem turns into a Partially Observable MDP, or POMDP. The computational complexity of POMDPs is often too high to provide useful results for real-world problems [21]. Some approaches such as POMCP [25] or DESPOT [26] use sampling methods to locally solve POMDPs. These approaches require significant resources and planning is limited to short horizons. Roy *et al.* [24] proposed the Augmented Markov Decision Process (A-MDP) to approximate a POMDP. A-MDPs formalize POMDPs as MDPs with an augmented state representation including the uncertainty. Hornung *et al.* [12] use A-MDPs to plan velocity commands for a mobile robot minimizing the motion blur of its camera, whereas Kawano [15] use A-MDPs to control under-actuated blimps under the effects of wind disturbance. 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 uncertainty into the planning process are usually referred to as planning in belief space. The belief roadmap [23] is a variant of a probabilistic roadmap algorithm that plans in belief space for linear Gaussian systems. Platt *et al.* [22] assume maximum likelihood observations and generate policies using linear quadratic regulation. The LQG-MP [2] combines an LQG controller with a Kalman filter to estimate the robot's state along paths and so selects the best candidate path. These approaches compute a path offline without considering the sensor or process noise during the execution. FIRM [1] generalizes probabilistic roadmaps over the belief space and assigns a unique belief to each node taking all possible future observations into account. However, FIRM does not account for reaching a node with a different belief. In contrast, our approach generates offline a policy that deals with different degrees of uncertainty, and selects online the optimal action given the current robot's belief.

Some approaches integrate planning under uncertainty into the SLAM framework. For example, Chaves *et al.* [7] plan revisit paths for loop-closure, whereas Fermin-Leon *et al.* [8] plan for re-localization in the Graph SLAM representation.

Other works such as Candido *et al.* [6], Indelman *et al.* [14] and Van Den Berg *et al.* [3] approach planning in belief space in the continuous domain. These approaches are computationally very expensive and, thus, not suited for city-scale domains. We consider a discrete space representation and use a compact representation of the robot's belief similar to Bopardikar *et al.* [5] to tackle larger environments and, thus, to take a step towards real world applications.

III. Planning and Localization in Road Networks

Typically, localization and planning for robot navigation rely on a map of the environment for estimating the position of the robot and for computing routes to the desired locations.

A. Metric-Topological Maps

Several probabilistic approaches for robot localization rely on occupancy grid maps, whereas topology graphs are an effective representation for planning. We combine these two representations using a metric-topological map, similar to the hierarchical maps [16]. We define our environment representation by extracting information about buildings, roads and their directions 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). Our localization system uses $\mathcal{X}$ to estimate the position of the robot. In addition to that, we consider a more abstracted representation of the environment for planning routes. We define a topological graph $\mathcal{G} = (V, E)$ defined over the discretized metric space $\mathcal{X}$ in which the vertexes $V \subset \mathcal{X}$ are the road intersections and the oriented edges E are the roads between them (Fig. 2c). Therefore, an edge of $\mathcal{G}$ corresponds to a sequences of traversable cells in $\mathcal{X}$.

B. Localization System

We consider a mobile robot equipped with a 360-degree range sensor that uses a Markov localization system [10] to localize in $\mathcal{X}$. Markov localization uses the scans and the odometry to estimate the robot's position using a discrete Bayes filter. The belief about the robot's position is represented by a probability distribution in form of a histogram over all cells of $\mathcal{X}$, without requiring probabilities to be restricted to any particular class of distributions.

C. Localizability Map

Given the buildings' footprints from OSM data and the sensor model of the laser range finder, we can estimate in advance of how scans fired at a given location affect the localization. We compute this prior using the method proposed by Vysotska and Stachniss [27]. It simulates at each location a virtual scan by ray-casting the map. Then, it translates/rotates the virtual sensor and estimates the error between the scan and the map. Considering the decay in the observation likelihood, 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 geometry, the covariance is small, whereas it is large if the surrounding 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 Ignoring Position Uncertainty

Given our representation of the environment $\mathcal{G}$, we can formalize the problem of planning a route as a Markov Decision Process in which the states are the road intersections V and the actions correspond to the roads E. The transition function allows for transitions between intersections connected by a road, and the rewards correspond to the length of the roads. Solving this MDP generates a navigation policy that leads the robot to the goal along the shortest path. However, the MDP assumes that the exact robot's position is always known, and this is often not the case in practice. Therefore, following a MDP policy in situations with large position uncertainty may lead the robot to take the wrong way and go for a long detour.

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 the robot's position provided by the localization system. A common formulation for this problem is a POMDP. However, POMDPs are typically hard to solve. We approximate a POMDP by designing an Augmented MDP (A-MDP) [24] for planning routes on road networks in which we augment the conventional MDP state with the robot's position uncertainty. Due to the augmented state representation, the transition function and the reward function become more complex. But, in their final formulation, A-MDPs have an analogous representation as MDPs, except for a larger number of states. Therefore, A-MDPs can be solved by using the same algorithms as for MDPs.

A. Augmented States

We define the state space of our A-MDP by augmenting the state of the MDP formulation in Sec. III-D with a statistic representing the robot's position uncertainty. Different statistics can be used to represent the uncertainty but, in general, the more compact a representation, the more efficient is the planner. Although our localization system can potentially generate any kind of belief, we assume that *during planning* we can approximate the belief by a Gaussian distribution with isotropic covariance, and we use the corresponding variance to represent the uncertainty. This representation augments the state space by only one dimension and, thus, avoids that planning explodes in complexity.

We define the set of augmented states S as:

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

where V are the road intersections, and W is a set of variances that discretizes the possible degrees of uncertainty. Each augmented state s corresponds to the normal distribution $\mathcal{N}(v, \Sigma)$, with $\Sigma = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$. We refer to s defined over the discrete space $\mathcal{X}$ as the probability mass function $p(x \mid \mathcal{N}(v, \Sigma))$ or, equivalently, $p(x \mid s)$.

B. Actions

In our A-MDP, actions correspond to take a direction at a road intersection, analogously as in the MDP. We assume that every intersection is a junction of up to 4 roads corresponding to the cardinal directions. Thus, $A = \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$ is the set of the actions. If needed, more actions can be added trivially.

C. Transition Function Considering Position Uncertainty

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

1) We compute the robot's position *posterior starting at an intersection*, $p(x \mid v, a)$, without considering any uncertainty in the input position, for all $v \in V$ and $a \in A$.
2) We compute the *posterior starting at a state* s, $p(x \mid s, a)$, by integrating all of the possible posteriors starting at intersections according to the belief corresponding to s.
3) We define the *state transition*, $T(s' \mid s, a)$, by mapping the posterior starting at s to our A-MDP state representation.

Posterior starting at an intersection: For computing the posterior probability $p(x \mid v, a)$ over $\mathcal{X}$, we *simulate* the robot taking action a at intersection v and moving along the corresponding road according to:

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

where g is a linearizable function, u_t is the one-step control corresponding to action a and M_t is the motion noise. We assume that the belief about the position of the robot while navigating along the road can be approximated by a Gaussian distribution, and we estimate it 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

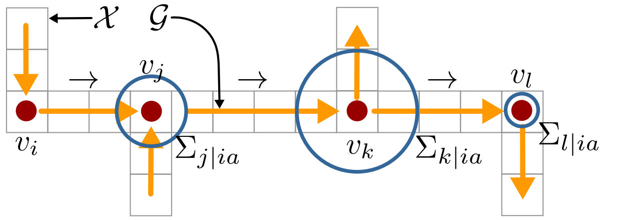


Fig. 3: Uncertainty propagation (blue circles) at intersections v_j, v_k, $v_l \in V$ (red dots) for taking action $a = \rightarrow$ at v_i.

do not have measurements to correct the EKF prediction. Instead, we compute the localizability map $\mathcal{Z}$ and use it to estimate how position uncertainty propagates along the road by combining the covariance from the EKF prediction $\hat{\Sigma}_t$ with the localizability covariance $\Sigma_{\hat{\mu}_t,\mathcal{Z}}$ at $\hat{\mu}_t$:

$$p(x_t \mid x_{t-1}, u_t, \mathcal{Z}) = \mathcal{N}(\mu_t, \Sigma_t). \tag{4}$$

where $\mu_t = \hat{\mu}_t$, and $\Sigma_t = (\hat{\Sigma}_t^{-1} + \Sigma_{\hat{\mu}_t,\mathcal{Z}}^{-1})^{-1}$. We use recursively this procedure to compute the belief about robot's position along each cell of the road. For example, in the topometric map depicted in Fig. 3, we estimate the posterior beliefs about robot's position of navigating from intersection v_i to v_j as the Gaussian distribution $\mathcal{N}(v_j, \Sigma_{j|ia})$ computed by using recursively Eq. (3) and Eq. (4) along the cells of $\mathcal{X}$ belonging to the corresponding road.

While navigating with large position uncertainty, we explicitly model the possibility that the robot misses an intersection and ends up in a successive one. For example, in Fig. 3, while navigating rightwards from v_i, the robot could miss v_j and end up in v_k or v_l. We compute the probability that the robot detects v_j such 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 $p(x \mid v, a)$ of taking action a at intersection v by considering the probability to end up in each of the reachable intersections:

$$p(x \mid v, a) = \sum_{j=1}^{|J|} \mathcal{N}(v_j, \Sigma_{j|ia})\, p_{\text{detect}}(v_j \mid v, a) \cdot \prod_{k=1}^{j-1} (1 - p_{\text{detect}}(v_k \mid v, a)), \tag{6}$$

where J is the ordered set of subsequent intersections that the robot may reach by missing a previous intersection. 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. The residual probability is assigned to the last intersection. If no road exists for executing a at v, we set the posterior to be equal to the input intersection v.

Posterior starting at a state: We consider now that the input position of the robot is represented by the belief corresponding to the A-MDP state $s \in S$. As the input position is described by a probability distribution, the posterior of taking an action should represent all of the possible transitions that might occur. Therefore, we compute the posterior probability of the robot's position $p(x \mid s, a)$ by weighting all of the possible posteriors starting at the intersections according to s:

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

where η is a normalization factor.

State Transitions: We define the transition probability of taking an action a from an A-MDP state s to another s' by computing a correspondence between the posterior belief about robot's position starting at s, $p(x \mid s, a)$, and the belief represented by s'. To this end, we use the Bhattacharyya distance D_B [4] which measures the amount of overlap between between two distributions over the same domain. We prefer the Bhattacharyya distance to the Kullback-Leibler divergence [17] as it is a symmetric measure of the similarity between distributions. We define the state transition $T(s' \mid s, a)$ according to the D_B over $\mathcal{X}$ between the posterior $p(x \mid s, a)$ and s', that we transform into the probability space using the softmax function:

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

where η is a normalization factor.

D. Reward Function Considering Transitions Uncertainty

We define the A-MDP reward function such that the resulting policy makes uncertainty-aware decisions that lead the robot to the goal minimizing in average the travel time. Equivalently, our reward function maximizes the negative time. Similarly as for the transition function, we first compute the rewards for navigating between intersections r that does not consider any uncertainty in the input and end position, and then we combine the rewards between the intersections to define the A-MDP reward function R.

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 as in the MDP:

$$r(v_i, a, v_j) = -\ell(v_i, a, v_j), \tag{9}$$

where ℓ indicates the length of the road and we assume that the robot moves with unitary velocity. If v_k is not reachable from v_i through a, we assign to r a constant penalty:

$$r(v_i, a, v_k) = -r_{\text{noroad}}, \quad \text{with} \quad r_{\text{noroad}} \gg 0. \tag{10}$$

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

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

We compute the reward $R(s', a, s)$ of taking action a from an A-MDP state s to another one s', with $s, s' \in S$, so that it reflects the uncertainty of the transitions. To this end, we combine the rewards between intersections r according to the beliefs corresponding to the input and end states, s and s':

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

E. Solving the A-MDP

The A-MDP formulation allows for computing the optimal policy by using the same tools as for MDPs. We solve our A-MPD by using the policy iteration algorithm [13]. Policy iteration presents a polynomial bound in the number of states and actions for solving MDPs with fixed discounted rewards [20], but in practice it is often much more efficient. Solving A-MDPs has the same complexity as MDPs but A-MDPs require a larger number of states, $|S| = |V| \cdot |W|$. On the contrary, POMDPs are PSPACE-complete [21]. Thus, A-MDPs are practically and theoretically more efficient than POMDPs.

F. Navigation Following an A-MDP Policy

Solving the A-MDP defined above, we obtain a policy π^* that, given the belief about robot's position, selects the optimal action to execute. During navigation, the localization system compute continuously an estimate $bel(x)$ over $\mathcal{X}$ about the robot's position as described in Sec. III-B. As the robot reaches an intersection, we compute the A-MDP state that presents the minimum distance to $bel(x)$:

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

and execute the corresponding action $a^* = \pi^*(s_{\text{bel}})$.

V. Experimental Evaluation

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

A. Experimental Setup and Baseline

All experiments presented here are conducted in a simulated environment that uses a grid map containing buildings and road information extracted from OSM. The robot navigates along the roads and uses the buildings 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 robot localization system implements Markov localization as described in Sec. III-B. The navigation actions at the intersections are non-deterministic, and the probability of missing an intersection is proportional to the variance of the belief about the robot's position.

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 the belief provided by the localization system. We compare our approach also against a safest decision policy that uses the localizability information for selecting always the actions that reduce the position uncertainty.

B. Situation-Aware Action Selection

The first experiment (Exp. 1) is designed to show that our approach reacts appropriately according to the situation and the position uncertainty. We consider the environment depicted in Fig. 4 and assume that the robot goes for a long detour if it navigates towards O, M or N. According to the localizability map $\mathcal{Z}$, 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 such as $\overline{\mathrm{AB}}, \overline{\mathrm{BC}}$ that cause a growth

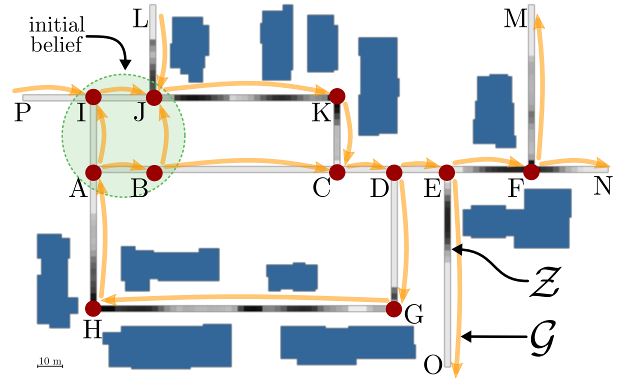


Fig. 4: Environment of Exp. 1. The intersections are the red dots denoted by letters, the roads are the orange arrows, and the buildings' footprints are in blue. The localizability $\mathcal{Z}$ along roads is colored such that the darker, the smaller the covariance.

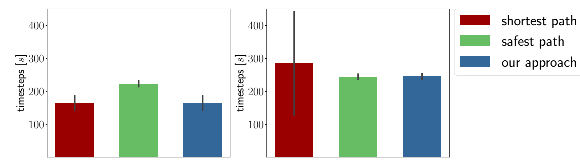


Fig. 5: Avg. travel time of Exp. 1 to the goal F (left) and G (right).

in the position uncertainty. Given the initial belief that the robot is at A, B, I, or J (green circle), we sample the initial true position with uniform probability, and consider two navigation tasks to show that our approach is able to adapt the action selection depending on the planning problem.

First, we set F as the goal location. The shortest path policy leads the robot rightwards to reach the goal fast, whereas the safest path policy seeks to go through $\overline{\mathrm{JK}}$ where the robot is expected to localize better. The policy generated by our approach follows a similar strategy as 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 by 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, outperforming the safest path policy.

The situation changes if we set G as the goal. 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 grows and so the probability that the robot takes the wrong turn or misses intersection D. This leads to an overall suboptimal performance, see Fig. 5 (right). As reaching D with large uncertainty may cause the robot to make mistakes that lead it to long detours, our planner seeks to reduce the uncertainty before making the turn. Thus, in this case, it behaves similarly to the safest path policy. This experiment showcases the ability of our planner to adapt to the situation and uncertainty by picking the best of both the shortest and the safest path worlds.

C. Uncertainty-Aware Action Selection

The second experiment (Exp. 2) illustrates the ability of our approach to deal with different degrees of uncertainty. To

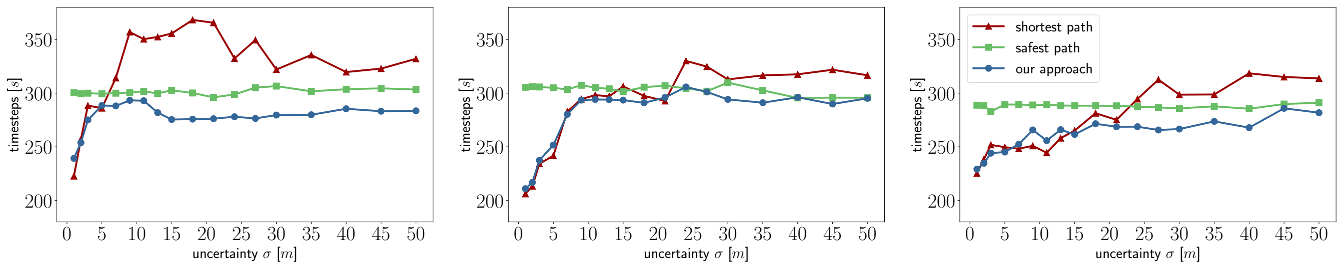


Fig. 6: Avg. travel time of Exp. 2 (Fig. 7) to reach the goal G starting from A, B, C respectively with different levels of uncertainty σ.

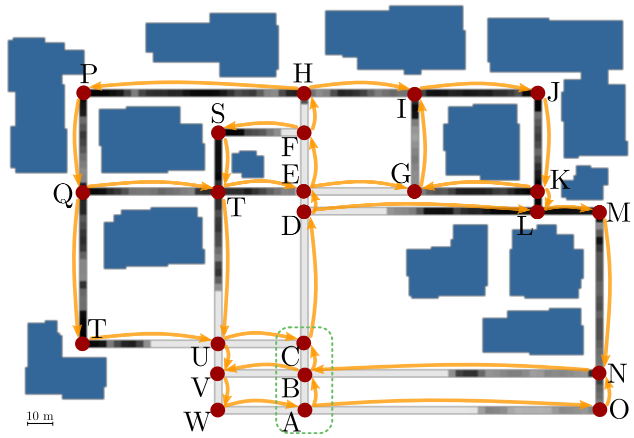


Fig. 7: Environment of Exp. 2: Same notation as in Fig. 4.

this end, we consider the environment depicted in Fig. 7. The robot starts from A, B, or C with different initial position uncertainties 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. If the robot is accurately localized, following this path leads it fast and safely to the goal. However, as there is little structure to localize along $\overline{\mathrm{AE}}$, the position uncertainty 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 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 may also cause the robot to miss E or 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 good. 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 position uncertainty of the robot while executing the plan. Therefore, it follows an overly conservative (and often longer) path also in the situations in which the uncertainty is small.

Our approach makes decisions by explicitly considering the whole belief provided by the localization system. Depending on the degree of uncertainty, it selects the action that leads the robot to the goal trading off safety and travel time. The performance of the three policies in Exp. 2 are shown in Fig. 6. We considered 18 different levels of uncertainty with σ ranging from 1 to 50 meters, and performed 200 runs for each initial location and uncertainty. 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.

D. Discussion

Even though our approach for planning routes considering position uncertainty is more efficient than a POMDP, the complexity to compute a policy at city-scale, where the number of roads and intersections is very large, is still high. However, planning under uncertainty is more relevant in practice at a local scale, where mistakes at intersections can lead to significant detours. Therefore, planning at a city-scale can be more effective by combining, in a hierarchical manner, our approach with a higher level planner that plans paths in a coarse representation, and uses our policy at a finer level to make appropriate, uncertainty-aware, local decisions where they have the highest impact.

VI. Conclusion

In this paper, we presented an approach for 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 without requiring to solve a full POMDP. We define the A-MDP transition function by estimating robot's belief propagation along the road network through the use of a localization prior. During navigation, we match the belief provided by the localization system with the A-MDP 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: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements. *Intl. Journal of Robotics Research*, 33(2):268–304, 2014.

[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] 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*, 31(11):1263–1278, 2012.

[4] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, 35:99–109, 1943.

[5] 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.

[6] 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.

[7] S. Chaves, A. Kim, and R. Eustice. Opportunistic sampling-based planning for active visual slam. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3073–3080, 2014.

[8] L. Fermin-Leon, J. Neira, and J. A. Castellanos. Path planning in graph slam using expected uncertainty. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 4594–4601, 2016.

[9] G. Floros, B. van der Zander, and B Leibe. Openstreetslam: Global vehicle localization using openstreetmaps. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2013.

[10] 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.

[11] M. Hentschel and B. Wagner. Autonomous robot navigation based on openstreetmap geodata. In *IEEE Trans. on Intelligent Transportation Systems (ITS)*, 2010.

[12] 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.

[13] R. Howard. Dynamic programming and markov processes. 1964.

[14] V. Indelman, L. Carlone, and F. Dellaert. Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments. *Intl. Journal of Robotics Research*, 34(7):849–882, 2015.

[15] 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.

[16] 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.

[17] S. Kullback and R. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[18] 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.

[19] 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.

[20] M. Littman, T. Dean, and L. Kaelbling. On the complexity of solving markov decision problems. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 394–402, 1995.

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

[22] 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.

[23] 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.

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

[25] D. Silver and J. Veness. Monte-carlo planning in large pomdps. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[26] A. Somani, N. Ye, D. Hsu, and W. S. Lee. Despot: Online pomdp planning with regularization. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[27] 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.