High-speed Safe Trajectory Planning in Confidence-rich Maps

Eric Heiden[1], Karol Hausman[1], Gaurav S. Sukhatme[1], Ali-akbar Agha-mohammadi[2]

I. Introduction

Despite significant progress in active perception methods, one of the greatest challenges toward full flight autonomy of UAVs remains the high-speed flight in unknown and unstructured environments.

Leveraging the richer representation of occupancy in confidence-aware maps [1], we address the problem of planning high-speed motions in unknown environments. While considering the future map uncertainty, the proposed simultaneous mapping and planning framework (SMAP) actively reduces the number of *surprises*, i.e. when the system observes an element of the environment that it did not consider before, leading to fewer trajectory re-planning steps and faster flight with better predictability and higher safety.

The key contributions of our approach are:

- A thorough analysis of a probabilistic safety measure for trajectories.
- Incorporating the confidence-rich map representation into a planning framework that considers future map predictions.
- A novel cost function that utilizes the estimate of the covariance of the map, and enables faster and safer plans.

II. Background and Problem Formulation

A. Confidence-rich maps

A confidence-rich map [1] is an alternative map representation for occupancy grid mapping which not only stores the expected occupancy but the probability distribution of such estimate in every voxel.

This richer map representation has several advantages over traditional occupancy grids: (i) it relaxes the assumption of fully independent voxels and considers the coupling between voxel occupancies along the measurement ray, (ii) it obviates the hand-engineering of an inverse sensor model and instead proposes the sensor cause model which can be derived from the forward sensor model, and most importantly, (iii) it provides consistent confidence values over occupancy estimates that can be used in planning to compute safer trajectories.

The complete mapping problem is defined as estimating the map m based on obtained depth measurements and robot poses. We denote the sensor measurement at the k-th time step by z_k and the sensor configuration at the k-th time step as xv_k. By formulating the problem in a Bayesian framework, we compress the information obtained from past measurements $z_{0:k} = \{z_0, \cdots, z_k\}$ and $xv_{0:k} = \{xv_0, \cdots, xv_k\}$ to create a probability distribution (belief) of the map m, i.e., $\bar{b}_k^m = p(m|z_{0:k}, xv_{0:k})$. Due to challenges in storing and updating such a high-dimensional belief, grid mapping methods typically start from individual cells such that the map probability density function (pdf) is represented by the collection of individual voxel pdfs (marginal pdfs), instead of the full joint pdf.

In [1] a method was proposed to recursively compute these marginals while taking into account the correlation between nearby cells. Here, we abstract this mapping mechanism to:

$$b_{k+1}^{m^i} = \tau^{m^i}(b_k^m, z_{k+1}, xv_{k+1}) \tag{1}$$

where τ (mapper) updates the current belief of the i-th voxel based on the last measurement and all surrounding voxels.

B. Trajectory Representation and Optimization

Similar to [2] and [3], we represent a trajectory by a piece-wise polynomial $\mathbf{x}(t)$ (see [4] for more details).

Since the system can be undetermined with a sufficiently high polynomial degree, we can use the left null-space of the constraint matrix as the optimization space. This converts an optimization problem over the space of waypoint- and continuity-satisfying piece-wise polynomials from a constrained problem into a smaller, unconstrained problem over the null-space weights $\mathbf{r}$.

III. Reachability

A. Traditional Measure of Reachability

The *reachability* R of a trajectory $\mathbf{x}$ can be computed as follows:

$$R = \Pr(S^1, S^2, \cdots, S^n) \tag{2}$$

where, S^i denotes the event of surviving voxel i, while the trajectory is assumed to intersect with voxels $1, 2, \cdots, n$ in the map. Using Bayes rule the reachability is computed as

$$R = \Pr(S^1)\Pr(S^2|S^1)\cdots\Pr(S^n|S^{n-1}, \cdots, S^1) \tag{3}$$

To survive the whole path, the robot needs to survive every one of these voxels, which leads to:

$$R = \prod_{i=1}^{n}\left(1 - m^i\right), \tag{4}$$

where m^i is the density of the i-th voxel. This is the traditional measure used in reachability calculations [5].

*This work was not supported by any organization

[1] Eric Heiden, Karol Hausman and Gaurav S. Sukhatme are with the Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA heiden@usc.edu

[2] Ali-akbar Agha-mohammadi is with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA

B. Reachability as a Product Integral

Reachability, as defined in Eq. 4, describes the binary case in which a robot can either be at voxel i or not. Hence, this measure is only informed based on the occupancy of intersecting voxels without taking into account what fraction of a voxel has been traversed. However, interpreting the voxel occupancy as a density provides a physically more accurate representation of the environment: if we subdivide a three-dimensional voxel with occupancy 0.5 into 8 smaller voxels, they have, without taking further information into account, the same density. Traversing the original (larger) voxel should therefore yield the same reachability as if the smaller voxels were traversed in the same way. In contrast to this interpretation, surviving the larger voxel would have a higher probability (0.5) than traversing its subdivided voxels (e.g. 0.25 if traversing along one dimension) as the voxel reachabilities are multiplied based on the traditional measure of reachability. To resolve this issue, we propose a new interpretation of reachability that accounts for the fraction of the voxel which was traversed by the robot following the trajectory.

We assume time to be continuous such that Eq. 4 is represented by the product integral $\prod_0^T (1 - m(\mathbf{x}(t)))^{\mathrm{d}t}$. By choosing n intermediate points at small time steps t_i ($i = 0..n;\ 0 \leq t_i \leq T$) with constant time interval Δt, this product integral is computed as:

$$R_t(\mathbf{x}) = \lim_{\Delta t \to 0} \prod_{i=0}^{n} (1 - m(\mathbf{x}(t_i)))^{\Delta t} \tag{5}$$

The continuous reachability representation must consider the *distance* travelled through a voxel which requires a geometric reformulation of the trajectory. The arc-length travelled along trajectory $\mathbf{x}(t)$ can be approximated by sampling at sufficiently narrow time steps t_i such that the reformulation of reachability is as follows:

$$R(\mathbf{x}) \approx \lim_{\Delta t \to 0} \prod_{i=0}^{n} (1 - m(\mathbf{x}(t_i)))^{||\mathbf{x}(t_i) - \mathbf{x}(t_i + \Delta t)||} \tag{6}$$

IV. Map Prediction using Confidence-rich Maps

A common practice in the belief space planning literature is to use the most likely future observations as the representative of the future observations to reason about the evolution of belief. Let us denote the most likely observation at the n-th step by:

$$z_n^{ml} = \arg\max_z p(z | b_n^m, xv_n) \tag{7}$$

Using the most likely observations, we can compute most likely future map beliefs:

$$b_{n+1}^{m^i,ml} = \tau^i(b_n^{m,ml}, z_{n+1}^{ml}, xv_{n+1}), \qquad n \geq k \tag{8}$$

where, $b_k^{m^i,ml} = b_k^{m^i}$.

These quantities can be used to compute the mean and the variance of the future reachability and therefore, serve as good measures of the final cost function that incorporates the active component into high-speed trajectory planning. For more details on map prediction, see [1].

V. Cost Function for High-speed Safe Trajectories

By treating the reachability of a trajectory as a product of random variables representing the individual voxel reachabilities from which the mean and the variance of reachability can be derived [4], we aim to find a measure that combines the two in a principled manner to obtain a utility function for reliable high-speed trajectories.

Resulting from the theory on multi-armed bandit problems, the *Lower Confidence Bound* (LCB) [6] is a commonly used metric to drive a no-regret decision making strategy. Similarly, our objective is to find the safest trajectory by combining the reachability estimate and our confidence in it, which can be formulated as follows:

$$LCB(\mathbf{x}) = \mathbb{E}[R(\mathbf{x})] - \kappa\sigma[R(\mathbf{x})]. \tag{9}$$

When incorporating the standard deviation into our objective, we are able to avoid regions where pure reachability-based methods would be overconfident by ignoring the high uncertainty of the reachability estimate.

A. Algorithm Summary

The proposed method instructs the robot to follow the current trajectory $\mathbf{x}$ parameterized by $\mathbf{r}$, take measurements and update the belief map [4]. This process continues until the time limit has been reached or the reachability of the next k trajectory positions $\mathbf{x}(t), \mathbf{x}(t + \Delta t), \ldots, \mathbf{x}(t + k\Delta t)$ falls below a predefined threshold $1 - \epsilon$. In this case, re-planning is triggered and the trajectory optimization step maximizes the Lower Confidence Bound through simulation of trajectories while taking most likely observations (Eq. 7) and updating an imaginary map $b^{m,ml}$ (Eq. 8).

VI. Simulation Experiments

We evaluate the proposed method in a simulated environment where two rectangular obstacles form a tunnel which the robot has to traverse in order to reach the goal. The robot is equipped with a depth camera which always points forward along the robot's trajectory, and utilizes 16 horizontally arranged range sensors that span a field of view of 90°. The voxels have a side length of $0.1m$ and are arranged in a 20×20 grid. Measurements are perturbed by zero-mean Gaussian noise with std dev $\sigma = 0.2m$.

As a baseline for comparison, we set κ from Eq. 9 to 0, causing the optimization based on LCB to become a pure reachability-based maximization. This method is compared to our LCB-based optimization using different values for κ on various metrics. Besides the consideration of reachability of the remaining part of the trajectory (Fig. 1 center), we compare the two methods by the measure of *surprise*:

$$\Delta\mathbb{E}[R(\mathbf{x}(t))] = \Big|\mathbb{E}[R(\mathbf{x}(t))] - \mathbb{E}[R(\mathbf{x}(t - \Delta t))]\Big| \tag{10}$$

which is computed at every time step.

As shown in Fig. 1, the trajectory that is solely optimized for reachability makes a sharp turn before entering the tunnel, which not only slows down the robot but also increases

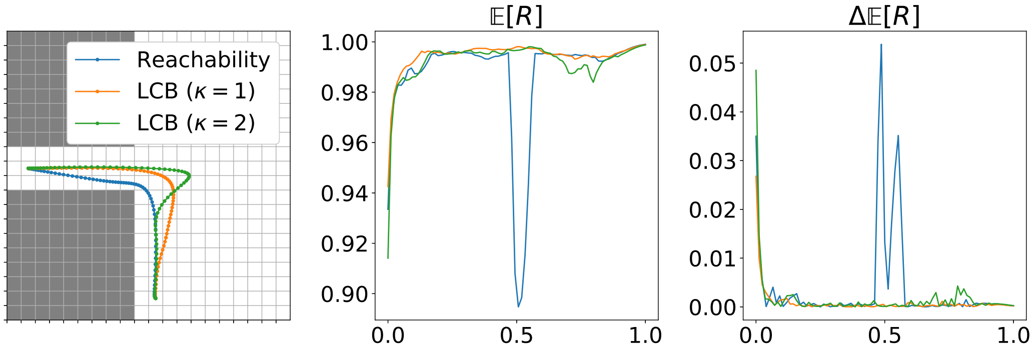

Fig. 1. Evaluation of pure reachability-based trajectory optimization vs. Lower Confidence Bound (LCB) as cost function with different κ parameters. Left: the obstacles (grey) and trajectories with measurement positions (dots along solid lines). Middle: the expected reachability of the remaining course of the trajectory starting from current time step t. Right: the surprise in reachability (Eq. 10) at time t.

the "surprise" since most of the remaining path is hidden behind the corner. Optimizing for the lower confidence bound resolves both issues: the robot makes a wider turn to reduce estimation uncertainty which allows faster trajectory execution and a significant reduction in surprise. Not only does our method outperform the reachability baseline w.r.t. surprise, but the overall expected reachability of the remaining trajectory is better most of the time as well.

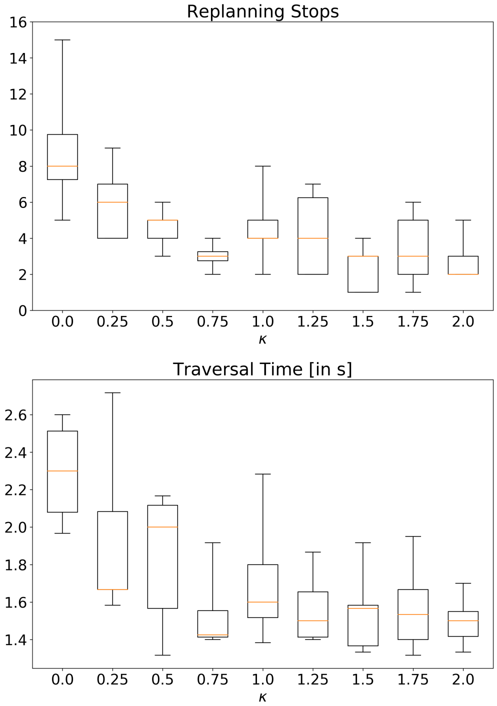

Fig. 2. Number of necessary stops for re-planning until the goal is reached (top) and total time for trajectory execution (bottom) over different values for κ (ranging from 0 to 2) in the environment from Fig. 1. Due to the stochasticity of the range sensor measurements and the compounding effects on the map belief and the generated trajectories, 5 sample runs have been simulated for every value of κ and reported in these two box plots.

Besides generating safer trajectories, the LCB-measure as basis for trajectory optimization also yields fewer stops for re-planning as well as faster traversal times. As shown in Fig. 2, compared to pure reachability-based planning, the proposed method utilizing a Lower Confidence Bound with $\kappa > 0$ led to a reduction in re-planning stops of up to 70% and a speed-up in traversal time by up to 34%.

VII. Conclusion

We present a method that enables planning of high-speed and safe motions through confidence-rich maps. Specifically, we predict the future uncertainty of the confidence-rich map and optimize trajectories with minimum surprise in reachability. Besides faster *and* safer trajectories, simulation experiments indicate that the presented trajectory optimization method leads to fewer re-planning stops than the traditional reachability-based optimization.

References

[1] A.-a. Agha-mohammadi, "SMAP: Simultaneous mapping and planning on occupancy grids," *arXiv preprint arXiv:1608.04712*, 2016.

[2] J. Müller and G. Sukhatme, "Risk-aware trajectory generation with application to safe quadrotor landing," in *International Conference on Intelligent Robots and Systems (IROS)*. Chicago, IL, USA: IEEE, September 2014.

[3] K. Hausman, J. Preiss, G. S. Sukhatme, and S. Weiss, "Observability-aware trajectory optimization for self-calibration with application to uavs," *IEEE Robotics and Automation Letters*, 2017.

[4] E. Heiden, K. Hausman, G. S. Sukhatme, and A.-a. Agha-mohammadi, "Planning high-speed safe trajectories in confidence-rich maps," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017.

[5] P. E. Missiuro and N. Roy, "Adapting probabilistic roadmaps to handle uncertain maps," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2006, pp. 1261–1267.

[6] P. Auer, "Using confidence bounds for exploitation-exploration trade-offs," *Journal of Machine Learning Research (JMLR)*, vol. 3, no. Nov, pp. 397–422, 2002.