Planning High-speed Safe Trajectories in Confidence-rich Maps

Author Names Omitted for Anonymous Review.

***Abstract*— Planning safe, high-speed trajectories in unknown environments remains a major roadblock in the way toward achieving fast autonomous flight. Current state-of-the-art planning approaches use sampling-based methods or trajectory optimization to obtain fast trajectories, whose safety is evaluated by taking into account the current state estimate of the environment. In unknown environments, however, this leads to numerous stops caused by the need for re-planning the trajectory due to unexpected obstacles. In this paper, we propose to use an active perception paradigm for planning. We predict the future uncertainty of the map and optimize trajectories to minimize re-planning risk. This leads to faster *and* safer trajectories. We evaluate the proposed planning approach in a series of simulation experiments, which show that we are able to achieve safer trajectories with a smaller number of re-planning stops and faster speeds.**

I. Introduction

Despite impressive progress in the area of UAVs, high-speed flight in unknown, unstructured environments still remains one of the biggest challenges towards full flight autonomy. In recent years, researchers have achieved impressive results on different aspects of this problem such as fast localization [17], novel sensors [14], aggressive low-level controllers [7] and self-calibration [11]. More recently, many groups have started looking into using active perception methods. These have led to results that were not possible with traditional techniques that consider the different aspects of the fast-flight problem in isolation [8, 9].

In this work, we develop the idea of using active perception to enable fast flight. In particular, we address the problem of planning high-speed motions in confidence-rich maps while considering the future map uncertainty. A confidence-rich map [2] is a planning-oriented representation of the environment and an extension to the traditional occupancy grip map. It provides a fully probabilistic map representation that has access to an accurate estimate of the variance of the voxel occupancy estimate, which is necessary for active perception techniques.

While the community has given a lot attention to different aspects of planning under uncertainty (e.g., planning under location uncertainty [3] or under target uncertainty [10]), there has been relatively less emphasis on the uncertainty of the map, which has arguably the biggest influence on high-speed flight. As an example, consider a quadrotor flying in an unknown, obstacle rich environment, trying to reach a designated goal as fast as possible. Every time the system is "surprised" by an element of the map that it did not consider before, it must to re-plan its motion entirely using the newly acquired information. This typically causes the system to slow down. The number of such "surprises" increases when using low-cost light sensors (e.g. stereo cameras with small baselines that are commonly used on agile micro UAVs, Fig. 1). In this work, we aim to limit the number of surprises, enabling faster flight with better predictability and higher safety.

Fig. 1. An example of a micro aerial vehicle capable of fast and dynamic flight up to 20mph. The vehicle is equipped with stereo cameras with a baseline of 80mm which are used as the main depth sensor.

The key contributions of our approach are:

- A thorough analysis of a probabilistic safety measure for a trajectory (including the proposed improvements).
- Incorporating the confidence-rich map representation into a planning framework that includes future map predictions.
- A novel cost function that utilizes the estimate of the covariance of the map and enables faster and safer plans.

We evaluate the proposed planning approach in a series of simulation experiments which serve as a proof of concept to demonstrate the performance of the method.

II. Related Work

Planning under uncertainty has enjoyed much success in recent years. Examples include planning under location uncertainty for multi-robot systems [10], planning under environment uncertainty for humanoids and medical robots [6, 12], and planning under motion and sensing uncertainty [3, 18]. There has been a relatively smaller body of work on planning in uncertain maps e.g., Missiuro and Roy [16] consider the map uncertainty by extending the probabilistic roadmaps framework [13] to cope with uncertain maps using different sampling strategies.

Planning in uncertain maps requires a probabilistic representation of the map in a format suitable for the robot. Even though there have been multiple approaches to fast

mapping for robotic vehicles [19, 24], there has been less work that includes the uncertainty of the map. Examples include O'Callaghan and Ramos [20] wherein a mapping approach is presented based on Gaussian Processes that includes the uncertainty and relaxes the voxel-independence assumption. Later on, the same group proposed an extension [21] that yields significantly better computation complexity while maintaining the probabilistic advantages previously developed. In this work, we adopt the method recently proposed by [2] that is able to represent the full probability distribution over occupancy levels while keeping the computationally-efficient voxel map representation.

Since we aim for not only incorporating the uncertainty of the map into the trajectory optimization framework but also high speed of the resulting trajectory, we present the related work in planning dynamic trajectories. Planning high-speed motions is an area that emerged recently with advances in mechanical design and robust low-level controllers for flying vehicles. In Mellinger and Kumar [15] a trajectory optimization framework is described that enables fast generation of minimum snap trajectories based on the time polynomial representation of the trajectory. A similar representation was used in [11, 18, 22] to generate safe trajectories that consider the uncertainty of the system. A different approach to planning high-speed motions in the presence of uncertainty was presented by [8], where the authors present an active sensing method to fly through narrow gaps. We believe that the idea of perception-oriented control presented in this work shows great potential for fast-flight planning and we use this paradigm in this paper.

There exist other approaches to the problem of planning under uncertainty that focus on learning. For example, Richter and Roy [23] train a guidance function to give the robot greater visibility in the unknown parts of the environment. Other learning methods use high-dimensional function approximators (such as neural networks) to cope with the difficulty of planning safe motions straight from the sensory output [25]. Although these approaches show great potential, they come with significant drawbacks such as requiring access to a lot of training data, limitation to the training dataset, and non-probabilistic interpretation of the results. In contrast, in this work, we show an approach that does not rely on training data, and instead uses trajectory optimization to plan for high-speed safe motions while considering the future uncertainty of the map.

III. Background and Problem Formulation

A. Confidence-rich maps

A confidence-rich map [2] is a planning-oriented representation of the environment. It provides an alternative map representation for occupancy grid mapping by storing richer data in each cell. This extension leads to a number of advantages over traditional occupancy grids: (i) it relaxes the full-independence assumption and incorporates the coupling between voxels within the measurement cone into the mapping scheme, (ii) it relaxes the need for hand-engineering an inverse sensor model and proposes the sensor cause model that can be derived from the forward sensor model, and most importantly, (iii) it provides consistent confidence values over occupancy estimates that can be reliably used in planning.

Next, we briefly discuss the theory behind confidence-rich maps. Let $G = [G^1, \cdots, G^n]$ be an n-voxel grid overlaid on the 3D (or 2D) environment. An occupancy map $m = [m^1, \cdots, m^n]$ is defined as a set of values over this grid. We adopt a more general definition of occupancy where $m^i \in [0, 1]$ denotes what fraction of the corresponding voxel is occupied. $m^i = 1$ when the i-th voxel is fully occupied and $m^i = 0$ when it is free.

The full mapping problem is defined as estimating the map m based on obtained 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 with xv_k. 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 (marginal distributions). In other words, the map probability density function (pdf) is represented by the collection of individual voxel pdfs (marginal pdfs), instead of the full joint pdf.

$$b_k^m \equiv (b_k^{m^i})_{i=1}^n, \qquad b_k^{m^i} = p(m^i|z_{0:k}, xv_{0:k}). \tag{1}$$

In [2] a method was proposed to recursively compute these marginals while taking the correlation with nearby cells into account. Here, we abstract this mapping mechanism to:

$$b_{k+1}^{m^i} = \tau^{m^i}(b_k^m, z_{k+1}, xv_{k+1}) \tag{2}$$

where τ (mapper) updates the current belief of the i-th voxel based on the last measurement and all surrounding voxels. The mapping system is designed for planning purposes, as it captures the full pdf of the map which includes the variance (i.e., the confidence) of information in the map. This confidence is crucial for the quantification of safety and reliability. We exploit this in this as a metric in planning.

B. Trajectory Representation and Optimization

Similar to [18] and [11], we represent a trajectory by a k-dimensional, d-degree, q-piece piecewise polynomial:

$$\mathbf{x}(t) = \begin{cases} P_1\mathbf{t}(t) & \text{if } t_0 \le t < t_1 \\ \vdots & \\ P_q\mathbf{t}(t) & \text{if } t_{q-1} \le t \le t_q, \end{cases}$$

where P_i is the $k \times (d+1)$ matrix of polynomial coefficients for the ith polynomial piece, and $\mathbf{t}$ is the time vector, i.e.:

$$\mathbf{t}(t) = \begin{bmatrix} t^0 & t^1 & \dots & t^d \end{bmatrix}^T.$$

Using this representation, we can represent simple constraints on positions and their derivatives at certain times with a system of linear equations, For example:

$$\begin{bmatrix} \mathbf{x}(0) & \dot{\mathbf{x}}(0) & \ddot{\mathbf{x}}(0) \end{bmatrix} = P_1 \begin{bmatrix} \mathbf{t}(0) & \dot{\mathbf{t}}(0) & \ddot{\mathbf{t}}(0) \end{bmatrix} \tag{3}$$

expresses initial position, velocity, and acceleration constraints on a trajectory. The vector $\dot{\mathbf{t}}$ is the trivial derivative such that $\dot{\mathbf{x}}(t) = P_i \dot{\mathbf{t}}(t)$.

Since we employ multiple polynomial pieces to extend the expressiveness of the trajectory representation, we must ensure the continuity of the trajectory at the knots. In particular, we enforce the trajectory to be continuous up to the β-th derivative:

$$\mathbf{T}_i = \begin{bmatrix} \mathbf{t}(t_i) & \dot{\mathbf{t}}(t_i) & \dots & \overset{(\beta)}{\mathbf{t}}(t_i) \end{bmatrix}. \tag{4}$$

We thus formulate smoothness constraints as a linear system which, in combination with equations in the form of Eq. 3, completely expresses the trajectory constraints:

$$\begin{aligned} \begin{bmatrix} P_1 & \dots & P_q \end{bmatrix} & \begin{bmatrix} \mathbf{T}_0 & \mathbf{T}_1 & & & & \\ & -\mathbf{T}_1 & \mathbf{T}_2 & & & \\ \vdots & & \ddots & \ddots & & \vdots \\ & & & -\mathbf{T}_{q-2} & \mathbf{T}_{q-1} & \\ & & & & -\mathbf{T}_{q-1} & \mathbf{T}_q \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{x}(0) & \dot{\mathbf{x}}(0) & \dots & \mathbf{0} & \dots & \ddot{\mathbf{x}}(T) \end{bmatrix}, \end{aligned} \tag{5}$$

or in short:

$$\bar{\mathbf{P}}\mathbf{T} = \mathbf{c}. \tag{6}$$

Since the system can be undetermined with a high enough degree of the polynomials, we can use the left nullspace of the constraint matrix as the optimization space. This converts an optimization problem over the space of waypoint- and continuity-satisfying piecewise polynomials from a constrained problem into a smaller, unconstrained problem over the null space weights. In particular:

$$\bar{\mathbf{P}}^* = \mathbf{c}\mathbf{T}^+ + \mathbf{r} Null(\mathbf{T}^T)^T, \tag{7}$$

where $\mathbf{T}^+$ is the pseudo inverse of the time matrix $\mathbf{T}$, $\mathbf{r}$ is a row vector of the null space weights, which are the optimization variables. Additional constraints related to the system's physical limits are incorporated into the optimization problem as nonlinear inequalities.

IV. Reachability

A. Traditional Measure of Reachability

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

$$R = \Pr(S^1, S^2, \cdots, S^n) \tag{8}$$

where, S^i denotes the event of surviving voxel i, and we assume the robot trajectory intersects with voxels $1, 2, \cdots, n$ in the map. Using Bayes rule we can compute the reachability:

$$R = \Pr(S^1) \Pr(S^2|S^1) \cdots \Pr(S^n|S^{n-1}, \cdots, S^1) \tag{9}$$

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

$$R = \prod_{i=1}^{n} \left(1 - m^i\right), \tag{10}$$

where m^i is the density of the i-th voxel. This is the traditional measure used in reachability calculations [16].

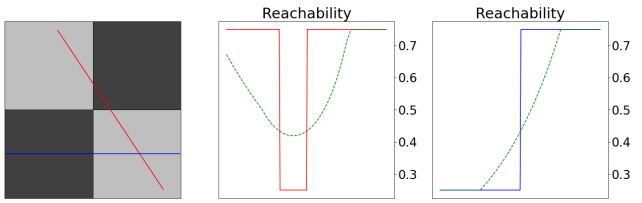

Fig. 2. Example of trajectories that traverse a 2×2 voxel grid (left), which are sampled for reachability along 200 steps. Middle: the reachability profile for the red trajectory as the map is sampled along the trajectory. The green, dashed curve represents the interpolated reachability based on Eq. 15. Right: the reachability profile for the blue trajectory and its corresponding interpolated reachability curve (green, dashed).

B. Reachability as a Product Integral

Reachability, as defined in Eq. 10, describes the binary case in which a robot can either be at voxel i or not. Hence, the trajectory is only considered based on the voxels it traverses without taking into account how much of a voxel has been traversed. 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 the binary case, however, 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. In the following, we propose a new interpretation of reachability that accounts for the fraction of time that a robot has stayed in a voxel while following a trajectory.

We assume time to be continuous such that Eq. 10 can be 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 defined as:

$$R_t(\mathbf{x}) = \lim_{\Delta t \to 0} \prod_{i=0}^{n} \left(1 - m(\mathbf{x}(t_i))\right)^{\Delta t} \tag{11}$$

Figure 2 exemplifies the computation of reachability using Eq. 11. The lighter and darker voxels have occupancies m of 0.25 and 0.75, respectively. According to Eq. 10, the reachability of the red trajectory is equal to 0.14, while Eq. 11 results in the reachability of 0.37 when taking 200 samples along the trajectory. In the latter case, the top right voxel of high occupancy is discounted because of the small number of samples that were taken in this cell. On the other hand, Eq. 10 treats the traversal of the top right voxel the same as if it was traversed through its diagonal, which leads to an underestimated measure of reachabilty. The horizontal trajectory (blue) traverses two voxels and samples over the whole length of both voxels (side length 1). In this special case, both reachability representations yield the same result.

While trajectories are generally defined with respect to time, the evaluation of traversed voxels by the trajectory requires a geometric representation that focuses on the

shape of the trajectory, i.e. the path. Instead of time t, the continuous reachability representation must consider the *distance* travelled through a voxel. The distance travelled along trajectory $\mathbf{x}(t)$ over the time interval $[t_1, t_2]$ is defined by the arc-length function:

$$L(t_1, t_2) = \int_{t_1}^{t_2} ||\dot{\mathbf{x}}(t)|| \, \mathrm{d}t, \tag{12}$$

which leads to the expression of reachabilty integrated over arc-length R_L:

$$R_L(\mathbf{x}) = \lim_{\Delta t \to 0} \prod_{i=0}^{n} \left(1 - m(\mathbf{x}(t_i))\right)^{L(t_i, t_i + \Delta t)} \tag{13}$$

As the time difference Δt between two consecutive time steps t_i, $t_i + \Delta t$ approaches zero, the arc-length between these positions can be approximated by the Euclidean distance $||\mathbf{x}(t_i) - \mathbf{x}(t_i + \Delta t)||$ such that

$$R_L(\mathbf{x}) \approx \lim_{\Delta t \to 0} \prod_{i=0}^{n} \left(1 - m(\mathbf{x}(t_i))\right)^{||\mathbf{x}(t_i) - \mathbf{x}(t_i + \Delta t)||} \tag{14}$$

For brevity, in the following, the notation $R(\mathbf{x})$ is assumed to express $R_L(\mathbf{x})$ from Eq. 14.

C. Map Interpolation

While the interpretation of reachability using a product integral alleviates the binary traversal of voxels assumption, it is still not continuous. The discontinuity comes from the fact that samples at time step t_i and $t_i + \Delta t$ are evaluated using the grid quantisation of $\mathbf{x}(t_i)$ and $\mathbf{x}(t_i + \Delta t)$, which leads to discontinuities if both positions fall into different voxels. In order to enable a smooth objective function of reachability, we introduce an interpolation method that is inspired by bilinear filtering [1]. A continuous point is sampled based on its relative position between two neighboring voxels in every dimension. While classical bilinear filtering uses a weighted sum to interpolate between given data points, we compute the weighted product to preserve the multiplicative property of the reachability (Eq. 14). In the two-dimensional case, the occupancy at point $\mathbf{x}(t) = \begin{bmatrix} x_t & y_t \end{bmatrix}^T$ in a $U \times V$ voxel grid is sampled as follows:

$$\begin{aligned} \mathfrak{r}(\mathbf{x}(t)) \approx \left((1 - m_{ij})^{1-\alpha} \cdot (1 - m_{i+1,j})^{\alpha}\right)^{1-\beta} \\ \cdot \left((1 - m_{i,j+1})^{1-\alpha} \cdot (1 - m_{i+1,j+1})^{\alpha}\right)^{\beta}, \end{aligned} \tag{15}$$

where:

$$\begin{aligned} i &= \lfloor x_t U \rfloor \qquad j = \lfloor y_t V \rfloor \\ \alpha &= \mathrm{frac}(x_t U) \qquad \beta = \mathrm{frac}(y_t V). \end{aligned}$$

In the special case where a trajectory traverses voxels completely, e.g. the green horizontal trajectory in Fig. 2, the proposed interpolation method yields the same reachability as the multiplied individual voxel reachabilities. This method provides a smoother representation of the map that can be efficiently used in the optimization, while maintaining the correct estimate of the reachability. As a result, we obtain a better measure of reachability, but do not have the estimate of the variance that can be used to actively plan high-speed trajectories. We address this problem in the next section.

V. Map Prediction using Confidence-rich Maps

In order to estimate the variance of reachability of the map, the planner needs to reason about the acquisition of future perceptual knowledge and incorporate this knowledge in planning. An important feature of the confidence-rich map is that it enables efficient prediction of the map evolution and map uncertainty.

The precise way of incorporating unknown future observations is to treat them as random variables and compute their future probability density function. However, 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:

$$z_n^{ml} = \arg\max_z p(z | b_n^m, x v_n) \tag{16}$$

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

$$b_{n+1}^{m^i, ml} = \tau^i(b_n^{m, ml}, z_{n+1}^{ml}, x v_{n+1}), \qquad n \geq k \tag{17}$$

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 [2].

VI. Cost Function for High-speed Safe Trajectories

Since confidence-rich mapping not only estimates voxel occupancies but also their variances, we are able to integrate the mapping uncertainty into our objective function. In the following, we first show how to compute the mean and the variance of reachability based on the voxel occupancy estimates, and then, we propose to use Lower Confidence Bound [4] as an effective approach to combine the mean and variance of reachability to obtain a cost function that can be used to generate safe, high-speed trajectories.

A. Expectation and Variance of Reachability

The expectation of reachability is defined based on the product integral of voxel occupancy expectations (Eq. 14):

$$\mathbb{E}[R(\mathbf{x})] = \lim_{\Delta t \to 0} \prod_{i=0}^{n} \left(1 - \hat{m}(\mathbf{x}(t_i))\right)^{||\mathbf{x}(t_i) - \mathbf{x}(t_i + \Delta t)||}, \tag{18}$$

where $\hat{m}(\mathbf{x}(t_i))$ is the expected occupancy of the voxel at position $\mathbf{x}(t_i)$. In the following, the voxel reachability estimates $X_1, \ldots, X_m$ are assumed to be independent random variables, for which we compute $\mathbb{E}[X_i] = 1 - \hat{m}_i$ and $\mathrm{var}[X_i] = \hat{v}_i$. In this case, the variance of the reachability,

i.e. the product of $X_1, \ldots, X_n$ is computed as:

$$\begin{aligned}\operatorname{var}[X_1 \cdots X_n] &= \prod_{i=1}^{n} \left(\operatorname{var}[X_i] + (\mathbb{E}[X_i])^2\right) - \prod_{i=1}^{n} (\mathbb{E}[X_i])^2 \\ &= \prod_{i=1}^{n} \left(\hat{v}_i + (1-\hat{m}_i)^2\right) - \prod_{i=1}^{n} (1-\hat{m}_i)^2 .\end{aligned}$$

While this representation of variance is correct under the binary voxel traversal assumption, it does not consider the distance travelled within voxels. Reintroducing the notion of a product integral over arc-length, $\operatorname{var}[R(\mathbf{x})]$ is computed as follows:

$$\begin{aligned}\operatorname{var}[R(\mathbf{x})] = \lim_{\Delta t \to 0} &\prod_{i=1}^{n} \left(\hat{v}(\mathbf{x}(t_i)) + (1-\hat{m}(\mathbf{x}(t_i)))^2\right)^{d_i} \\ &- \prod_{i=1}^{n} (1-\hat{m}(\mathbf{x}(t_i)))^{2d_i} ,\end{aligned} \tag{19}$$

where $d_i = ||\mathbf{x}(t_i) - \mathbf{x}(t_i + \Delta t)||$.

B. Lower Confidence Bound

Having computed the mean and the variance of reachability, we aim to find a measure that combines the two in a principled manner to obtain a cost function for reliable high-speed trajectories.

In order to do so, we follow the theory developed for multi-armed bandit problems. In particular, we use the *Lower Confidence Bound* [4], which is used to drive an efficient decision making strategy that balances exploitation and exploration to maximize the reward. In Bayesian Optimization the objective function is modeled by a Gaussian Process, over which the acquisition component selects the next point at which to evaluate the objective function [5]. This method performs well as it allows to directly tune the trade-off between exploitation and exploration. In this setting, our objective is to find the safest trajectory by combining the reachability value and our confidence in it, which can be formulated as the lower confidence bound:

$$LCB(\mathbf{x}) = \mathbb{E}[R(\mathbf{x})] - \kappa\sigma[R(\mathbf{x})]. \tag{20}$$

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

C. Algorithm Summary

Initially, the robot takes one measurement and updates the map. Based on this initialization, an optimizer simulates trajectories (Alg. 2) with different parameters $\mathbf{r}$ to maximize the lower confidence bound. The simulation takes place by taking most likely observations and updating an imaginary map $b^{m,ml}$, which is a local copy of the current-state belief map b^m.

The proposed method (for an overview of the algorithm see Alg. 1) instructs the robot to follow the current trajectory $\mathbf{x}$ parameterized by the previously obtained $\mathbf{r}$, take measurements and update the belief map. This process continuous 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 a new best trajectory is computed.

Algorithm 1 Re-planning given constraints $\mathbf{c}$

procedure PLAN($\mathbf{c}, b^m$)
  $t \leftarrow 0$
  Initialize trajectory parameter $\mathbf{r} \leftarrow \mathbf{0}$
  Compute trajectory $\mathbf{x}$ using $\mathbf{r}$ (Eqn. 7)
  Take measurement, update b^m (Eq. 2)
  Compute $\mathbf{T}$ (Eq. 4)
  $\mathbf{r} \leftarrow \underset{\mathbf{r}}{\operatorname{argmax}}\ \text{EVALUATE}(\mathbf{c}, \mathbf{T}, \mathbf{r}, b^m)$
  Compute trajectory $\mathbf{x}$ using $\mathbf{r}$ (Eqn. 7)
  while $t < T$ **do**
    Move to position $\mathbf{x}(t)$ and orient forward-facing
    Take measurement, update b^m (Eq. 2)
    $r_t \leftarrow R\left(x(t : t + k\Delta t)\right)$
    if $r_t < 1 - \epsilon$ **then** ▷ Replanning necessary
      Compute $\mathbf{T}$ (Eq. 4)
      $\mathbf{r} \leftarrow \underset{\mathbf{r}}{\operatorname{argmax}}\ \text{EVALUATE}(\mathbf{c}, \mathbf{T}, \mathbf{r}, b^m)$
      Compute trajectory $\mathbf{x}$ using $\mathbf{r}$ (Eqn. 7)
    end if
    $t \leftarrow t + \Delta t$
  end while
end procedure

VII. Simulation Experiments

We evaluate the proposed method in different environments. In the first scenario, two rectangular obstacles form a tunnel through which the robot has to traverse in order to reach the goal. The robot is equipped with a depth camera of 16 horizontally arranged range sensors that span a field of view of 90° and always points forward along the robot's trajectory. 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. 20 to 0, causing the optimization based on lower confidence bound (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. The first metric considers the reachability of the points along the trajectory after Alg. 1 has finished, which directly corresponds to the path's safety. In addition, we compare the two methods by the measure of *surprise*:

$$\Delta\mathbb{E}[R(\mathbf{x}(t))] = \left|\mathbb{E}[R(\mathbf{x}(t))] - \mathbb{E}[R(\mathbf{x}(t-\Delta t))]\right| \tag{21}$$

which is computed at every time step by taking the absolute difference between the expected reachability of the remaining trajectory at the current and the previous time step.

As it can be seen in Fig. 3, the trajectory that is solely optimized for reachability makes a sharp turn before entering

Algorithm 2 Simulating trajectory to compute LCB

function EVALUATE($\mathbf{c}, \mathbf{T}, \mathbf{r}, b^m$)
▷ Simulates and computes LCB of trajectory $\mathbf{x}$ parametrized by $\mathbf{r}$
Compute trajectory $\mathbf{x}$ using $\mathbf{r}$ (Eqn. 7)
$t \leftarrow \Delta t$ ▷ Move one step ahead
$b^{m,ml} \leftarrow b^m$ ▷ Copy current state of belief map
while $t < T$ **do**
Move to position $\mathbf{x}(t)$ and orient forward-facing
Obtain z^{ml} using Eq. 16 from $b^{m,ml}$
Update map $b^{m,ml}$ according to Eq. 17
$d \leftarrow ||\mathbf{x}(t_i) - \mathbf{x}(t_i + \Delta t)||$
$\mathcal{R} \leftarrow R \cdot (1 - m(\mathbf{x}(t)))^d$
$\mathcal{V}_l \leftarrow \mathcal{V}_l \cdot (v(\mathbf{x}(t)) + (1 - m(\mathbf{x}(t))))^d$
$\mathcal{V}_r \leftarrow \mathcal{V}_r \cdot (1 - m(\mathbf{x}(t)))^{2d}$
$t \leftarrow t + \Delta t$
end while
$\mathcal{V} \leftarrow \mathcal{V}_l - \mathcal{V}_r$
$LCB \leftarrow \mathcal{R} - \kappa\sqrt{\mathcal{V}}$
return LCB
end function

the tunnel, which not only slows down the robot but also increases "the surprise factor" 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 decrease in "the surprise factor". Not only does our method outperform the reachability baseline based on surprise but also the overall expected reachability of the remaining trajectory is higher most of the time. The same holds true for the expected reachability $\mathbb{E}[R(\mathbf{x}(t:t+10\Delta t))]$ (rightmost plot in Fig. 3) of the next 10 steps, which corresponds to the path's safety for the immediate future.

In another scenario, depticted in Fig. 4, the proposed method yields a trajectory that again preemptively circumvents obstacles ahead of time to keep surprise to a minimum, which additionally results in a better reachability, i.e. a safer path. Furthermore, the traversal time is shorter in the LCB-optimized trajectory ($1.512s$) compared to the baseline ($1.653s$).

VIII. Discussion and Future Work

We presented a method that enables planning of high-speed motions using confidence-rich maps and the explicit consideration of future map uncertainty. Specifically, we predict the future uncertainty of the confidence-rich map and optimize trajectories to minimize re-planning. The simulation experiments indicate that the presented trajectory optimization method leads to faster *and* safer trajectories with a smaller number of re-planning stops than the traditional reachability-based optimization.

There are a number of interesting extensions of the proposed method that we plan to investigate in the future. We are working on extending the proof-of-concept simulations to a 3D environment that will be later used to perform tests on real robots. Further, we plan to incorporate the yaw of the vehicle into the trajectory optimization so that we can achieve behaviors where the robot looks ahead at the part of the map that will be relevant for the future part of the trajectory. We believe that this will further reduce the variance of our estimates and potentially allow even faster flight.

References

[1] Milton Abramowitz, Irene A Stegun, et al. Handbook of mathematical functions. *Applied mathematics series*, 55(62):39, 1966.

[2] Ali-akbar Agha-mohammadi. SMAP: Simultaneous mapping and planning on occupancy grids. *arXiv preprint arXiv:1608.04712*, 2016.

[3] Ali-Akbar Agha-Mohammadi, Suman Chakravorty, and Nancy M Amato. FIRM: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements. *The International Journal of Robotics Research*, 33(2):268–304, 2014.

[4] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

[5] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[6] Yan Duan, Sachin Patil, John Schulman, Ken Goldberg, and Pieter Abbeel. Planning locally optimal, curvature-constrained trajectories in 3d using sequential convex optimization. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5889–5895. IEEE, 2014.

[7] Matthias Faessler, Davide Falanga, and Davide Scaramuzza. Thrust mixing, saturation, and body-rate control for accurate aggressive quadrotor flight. *IEEE Robotics and Automation Letters*, 2016.

[8] Davide Falanga, Elias Mueggler, Matthias Faessler, and Davide Scaramuzza. Aggressive quadrotor flight through narrow gaps with onboard sensing and computing. *arXiv preprint arXiv:1612.00291*, 2016.

[9] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Appearance-based active, monocular, dense reconstruction for micro aerial vehicle. In *2014 Robotics: Science and Systems Conference*, number EPFL-CONF-203672, 2014.

[10] Karol Hausman, Jörg Müller, Abishek Hariharan, Nora Ayanian, and Gaurav S Sukhatme. Cooperative multi-robot control for target tracking with onboard sensing. *The International Journal of Robotics Research*, 34(13): 1660–1677, 2015.

[11] Karol Hausman, James Preiss, Gaurav Sukhatme, and Stephan Weiss. Observability-aware trajectory optimization for self-calibration with application to uavs. *arXiv preprint arXiv:1604.07905*, 2016.

[12] Gregory Kahn, Peter Sujan, Sachin Patil, Shaunak Bopardikar, Julian Ryde, Ken Goldberg, and Pieter

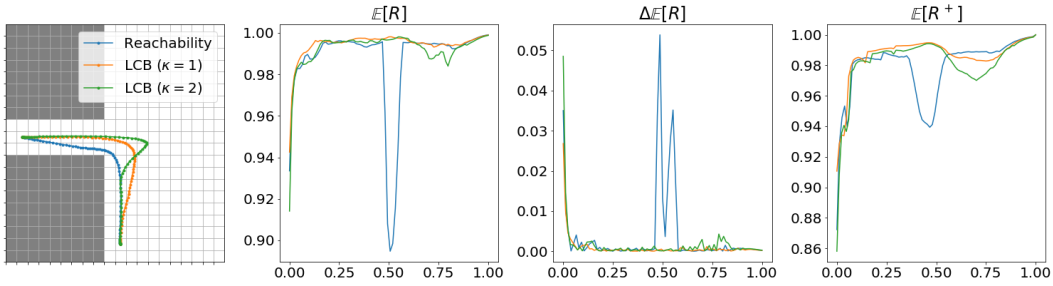

Fig. 3. Evaluation of reachability-based trajectory optimization vs. Lower Confidence Bound (LCB) as cost function with different κ parameters. Left: the obstacles (grey) and trajectories with sampling times (dots along solid lines). Middle-left: the expected reachability of the remaining trajectory starting from the current time step t is . Middle-right: the surprise (Eq. 21) at time t. Right" the evolution of the reachability of only the next 10 time steps starting from current time t.

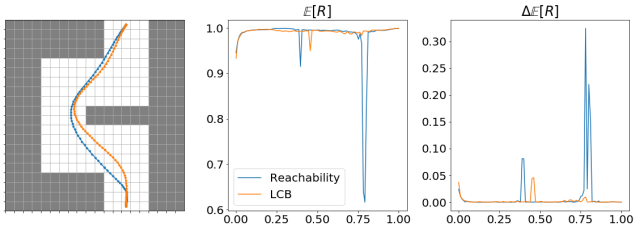

Fig. 4. Evaluation of trajectory optimization based on reachability vs. Lower Confidence Bound (LCB) in corridor environment. Left: the orange trajectory has been optimized via LCB with $\kappa = 1$, while the blue line represents a reachability-optimized trajectory. Middle, the evolution of the expected reachability of the remaining trajectory. Right" the surprise at every time step is computed for both trajectories according to Eq. 21.

Abbeel. Active exploration using trajectory optimization for robotic grasping in the presence of occlusions. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 4783–4790. IEEE, 2015.

[13] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4): 566–580, 1996.

[14] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128 x 128 120 db 15 mu s latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576, 2008.

[15] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *In Int. Conf. on Robotics and Automation (ICRA)*, pages 2520–2525. IEEE, 2011.

[16] Patrycja E Missiuro and Nicholas Roy. Adapting probabilistic roadmaps to handle uncertain maps. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1261–1267. IEEE, 2006.

[17] Elias Mueggler, Basil Huber, and Davide Scaramuzza. Event-based, 6-dof pose tracking for high-speed maneuvers. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2761–2768. IEEE, 2014.

[18] J. Müller and G.S. Sukhatme. Risk-aware trajectory generation with application to safe quadrotor landing. In *In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Chicago, IL, USA, September 2014.

[19] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[20] Simon T O'Callaghan and Fabio T Ramos. Gaussian process occupancy maps for dynamic environments. In *Experimental Robotics*, pages 791–805. Springer, 2016.

[21] Fabio Ramos and Lionel Ott. Hilbert maps: scalable continuous occupancy mapping with stochastic gradient descent. *The International Journal of Robotics Research*, 35(14):1717–1730, 2016.

[22] C. Richter, A. Bry, and N. Roy. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In *In Proc. of the International Symposium on Robotics Research (ISRR)*, 2013.

[23] Charles Richter and Nicholas Roy. Learning to plan for visibility in navigation of unknown environments. 2016.

[24] Kai M Wurm, Armin Hornung, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems. In *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, volume 2, 2010.

[25] T. Zhang, G. Kahn, S. Levine, and P. Abbeel. Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 528–535. IEEE, 2016.