2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

Uncertainty–aware Receding Horizon Exploration and Mapping using Aerial Robots

Christos Papachristos, Shehryar Khattak, and Kostas Alexis

***Abstract*— This paper presents a novel path planning algorithm for autonomous, uncertainty–aware exploration and mapping of unknown environments using aerial robots. The proposed planner follows a two–step, receding horizon, belief space–based approach. At first, in an online computed tree the algorithm finds the branch that optimizes the amount of space expected to be explored. The first viewpoint configuration of this branch is selected, but the path towards it is decided through a second planning step. Within that, a new tree is sampled, admissible branches arriving at the reference viewpoint are found and the robot belief about its state and the tracked landmarks of the environment is propagated. The branch that minimizes the expected localization and mapping uncertainty is selected, the corresponding path is executed by the robot and the whole process is iteratively repeated. The proposed planner is capable of running online onboard a small aerial robot and its performance is evaluated using experimental studies in a challenging environment.**

I. INTRODUCTION

Aerial robotics face an unprecedented process of wide integration in many critical applications. In fields like infrastructure inspection, precision agriculture, disaster response and security monitoring, aerial robots correspond to novel tools for rapid data collection at low cost. But for such systems to demonstrate their full potential, they should be able to autonomously handle complex missions in order to eliminate the need for manual operation. This work focuses on the fundamental need to enable fully autonomous exploration, and simultaneous accurate 3D mapping of previously unknown environments using aerial robots.

A great body of work exists in the field of robotic exploration [1–5]. But exploring an environment in the sense of ensuring that a sensor "covers" the whole space from relevant viewpoints does not ensure mapping quality. The probabilistic process of localization and mapping, especially when operating without GPS or other external positioning support, depends on many factors, including light conditions and the order of observations, that are not considered during traditional exploration planning. To account for the correlation of viewpoint selection and the quality of onboard mapping, the so–called fields of active Simultaneous Localization And Mapping (SLAM) [6–9] and belief–space planning [10–13] have emerged.

The approach described in this paper aims to simultaneously address the often combined need for efficient online exploration of unknown environments, with their accurate mapping on–the–go. To achieve this goal, it proposes a new *receding horizon*, *belief uncertainty–aware* exploration and mapping path planning strategy. Within this method, an optimized sequence of viewpoints for exploration of unknown spaces is computed and its first viewpoint is selected to be visited, but the path to this new viewpoint is computed through a second planning layer that aims to optimize the probabilistic mapping behavior of the robot. Subsequently, the whole process is repeated in a receding horizon fashion.

Fig. 1. An instant of the uncertainty–aware exploration and mapping of a room with complex geometry using an aerial robot.

More specifically, the environment is represented using an occupancy map [14], along with the currently tracked landmarks. Within the resulting known free space at every iteration, the first planning step samples a finite–iteration random tree using a relevant algorithm (e.g. RRT [15]) and each branch of it is evaluated for the amount of unmapped space that it can explore. The best branch is found and its first edge is selected. Knowing what is the next viewpoint configuration to optimize for exploration, the algorithm then runs the second, nested, planning step which samples a tree to find multiple ways to go to this vertex. Belief propagation about the robot state and tracked landmarks takes place along the edges of this tree and the branch that minimizes the robot's pose and landmarks uncertainty is selected. This belief–optimized branch is fully executed, the robot arrives at the first vertex of the exploration planning step and the whole process is repeated. By first selecting a viewpoint to optimize the exploratory behavior of the robot, and then optimizing for paths that lead to minimized uncertainty, a behavior that does not sacrifice exploration efficiency, while supporting mapping quality is achieved.

The proposed uncertainty–aware exploration and mapping

This material is based upon work supported by the Department of Energy under Award Number [DE-EM0004478].

The authors are with the Autonomous Robots Lab, University of Nevada, Reno, 1664 N. Virginia, 89557, Reno, NV, USA cpapachristos@unr.edu

planning strategy is evaluated in a challenging experimental environment, while computational complexity analysis is also provided. The experimental studies were conducted using an aerial robot performing visual–inertial localization indoors without support of motion capture systems. The planned paths, and the derived volumetric and dense maps are presented along with analysis plots and visual comparison with an offline reconstructed map. The algorithm implementation is released as an open source package accompanied with a relevant dataset [16].

The remainder of the paper is structured as follows: Section II provides an overview of related work, while the considered problem is defined in III. The proposed algorithm is detailed in Section IV, followed by evaluation studies in Section V. Finally, conclusions are drawn in Section VI.

II. RELATED WORK

The research community has contributed a set of methods for the problems of exploration and mapping. In the field of exploration from the perspective of finding a set of viewpoints that geometrically identify the non–free space, early work includes the sampling of "next–best–views" [1], frontiers–based exploration [2] and more recent efforts [3, 5]. However, exploration in this geometric sense cannot alone ensure mapping quality. The computer vision community has long considered the concept of accounting for the coupling between viewpoint selection and the quality of the derived map. Early works describe the importance of active vision for problems like structure from motion [8], while the work in [17] focused on integrating trajectory planning with SLAM. The term active SLAM was introduced by [18], to be followed by a set of works aiming to optimize the robot navigation robustness [6–9, 19]. Belief–space planning emerged to allow the integration of the expected robot belief into its motion planning [10–13, 20] and led to powerful tools such as the belief roadmap [11] and belief trees [10]. Further work examines how to find optimized navigation paths [20, 21], while some authors have approached the problem of exploration and mapping [22–27]. The contribution proposed in this paper differs from the above in the sense that it proposes a nested, receding horizon planning strategy to combine efficient exploration of unknown environments with belief–space based planning to enhance the on–the–go mapping behavior of the robot in a framework that remains computationally tractable and runs online on a small robot, alongside all its other tasks (e.g. control, SLAM).

III. PROBLEM DESCRIPTION

The overall problem considered in this work is that of exploring a bounded 3D volume $V^E \subset \mathbb{R}^3$ while aiming to minimize the localization and mapping uncertainty as evaluated through a metric over the robot pose and landmarks probabilistic belief. The exploration problem may be casted globally as that of starting from an initial collision free configuration and deriving viewpoints that cover the a priori unknown volume by determining which parts of the initially unexplored space $V^E_{une} \overset{init.}{=} V^E$ are free $V^E_{free} \subseteq V^E$ or occupied $V^E_{occ} \subseteq V^E$. For this process, the volume is discretized in an occupancy map $\mathcal{M}$ consisting of cubical voxels $m \in \mathcal{M}$ with edge length r. The operation is subject to vehicle dynamic constraints, localization uncertainty and limitations of the sensing system. As for most sensors the perception stops at surfaces, hollow spaces or narrow pockets can sometimes not be explored leading to a residual volume:

Definition 1 (Residual Volume): Let Ξ be the simply connected set of collision free configurations and $\bar{\mathcal{V}}^E_m \subseteq \Xi$ the set of all configurations from which the voxel m can be perceived. Then the residual volume is given as $V^E_{res} = \bigcup_{m\in\mathcal{M}}(m|\ \bar{\mathcal{V}}^E_m = \emptyset)$.

The exploration problem is then globally defined as:

Problem 1 (Volumetric Exploration Problem): Given a bounded volume V^E, find a collision free path σ starting at an initial configuration $\xi_{init} \in \Xi$ that leads to identifying the free and occupied parts V^E_{free} and V^E_{occ} when being executed, such that there does not exist any collision free configuration from which any piece of $V^E \setminus \{V^E_{free}, V^E_{occ}\}$ could be perceived. Thus, $V^E_{free} \cup V^E_{occ} = V^E \setminus V^E_{res}$.

While the exploration problem is globally defined, the problem of localization and mapping uncertainty–aware planning in a priori unknown environments can only be casted locally as new features are tracked from the perception system. At every planning step, let $V^M \subset V^E$ be a local volume enclosing the current pose configuration and the next exploration viewpoint. The problem is that of minimizing the covariance of the propagated robot pose and tracked landmarks belief as the vehicle is moving from its current configuration to the next exploration viewpoint. By minimizing the robot belief uncertainty, a good prior for high–quality mapping is achieved. More formally:

Problem 2 (Belief Uncertainty–aware Planning): Given a $V^M \subset V^E$, find a collision free path σ^M starting at an initial configuration $\xi_0 \in \Xi$ and ending in a configuration ξ_{final} that aims to improve the robot's localization and mapping confidence by following paths of optimized robot pose and tracked landmarks covariance.

IV. PROPOSED APPROACH

Considering the problem of belief uncertainty–aware exploration and mapping of unknown space, the proposed algorithm employs a two–step sampling–based receding horizon planning paradigm, to iteratively generate paths that can cover the whole space V^E, while locally aiming to improve mapping confidence through propagation and evaluation of the robot belief about its pose and the tracked landmarks. The first planning step identifies a finite–steps path that maximally explores new, previously unmapped space. Subsequently, the first new viewpoint configuration of this path becomes the goal point of a second, nested, planning step which computes a path to go to this viewpoint, while ensuring that good landmarks are tracked and low–uncertainty belief is maintained by the robot. The robot follows this path and the whole process is iteratively repeated. A visualization of this functionality is shown in Figure 2. By minimizing the robot uncertainty while maintaining optimized exploratory

behavior, a good prior for accurate 3D mapping is provided. It is noted that the code implementation of the proposed method is released as an open source package available at [16].

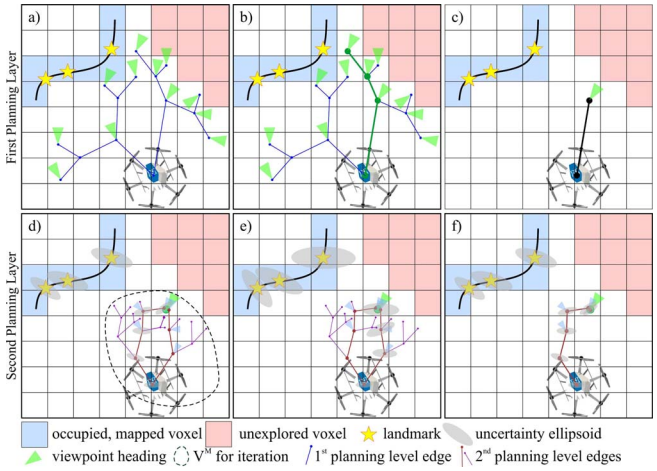

Fig. 2. 2D representation of the two–steps uncertainty–aware exploration and mapping planner. The first planning layer samples the path with the maximum information gain in terms of space to be explored. The viewpoint configuration of the first vertex of this path becomes the reference to the second planning layer. Then this step, samples admissible paths that arrive to this configuration and selects the one that provides minimum uncertainty of the robot belief about its pose and the tracked landmarks. To achieve this step, belief propagation along the edges of this planning level takes place.

A sensor such as a camera, or in the case of this work a stereo visual–inertial unit, is used to enable robot localization and map building functionality. More specifically, estimates and uncertainty metrics for the robot pose and tracked landmarks are provided, along with a volumetric map of the environment derived using the stereo depth sensing and the estimated robot poses. The robot belief and the volumetric map are used for collision free navigation and path planning.

The employed volumetric representation is an occupancy map $\mathcal{M}$ [14] dividing the space V^E in cubical volumes $m \in \mathcal{M}$, that can either be marked as free, occupied with probability $\mathcal{P}(m)$ or unmapped. The resulting array of voxels is saved in an octree to enable efficient access for operations like occupancy checking. Paths are only planned inside the iteratively explored free space V^E_{free}, relevant vehicle configurations are denoted as $\xi \in \Xi$, while paths are given by $\sigma : \mathbb{R}^n \rightarrow \xi$ and specifically from ξ_{k-1} to ξ_k by $\sigma_{k-1,k}(s),\ s \in [0,1]$ with $\sigma_{k-1,k}(0) = \xi_{k-1}$ and $\sigma_{k-1,k}(1) = \xi_k$. The sampled paths have to be such that they are collision free and can be tracked by the vehicle, given possible motion constraints. Within the algorithm implementation, a robot configuration is defined by the state $\xi = [x, y, z, \psi]^T$ with roll (ϕ) and pitch (θ) considered around zero. For slow maneuvering it is considered that the robot flies straight paths $\sigma_{k-1,k} = s\xi_k + (s-1)\xi_{k-1},\ s \in [0,1]$. Subsequently, the role of each planning step is detailed.

A. Exploration Planning

At each iteration, for the occupancy map representing the world $\mathcal{M}$, the set of visible but unmapped voxels from a configuration ξ is denoted as $\mathbf{VisibleVolume}(\mathcal{M}, \xi)$. Every voxel m in this set lies in the unexplored area V^E_{unm}, the direct line of sight does not cross occupied voxels and complies with the sensor model. Within this work, the camera sensor is mounted with a fixed pitch, has a Field of View (FoV) described by vertical and horizontal opening angles $[a_v, a_h]$, and a maximum effective sensing distance. While the sensor limitation is $d^{\text{sensor}}_{\max}$, the planner uses a value $d^{\text{planner}}_{\max} < d^{\text{sensor}}_{\max}$. Moreover, for the explored, mapped as occupied with probability $\mathcal{P}(m) < p_{thres}$ voxels, the $\mathbf{ReobservationGain}(\mathcal{M}, \mathcal{P}, \xi)$ is computed and refers to their volume weighted by $(1 - \mathcal{P}(m))$. Starting from the current configuration ξ_0, a geometric tree $\mathbb{T}^E$ with maximum edge length ℓ_E is incrementally built in the configuration space using the RRT algorithm [28]. The resulting tree contains $N^E_{\mathbb{T}}$ nodes n^E and its edges are given by collision free paths σ^E. A minimum of $N^E_{\max}$ but no more than N^E_{TOT} nodes are sampled. The collected information gain of a node $\mathbf{ExplorationGain}(n^E)$ is the cumulative unmapped volume that can be explored from the nodes along the branch, added to the $\mathbf{ReobservationGain}(\mathcal{M}, \mathcal{P}, \xi)$. For node k:

$$\begin{aligned}\mathbf{ExplorationGain}(n^E_k) = \mathbf{ExplorationGain}(n^E_{k-1}) + \\ \mathbf{VisibleVolume}(\mathcal{M}, \xi_k)\exp(-\lambda c(\sigma^E_{k-1,k})) + \\ \mathbf{ReobservationGain}(\mathcal{M}, \mathcal{P}, \xi_k)\exp(-\lambda c(\sigma^E_{k-1,k}))\end{aligned} \quad (1)$$

where $c(\sigma^E_{k-1,k})$ is the length of the path, and λ is a parameter to penalize long paths. At the end of this planning level iteration, the first segment σ^E_{RH} of the branch to the best node $\mathbf{ExtractBestPathSegment}(n^E_{best})$, the vertex at the end of this segment n^E_{RH}, and the associated pose configuration ξ_{RH} are extracted. If no positive gain can be found, the exploration process is terminated. This step extends the concept of previous own work [3] by integrating the occupancy map probabilities into the gain calculation.

B. Belief Uncertainty–aware Planning

The second planning step is responsible to support the on–the–go mapping behavior of the robot by minimizing its localization and tracked landmarks uncertainty. The robot is considered to be equipped with a perception system providing the estimates of the robot pose and the landmarks of the environment. At every call of this nested planning level, a new random tree $\mathbb{T}^M$ with maximum edge length ℓ_M is spanned within a local volume $V^M \subset V^E$, enclosing the current robot pose and ξ_{RH}. A total set of $N^M_{\mathbb{T}}$ vertices within V^M are sampled with $N^M_{\mathbb{T}} < N^M_{\max}$ and admissible paths (σ^E_{RH} treated as one of them) that a) start from the current robot configuration and arrive in a local set $\mathbb{S}_{\xi_{RH}}$, around the reference ξ_{RH} provided from the first planning layer, and b) have an overall length $c \leq \delta c(\sigma^E_{RH})$ where $\delta \geq 1$ a tuning factor, are derived. Since an admissible path is one that arrives into the local set $\mathbb{S}_{\xi_{RH}}$ ($\mathbb{V}(\mathbb{T}^M) \cap \mathbb{S}_{\xi_{RH}} \neq \emptyset$) and not necessarily on ξ_{RH} an additional connection takes place to guarantee that the robot arrives exactly on the configuration sampled by the previous planning layer. For each of the $N^M_{\mathbb{T}} - 1$ edges of the tree (including the additional connections to n_{RH}), belief propagation takes

place to estimate the expected robot belief about its state and the tracked landmarks along the sampled paths. Given the updated estimates of the state and landmarks covariance, the admissible tree branches are evaluated regarding their "belief gain" which is the D-optimality [7] calculation of the robot pose and tracked landmarks covariance matrix at the end–vertex of each admissible path. The admissible branch with the best belief gain is then selected to be executed by the robot. Algorithm 1 details the steps of the complete planning process, while subsections IV-B.1, IV-B.2 and IV-B.3 detail the employed localization framework, the belief propagation and the belief gain calculations respectively.

Algorithm 1 Proposed Planner - Iterative Step

$\xi_0 \leftarrow$ current vehicle configuration
Initialize $\mathbb{T}^E$ with ξ_0
$g^E_{best} \leftarrow 0$ ▷ Set best exploration gain to zero
$n^E_{best} \leftarrow n_0(\xi_0)$ ▷ Set best exploration node to root
$N^E_{\mathbb{T}} \leftarrow$ Number of initial nodes in $\mathbb{T}^E$
while $N^E_{\mathbb{T}} < N^E_{\max}$ **or** $g^E_{best} = 0$ **do**
  Incrementally build $\mathbb{T}^E$ by adding $n^E_{new}(\xi_{new})$
  $N^E_{\mathbb{T}} \leftarrow N^E_{\mathbb{T}} + 1$
  if $\mathbf{ExplorationGain}(n^E_{new}) > g^E_{best}$ **then**
    $n^E_{best} \leftarrow n^E_{new}$
    $g^E_{best} \leftarrow \mathbf{ExplorationGain}(n^E_{new})$
  end if
  if $N^E_{\mathbb{T}} > N^E_{TOL}$ **then**
    Terminate planning
  end if
end while
$\sigma^E_{RH}, n^E_{RH}, \xi_{RH} \leftarrow \mathbf{ExtractBestPathSegment}(n^E_{best})$
$\mathbb{S}_{\xi_{RH}} \leftarrow \mathbf{LocalSet}(\xi_{RH})$
Propagate robot belief along σ^E_{RH}
$\alpha \leftarrow 1$ ▷ number of admissible paths
$g^M_\alpha \leftarrow \mathbf{BeliefGain}(\sigma^E_{RH})$
$g^M_{best} \leftarrow g^M_\alpha$ ▷ straight path belief gain
$\sigma^M_{best} \leftarrow \sigma^M_{RH}$ ▷ Set best belief path
while $N^M_{\mathbb{T}} < N^M_{\max}$ **or** $\mathbb{V}(\mathbb{T}^M) \cap \mathbb{S}_{\xi_{RH}} = \emptyset$ **do**
  Incrementally build $\mathbb{T}^M$ by adding $n^M_{new}(\xi_{new})$
  Propagate robot belief from current to planned vertex
  if $\xi_{new} \in \mathbb{S}_{\xi_{RH}}$ **then**
    Add new vertex n^M_{new} at ξ_{RH} and connect
    $\alpha \leftarrow \alpha + 1$
    $\sigma^M_\alpha \leftarrow \mathbf{ExtractBranch}(n^M_{new})$
    $g^M_\alpha \leftarrow \mathbf{BeliefGain}(\sigma^M_\alpha)$
    if $g^M_\alpha < g^M_{best}$ **then**
      $\sigma^M \leftarrow \sigma^M_\alpha$
      $g^M_{best} \leftarrow g^M_\alpha$
    end if
  end if
end while
return σ^M

1) Visual–Inertial Localization: For the purposes of robot navigation, a visual–inertial odometry framework is employed due to the superior robustness and accuracy such methods provide. In particular, the open–source Robust Visual Inertial Odometry (ROVIO) is utilized [29]. Within this paper, a necessarily brief summary will be provided as its formulation is also used for belief propagation. ROVIO closely couples the tracking of multilevel image patches with the Extended Kalman Filter (EKF) through the direct use of image intensity errors to derive the filter innovation term and uses a QR–decomposition to reduce the dimensionality of the error terms, therefore keeping the Kalman filter update step computationally tractable. Its formulation is robocentric, therefore the landmarks are estimated with respect to the camera pose. The estimated landmarks are decomposed into a bearing vector, as well as a depth parametrization. The Inertial Measurement Unit (IMU) fixed coordinate frame ($\mathcal{B}$), the camera fixed frame ($\mathcal{V}$) and the inertial frame $\mathcal{W}$ are considered and the employed state vector with dimension l and associated covariance matrix $\mathbf{\Sigma}_l$ is:

$$\mathbf{x} = [\overbrace{\underbrace{\mathbf{r}\ \mathbf{q}}_{\text{pose, } l_p}\ \boldsymbol{v}\ \mathbf{b}_f\ \mathbf{b}_\omega\ \mathbf{c}\ \mathbf{z}}^{\text{robot states, } l_s} \Big|\ \underbrace{\boldsymbol{\mu}_0,\ \cdots\ \boldsymbol{\mu}_J\ \rho_0\ \cdots\ \rho_J}_{\text{features states, } l_f}]^T \quad (2)$$

where l_p, l_s, l_f are dimensions, $\mathbf{r}$ is the robocentric position of the IMU expressed in $\mathcal{B}$, $\boldsymbol{v}$ represents the robocentric velocity of the IMU expressed in $\mathcal{B}$, $\mathbf{q}$ is the IMU attitude represented as a map from $\mathcal{B} \rightarrow \mathcal{W}$, $\mathbf{b}_f$ represents the additive accelerometer bias expressed in $\mathcal{B}$, $\mathbf{b}_\omega$ stands for the additive gyroscope bias expressed in $\mathcal{B}$, $\mathbf{c}$ is the translational part of the IMU–cameras extrinsics expressed in $\mathcal{B}$, $\mathbf{z}$ represents the rotational part of the IMU–cameras extrinsics and is a map from $\mathcal{B} \rightarrow \mathcal{V}$, while $\boldsymbol{\mu}_j$ is the bearing vector to feature j expressed in $\mathcal{V}$ and ρ_j is the depth parameter of the j^{th} feature such that the feature distance d_j is $d(\rho_j) = 1/\rho_j$. The relevant state propagation and update steps are briefly summarized in Table I, and are detailed in [29].

2) Belief Propagation: Within the second planning step, belief propagation takes place and the sampled paths contain relevant expected values and covariance estimates of both the robot state and the landmarks corresponding to the latest tracked features from ROVIO. This is achieved through the following: At first, for every path segment (an edge $\sigma_{k-1,k}$ of $\mathbb{T}^M$) the expected IMU trajectories are derived through simulation of the identified closed–loop dynamics of our robot $\dot{\boldsymbol{\chi}} = f(\boldsymbol{\chi}, \boldsymbol{\chi}^r)$ ($\boldsymbol{\chi}$ being the robot pose $[\mathbf{r}\ \mathbf{q}]^T$ expressed in $\mathcal{W}$ and $\boldsymbol{\chi}^r$ its reference) sampled every T_s. Using these IMU trajectories, prediction of the robot belief takes place by running the state propagation step in the EKF–fashion of ROVIO shown in Table I, Equations (3). In order to then conduct the filter update step, the landmarks corresponding to the features tracked at the initialization of the planner are projected in world coordinates $\mathcal{V} \rightarrow \mathcal{W}$ using the estimated feature distance, extrinsic and intrinsic camera parameters. For feature $\boldsymbol{\mu}_j$ this takes the form:

$$\begin{bmatrix} X_j \\ Y_j \\ Z_j \end{bmatrix} = \mathbf{R}^W_C \left(\mathbf{K}_I^{-1} \frac{1}{\rho_j} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \right) + \mathbf{T}^W_C \quad (5)$$

where X_j, Y_j, Z_j are their calculated world coordinates, $\mathbf{R}^W_C$ and $\mathbf{T}^W_C$ represent extrinsic camera pose in world

TABLE I

ROVIO STATE PROPAGATION & FILTER UPDATE STEPS.

State Propagation Step - Equations (3)

$$\dot{\mathbf{r}} = -\hat{\boldsymbol{\omega}}^{\times}\mathbf{r} + \boldsymbol{v} + \mathbf{w}_r$$
$$\dot{\boldsymbol{v}} = -\hat{\boldsymbol{\omega}}^{\times}\boldsymbol{v} + \hat{\mathbf{f}} + \mathbf{q}^{-1}(\mathbf{g})$$
$$\dot{\mathbf{q}} = -\mathbf{q}(\hat{\boldsymbol{\omega}})$$
$$\dot{\mathbf{b}}_f = \mathbf{w}_{bf}$$
$$\dot{\mathbf{b}}_\omega = \mathbf{w}_{bw}$$
$$\dot{\mathbf{c}} = \mathbf{w}_c$$
$$\dot{\mathbf{z}} = \mathbf{w}_z$$
$$\dot{\boldsymbol{\mu}}_j = \mathbf{N}^T(\boldsymbol{\mu}_j)\hat{\boldsymbol{\omega}}_{\mathcal{V}} - \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{N}^T(\boldsymbol{\mu}_j)\frac{\hat{\boldsymbol{v}}_{\mathcal{V}}}{d(\rho_j)} + \mathbf{w}_{\mu,j}$$
$$\dot{\rho}_j = -\boldsymbol{\mu}_j^T\hat{\boldsymbol{v}}_{\mathcal{V}}/d'(\rho_j) + w_{\rho,j}$$

$$\hat{\mathbf{f}} = \tilde{\mathbf{f}} - \mathbf{b}_f - \mathbf{w}_f$$
$$\hat{\boldsymbol{\omega}} = \tilde{\boldsymbol{\omega}} - \mathbf{b}_\omega - \mathbf{w}_\omega$$
$$\hat{\boldsymbol{v}}_{\mathcal{V}} = \mathbf{z}(\boldsymbol{v} + \hat{\boldsymbol{\omega}}^{\times}\mathbf{c})$$
$$\hat{\boldsymbol{\omega}}_{\mathcal{V}} = \mathbf{z}(\hat{\boldsymbol{\omega}})$$

Filter Update Step - Equations (4)

$$\mathbf{y}_j = \mathbf{b}_j(\boldsymbol{\pi}(\hat{\boldsymbol{\mu}}_j)) + \mathbf{n}_j$$
$$\mathbf{H}_j = \mathbf{A}_j(\boldsymbol{\pi}(\hat{\boldsymbol{\mu}}_j))\frac{d\boldsymbol{\pi}}{d\boldsymbol{\mu}}(\hat{\boldsymbol{\mu}}_j)$$

By stacking the above terms for all visible features, standard EKF update step is directly performed to derive the new estimate of the robot belief for its state and the tracked features.

Notation

$^{\times} \rightarrow$ skew symmetric matrix of a vector, $\tilde{\mathbf{f}} \rightarrow$ proper acceleration measurement, $\tilde{\boldsymbol{\omega}} \rightarrow$ rotational rate measurement, $\hat{\mathbf{f}} \rightarrow$ biased corrected acceleration, $\hat{\boldsymbol{\omega}} \rightarrow$ bias corrected rotational rate, $\mathbf{N}^T(\boldsymbol{\mu}) \rightarrow$ projection of a 3D vector onto the 2D tangent space around the bearing vector, $\mathbf{g} \rightarrow$ gravity vector, $\mathbf{w}_\star \rightarrow$ white Gaussian noise processes, $\boldsymbol{\pi}(\boldsymbol{\mu}) \rightarrow$ pixel coordinates of a feature, $\mathbf{b}_i(\boldsymbol{\pi}(\hat{\boldsymbol{\mu}}_j)) \rightarrow$ a 2D linear constraint for the j^{th} feature which is predicted to be visible in the current frame with bearing vector $\hat{\boldsymbol{\mu}}_j$

coordinates, $\mathbf{K}_I$ represents the intrinsic camera matrix, and $\boldsymbol{\pi}(\boldsymbol{\mu}_j) = [u, v]$ are the feature pixel coordinates. At each step of the belief propagation, only the landmarks that are visible are considered. For a landmark to be considered as visible from a planned viewpoint, it has to be within the camera frustum view and to be lying on the visible side of the occupany map $\mathcal{M}$ as tested using ray–casting and collision checking. As bearing vectors for all the landmarks are propagated in the state propagation step, the algorithm subsequently utilizes those predicted to be visible in the new planned configuration in order to update the covariance estimates through the filter update step. The calculated Jacobians are used to update the robot covariance estimate in standard EKF fasion, that is $\boldsymbol{\Sigma}_{t,l} = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\bar{\boldsymbol{\Sigma}}_{t,l}$ with $\mathbf{K}_t$ being the current gain, $\mathbf{H}_t$ the Jacobians, $\mathbf{I}$ the identity matrix, $\boldsymbol{\Sigma}_{t,l}$ the updated covariance, and $\bar{\boldsymbol{\Sigma}}_t$ denoting the covariance estimate of the state propagation step. This is analogous to ROVIO covariance update operation following the filter update of Table I, Equations (4). By conducting this process iteratively for all simulated trajectories and the corresponding sampled locations along the tree $\mathbb{T}^M$ branches, belief propagation takes place and the estimates for the covariance matrix $\boldsymbol{\Sigma}_l$ and its pose–features subset $\boldsymbol{\Sigma}_{p,f}$ become available. These estimates are then used in the belief–based gain calculation as shown in the next subsection.

3) Belief Uncertainty–based Gain Calculation: Among the key requirements for the visual–inertial odometry to perform robustly is that it a) must reobserve landmarks with good confidence and b) is better to follow trajectories that appropriately excite the inertial sensors. This has two effects, namely that of improving the location estimate of the features and that of improving the pose estimate of the robot due to the statistical correlations that link the vehicle to the features. Especially when the robot explores an unknown environment, new features are initialized into the map. This imposes the need to reobserve previous features in order to reduce the growth in localization and mapping error. To encode this behavior into the planning process, each admissible branch of $\mathbb{T}^M$ is evaluated regarding the uncertainty of the robot belief about its pose and the tracked landmarks along its edges as explained in subsection IV-B.2. Then for the derived pose and tracked landmarks covariance matrix $\boldsymbol{\Sigma}_{p,f}$ (subset of $\boldsymbol{\Sigma}_l$), $\mathbf{BeliefGain}$ computes its *D-optimality* (*D-opt*) [7] metric which evaluates the area of the uncertainty ellipsoids and has its roots in the theory of optimal experiments. Therefore, for two robot path policies σ_1^M and σ_2^M, D-opt is used to evaluate which of the two $(l_p + l_f) \times (l_p + l_f)$ covariance matrices $\boldsymbol{\Sigma}_{p,f}(\sigma_1^M)$ and $\boldsymbol{\Sigma}_{p,f}(\sigma_2^M)$ corresponds to more confident belief at the *end–vertex* of each path. Following the unifying uncertainty criteria definitions of Kiefer [30], the particular formulation of [7] is employed:

$$D_{opt}(\sigma^M) = \exp(\log([\det(\boldsymbol{\Sigma}_{p,f}(\sigma^M)]^{1/(l_p+l_f)})) \tag{6}$$

As discussed in [7], this formulation addresses the concerns regarding the capacity of D-opt to enable task completion checking, and its rate of being driven to zero. Accordingly, $\mathbf{BeliefGain}$ computes the D-opt at the end–vertex ξ_{RH} for each admissible branch σ_α^M(where α indexes the admissible branches extracted using the $\mathbf{ExtractBranch}$ function):

$$\mathbf{BeliefGain}(\sigma_\alpha^M) \quad = \quad D_{opt}(\sigma_\alpha^M) \tag{7}$$

Therefore, selection of the path that has optimized belief propagation properties, while arriving at the viewpoint provided by the exploration planning level, is achieved.

C. Computational Complexity

The following provides an approximate analysis of the dominant computational complexity terms of the proposed planner *per iteration step* aiming to reveal how the algorithm can be tuned for optimized performance and tractable computational costs. The planner explores in a volume V^E, the nested planning step takes place in $V^M \subset V^E$, $d_{\max}^{planner}$ is the maximum sensor range for planning, and $d_{\max}^{sensor}$ the maximum sensing distance of the sensor. Furthermore, $N_{\mathbb{T}}^E, N_{\mathbb{T}}^M$ are the number of nodes in the first and second planning level respectively, and l_s, l_p, l_f denote the number of robot states, pose and tracked landmarks respectively. Checking the status of a voxel, and other queries for the full occupancy map, have logarithmic complexity in the

number of voxels [14], that is $\mathcal{O}(\log(V^E/r^3))$. The geometric construction of RRTs in fixed environments scales with $\mathcal{O}(N_{\mathbb{T}}^S \log N_{\mathbb{T}}^S),\ S \to E, M$ and the query for the best node in an RRT scales with $\mathcal{O}(N_{\mathbb{T}}^S),\ S \to E, M$. Moreover, the number of voxels in the fixed volume around an edge to check for collision scales with $1/r^3$ and the complexity to check the $N_{\mathbb{T}}^S - 1,\ S \to E, M$ edges in the occupancy map takes the form $\mathcal{O}(N_{\mathbb{T}}^S/r^3 \log(V^E/r^3)),\ S \to E, M$.

For the first planning level, for every node of the tree $N_{\mathbb{T}}^E$, the relevant gain has to be computed. The volume covered by the sensor takes the form $V_{sensor} \propto (d_{\max}^{planner})^3$ and therefore, the number of voxels to test is approximately $N_{vox} = V_{sensor}/r^3$. As a ray cast for visibility checking is performed for every voxel and its complexity scales with the number of voxels on the ray $\mathcal{O}(d_{\max}^{planner}/r)$, the total complexity for visibility check is $\mathcal{O}(d_{\max}^{planner}/r \log(V^E/r^3))$. This leads to a complexity for the first–level planning gain calculation that takes the form $\mathcal{O}((d_{\max}^{planner}/r)^4 \log(V^E/r^3))$.

For the second planning level, belief propagation contributes to the computational complexity. For every edge of this tree, forward simulation of the robot motion model $\dot{\boldsymbol{\chi}} = f(\boldsymbol{\chi}, \boldsymbol{\chi}^r)$ takes place, as well as the relevant filter state propagation and update steps. Within that, filter propagation and update are the dominant component, compared to cost of simulation, and have a complexity $\mathcal{O}(l_s^{2.4} + l_f^2)$ [31]. That is accounting for sparsity of the Jacobians and efficient matrix inversion but not specific to the implementation details of ROVIO. The required visibility check for each of the possibly visible l_f landmarks takes a constant $\mathcal{O}(5)$ given a pyramid frustum model, and the test for a feature being on the visible side of the occupancy map comes with a $\mathcal{O}(d_{\max}^{sensor}/r \log(V^E/r^3))$ for the maximum considered distance to features. Belief propagation takes place along the branches of $\mathbb{T}_M$. Considering a constant traversing speed v_t the amount of filter steps would be $n_M = \mathbb{N}_T^M \ell_M / v_t / T_s$. Therefore, for analysis purposes, the simplified expression $n_M \propto \mathbb{N}_T^M \ell_M / T_s$ is derived, while it is noted that the exact n_M will depend on the velocity profile of the robot. Then for the propagated belief, the calculation of the D-opt has a complexity $\mathcal{O}(l_p + l_f)$ due to the determinant term and given that only the diagonal terms are considered ($\mathcal{O}((l_p + l_f)^{2.4})$ otherwise) [32]. Finally, it is noted that the construction of the RRT trees for each step dominate their query.

The combined complexity terms per key algorithm functionality are summarized in Table II, while their addition leads to the complete complexity form. As shown, the overall order depends logarithmically on the volume of the planning levels, the number of samples, on the 2.4, 2–powers of the robot state and features belief respectively, linearly on the sum of pose and landmarks vector length, and on the inverse of the sampling time. Therefore, its scalability is maintained given the correlation of V^E, V^M with r, a selection of a reasonable feature set size and the use of an appropriate sampling time given that the simulated system is the closed–loop one. As shown in Section V, good exploration and mapping performance is achieved for online operation in environments of significant size and complexity.

TABLE II
MAIN COMPUTATIONAL COMPLEXITY TERMS

RRT construction	Collision checking
$\mathcal{O}(N_{\mathbb{T}}^S \log(N_{\mathbb{T}}^S)),\ S \to E, M$	$\mathcal{O}(N_{\mathbb{T}}^S/r^3 \log(V^E/r^3)),\ S \to E, M$
1^{st} planning level gain computation	
$\mathcal{O}(N_{\mathbb{T}}^E (d_{\max}^{\text{planner}}/r)^4 \log(V^E/r^3))$	
2^{nd} planning level gain computation	
$\mathcal{O}(N_{\mathbb{T}}^M (d_{\max}^{\text{sensor}}/r)^4 \log(V^E/r^3) l_f + n_M (l_s^{2.4} + l_f^2) + n_M (l_p + l_f))$	

V. EXPERIMENTAL EVALUATION

A challenging experiment was conducted to evaluate and demonstrate the performance of the proposed uncertainty–aware exploration and mapping strategy. The experiment took place in a geometrically complex environment and all the calculations were conducted onboard the aerial robot without any support from motion capture systems.

A. Platform Overview

A custom–built hexarotor platform is employed and has a weight of 2.6kg. The system is relying on a Pixhawk–autopilot for its attitude control, while further integrating an Intel NUC5i7RYH, and the Visual–Inertial Sensor (VI–Sensor) developed at the Autonomous Systems Lab and described in [33]. The VI–Sensor provides time–synchronization of the data from the stereo camera pair (2×Aptina MT9V034) and the IMU (ADIS16448). An overview of the main robot functionalities is depicted in Figure 3. As shown, the planning, localization and mapping, and position control loops are running on the NUC5i7RYH. The system performs visual–inertial odometry using ROVIO and dense mapping through the depth point cloud from the VI–Sensor and the robot pose estimates. For the position control task, the linear model predictive controller described in [34] is utilized. The complete set of high–level tasks run with the support of the Robot Operating System (ROS). Our team has used similar systems before for relevant operations [35–38].

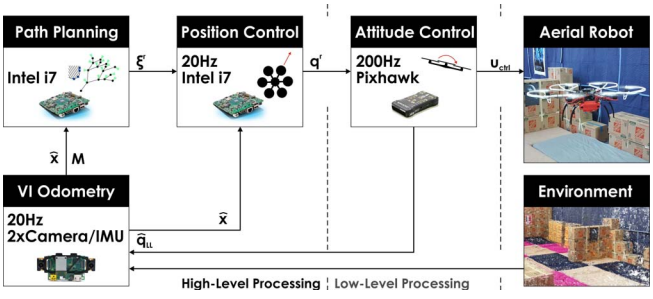

Fig. 3. Overview of the key robot functionalities.

B. Experimental Studies

The experimental scenario refers to the mapping of an indoors environment with dimensions $12 \times 6.5 \times 2$m. Using 300 boxes with size $0.4 \times 0.3 \times 0.3$m, vertical and T–shaped walls, as well as other structural elements are created to complexify the robot exploration and mapping mission.

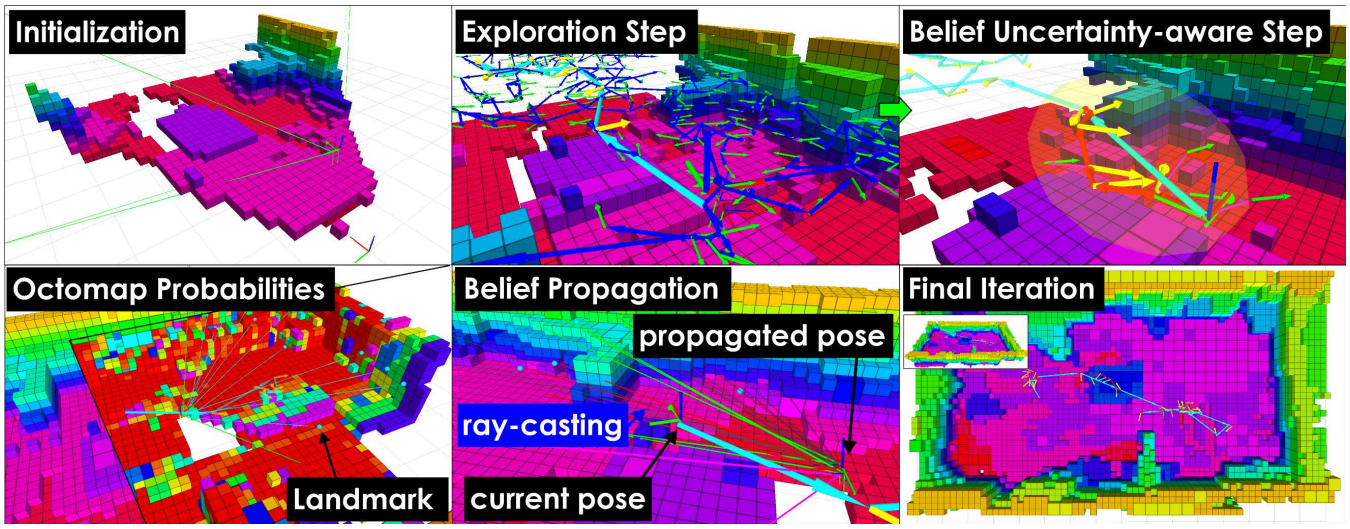

Fig. 4. Instances of an exploration and mapping experiment in a closed room with a challenging geometry. The initial phase of the exploration is dominated by yawing motions. Especially when long paths are selected, the second planning layer identifies alternative paths that optimize the robot belief. Furthermore, as shown the probabilistic backend of octomap is maintained to allow the computation of the **ReobservationGain**, while during belief propagation, visibility check for the tracked landmarks takes place. The result is a consistent 3D map despite the size and the challenges of the environment.

TABLE III
EXPERIMENTAL PARAMETERS

Parameter	Value	Parameter	Value
Volume	12x6.5x2m	Map resolution r	0.2m
$v_{\max}$	0.75m/s	$\dot{\psi}_{\max}$	$\pi/4$rad/s
FoV	$[60, 90]^\circ$	Mounting pitch	13.5°
$d_{\max}^{\text{planner}}$	3.5m	$d_{\max}^{\text{sensor}}$	7.5m
λ	0.35	ℓ_E	1.5m
$N_{\max}^E$	250	ℓ_M	0.375m
$N_{\max}^M$	50	Collision box	1.2x1.2x0.6m
δ	1.5	l_f	25
T_s	0.1	p_{thres}	0.97

Fig. 5. Online reconstructed dense point cloud (right) based on the estimated robot poses and the stereo depth sensing alongside with accurate offline mapping (left) using the Pix4D software and VICON–based pose–annotated HD camera frames.

During the experiment, the robot is constrained to fly with a maximum velocity of $v_{\max} = 0.75$m/s and maximum yaw rate $\dot{\psi}_{\max} = \pi/4$rad/s. The complete set of algorithm parameters and scenario–specific values are summarized in Table III. Multiple instances of the experiment demonstrating the exploratory and belief space–based mapping behavior of the robot are shown in Figure 4. As shown, the second path planning step identifies paths that reduce the robot uncertainty, while the efficient exploratory behavior of the robot is maintained. The result is a visually appealing 3D reconstruction as derived from the stereo depth estimates and their combination based on the robot poses provided from the visual–inertial localization framework. A comparison of this result against an offline reconstruction using the Pix4D software and a set of VICON pose–annotated images is shown in Figure 5. Plots on the exploration evolution, as well as on the evolution of the D-opt metric along the experiment are shown in Figure 6. The complete set of recorded data is released online [16].

VI. CONCLUSIONS

Within this work, an uncertainty–aware exploration and mapping planning strategy is proposed and was demonstrated to be able to efficiently explore unknown environments, while minimizing the robot's localization and tracked landmarks uncertainty. The algorithm employs a receding horizon, two–step, planning paradigm within which it first samples a path for optimized exploration, extracts its first vertex and subsequently identifies a new path to the relevant viewpoint that minimizes the robot's belief uncertainty. Experimental results are presented including specific cases where the proposed planner identified paths of reduced uncertainty. Complexity analysis and comparison with offline 3D maps are further provided and an open dataset is released.

REFERENCES

[1] C. Connolly *et al.*, "The determination of next best views," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2. IEEE, 1985, pp. 432–435.

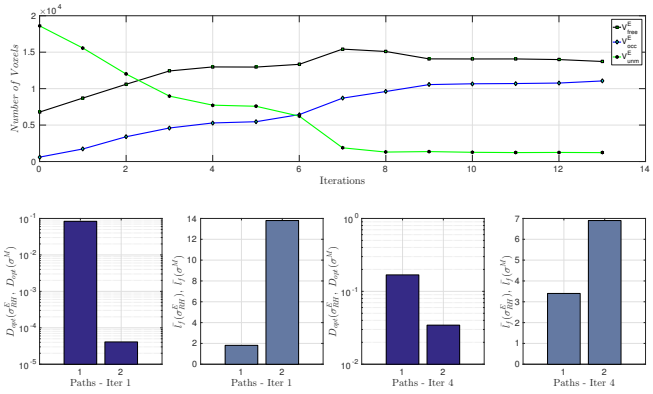

Fig. 6. Rate of exploration per iteration (upper plot, black line for explored free space, blue for explored occupied space and green for the unmapped space), best 2 D-opt gain calculations for the second, nested, planning layer during the 1^{st} and 4^{th} iteration of the algorithm (lower plots, first and third from the left, logarithmic scale of y-axis), and average number of features $\bar{l}_f$ per sample of the path for these paths. As shown, based on belief propagation, the robot often identifies new paths with significantly improved D-opt uncertainty metric which is consistent with tracking in average more features.

[2] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*. IEEE, 1997, pp. 146–151.

[3] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova and R. Siegwart, "Receding horizon "next-best-view" planner for 3d exploration," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2016. [Online]. Available: https://github.com/ethz-asl/nbvplanner

[4] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon path planning for 3d exploration and surface inspection," *Autonomous Robots*, pp. 1–16, 2016.

[5] L. Yoder and S. Scherer, "Autonomous exploration for infrastructure modeling with a micro aerial vehicle," in *Field and Service Robotics*. Springer, 2016, pp. 427–440.

[6] M. Chli and A. J. Davison, "Active matching," in *European conference on computer vision*. Springer, 2008, pp. 72–85.

[7] H. Carrillo, I. Reid, and J. A. Castellanos, "On the comparison of uncertainty criteria for active slam," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012.

[8] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *International journal of computer vision*, vol. 1, no. 4, 1988.

[9] A. J. Davison, and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 865–880, 2002.

[10] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 723–730.

[11] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *The International Journal of Robotics Research*, 2009.

[12] V. Indelman, L. Carlone, and F. Dellaert, "Towards planning in generalized belief space," in *Robotics Research*. Springer, 2016.

[13] W. Burgard, C. Stachniss, and G. Grisetti, "Information gain-based exploration using rao-blackwellized particle filters," in *Proc. of the Learning Workshop*, 2005.

[14] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013.

[15] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006.

[16] C. Papachristos, S. Khattak, and K. Alexis, "Uncertainty–aware Receding Horizon Exploration and Mapping Planner Open Source Package." [Online]. Available: https://github.com/unr-arl/rhem_planner

[17] H. J. S. Feder, J. J. Leonard, and C. M. Smith, "Adaptive mobile robot navigation and mapping," *The International Journal of Robotics Research*, vol. 18, no. 7, pp. 650–668, 1999.

[18] C. Leung, S. Huang, and G. Dissanayake, "Active slam using model predictive control and attractor based exploration," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 5026–5031.

[19] L. Carlone, J. Du, M. K. Ng, B. Bona, and M. Indri, "An application of kullback-leibler divergence to active slam and exploration with particle filters," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 287–293.

[20] G. Costante, C. Forster, J. Delmerico, P. Valigi, and D. Scaramuzza, "Perception-aware path planning," *arXiv:1605.04151*, 2016.

[21] M. W. Achtelik, S. Lynen, S. Weiss, M. Chli, and R. Siegwart, "Motion-and uncertainty-aware path planning for micro aerial vehicles," *Journal of Field Robotics*, vol. 31, no. 4, pp. 676–698, 2014.

[22] M. Bryson and S. Sukkarieh, "Observability analysis and active control for airborne slam," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 1, pp. 261–280, 2008.

[23] R. Sim and N. Roy, "Global a-optimal robot exploration in slam," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 661–666.

[24] A. A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte, "An experiment in integrated exploration," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 1. IEEE, 2002, pp. 534–539.

[25] R. Sim, G. Dudek, and N. Roy, "Online control policy optimization for minimizing map uncertainty during exploration," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 2. IEEE, 2004, pp. 1758–1763.

[26] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, "Information based adaptive robotic exploration," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 1. IEEE, 2002, pp. 540–545.

[27] S. Huang, N. M. Kwok, G. Dissanayake, Q. P. Ha, and G. Fang, "Multi-step look-ahead trajectory planning in slam: Possibility and necessity," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 1091–1096.

[28] S. LaValle and J. Kuffner, J.J., "Randomized kinodynamic planning," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 1, 1999, pp. 473–479 vol.1.

[29] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 298–304.

[30] J. Kiefer, "General equivalence theory for optimum designs (approximate theory)," *The annals of Statistics*, pp. 849–879, 1974.

[31] J.-A. Fernández-Madrigal, *Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods: Introduction and Methods*. IGI Global, 2012.

[32] R. P. Brent and P. Zimmermann, *Modern computer arithmetic*. Cambridge University Press, 2010, vol. 18.

[33] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. T. Furgale and R. Siegwart, "A synchronized visual-inertial sensor system with fpga pre-processing for accurate real-time slam," in *Robotics and Automation (ICRA), IEEE International Conference on*. IEEE, 2014.

[34] M. Kamel, T. Stastny, K. Alexis, and R. Siegwart, "Model predictive control for trajectory tracking of unmanned aerial vehicles using ros," *Springer Book on Robot Operating System (ROS)*.

[35] A. Bircher, M. Kamel, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel and R. Siegwart, "Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots," *Autonomous Robots*, pp. 1–25, 2015.

[36] A. Bircher, K. Alexis, U. Schwesinger, S. Omari, M. Burri, and R. Siegwart, "An incremental sampling-based approach to inspection planning: The rapidly-exploring random tree of trees," 2015.

[37] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel and R. Siegwart, "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 6423–6430. [Online]. Available: https://github.com/ethz-asl/StructuralInspectionPlanner

[38] K. Alexis, C. Papachristos, R. Siegwart, and A. Tzes, "Uniform coverage structural inspection path-planning for micro aerial vehicles," September 2015.