Spatio-Temporally Smooth Local Mapping and State Estimation inside Generalized Cylinders with Micro Aerial Vehicles

Tolga Özaslan[1], Giuseppe Loianno[2], James Keller[1], Camillo J. Taylor[1] and Vijay Kumar[1]

Abstract—**In this paper, we consider state estimation and local mapping with a Micro Aerial Vehicle (MAV) inside a tunnel that can be modeled as a generalized cylinder, using a 3D lidar and an Inertial Measurement Unit (IMU). This axisymmetric environment poses unique challenges in terms of localization and mapping. The point cloud data returned by the sensor consists of indiscriminate partial cylindrical patches complicating data association. The proposed method reconstructs an egocentric local map through an optimization process on a nonlinear manifold, which is then fed into a constrained Unscented Kalman Filter (UKF). The proposed method easily adapts to different diameters, cross-sections and changes in center line curves. The proposed approach outperforms our previous contribution [1] in terms of mapping quality and robustness to non-cylindrical cross-sections. Our motivation is to automate the labor-intensive, dangerous and expensive inspection of penstocks with the least operator intervention. We present experimental results obtained in Center Hill Dam, TN to validate the proposed approach.**

Index Terms—**Aerial System Applications, Field Robots, Mapping & Localization**

I. Introduction

MULTI-ROTOR Micro Aerial Vehicles (MAV) have become popular robotic platforms in the last decade due to their manufacturability, agility and diverse payload options [2], [3], [4]. MAVs are well suited for a number of application areas including inspection, air delivery, surveillance, search and rescue, real estate, entertainment and photography to name a few. MAVs are especially attractive for inspection tasks because of their ability to access hard to reach spots inside mines, penstocks, bridges, ship vessels and other industrial infrastructure.

Simultaneous Localization and Mapping (SLAM) [5], [6], nonlinear control [7], path planning and navigation [8] problems have been extensively studied on these platforms. MAVs optimized for higher payload capacity, yet lightweight and small enough for indoor applications, allow researchers to equip them with different types of sensors such as color and RGB-D cameras, 2D/3D lidars, IMU and GPS units [4], [9]. Unlike earlier studies on SLAM which focus on inferring the world state from a single type of sensor [6], [10], the emergence of MAVs that can move freely through 3D space equipped with a range of sensor modalities, has led the robotics community to develop more sophisticated algorithms [1], [11], [12].

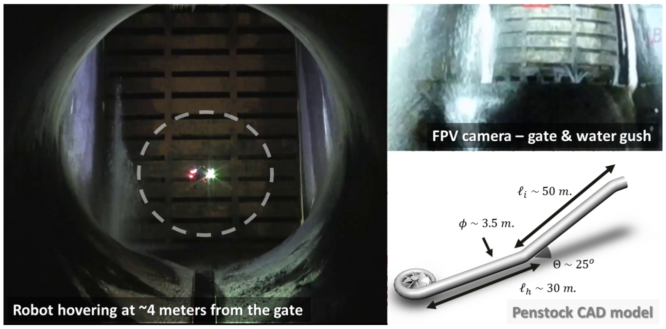

Fig. 1: *(Left)* Custom-designed hex-rotor platform hovering ~ 4 meters from the gate in shared-control mode. *(Top-right)* FPV camera snapshot showing the gate and the water gush. *(Bottom-right)* CAD model of a penstock at Center Hill Dam, TN.

This work focuses on state estimation and local mapping for an MAV equipped with a 3D Lidar and an IMU for autonomous navigation inside penstocks (Fig. 1). We model the environment as a parametric *piecewise-smooth-generalized-cylinder (PSGC)* [13]. The significance of this assumption hence the particular solution we propose is due to our conviction that most of the range-based methods that rely on the presence of geometric cues would fail in similar environments [11], [14]. Although we present results from a single penstock, the proposed principles should not be thought to be peculiar to this infrastructure. They can be used in other environments which exhibit PSGC structure such as mine shafts, caves and sinkholes, highway tunnels and building corridors (Fig. 2).

Penstocks are pitch-dark tunnels with diameters ranging between 3.5 and 20 meters and lengths between 70 and 250 meters with no geometric features except for changes in the bending profile and tunnel diameter (Fig. 1). Our claim is that *uniaxial*, *axisymmetric* and *featureless* tunnels can be mapped with better accuracy and robustness if represented as

Manuscript received: February 24, 2018; Revised May 23, 2018; Accepted July 10, 2018. This paper was recommended for publication by Editor Jonathan Roberts upon evaluation of the Associate Editor and Reviewers' comments.

T. Özaslan acknowledges the fellowship from The Republic of Turkey, Ministry of National Education. This work was supported by RCTA Army Research Office USACE grant W911NF-10-2-0016, ARL grant W911NF-17-2-0181, ONR grants N00014-07-1-0829, N00014-14-1-0510, ARO grant W911NF-13-1-0350, NSF grants IIS-1426840, IIS-1138847, DARPA grants HR001151626, HR0011516850.

[1] The authors are with the GRASP Lab, University of Pennsylvania, 3330 Walnut Street, 19104 Philadelphia, USA.{ozaslan, jkeller, cjtaylor, kumar}@seas.upenn.edu.

[2] The author is with the New York University, Tandon School of Engineering, 6 MetroTech Center, 11201 Brooklyn NY, USA. email: {loiannog}@nyu.edu.

Digital Object Identifier (DOI): see top of this page.

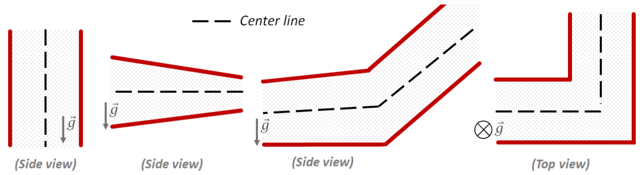

Fig. 2: Illustrations for various piecewise-smooth-generalized-cylinder (PSGC) shaped environments. The center line curves of the right two topologies are piece-wise functions.

a parametric PSGC compared to the common choice of raw point cloud representations. We make use of the geometric properties of the assumed cylinder environment to impose consistency between the segments of a PSGC. We attain this objective through a constrained UKF [15] that allows us to estimate the pose of the vehicle and the local map of the environment on a nonlinear manifold, through a constrained nonlinear optimization on $\mathbb{S}^2 \times \mathbb{S}^2$ for fitting parametric segments from raw point cloud data and a Bernstein interpolation for data association. A novel feature of this work is the use of spherical data analysis tools from the *directional statistical literature* for noise filtering and constrained optimization on $\mathbb{S}^2$ [16], [17].

In our previous work [18], we used a single 2D lidar rectified with a mirror setup to reflect a subset of its rays to the ceiling and floor, and in our other study [9] two 2D lidars one of them sweeping the ground and the other facing forward. However, with only 2D slices from a 3D environment, the map and state inference problems become infeasible unless the map is known *a priori*. In our recent work [1], we used the same 3D lidar as in this work without imposing smoothness constraints on the PSGC segments.

In this work, we overcome several fundamental limitations of our previous work. *First*, we model the uniaxial, axisymmetric tunnel as a parametric, deformable *piecewise-smooth-generalized-cylinder* (PSGC). This imposes constraints on the measurement model and prevents the state from diverging. *Second*, we use the Watson distribution, a statistical tool that can model axially symmetric distributions on $\mathbb{S}^p$, which has not been exploited by the robotics community to our knowledge. We use this tool for outlier elimination on $\mathbb{S}^2$ and to preserve consistency between local map segments especially when the circularity assumption does not hold. *Third*, we integrate sensory information for both pose and local map estimation in a UKF running on a manifold. Finally, to our knowledge, this is the only MAV system that works in PSGC environments.

II. Related Work

After more than a decade of research on MAVs, they have become an important robotic platform employed in many real-life applications. The authors of [19] describe an interesting field study which uses an MAV to inspect ship vessels. This study proposes an automated solution for visual inspection of internal and external regions of a ship hull using a 2D lidar, IMU and a downward facing camera. However, this work makes strong assumptions about the planarity of the floor to simplify the visual odometry problem which may cause issues when flying on curved surfaces.

In an industrial boiler inspection scenario, [20] uses a quadrotor equipped with an IMU and a stereo rig tightly coupled to estimate 6 Degrees Of Freedom (DOF) robot state. Equipped with on-board power-LEDs, the platform does not require any external illumination for the on-board cameras to work. However, the researchers rely on the existence of significant texture on the boiler walls which is not the case inside a penstock. Penstocks are usually coated with tar to prevent rusting and offer very weak texture. In a similar study, the authors of [21] inspect mine shafts with an MAV controlled manually to evaluate its performance under challenging conditions. The authors post-process the data collected with a 2D lidar and a stereo camera rig to reconstruct the vertical mine shaft. This methodology requires an expert pilot and is applicable as long as the pilot has clear view of the platform. In another inspection study [14], the researchers use a pair of rotating 2D lidars to map a 17 km long mine shaft. This is perhaps the closest study to ours, but in this case the mine walls have enough geometric texture to offer information about the motion along the shaft axis which is not the case inside a steel or concrete penstock.

3D mapping of cluttered environments and state estimation of MAVs is often studied using either a 3D lidar or a 2D lidar mounted on a rotating base. [22] uses the latter hardware to map both indoor and outdoor environments with a feature-based point cloud registration technique. The authors report that their estimator experiences very low drift and is not demanding in terms of CPU power despite the complexity induced by integrating each 2D range reading at different times due to the hardware setup. This method is, however, not applicable in our case due to the lack of geometric features inside a tunnel. In [23], the authors propose an MAV system that relies on an on-board RGB-D sensor and an IMU to estimate the robot state and map its surroundings. The proposed system is mostly relevant to indoor settings due the limited sensing capability of the RGB-D sensor. This work also demonstrates path planning and navigation using the 3D occupancy grid map previously generated by the robot.

III. Preliminaries

A. Environment

A penstock has a well-defined geometry rarely encountered in other indoor settings that needs to be characterized in particular. Binford first introduced the notion of a *generalized cylinder (GC)* in 1973 [24]. This became popular in robotics after Brooks' work in 1983 [13]. In this work, we have chosen to model penstocks, as a generalization of an ordinary cylinder where the center axis can be any continuous spine and its radius may change along the axis (Fig. 2). This environment poses unique challenges for an on-board state estimation system because of the axial symmetries. Except for a small volume close to the distal ends of the tunnel, a lidar-based estimator cannot sense position along the tunnel axis. Also the only source of roll and pitch angles is the IMU due to symmetry. Lastly, auxiliary sensors such as barometers,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2018.2861888, IEEE Robotics and Automation Letters

magnetometers are often affected by vortices generated by propeller down-wash and metal structure respectively. We showed in [9] that cameras with on-board illumination can be an option if the tunnel surface has visual texture.

There is typically a vertical gate at the upper end of a penstock which the robot should sense to infer completion of an autonomous traversal. A sample CAD model of a penstock is given in Fig. 1.

B. MAV Platform

The challenges and specifications of the task require design of a custom MAV platform. We use DJI F550 frame[1] with extended arms that allow using DJI E600 propulsion system[2] to obtain a high weight to thrust ratio. The total weight of the platform is approximately 4.5 kg when equipped with a Velodyne Puck LITE [3], a Pixhawk autopilot [4], a 5^{th} generation Intel i7 NUC board, a 6S 5000mAh 50C LiPo battery and a custom-designed power distribution board. Although we do not use in this specific scenario, the robot also carries 4 Chameleon3 1.3MP color cameras[5] and high-power Cree LEDs[6] for on-board illumination and imagery collection (Fig. 1). The total flight time is more than 12 minutes for non-aggressive flights.

C. Robot State and Local Map

This section explains the choice of robot state and local map representations. The reader should be aware that this work estimates the *local map* only. The robot state is defined as

$$\mathbf{x}^\top = [\, \mathbf{r}^\top,\ \dot{\mathbf{r}}^\top,\ \mathbf{q}^\top \,] \tag{1}$$

consisting of position, velocity both along the tunnel cross-section and orientation respectively. State uncertainty is denoted as $\mathbf{\Sigma_x}$.

We model the local map, $\mathcal{M}$, as a *deformable* PSGC consisting of segments, $\{\mathcal{S}^i\}$, with fixed inter-segment Euclidean distance. A segment is defined by its position, local tunnel axis and radius which respectively is written

$$\mathcal{S}^\top = [\, \mathbf{k}^\top,\ \hat{\mathbf{t}}^\top,\ \rho \,] \tag{2}$$

where $\mathbf{k} \in \mathbb{R}^3$, $\hat{\mathbf{t}} \in \mathbb{S}^2$ and $\rho \in \mathbb{R}$. Segment uncertainty is denoted as $\mathbf{\Sigma}_\mathcal{S}$.

We index the segment closest to the lidar with 0 and others after and before this with positive and negative indices respectively. Thus a local map is written as $\mathcal{M} = \{\cdots, \mathcal{S}^{-2}, \mathcal{S}^{-1}, \mathcal{S}^0, \mathcal{S}^1, \mathcal{S}^2, \cdots\}$ (Fig. 3). To refer to adjacent segments to $\mathcal{S}^i$, we use the short-hand notation of $\mathcal{S}^{i^+}$ and $\mathcal{S}^{i^-}$ where $i^+ = i + sgn(i)$ and $i^- = i - sgn(i)$.

There are three frames we use throughout the paper which are the world, $\mathcal{W}$, local map, $\mathcal{M}$, the body, $\mathcal{B}$, frames. We assume that the IMU, 3D lidar and body origins are the same for the sake of clarity. We often explicitly denote the frame at which a variable is represented at as $\mathbf{x}^\mathcal{F}$. The reader should assume $\mathcal{B}$ when a frame is not explicitly mentioned. Fig. 3 illustrates these frame definitions.

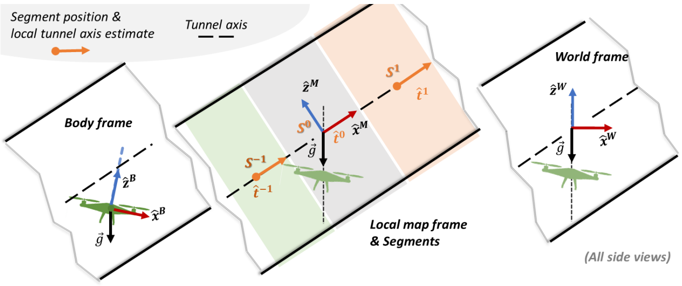

Fig. 3: An illustration of the three frame definitions, $\mathcal{B}, \mathcal{M}$ and $\mathcal{W}$ with segments shown with shaded boxes and their corresponding position and local axis tangent estimates.

Since we cannot estimate robot position along the tunnel axis with the current sensor package, $\mathcal{W}$ cannot be set to a fixed physical point in space. The only globally sensible DOFs are roll and pitch due to the IMU. Hence $\mathcal{W}$ and $\mathcal{M}$ differ only by roll and pitch angles.

D. Watson Distribution on $\mathbb{S}^2$

The statistical tool most often used in regression and estimation problems is the normal distribution which is designed for Euclidean space. Although under certain circumstances and with appropriate assumptions, random variables on manifolds can be analyzed with this tool, we believe that the directional statistics literature has better tools deliberately designed for spherical data. Because of this, we use Watson distribution on $\mathbb{S}^2$ (Section 9.4 in [16]) to describe the set of possible tunnel axis tangent estimates as well as to perform outlier elimination which will be covered in the subsequent sections. For this, we keep track of surface normal distributions of each segment, $\mathcal{S}^i$. We refer to the Watson distribution parameters corresponding to $\mathcal{S}^i$ as $\{\mu^i, \kappa^i\}$ or shortly use W^i and W^i_m.

The density of Watson distribution which yields the probability density over the set of direction vectors $\mathbf{x}$ is written as

$$W(\pm\mathbf{x}, \boldsymbol{\mu}, \kappa) = M\left(\frac{1}{2}, \frac{d}{2}, \kappa\right)^{-1} \exp\left(\kappa(\boldsymbol{\mu}^\top \mathbf{x})^2\right) \tag{3}$$

where μ is the mode, κ is the concentration parameter and $M\left({}^1\!/_2, {}^d\!/_2, \kappa\right)$ is the Kummer function which is

$$M\left(\frac{1}{2}, \frac{d}{2}, \kappa\right) = \beta\left(\frac{d-1}{2}, \frac{1}{2}\right)^{-1} \int_{-1}^{1} e^{\kappa t^2}(1-t^2)^{\frac{(d-3)}{2}} dx \tag{4}$$

where $d = 3$ for the $\mathbb{S}^2$ case (we will use M for clarity) and β is the beta function. The Watson distribution takes the form of a bipolar distribution for $\kappa > 0$ and becomes a girdle distribution for $\kappa < 0$ [16]. Different synthetic data distributions are shown in Fig. 4.

The problem of estimating the most likely local tunnel axis tangent relates to finding the mode of a Watson distribution. When $\kappa \geq 0$, the likelihood of a point, $\mathbf{p}$, being the mode is the same as the distribution itself, i.e. $W_m = W(\pm\mathbf{p}, \boldsymbol{\mu}, \kappa)$.

[1] https://www.dji.com/flame-wheel-arf
[2] https://www.dji.com/e600
[3] http://velodynelidar.com/vlp-16-lite.html
[4] https://pixhawk.org/modules/pixhawk
[5] https://www.ptgrey.com/chameleon3-usb3-vision-cameras
[6] http://www.cree.com/xlamp/xhp50_2

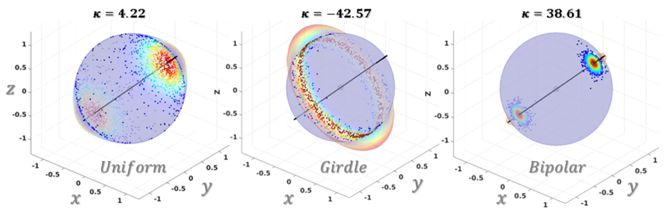

Fig. 4: Synthetic data on $\mathbb{S}^2$ sampled from uniform, girdle and bipolar distributions.

However when $\kappa < 0$, the mode likelihood becomes

$$W_m(\pm\mathbf{p},\boldsymbol{\mu},\kappa) = \frac{2}{M}\int_0^\pi \exp\left\{\kappa\left(1-\left[\boldsymbol{\mu}^\top\mathbf{p}\right]^2\right)\cos(\theta)^2\right\} d\theta \tag{5}$$

which is an integral over the great circle perpendicular to $\mathbf{p}$. Since these integrals are prohibitively time demanding, we use lookup tables in our implementation.

E. Point Cloud Processing

Accurate surface normal estimation plays an important role in the proposed method since segment fitting is formulated as an optimization problem with its cost function dependent on normals and local curvature as explained in the subsequent sections. We use the method presented in [25] for surface normal and curvature estimation. This method calculates the mean and the covariance of the k-nearest neighbors for a given point and takes the eigenvector corresponding to the smallest eigenvalue as its local surface normal. The ratio of the smallest eigenvalue to the summation of all three is taken as the curvature at the point of interest [1]. Nearest neighbor search is sped up using space partitioning techniques. We also apply a voxel filter on the raw point cloud to reduce the point count and save CPU time.

A single frame, raw point cloud reading from the lidar is denoted as $\mathcal{P}$ and its subset, not necessarily exclusive, used in fitting a segment, $\mathcal{S}^i$, is denoted as $\mathcal{P}^i$. This consists of points that are inside a rotated-box centered at an initial guess for $\mathcal{S}^i$. Similarly, we denote normals and curvature which are a part of $\mathcal{P}^i$ as $\mathcal{N}^i$ and $\mathcal{K}^i$ as a short-hand notation.

IV. Segment Fitting

The proposed method processes the raw point cloud data to obtain a set of segments (i.e. a local map, $\mathcal{M}$) which are then fed into a UKF for robustness against either noise in the raw data or erroneous segment fits. This section describes the segment fitting process formulated as a constrained optimization on $\mathbb{S}^2 \times \mathbb{S}^2$.

This method uses the point coordinates and their surface normals to define cost functions to be minimized. For robustness against noisy data points which may be due to scaffolding or other obstacles, we weight the data points as a function of local curvature. Furthermore, the likelihood of a data point to belong to a particular segment is determined using the segment fit at the previous time step as well as the neighboring segments. We apply a similar strategy for evaluating the likelihood of the axis tangent hypotheses which we will explain shortly.

A. Optimizer Definition

The cost function is defined as $C = C_{\mathbf{k}} + C_{\hat{\mathbf{t}}} + C_W$ where

$$C_{\mathbf{k}} = \frac{1}{\sum w_i}\sum w_i \left|\bar{\rho} - \|\boldsymbol{\Delta}\mathbf{p}_i\|\right| \tag{6}$$

$$C_{\hat{\mathbf{t}}} = \frac{1}{\sum w_i}\sum w_i \left(\hat{\mathbf{n}}_i^\top \hat{\mathbf{t}}\right)^2 \tag{7}$$

$$C_W = -\log\left(W_m(\hat{\mathbf{t}},\mu,\kappa)\right) \tag{8}$$

with all the vectors written in the body frame and summations running over all the points of the segment optimized. For clarity we omit segment indices, but the reader should be aware that C is written for a particular segment, $\mathcal{S}^i$, using its corresponding point cloud $\mathcal{P}^i \subset \mathcal{P}$ and spherical distribution, W_m^i. The terms in the above equations penalize respectively *(1)* the discrepancy between each point-axis distance and the mean value, *(2)* incompatibility between surface normals and the tunnel axis, and *(3)* the difference between the local axis tangent and the mode of the corresponding segment's Watson distribution. The mean radius $\bar{\rho}$ is calculated as

$$\bar{\rho} = \frac{1}{\sum w_i}\sum w_i \|\boldsymbol{\Delta}\mathbf{p}_i\| \tag{9}$$

where summations run over all the points of the segment optimized, $\boldsymbol{\Delta}\mathbf{p}$ is the shortest vector from point $\mathbf{p}$ to the local axis, $\mathbf{k} + \alpha\hat{\mathbf{t}}$ for $\alpha \in \mathbb{R}$, and is given by

$$\boldsymbol{\Delta}\mathbf{p} = (\mathcal{I} - \hat{\mathbf{t}}\ \hat{\mathbf{t}}^\top)(\mathbf{p} - \mathbf{k}). \tag{10}$$

The solution to the fitting problem can be written as

$$\{\mathbf{k}^*, \hat{\mathbf{t}}^*\} = \underset{\mathbf{k},\hat{\mathbf{t}}}{\operatorname{argmin}}\ C. \tag{11}$$

However this choice of free parameters does not impose any constraint between segment positions hence may undesirably cause $\mathbf{k}$ end up at a far point from its neighboring segments. For this, we rewrite this parameter for the segment $\mathcal{S}^i$ as

$$\mathbf{k}^i = \mathbf{k}^{i^-} + \ell^i \hat{\mathbf{k}}^i \tag{12}$$

where $\ell^i = \|\mathbf{k}^i - \mathbf{k}^{i^-}\|$ and $\hat{\mathbf{k}}^i = \frac{\mathbf{k}^i - \mathbf{k}^{i^-}}{\ell^i}$ is the direction vector between consecutive segment positions with the abuse of notation. Fig. 6 shows these two variables on a sample map. We can rewrite the solution as

$$\{\hat{\mathbf{k}}^*, \hat{\mathbf{t}}^*\} = \underset{\hat{\mathbf{k}},\hat{\mathbf{t}}}{\operatorname{argmin}}\ C \tag{13}$$

with ℓ being constant and both free parameters belonging to $\mathbb{S}^2$.

In a real-life scenario, a penstock does not bend sharply. This constraint can be easily imposed on the solver with this choice of parameters. However the root segment, $\mathcal{S}^0$, is an exception to this where the change of free parameters cannot be applied. In that case we stick with the model in Equ. 11 and apply the constraint $\mathbf{k}^0 \cdot \hat{x}^{\mathcal{M}} = 0$.

The optimizer uses the Levenberg-Marquardt method with adaptive step size to minimize C. The update is written as

$$\begin{bmatrix} \Delta\hat{\mathbf{k}}^\top, & \Delta\hat{\mathbf{t}}^\top \end{bmatrix}^\top = \left(\mathbf{J}\ \mathbf{J}^\top + \lambda\mathcal{I}\right)^{-1} \mathbf{J} \tag{14}$$

where $\mathbf{J} = \nabla C$ and $\lambda > 0$ is the damping parameter that determines the step size. Each update is first projected onto $\mathbb{S}^2$ at the most recent position as

$$\hat{\mathbf{k}}_{new} = \hat{\mathbf{k}} + (\mathcal{I} - \hat{\mathbf{k}}\ \hat{\mathbf{k}}^\top)\ \Delta\hat{\mathbf{k}} \tag{15}$$

$$\hat{\mathbf{t}}_{new} = \hat{\mathbf{t}} + (\mathcal{I} - \hat{\mathbf{t}}\ \hat{\mathbf{t}}^\top)\ \Delta\hat{\mathbf{t}} \tag{16}$$

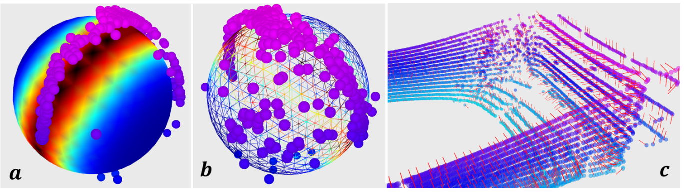

Fig. 5: Sample surface normal data from Center Hill Dam experiments and the corresponding girdle distribution, i.e. $\kappa < 0$, fit with the method explained in Sec. IV. *(a)* Likelihoods are color-coded with blue and black corresponding to low and high values respectively. *(b)* Vertices of the mesh are used as hypothesis. *(c)* Point cloud from which the distribution is obtained.

followed by projection onto the unit sphere (not shown). The radius is automatically estimated as in Equ. 9 once the optimization converges. The uncertainty of the segment is taken as the inverse of the Hessian of C. We disregard the results of the optimization if the resultant axis tangent is outside the 95% confidence interval of W_m.

B. Point Weights and Fitting a Watson Distribution

The two other important components of the segment fitter are the Maximum Likelihood Estimate (MLE) of the spherical distributions, $\{\mu^i, \kappa^i\}$, and the calculation of weights, w_j. The MLE of a Watson distribution is obtained by using both the point cloud data and the distributions from the previous time step. Watson parameters corresponding for the i^{th} segment can be written as $p(\mu^i, \kappa^i|\mathcal{P}^i, \{W^j\})$. While the former condition steers MLE towards the current data, the latter helps preserving inter-segment consistency and prevents the optimizer from diverging. As explained in Chapter 10 of [16] concentration of a Watson distribution is a function of its mode and data points. Hence $p(\mu^i|\mathcal{P}^i, \{W^j\}) = p(\mu^i, \kappa^i|\mathcal{P}^i, \{W^j\})$. We use a fixed, large number of hypotheses (i.e. points) uniformly placed on an icosphere of unit radius as shown in Fig. 5. These hypotheses are generated by iteratively subdividing a convex regular icosahedra. A hypothesis, h, for a specific segment's potential Watson distribution mode is scored as

$$-\sum\left\{k_i^2+\left(h^\top\hat{\mathbf{n}}_i\right)^2\right\}+\sum\gamma^j\, log\left(W_m^j(h)\right) \qquad (17)$$

where the first summation runs over all of the surface normals, $\hat{\mathbf{n}}_i$, and curvatures, k_i, of the specified segment, and the second summation runs over all of the segments. The multiplicative factor, γ^j, is a decreasing function of the distance between the segment of concern and its neighboring segments. We can liken this strategy to the averaging operation on signals using smoothing kernels.

The hypothesis with the highest score is chosen as the mode, μ^*, of the corresponding distribution. The effective eigenvalue, e, and the weight of a given point, w_i are calculated as

$$e = \frac{\sum w_i|\hat{\mathbf{n}}_i^\top\mu^*|}{\sum w_i} \qquad (18)$$

$$w_i = -k_i^2 - \left(\hat{\mathbf{n}}_i^\top\mu^*\right)^2 + \sum\gamma^j\, log\left(W^j(\hat{\mathbf{n}}_i)\right). \qquad (19)$$

where the summation runs over all of the segments. Finally we find the concentration parameter, κ^*, by plugging e into the inverse of Equ. 10.3.32 in [16]. If $-5 < \kappa$ we conclude that the surface normal data is extremely noisy and we terminate the process. This corresponds to either a uniform distribution if $|\kappa|$ is small, or bipolar distribution if $\kappa > 0$.

One can argue that the mode of a spherical distribution as we proposed can be directly used as the local axis tangent avoiding the need for an optimization procedure. However, we think that a single distribution, such as Watson, compresses the point cloud information into two parameters, μ and κ, which blurs most of the details in the raw data. Hence we use the Watson distribution as a tool for guiding the optimizer in our procedure.

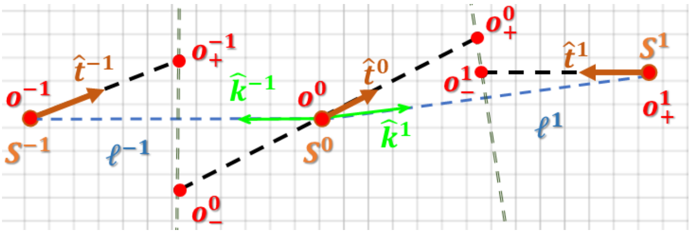

Fig. 6: Generation of Bernstein knots from tunnel axis tangents illustrated on a sample three segment local map.

V. Bernstein Interpolation on Local Map

The method explained in the previous section provides a discrete representation of the environment. However, given that a penstock is smooth, we can interpolate the sections between segments using a Bernstein spline. This has several benefits such as the total curvature of the tunnel centerline provides a metric for the fitness of the local map, a 3D mesh of the tunnel can be reconstructed which then may be used for path planning etc. More importantly, two maps can be compared using their corresponding Bernstein splines to quantify how *similar* or *close* they are to each other.

A Bernstein spline is defined by an ordered set of knots, o^i. In our case, only the positions of segments could have been used as the knots which would give an acceptable approximation. However, we would like also to incorporate the axis tangent information for a more accurate interpolation. For this, we add two more knots to the opposite sides of each segment (*one* to the end segments) which are obtained as

$$o_-^i = \mathbf{k}^i + sgn(i)\frac{\ell^i}{2}\,\hat{\mathbf{t}}^i\left(\hat{\mathbf{t}}^i\cdot\hat{\mathbf{k}}^i\right)^{-1} \qquad (20)$$

$$o_+^i = \mathbf{k}^i + sgn(i)\frac{\ell^{i+}}{2}\,\hat{\mathbf{t}}^i\left(\hat{\mathbf{t}}^i\cdot\hat{\mathbf{k}}^{i+}\right)^{-1} \qquad (21)$$

where ℓ and $\hat{\mathbf{k}}$ are defined in Sec. IV (Fig. 6). With these additional knots, a local map consisting of n_s segments is approximated with a Bernstein of $3n_s - 2$ knots. The corresponding Bernstein spline for the local map, $\mathcal{B}_\mathcal{M}$, is obtained as

$$\mathcal{B}_\mathcal{M}(t) = \sum_{v=0}^{n-1} p_v\; B_{v,n}(t) \qquad (22)$$

$$B_{v,n}(t) = \binom{n}{v} t^v(1-t)^{n-v} \qquad (23)$$

where $n = 3n_s - 3$, $t \in [0, 1]$, the second equation is the v^{th} Bernstein basis polynomial of degree n and $\{p_i\}$ is the ordered array of knots given as

$$\{p_i\} = \left\{\cdots, o_{+}^{-1}, o^{-1}, o_{-}^{-1}, o_{-}^{0}, o^{0}, o_{+}^{0}, o_{-}^{1}, o^{1}, \cdots\right\}. \quad (24)$$

We can define a distance metric between two segments using their positions and axis tangents as

$$\|\mathcal{S}^i, \mathcal{S}^j\| = \alpha_p \|\mathbf{k}^i - \mathbf{k}^j\| + \alpha_o \left(1 - |\hat{\mathbf{t}}^i \cdot \hat{\mathbf{t}}^j|\right) \quad (25)$$

where α_p and α_o are weights assigned to position and orientation difference. With this metric we can define an operation which gives the closest point to a segment on a spline as

$$\mathcal{B}(\mathcal{S}) = \left\{\mathcal{B}(t^*) \middle| t^* = \underset{t}{\operatorname{argmin}} \|\mathcal{B}(t), \mathcal{S}\|, t \in [0, 1]\right\}. \quad (26)$$

The reader should notice that we used a point on the spline rather than a segment in the distance calculation. But this is a valid operation, since a point on a spline has both a position and a tangent which equals ${}^{d\mathcal{B}(t)}/_{dt}$ followed by normalization. We will use this operation during the calculation of innovation in the measurement update step of a UKF.

VI. Filter Design

The method explained in the previous section gives noisy or, in the worst case, completely wrong segment fits. For this, we use a UKF [15] to handle noise from both the IMU and the lidar. As will be explained shortly, nonlinearities due unit vector components of the local map prohibit the use of a linear Kalman Filter (KF). Although Extended Kalman Filter (EKF) offers a solution for such systems, linearization of the process and the measurement models can get very complicated and prohibitively time consuming. Furthermore, a UKF is capable of capturing nonlinearities up to third order [15].

A. The State Vector

The system state vector can be written as

$$\mathbf{s}^\top = \left[\hat{\mathbf{g}}^\top, \mathbf{b}^\top, \mathbf{v}^\top, {\mathcal{S}^0}^\top, ..., {\mathcal{S}^i}^\top, ...\right] \quad (27)$$

where $\hat{\mathbf{g}} \in \mathbb{S}^2$ is the gravity vector direction, $\mathbf{b} \in \mathbb{R}^6$ is accelerometer and gyroscope biases, $\mathbf{v} \in \mathbb{R}^3$ is the linear velocity of the map along the local tunnel cross-section all defined in the body frame. The size of $\mathbf{s}$ changes depending on the visible range of the tunnel by addition and removal of segments. The *unconventional* state vector $\mathbf{s}$ has the *minimal* set of variables that *fully* represents the robot state and the local map in this unique environment. Although the robot state is not explicitly included in $\mathbf{s}$, it can be referred from $\mathcal{M}$ as explained in Sec. VI-E. Finally the system state covariance is denoted as $\mathbf{\Sigma}_s$.

B. The Process Model

The process model $\mathcal{A}$ predicts the evolution of the system state with the control input, $\mathbf{u}$, and the process noise, $\mathbf{w}$. The form of the process models is

$$\mathbf{s}_{k+1} = \mathcal{A}(\mathbf{s}_k, \mathbf{u}_k, \mathbf{w}_k) \quad (28)$$

$$\mathbf{u}^\top = \left[\ {\mathbf{u}_a}^\top, \quad {\mathbf{u}_\omega}^\top\ \right] \quad (29)$$

$$\mathbf{w}^\top = \left[\ {\mathbf{w}_a}^\top, \quad {\mathbf{w}_\omega}^\top, \quad {\mathbf{w}_{\mathbf{b}_a}}^\top, \quad {\mathbf{w}_{\mathbf{b}_\omega}}^\top\ \right] \quad (30)$$

where $\mathbf{u}$ is the control input comprising of acceleration and rotational velocity data from the IMU, $\mathbf{w} \sim \mathcal{N}(0, \mathbf{Q})$ is the process noise consisting of noise in acceleration, angular velocity, acceleration bias, gyroscope bias, and $\mathbf{Q}$ is the process noise covariance.

The process model $\mathcal{A}$ is defined as

$$\hat{\mathbf{g}}_{k+1} = \exp(\boldsymbol{\omega}_k\ \Delta t)\hat{\mathbf{g}}_k \quad (31)$$

$$\mathbf{b}_{k+1} = \mathbf{b}_k + \mathbf{w}_{\mathbf{b},k}\Delta t \quad (32)$$

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \tilde{\mathbf{a}}_k\Delta t \quad (33)$$

$$\mathbf{k}_{k+1}^i = \exp(\boldsymbol{\omega}_k\ \Delta t)\left(\mathbf{k}_k^i + \mathbf{v}_k\Delta t + \frac{1}{2}\tilde{\mathbf{a}}_k\Delta t^2\right) \quad (34)$$

$$\hat{\mathbf{t}}_{k+1}^i = \exp(\boldsymbol{\omega}_k\ \Delta t)\hat{\mathbf{t}}_k^i \quad (35)$$

where $\tilde{\mathbf{a}} = -(\mathbf{u}_a + g\hat{\mathbf{g}} - \mathbf{b}_a + \mathbf{w}_a)$, g is the gravitational acceleration, $\boldsymbol{\omega} = -(\mathbf{u}_\omega - \mathbf{b}_\omega + \mathbf{w}_\omega)$.

C. Measurement Model

The measurement model, $\mathcal{H}$, relates a measurement, $\mathbf{z}$, to the system state, $\mathbf{s}$. It also integrates the effect of the measurement noise, $\mathbf{m}$, into the expected measurement at a specific state.

In this work, we use two measurement models $\mathcal{H}_{\hat{\mathbf{g}}}$ and $\mathcal{H}_{\mathcal{S}}$ for gravity vector estimation using the IMU and to update the whole system state using the point cloud data respectively. Input to the latter model is an array of *segment measurements*, $\{\tilde{\mathcal{S}}^i\}$, obtained using the method explained in Sec. IV.

The general form of a measurement models is

$$\mathbf{z}_k = \mathcal{H}(\mathbf{s}_k, \mathbf{m}_k) \quad (36)$$

where $\mathbf{z}$ is either $\mathbf{u}_a$ or $\tilde{\mathcal{S}}$. $\mathbf{m} \sim \mathcal{N}(0, \mathbf{Y})$ is the sensor noise defined separately for each measurement model. The former measurement model is defined as

$$\mathcal{H}_{\hat{\mathbf{g}}} : \mathbf{u}_a = -g\hat{\mathbf{g}} + \mathbf{b}_a + \mathbf{m}_{\hat{\mathbf{g}}} \quad (37)$$

where $\mathbf{m}_{\hat{\mathbf{g}}} \sim \mathcal{N}(0, \mathbf{Y}_{\hat{\mathbf{g}}})$. The latter measurement model is defined as

$$\mathcal{H}_{\mathcal{S}} : \tilde{\mathcal{S}} = \mathcal{B}_{\mathcal{M}}(\tilde{\mathcal{S}}) + \mathbf{m}_{\mathcal{S}} \quad (38)$$

where the left-hand-side is a segment measurement obtained through optimization as explained in Sec. IV, and the right-hand-size is the point-tangent-radius tuple closest to $\tilde{\mathcal{S}}$ along $\mathcal{B}_{\mathcal{M}}$ in the sense of Equ. 26. Finally, $\mathbf{m}_{\mathcal{S}} \sim \mathcal{N}(0, \mathbf{Y}_{\mathcal{S}})$ is the measurement noise.

D. A Constrained UKF on a Nonlinear Manifold

The unit vectors in $\mathbf{s}$ belong to a nonlinear manifold hence prohibit using a linear KF. For this, we use a UKF since it neither requires process or measurement models to be linear nor linearization of these as in the case of EKF. Through application of process and measurement models on the sigma points, one can attain significantly better results compared to alternative KF variations. We leave the details of UKF to the seminal paper [15] and explain issues specific to our choice of state definition.

The state covariance, denoted as $\mathbf{\Sigma}_s$, is a $12 + 7n_s$ square, positive definite matrix where n_s is the number of segments. Sigma points are obtained as $\mathcal{X}_i = \mathbf{s} \pm \mathcal{W}_i$ where $\{\mathcal{W}_i\}$ are the columns of $\sqrt{\mathbf{\Sigma}_s}$ obtained through *Cholesky* decomposition.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2018.2861888, IEEE Robotics and Automation Letters

The blocks of $\mathbf{\Sigma}_s$ corresponding to $\mathbf{b}, \mathbf{v}, \mathbf{k}^i$ and ρ^i hold the corresponding uncertainties which belong to Euclidean space. However, the uncertainties of $\hat{\mathbf{g}}$ and $\hat{\mathbf{t}}$ are represented as 3D rotation vectors [26]. In sigma point generation, mean and covariance calculation steps, these elements have to be handled properly. That is, we have to define summation, difference and mean operations on these elements.

We define the *summation* of a unit vector, $\hat{\mathbf{a}}$, with a rotation vector, $\mathbf{b}$, as $\hat{\mathbf{a}} \oplus \mathbf{b} = \exp(\mathbf{b})\hat{\mathbf{a}}$. Similarly the subtraction operation is defined as $\hat{\mathbf{a}} \ominus \mathbf{b} = \exp(-\mathbf{b})\hat{\mathbf{a}}$. Finally, the mean of a set of unit vectors, $\{\hat{\mathbf{a}}\}$, is $\hat{\bar{\mathbf{a}}}$ which minimizes $\sum_{\{\hat{\mathbf{a}}\}} \hat{\mathbf{a}} \times \bar{\mathbf{a}}$ [26].

The innovation of UKF does not respect the constant distance constraints, ℓ^i, between adjacent segments. Furthermore, since the motion along the tunnel axis is not observable, velocity components along the tunnel axis should be always zero, i.e. $\mathbf{v}^{\mathcal{M}} \cdot \hat{\mathbf{x}}^{\mathcal{M}} = 0$. We also want to keep the the position of the root segment, $\mathbf{k}^0$, along the tunnel axis fixed for the same reason. This can written as $\mathbf{k}^0 \cdot \hat{\mathbf{x}}^{\mathcal{M}} = 0$.

We use the method explained Section 3 of [27] to integrate the effect of these constraints on the segment positions at the end of each UKF cycle.

E. Robot State

Although we do not explicitly include the robot state in the UKF, it is indirectly estimated together with the local map. The position of the root segment, which is also the origin of world frame, is contained by the UKF as explained in the previous section. Hence $-\mathbf{k}^{0,\mathcal{W}}$ gives the position of the robot in the world frame. Furthermore, roll and pitch angles can be inferred from $\hat{\mathbf{g}}$ and yaw from the difference between $\hat{\mathbf{x}}^{\mathcal{B}}, \hat{\mathbf{x}}^{\mathcal{W}}$ pair. Finally, corresponding uncertainties of these variables can be extracted from the state covariance to obtain robot state uncertainty.

VII. Experimental Results

In this section, we report on the experiments that we performed inside a penstock in Center Hill Dam, TN. We compare this approach with our previous work [1] to demonstrate how the new formulation and filtering mechanisms improve robustness around critical regions such at the top of the penstock. This penstock has a diameter of ~ 5 meters and is about 70 meters long. We could experiment only along certain parts of the penstock due to an on-going construction on the site. Because of this we could not collect data around the turbine area which is the lower part of a penstock shown at the bottom-right of Fig. 1. The data we present is from the horizontal and inclined sections of the tunnel.

We used our custom-designed platform, which is explained in Sec. III-B equipped with a 3D lidar and an IMU. Our algorithms are implemented in ROS, optimized for real-time performance and uses less than a single core. Since our tests were conducted in confined space, we do not have ground truth data for comparison. A tunnel reconstruction that shows PSGC geometry also means that the state is estimated accurately since the robot pose is a obtained through inverting map parameters.

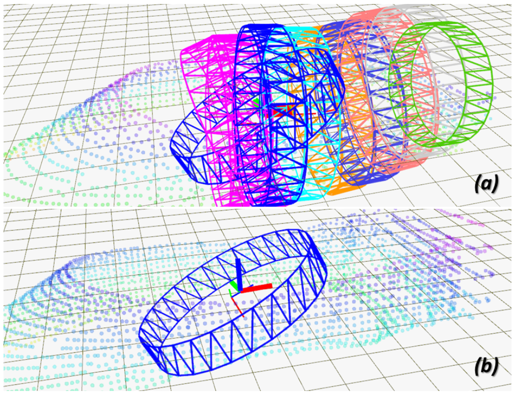

Fig. 7: These RViz snapshots show failure of the approach in [1] failing while hovering close to the gate where the circular cross-section assumption does not hold. *(a)* Orientation of some segments are estimated wrong and the spacing between segments are not uniform. *(b)* Since the first segment is estimated wrong, the fitting process is early terminated.

We compare the proposed approach with our previous work [1]. Our previous work follows a similar approach and reconstructs the tunnel by fitting segments. However, it does not impose any constraints between segments and also only uses the point curvature to weight the reliability of point cloud data. This often results in segments either far from their neighboring segment hence does not preserve continuity. Also when the robot flies in non-cylindrical regions, segments may end up at angles off by 90 degrees compared to neighbor segments. Two cases where the previous approach fails are shown in Fig. 7. In these figures, the robot is close to the gate where the tunnel cross-section is not cylindrical.

We compare also our results with the filtering enabled and disabled. Fig. 8 shows two cases where the current method solely using the optimizer with filtering and outlier elimination are both disabled. In this case the results are almost the same as our previous work. However as in Fig. 9, after enabling the filtering, the current method performs successfully both when the PSGC assumption is violated or holds. Finally we provide a video and supplementary technical details at [28].

VIII. Conclusion

In this paper, we present a new method for state estimation and local mapping inside cylindrical geometries, which we show to be superior to the methods proposed in literature. The current approach models the environment as a piecewise-smooth-generalized-cylinder . A constrained Unscented Kalman Filter tracks the segments and the gravity vector on a manifold using the information from an IMU and a 3D lidar. The raw point cloud data is processed by an optimizer running on a manifold constraining relative segment poses to preserve the consistency of the local map. We also use the Watson distribution to analyze the raw data and do outlier elimination on $\mathbb{S}^2$. We show our results on the data obtained during experiment in Center Hill Dam, TN to support our claims.

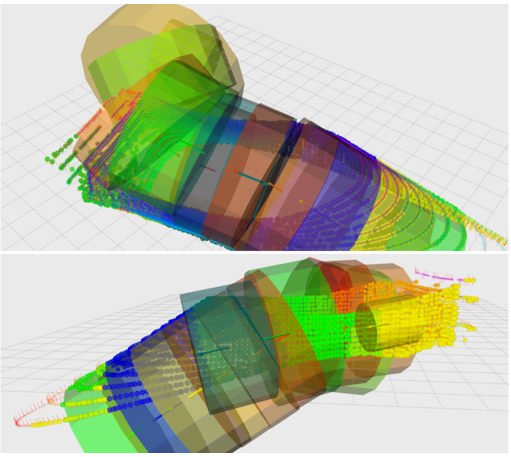

Fig. 8: These RViz snapshots show failure of the current approach while the robot is hovering at the same region as in Fig. 7. Here, we excluded Watson distributions from the optimizer and disabled outlier elimination to demonstrate their effects.

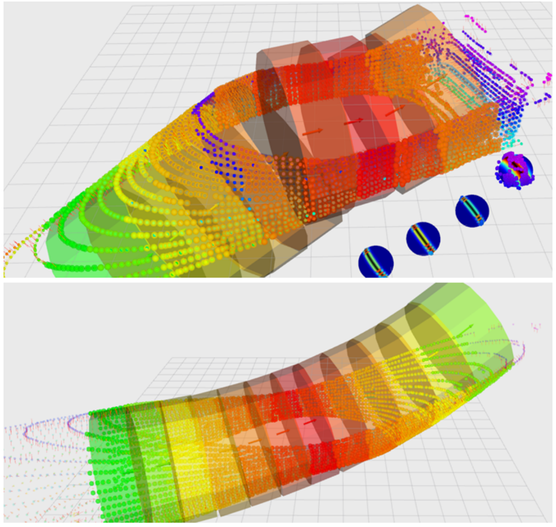

Fig. 9: Two cases showing that the current method performs successfully when the robot is hovering close to the gate where the previous method and this method with the filtering disabled fails. The second snapshot also show the map along the inclined section of the tunnel.

References

[1] T. Ozaslan, G. Loianno, J. Keller, C. J. Taylor, V. Kumar, J. M. Wozencraft, and T. Hood, “Autonomous Navigation and Mapping for Inspection of Penstocks and Tunnels With MAVs,” *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1740–1747, jul 2017.

[2] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, “The GRASP multiple micro-UAV testbed,” *IEEE Robot. Autom. Mag.*, vol. 17, no. 3, pp. 56–65, sep 2010.

[3] S. Bouabdallah, “Design and control of quadrotors with application to autonomous flying,” Ph.D. dissertation, LSA, EPFL, Lausanne, 2007.

[4] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, and D. Burschka, “Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue,” *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 46–56, sep 2012.

[5] S. Weiss, D. Scaramuzza, and R. Siegwart, “Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments,” *J. F. Robot.*, vol. 28, no. 6, pp. 854–874, nov 2011.

[6] “Using laser range data for 3D SLAM in outdoor environments,” in *Proc. 2006 IEEE Int. Conf. Robot. Autom. 2006. ICRA 2006.* IEEE, 2006, pp. 1556–1563.

[7] G. Loianno, C. Brunner, G. McGrath, and V. Kumar, “Estimation, Control, and Planning for Aggressive Flight With a Small Quadrotor With a Single Camera and IMU,” *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 404–411, apr 2017.

[8] S. Liu, M. Watterson, S. Tang, and V. Kumar, “High speed navigation for quadrotors with limited onboard sensing,” in *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2016-June. IEEE, may 2016, pp. 1484–1491.

[9] T. Özaslan, K. Mohta, J. Keller, Y. Mulgaonkar, C. J. Taylor, V. Kumar, J. M. Wozencraft, and T. Hood, “Towards fully autonomous visual inspection of dark featureless dam penstocks using MAVs,” in *2016 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, oct 2016, pp. 4998–5005.

[10] E. Eade and T. Drummond, “Scalable Monocular SLAM,” in *CVPR*, vol. 1. IEEE, 2006, pp. 469–476.

[11] J. Zhang and S. Singh, “Visual-lidar odometry and mapping: Low-drift, robust, and fast,” in *Robot. Autom. (ICRA), 2015 ....* IEEE, may 2015, pp. 2174–2181.

[12] S. Shaojie, “Autonomous Navigation in Complex Indoor and Outdoor Environments with Micro Aerial Vehicles,” Ph.D. dissertation, 2014.

[13] R. Brooks, “Solving the Find-Path Problem by Good Representation of Free Space,” *IEEE Trans. Syst., Man, Cybern.*, vol. 13, no. 3, pp. 190 – 197, 1983.

[14] R. Zlot and M. Bosse, “Efficient Large-Scale 3D Mobile Mapping and Surface Reconstruction of an Underground Mine.” Springer, Berlin, Heidelberg, 2014, pp. 479–493.

[15] R. Van der Merwe and E. Wan, “The square-root unscented Kalman filter for state and parameter-estimation,” in *2001 IEEE Int. Conf. Acoust. Speech, Signal Process. Proc. (Cat. No.01CH37221)*, vol. 6. IEEE, pp. 3461–3464.

[16] K. V. Mardia and P. E. Jupp, *Directional Statistics [Hardcover]*. J. Wiley, 1999.

[17] G. S. Chirikjian and A. B. Kyatkin, *Harmonic Analysis for Engineers and Applied Scientists: Updated and Expanded Edition*. Courier Dover Publications, 2016.

[18] T. Özaslan, S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, “Inspection of Penstocks and Featureless Tunnel-like Environments Using Micro UAVs,” in *Field Service Robotics*, vol. 105, jan 2015, pp. 123–136.

[19] A. Ortiz, F. Bonnin-Pascual, and E. Garcia-Fidalgo, “Vessel Inspection: A Micro-Aerial Vehicle-based Approach,” *J. Intell. Robot. Syst.*, vol. 76, no. 1, pp. 151–167, sep 2014.

[20] M. Burri, J. Nikolic, C. Hrzeler, G. Caprari, and R. Siegwart, “Aerial service robots for visual inspection of thermal power plant boiler systems,” *Appl. Robot. Power Ind. (CARPI), 2012 2nd Int. Conf.*, pp. 70–75, 2012.

[21] P. Gohl, M. Burri, S. Omari, J. Rehder, J. Nikolic, M. Achtelik, and R. Siegwart, “Towards autonomous mine inspection,” in *Proc. 3rd Int. Conf. Appl. Robot. Power Ind. CARPI 2014*, 2015.

[22] J. Zhang and S. Singh, “Low-drift and real-time lidar odometry and mapping,” *Auton. Robots*, vol. 41, no. 2, pp. 401–416, feb 2017.

[23] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, “Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera,” in *Proc. Int. Symp. Robot. Res.*, 2011, vol. 100, pp. 235–252.

[24] G. J. Agin and T. O. Binford, “Computer description of curved objects,” in *Proceedings of the 3rd international joint conference on Artificial intelligence*. Morgan Kaufmann Publishers Inc., 1973, pp. 629–640.

[25] R. B. Rusu, “Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments,” Ph.D. dissertation, Institut für Informatik der Technische Universität Münchne, nov 2010.

[26] E. Kraft, “A quaternion-based unscented Kalman filter for orientation tracking,” in *Proc. 6th Int. Conf. Inf. Fusion, FUSION 2003*, vol. 1, 2003, pp. 47–54.

[27] D. Simon, “Kalman filtering with state constraints: a survey of linear and nonlinear algorithms,” *IET Control Theory Appl.*, vol. 4, no. 8, pp. 1303–1318, aug 2010.

[28] T. Özaslan, G. Loianno, J. Keller, C. J. Taylor, and V. Kumar. (2018) Spatio-temporally smooth local mapping and state estimation inside generalized cylinders with micro aerial vehicles. https://fling.seas.upenn.edu/~ozaslan/wiki/doku.php?id=ral2018