2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Daejeon Convention Center
October 9-14, 2016, Daejeon, Korea

Towards Fully Autonomous Visual Inspection of Dark Featureless Dam Penstocks using MAVs

Tolga Özaslan*, Kartik Mohta*, James Keller*, Yash Mulgaonkar*, Camillo J. Taylor* and Vijay Kumar*
Jennifer M. Wozencraft† and Thomas Hood†

***Abstract*— In the last decade, multi-rotor Micro Aerial Vehicles (MAVs) have attracted great attention from robotics researchers. Offering affordable agility and maneuverability, multi-rotor aircrafts have become the most commonly used platforms for robotics applications. Amongst the most promising applications are inspection of power-lines, cell-towers, large and constrained infrastructures and precision agriculture. While GPS offers an easy solution for outdoor autonomy, using onboard sensors is the only solution for autonomy in constrained indoor environments. In this paper, we present our results on autonomous inspection of completely dark, featureless, symmetric dam penstocks using cameras and range sensors. We use a hex-rotor platform equipped with an IMU, four cameras and two lidars. One of the cameras tracks features on the walls using the on-board illumination to estimate the position along the tunnel axis unobservable to range sensors while all of the cameras are used for panoramic image construction. The two lidars estimate the remaining degrees of freedom (DOF). Outputs of the two estimators are fused using an Unscented Kalman Filter (UKF). A moderately trained operator defines waypoints using the Remote Control (RC). We demonstrate our results from Carters Dam, GA and Glen Canyon Dam, AZ which include panoramic images for cracks and rusty spot detection and 6-DOF estimation results with ground truth comparisons. To our knowledge ours is the only study that can autonomously inspect environments with no geometric cues and poor to no external illumination using MAVs.**

I. INTRODUCTION

There is extensive literature on control, motion planning and navigation, and state estimation of multi-rotor MAVs. [2], [3] designed low-level controllers for agile and aggressive maneuvering using motion tracking systems such as Vicon. New motion planning and navigation methods such as [4], [5] exploited the nonholonomic characteristics of these small robots. Finally, the major but difficult problem of Simultaneous Localization and Mapping (SLAM) is tackled by [6], [7], [8] using range and image sensors applied on MAVs.

Compared to ground robots, MAVs have additional DOFs which further complicate pose estimation and safe navigation problems. Despite this fact, due to their agility, maneuverability and simple design with affordable costs, MAVs are platforms cut out for real-life applications. Furthermore, as the theoretical foundations saturate, MAVs are becoming more common in military applications and also in civilian applications such as maintenance of power-lines, cell-towers and precision agriculture. [9], [10] motivate use of MAVs for precision agriculture and inspection of large structure without offering autonomous solutions. In their recent publication similar to ours, Hansen *et al.* [11] contributed to inspecting gas pipelines but using a wheeled robot.

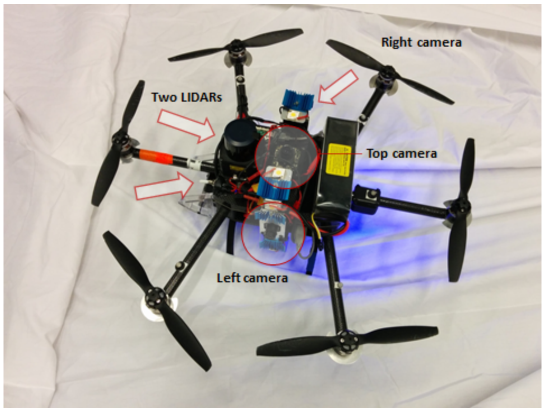


Fig. 1: The hex-rotor platform equipped with an Intel i7 computer, IMU, two Hokuyo UST20-LX lidars and four Bluefox XGA cameras. This design uses eight 10 W LEDs placed around the cameras to provide on-board illumination. The KHex [1] weighs 2.6kg and can fly about 8 minutes with a four cell 4500 mAh battery.

An important application field is inspection of large infrastructures such as dam locks, gates and penstocks. These structures are exposed to huge, oscillating loads for long periods due to which continuous maintenance is vital. If crack formation and rusty spots are not treated timely, catastrophic consequences are inevitable such as collapse of the dam or water discharge tunnels. Current inspection and maintenance practices are carried manually by dam workers either by swinging from the reservoir-side gates, tethering carts or building scaffolds. In either case, the workers manually inspect the tunnel walls for cracks and rusty spots.

In this study, we propose a complete system design to collect detailed imagery from inside dam penstocks for inspection and maintenance purposes using fully autonomous multi-rotor MAVs. We propose solutions to complete pose estimation in settings with no geometric features and in complete darkness. The system is designed for moderately skilled operators that can fly using a simple joystick in pitch darkness without line of sight to collect data for inspection. The operator can command position or velocity

-This work was supported by the MAST Collaborative Technology Alliance - Contract No. W911NF-08-2-0004

* GRASP Lab., University of Pennsylvania, 19104, Philadelphia, PA. email : {ozaslan}{kmohta}{jkeller}{yashm}{cjtaylor}{kumar} @ seas.upenn.edu

† The U.S. Army Corps of Engineers (USACE)

978-1-5090-3762-9/16/$31.00 ©2016 IEEE

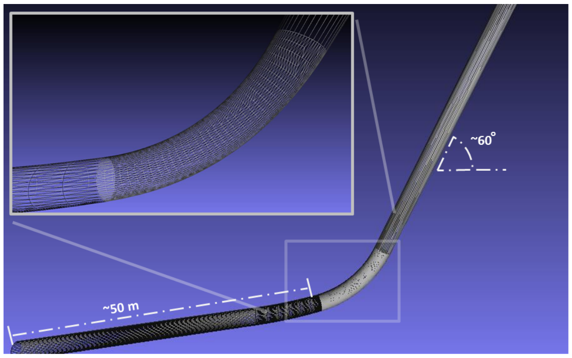


Fig. 2: A CAD model of Glen Canyon Dam penstock with close-up view of the bending section. The diameter of the tunnel varies from 4.5m to 5.5m. The horizontal section is more than 50m. long. The inclination climbs up longer than 100m with a slope of 60 degrees. The horizontal part makes a gentle left turn just before the inclination starts due to the narrow river bed. We present estimation results and panoramic images from the experiments at this site and at Carters Dam.

in all directions with soft constraints to prevent pilot error. The main motivation is to reduce the risk of accidents, the man power requirement and the cost of inspection.

By their design, MAVs are highly unstable and nonlinear platforms. Furthermore, penstock interiors are usually wet, partially covered with mud which adversely affects the sensor performance hence pose estimation [12]. Lidars especially suffer from wet tunnel walls and water puddles on the floor. Also, textureless tunnel walls prohibit vision-based estimators under weak illumination. Consequently, vision-only [7] or range-only [13], [14] methods do not offer a solution to the challenges of this problem. In order to attain complete autonomous control, eliminate possible sensor related failures and reduce the training required for operating the platform, penstock inspection MAVs have to be equipped with redundant sensors (Figure 1).

Using a similar sensor package to ours, [15] proposes the use of range sensors in combination with cameras for solving the data association problem and estimate incremental odometry. However they assume that the image is texture-rich which does not apply to our case due to low illumination. In another paper Lui [16] uses a sensor backpack system consisting of IMUs and lidars for indoor localization. While their method offers a solution for localization and mapping in corridors of length smaller than the range of the laser scanners, symmetric and very long penstocks will fail their algorithm.

The two experiment sites that we visited to evaluate our work exhibit similar characteristics with a few minor differences. Penstocks of Carters Dam, GA slope upwards at a single location whereas Glen Canyon Dam, AZ bend laterally due to the narrow river bed (Figure 2). The inclination of the tunnel ranges from gentle slopes to close-to-vertical case as is the case at Glen Canyon Dam. For this reason, we stress that ground vehicles such as proposed in [11] cannot attain sufficient traction on the slippery walls along the inclination.

We estimate the lateral and vertical coordinates of the robot using the two lidars and the map of the penstock. Using the IMU and the lidars we can also estimate the 3-DOF orientation [12], [6]. However, the position of the robot along the axis of the tunnel is unobservable to range sensors except at the start of the inclination. Lack of geometric cues prohibits the use of range sensors in localizing the robot along the tunnel axis. We overcome this problem by using the cameras to track features on the walls. The optical flow field is used to incrementally update the position estimate along the tunnel axis. Since the tunnel is not externally illuminated, on-board illumination plays an important role on the visual odometry (VO) accuracy. Partial estimates from the VO and the range-based estimator are fused using a UKF to obtain 6-DOF pose estimate. Finally, the metal tunnel structure completely eliminates the use of GPS as an alternative sensor. To our knowledge, except our previous study [12], this work is the only fully autonomous inspection application in such completely dark, featureless challenging environments.

II. System Description

A. Notation

We define the world frame with the standard reference triad $\{\hat{x}^{\mathcal{W}}, \hat{y}^{\mathcal{W}}, \hat{z}^{\mathcal{W}}\}$ where $\hat{x}^{\mathcal{W}}$ is aligned with the axis of the tunnel. It is defined to be pointing towards the inclined section. $\hat{z}^{\mathcal{W}}$ is aligned with the gravity vector pointing in the opposite direction which completes the definition of the world reference frame. The body frame is attached to the geometric center of the robot body which we assume to be coincident with the mass center. It is denoted as $\{\hat{x}^{\mathcal{B}}, \hat{y}^{\mathcal{B}}, \hat{z}^{\mathcal{B}}\}$. $\hat{x}^{\mathcal{B}}$ is the forward direction of the robot pointing in the same direction with the lidars. Finally $\hat{z}^{\mathcal{B}}$ is aligned with $\hat{z}^{\mathcal{W}}$ at hover state.

Transformations between frames are carried with the rotation matrix $\mathbf{R} \in SO(3)$. We use the ${}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}$ notation for transformations from the $\mathcal{W}$orld frame to the $\mathcal{B}$ody frame. In the rest of the paper, scripts such as $\mathcal{W}$ or $\mathcal{B}$ are used to denote the frame in which a vector is *represented*.

In our formulations, we use Euler angles to represent rotation with the ZXY order. Roll, pitch, yaw angles are denoted as ϕ, θ and ψ respectively. The rotation matrix ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}}$ is successive application of elementary rotations around the body frame axis such that ${}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}} = \mathbf{R}_{\phi}\mathbf{R}_{\theta}\mathbf{R}_{\psi}$. Finally, we define the 6-DOF robot state vector as $\mathbf{r} = [x, y, z, \phi, \theta, \psi]^{\mathcal{W}}$. $\mathbf{r}_i$ where $i \in \{x, y, z, \phi, \theta, \psi\}$ is used to refer to the corresponding state coordinates.

B. The KHex Platform

In this work we use the KHex platform designed by KMel Robotics [1]. The KHex can fly approximately 8 minutes with a four-cells 4500 mAh battery with the total payload of 2.6 kilograms. Figure 1 shows the KHex platform equipped with an Intel i7 board, two Hokuyo UST20-LX lidars and four XGA resolution BlueFox cameras. KHex is redundantly equipped with sensors in order to reduce the

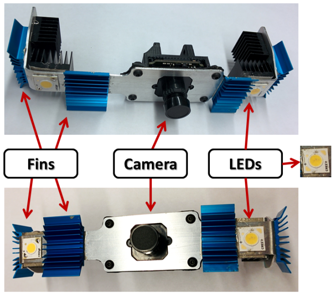


Fig. 3: The Camera-LED setup. We use eight Cree power-LEDs to provide on-board illumination for VO. Each LED is 10 W and has 5000 K color temperature.

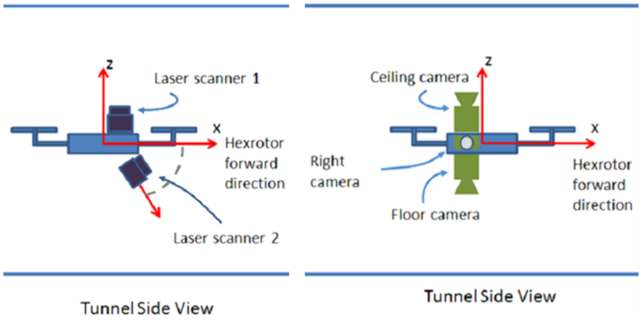


Fig. 4: This figure shows schematics for the sensor placements. Both figures are side views. The left figure shows one of the lidars tilted slightly downwards to measure the elevation while the other scanning in the $\hat{x}^{\mathcal{B}} - \hat{y}^{\mathcal{B}}$ plane. The right image shows placements of the four cameras. One of the cameras is used to track salient features on the wall. At the same time, all cameras are used to grab images to generate panoramic images.

risk of sensor related failures and collect detailed imagery from inside the penstock. In our previous work [12], we retrofitted the robot with a mirror setup to redirect a subset of the lidar rays to the floor and the ceiling to measure the elevation. However puddles, continuous water drainage and wet surfaces often cause failure of height measurements. This problem is solved by dedicating a lidar tilted downwards to measure the elevation.

We use KMel's proprietary on-board attitude estimator. After gravity correction, roll (θ), and pitch (ϕ), estimates exhibit low drift and noise so we directly feed these to the UKF prediction step at 100 Hz (Figure 5). The two lidars send scan data through two separate Ethernet ports at 40 Hz with a span of $\leq$270 degrees. Landing gears and booms partially occlude the view of the bottom lidar. Only one of the cameras is used for pose estimation due to the bandwidth constrain of the USB 2 bus. The onboard Intel i7 NUC board can only transfer 24 FPS XGA resolution frames in total. Since on-board processing is not supported on the BlueFox cameras, camera driver resizes the raw images to VGA size on the CPU for faster image processing. We sacrificed use of multiple cameras for higher frame rates since the image quality at lower frame rates is bad for optical flow calculation. The schematic showing the sensor placement and the preferred robot orientation during flight is shown in Figure 4.

In order to obtain sufficiently bright and textured images for both inspection and VO, we equipped the robot with power LEDs (Figure 3). This removes the requirement of external illumination and reduces the labor requirement significantly.

We define the transformation of sensor data to the body frame with a rotation matrix and a translation vector given as ${}^{\mathcal{B}}\mathbf{R}_{\mathcal{F}}$ and $\mathbf{t}^{\mathcal{F}}$ where $\mathcal{F}$ is the corresponding sensor frame. For the top and bottom lidars we use $\mathcal{L}_t$ and $\mathcal{L}_b$; and $\mathcal{C}_r$, $\mathcal{C}_t$, $\mathcal{C}_l$ and $\mathcal{C}_b$ for the right, top, left and bottom cameras. We use the notation $\{j, \mathcal{F}\}$ to refer to the j^{th} image feature point or laser beam from the sensor with the frame label $\mathcal{F}$.

C. Environment Assumptions

The method we propose relies on the map, $\mathcal{M}$, of the tunnel. $\mathcal{M}$ is a 3D occupancy grid approximation of the tunnel with 5 cm resolution. We assume that the cross-section of the tunnel is convex. This is basically in order to avoid local minima in the range-based iterative closest point (ICP [17]) algorithm. The tunnel has a single axis and does not branch off as in the case of city sewer systems.

D. Controller & Navigation

We use the PD controller explained in [2] as is. This method linearizes the equations of motion of a quadrotor MAV at its hover state. Because we are only interested in flights at slow speeds, close to the hover configuration, the linear controller proposed in [2] fits well to our case.

Since the 6-DOF pose estimator handles low level control, the system requires only moderate operator training. The operator defines waypoints and the speed using an RC interface. We visualize the robot state and the waypoints in the ROS visualization software, RViz, to visually assist the operator.

III. METHODOLOGY

A. Software Architecture

Robot Operating System (ROS) by Willow Garage is a pseudo-operating system which implements tasks as separate processes with a central mechanism for inter-process message exchange. Each ROS process handles certain tasks such as data acquisition, pose estimation or decision making. In Figure 5 we show the software architecture with each box corresponding to a process in a data flow diagram.

The inputs to the system are the map, $\mathcal{M}$, IMU data, frames from the right camera and range measurements from the two lidars. We assume that the engineering drawings of the penstock are given and converted to occupancy grid map, $\mathcal{M}$. These experiments choose $\mathcal{M}$ with 5 cm resolution. The UKF node outputs 6-DOF pose estimates to be fed to the PD controller. The operator gives waypoints to the trajectory generator using an RC. Finally, the controller generates

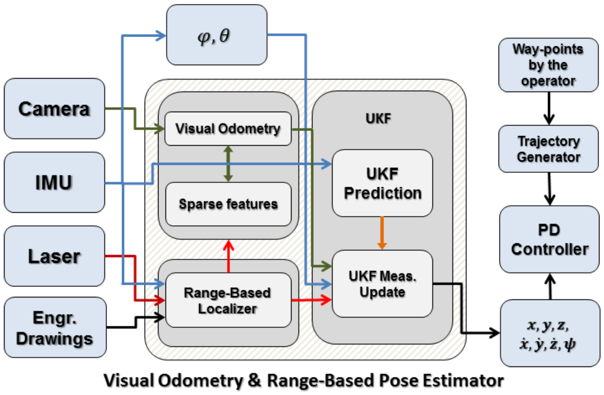


Fig. 5: This figure shows processes in a data flow diagram. The inputs to the system are the IMU, lidar and camera measurements, and the map of the tunnel. Partial pose estimates from range-based localizer and visual odometry are fused in the UKF node. The operator gives waypoints using an RC to the trajectory generator output of which is fed to the PD controller.

low-level controller commands in accordance with the pose estimate and the trajectory.

One of the four cameras is used to track salient features on the walls and incrementally update the robot position along the tunnel axis. The fusion of the range-based estimator with VO estimates the 6-DOF pose of the robot. All four cameras are used to grab images from the four sides of the robot to form 360 degrees image panoramas. These images can later be used to locate cracks and rusty spots by the maintenance engineers.

The frequency of the estimator is determined with sensor rates. The prediction step of UKF runs at a rate of 100 Hz while the range-based measurement update runs at 40 Hz. Optical flow based position increments are integrated to UKF at 24 Hz.

B. Range-Based Estimator

As shown in Figure 4 one of the lidars is tilted downwards by 60 degrees. The relative pose of each lidar with respect to $\mathcal{B}$ is represented with rotation-translation pairs ${}^{\mathcal{B}}\mathbf{R}_{\mathcal{L}_t} - \mathbf{t}^{\mathcal{B}}_{\mathcal{L}_t}$ and ${}^{\mathcal{B}}\mathbf{R}_{\mathcal{L}_b} - \mathbf{t}^{\mathcal{B}}_{\mathcal{L}_b}$ for the top and the bottom lidars respectively. The ICP scan matcher compares the raw data from both of the lidars to $\mathcal{M}$ and iteratively reduces the error to refine yaw, vertical and lateral position estimates. Lidar data is preprocessed to exclude indefinite measurements and rays with range larger than r_{th}. The latter filter reduces the effect of noisy range readings in yaw $(\mathbf{r}_\psi)$ and lateral position $(\mathbf{r}_y)$ estimates in the iterative least squares formulation that is explained below. In order to suppress noise, we apply median filtering in the range space. In addition, the scanner data are downsampled at the ray tips to exclude uninformative repetitive data and save CPU time.

This work chooses the median filter window to be 5, range threshold $r_{th} = 6m$ and the scanner ray tip downsampling resolution as $3cm$.

The ICP algorithm solves the data association problem by projecting each laser beam onto the grid map $\mathcal{M}$ and assigning the first hit voxel center to the corresponding beam. In other words, we define the closest map point to be the voxel center closest to the beam origin and intersecting with the lidar beam represented as

$$\mathcal{V}_{j,\mathcal{L}_i} = \pi(\mathbf{r}_t, \alpha_{j,\mathcal{L}_i}; \mathcal{M}) \tag{1}$$

$$\mathcal{V}^{*}_{j,\mathcal{L}_i} = \underset{\mathcal{V}_k \in \mathcal{V}_{j,\mathcal{L}_i}}{\operatorname{argmin}} \left(\| \mathbf{r}_t + {}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}} \mathbf{t}^{\mathcal{B}}_{\mathcal{L}_i} - \mathcal{V}_k \|_2 \right) \tag{2}$$

where $\mathbf{r}_t$ is the translational component of the state vector $\mathbf{r}$ as defined in $\mathcal{W}$, $i \in \{t, b\}$, $\alpha_{j,\mathcal{L}_i}$ is the angle of the ray $\{j, \mathcal{L}_i\}$ in the scanner's sweeping plane, $\mathcal{M}$ is the 3D occupancy grid approximation of the point cloud and $\| \bullet \|_2$ is the L_2 norm. The projection function π casts a ray along $\{j, \mathcal{L}_i\}$'s direction emanating from the origin of the corresponding lidar frame. This ray is denoted as $\rho_{j,\mathcal{L}_i}$ and the origin of lidar i is written as $\mathbf{r}_t + {}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}} \mathbf{t}^{\mathcal{B}}_{\mathcal{L}_i}$. The list $\mathcal{V}_{j,\mathcal{L}_i}$ consists of all the voxel centers that the projection function π intersects the corresponding laser beam. Finally, the beam $\{j, \mathcal{L}_i\}$ is associated with the voxel centered at $\mathcal{V}^{*}_{j,\mathcal{L}_i}$. Figure 6 depicts these parameters and vectors in a schematic.

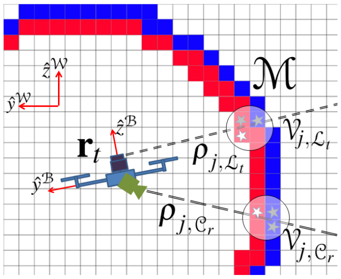


Fig. 6: This schematic depicts the parameters and vectors explained in Equations 1-2 and 13-14. The bright stars represent $\mathcal{V}^{*}_{j,\mathcal{L}_i}$ and $\mathcal{V}^{\tau*}_{j,\mathcal{C}_r}$ in these equations.

Having the data association defined, we formulate the problem as an iterative weighted least squares problem without regularization written as $WAx = Wb$ where $x = [\, \cos(\Delta \mathbf{r}_\psi),\ \sin(\Delta \mathbf{r}_\psi),\ \Delta \mathbf{r}_y \,]^T$, W is the diagonal weight matrix. We assume that the on-board attitude estimator gives accurate roll and pitch so it is excluded from the ICP. In fact, due to the symmetric tunnel geometry, roll cannot be measured except with an IMU or magnetometer. A is a $N \times 3$ matrix and b is an N vector where N is the total number of data points from both of the lidars. The vector corresponding to the ray $\{j, \mathcal{L}_i\}$ with the measurement $r_{j,\mathcal{L}_i}$ is $\rho^{\mathcal{L}_i}_{j,\mathcal{L}_i} = r_{j,\mathcal{L}_i} * [\, \cos(\alpha_{j,\mathcal{L}_i}),\ \sin(\alpha_{j,\mathcal{L}_i}),\ 0 \,]^T$. We define A and b as

$$\rho^{*\mathcal{W}}_{j,\mathcal{L}_i} := {}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}} (\rho^{\mathcal{B}}_{j,\mathcal{L}_i} + \mathbf{t}^{\mathcal{B}}_{\mathcal{L}_i}) \tag{3}$$

$$A_n = [\, \rho^{*\mathcal{W}}_{j,\mathcal{L}_i,y},\ -\rho^{*\mathcal{W}}_{j,\mathcal{L}_i,x},\ 1 \,] \tag{4}$$

$$b_n = \mathcal{V}^{*}_{j,\mathcal{L}_i,y} - \mathbf{r}_y \tag{5}$$

$$W_{n,n} = w_n \tag{6}$$

where A_n is the n^{th} row, w_n is the n^{th} diagonal element of the weight matrix W and each $\{j, \mathcal{L}_i\}$ tuple matches to unique

n. Each data point is assigned a weight w_n as a function of the alignment error ε_n. These parameters are defined as

$$\varepsilon_n = \rho^{*\mathcal{W}}_{j,\mathcal{L}_i} + \mathbf{r}_t - \mathcal{V}^{*}_{j,\mathcal{L}_i} \tag{7}$$

$$w_n = e^{-\|\varepsilon_n\|_2^{\gamma}} \tag{8}$$

where we choose $\gamma = 3$. This way, correspondences with large initial residuals are penalized more and lose their contribution to the least squares solution. Finally the partial solution becomes

$$x = (A^T W A)^{-1} A^T W b \tag{9}$$

$$\Delta\mathbf{r}_y = x_3 \tag{10}$$

$$\Delta\mathbf{r}_\psi = atan2(x_2^*, x_1^*). \tag{11}$$

Since x_2 and x_1 might not be valid cosine and sine values, we clamp them to the $[-1, 1]$ inclusive range which is denoted by $\bullet^*$.

The above formulation solves only for $\mathbf{r}_y$ and $\mathbf{r}_\psi$ simultaneously because of their strong coupling. Whereas, due to the geometry of $\mathcal{M}$ and the way we formulate the least squares solution, $\mathbf{r}_z$ and $\mathbf{r}_\psi$ do not correlate significantly. $\mathbf{r}_z$ is mostly a function of the ranges from the bottom lidar. For each iteration of the pose refinement we define the $\mathbf{r}_z$ update as

$$\Delta\mathbf{r}_z = -\frac{1}{N}\sum_{n=1}^{N} \varepsilon_{n,z} * \exp\left(\frac{|\rho^{*\mathcal{W}}_{j,\mathcal{L}_i,z}|}{\|\rho^{*\mathcal{W}}_{j,\mathcal{L}_i}\|_2} - 1\right). \tag{12}$$

As we noted earlier, there is an implicit one-to-one mapping between n and $\{i, \mathcal{L}_i\}$.

It should have been noticed that the range-based pose estimator does not update $\mathbf{r}_x$ since robot position along the tunnel axis is not observable to range sensors. This is because of the geometry of the tunnel and the maximum lidar range. When the robot hovers close to the bend, top lidar can take measurements from the inclination. This gives an estimate for $\mathbf{r}_x$ for a very small subspace of the tunnel, hence we discard it. We start the iteration from the last valid pose estimate, cast rays using the $\pi(\bullet)$ function, estimate $\Delta\mathbf{r}_{y,z,\psi}$ and update the robot pose until it stabilizes.

C. Visual Odometry (VO)

The shape of the environment does not permit use of range sensors for position estimate along the tunnel axis, that is $\mathbf{r}_x$. In order to solve this problem, we propose using cameras to track salient features on the tunnel walls. $\mathbf{r}_x$ can be incrementally updated over time using the optical flow field given $\mathbf{r}_{y,z,\phi,\rho,\psi}$ estimates and the map $\mathcal{M}$.

The VO method we design is heavily dependent on the quality of the on-board illumination since we don't have any external light sources in the completely dark penstock. As shown in Figure 7 the camera sees very pale images with very little texture. None of the feature extraction algorithms among FAST, Harris and Shi-Tomasi [18][19] could find any features or they fail persistence. Furthermore, the nonuniform lighting pattern and lens glare generate an artificial intensity gradient which adversely affects both feature extraction and tracking performance. We overcome these problems by applying a set of image filters as shown in Figure 7 to amplify the weak texture. In order to enhance contrast of the raw image we apply histogram equalization followed by Gaussian blur of 5 pixels kernel size. The blurring operation smooths the noise formed after histogram equalization. Then we apply the built-in OpenCV adaptive threshold on the blurred image which gives a black & white image. An interesting property of this image is, the white pixels adhere to each other and form small blobs which persist for a couple of frames. These blobs define small hills and the black regions form valleys around that blob which Kanade-Lucas-Tomasi (KLT) tracker [20] can easily track. Despite the short blob adhesion duration, optical flow field density is preserved due to the newly formed blobs. In some cases, only a small subset of the image exhibit sufficient quality even after applying the above filter pipeline. In these cases, we set the region of interest to the brighter regions.

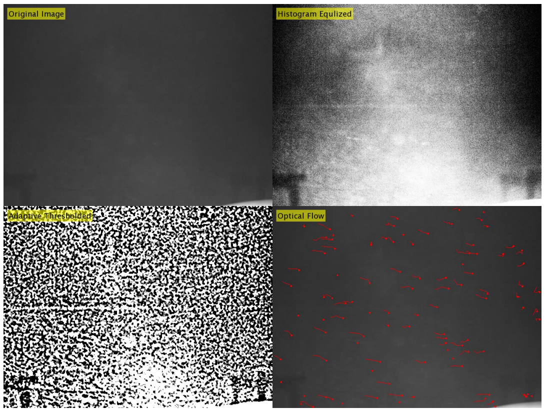


Fig. 7: This figure shows the output of the image processing pipeline at each step and the resultant optical flow field from our visit to Carters Dam, GA. At the top-left is the raw image. This is a pale image with almost no significant texture. In order to amplify the texture gradient, we used histogram equalization as shown on the top-right. Next, an adaptive threshold is applied to get a black-white image as in the bottom-left. FAST features and KLT tracker are used to extract and track features on this image. The corresponding video can be found at http://mrsl.grasp.upenn.edu/tolga/iros2016/

The built-in feature extraction functions of OpenCV return features with higher responses first. In some cases, this results in accumulation of features around small patches with significant texture. However, for better estimator performance, uniform feature distribution over the whole image plane is preferable. To achieve uniform feature distribution, we divide the image into a grid of 3 rows and 4 columns, and force equality of the number of features in each grid cell.

We eliminate flow vectors which are longer than a threshold. The typical value is between 30 and 100 pixels depending on the flight speed. These cases correspond to KLT tracker failures due to either rapid camera motion or a feature point at plain, low texture image region. Another criterion we implement for discarding bad features is to only use features with lifetime ≥ 3 frames.

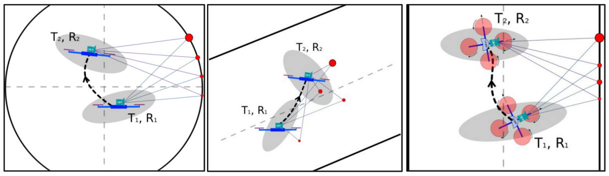


Fig. 8: These figures show snapshots from three perspectives of the two-step VO displacement estimation. From left to right are back, right and top views of the tunnel along the inclined section. The robot poses are denoted as rotation-translation pairs (R_i, T_i) $i \in \{1,2\}$. Grey shades represent the pose uncertainty. The red dots are the back-projected feature points. The range-based localizer can only provide lateral and vertical position estimates. The missing DOF is estimated using VO.

We estimate the inter-frame displacement along the tunnel axis by analyzing the back-projection of the tracked features onto $\mathcal{M}$. This operation is similar to the ray-casting function $\pi(\bullet)$ defined for the range-based localizer. Features in the 1^{st} and 2^{nd} frames are referenced by $\{j, \mathcal{C}_i\}^1$ and $\{j, \mathcal{C}_i\}^2$ respectively. The 2^{nd} frame is the most recently grabbed frame and the 1^{st} frame is the previous one. Note that, since lidars and cameras are not working with the same rate, we do not use a common time index. j is the tracked feature index in the camera with frame $\mathcal{C}_i$ where $i \in \{r,t,l,b\}$. In fact, due to low USB 2 bandwidth, we use only the right camera. The 3D projected point on $\mathcal{M}$ for each feature is found as

$$\mathcal{V}^{\tau}_{j,\mathcal{C}_r} = \pi(\mathbf{r}^{\tau}_t, \{j, \mathcal{C}_r\}; \mathcal{M}) \tag{13}$$

$$\mathcal{V}^{\tau*}_{j,\mathcal{C}_r} = \underset{\mathcal{V}_k \in \mathcal{V}_{j,\mathcal{C}_r}}{\text{argmin}} \left(\|\mathbf{r}^{\tau}_t + {}^{\mathcal{W}}\mathbf{R}^{\tau}_{\mathcal{B}} \mathbf{t}^{\mathcal{B}}_{\mathcal{C}_r} - \mathcal{V}_k\|_2\right) \tag{14}$$

where $\tau \in \{1,2\}$ denotes the frame index. The behavior of the $\pi(\bullet)$ function and the definition of the $\mathcal{V}$ lists are the same as explained previously. Figure 8 shows an instance of the robot from three perspectives with the back-projected rays. As shown in the system architecture (Figure 5), the initial pose of VO is input from the range-based localizer. In the above equations these are denoted by $\mathbf{r}^{\tau}_t$. $\{j, \mathcal{C}_r\}$ defines a vector $\rho_{j,\mathcal{C}_r}$ written in the $\mathcal{C}_r$ frame as

$$\rho^{\mathcal{C}_r}_{j,\mathcal{C}_r} := \mathbf{K}^{-1} \begin{bmatrix} \{j, \mathcal{C}_r\}_x \\ \{j, \mathcal{C}_r\}_y \\ 1 \end{bmatrix} \tag{15}$$

which is also depicted in Figure 6. Here $\mathbf{K}$ is the camera calibration matrix and $\{j, \mathcal{C}_r\}_x - \{j, \mathcal{C}_r\}_y$ are the $x-y$ image coordinates of the corresponding image feature. We use $\hat{\rho}^{\mathcal{C}_r}_{j,\mathcal{C}_r}$ for the normalized version of this vector.

Since $\mathcal{M}$ is a finite resolution occupancy grid, the back-projected points always correspond to voxel centers. However, discretization loses the precision of the continuous robot pose that definitely affects the VO precision adversely. We modify $\mathcal{V}^{\tau*}_{j,\mathcal{C}_r}$ as follows

$$\mathcal{V}^{\tau**}_{j,\mathcal{C}_r} = \|\mathcal{V}^{\tau*}_{j,\mathcal{C}_r} - \mathbf{r}^{\tau}_t\|_2 * \hat{\rho}^{\mathcal{C}_r}_{j,\mathcal{C}_r} + \mathbf{r}^{\tau}_t. \tag{16}$$

Since the voxel size is small (5cm) and the tunnel surface is smooth, we assume the norm of the back-projected ray (the term the L_2 operator is applied) is accurate. We then correct the orientation of this ray using the exact feature coordinates encoded by $\hat{\rho}^{\mathcal{C}_r}_{j,\mathcal{C}_r}$.

The motion unobservable to the range-based localizer is along the tunnel axis. We denote this direction as $\hat{t}$ which is equivalent to $\hat{x}^{\mathcal{W}}$ along the horizontal section and $\hat{x}^{\mathcal{W}}\cos(\beta) + \hat{z}^{\mathcal{W}}\sin(\beta)$ along the inclined section β being the inclination angle. The projection norm of the vector difference $\mathcal{V}^{2**}_{j,\mathcal{C}_r} - \mathcal{V}^{1**}_{j,\mathcal{C}_r}$ onto $\hat{t}$ gives an estimate of the corresponding displacement.

A histogram h of the projections $\left(\mathcal{V}^{2**}_{j,\mathcal{C}_r} - \mathcal{V}^{1**}_{j,\mathcal{C}_r}\right) \cdot \hat{t}$ is generated with 5 bins. Features in the highest percentage bin are regarded as inlier. The mean of the projections in the highest ranking bin is assigned to be $\Delta\mathbf{r}_{\hat{t}}$.

D. Measurement Model Uncertainty

In calculating the uncertainty of the range-based estimator, we use the method proposed by Censi [21]. [21], details of which we leave to the original work, assumes a single polygonal environment and uses an information theoretic approach to estimate the information carried by the each scan. However, as shown in Figure 9 assuming a single polygonal environment does not work for some cases. For this reason, we first segment the scan into clusters according to the discrepancy between consecutive readings and represent the environment with multiple polygons. The Fisher Information Matrix (FIM) is estimated for each polygon which are then summed to give a more accurate covariance estimate. The symmetric FIM is defined as

$$\mathcal{I}(\mathcal{L}_i)^{\mathcal{L}_i} = \begin{bmatrix} \mathcal{I}_{x,x} & \mathcal{I}_{x,y} & \mathcal{I}_{x,\psi} \\ \mathcal{I}_{y,x} & \mathcal{I}_{y,y} & \mathcal{I}_{y,\psi} \\ \mathcal{I}_{\psi,x} & \mathcal{I}_{\psi,y} & \mathcal{I}_{\psi,\psi} \end{bmatrix}. \tag{17}$$

written in $\mathcal{L}_i$ frame for the i^{th} lidar. Information from multiple lidars is merged by projecting each FIM onto the

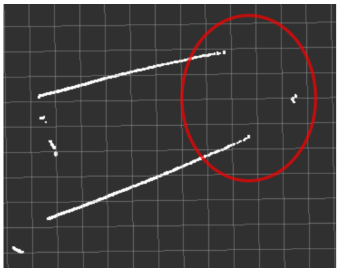


Fig. 9: Sample laser scanner contour from inside the Carters Dam penstock. The two straight segments are from the walls of the tunnel. Since the lidar cannot see the end of the tunnel, contour interrupts (circled). The FIM is estimated as proposed in [21] separately for each segment and summed to give the measurement covariance.

body frame, $\mathcal{B}$, shown as $\mathcal{I}(\mathcal{L}_i)^{\mathcal{B}}$. The upper 2×2 block, denoted as $\mathcal{I}(\mathcal{L}_i)^{\mathcal{L}_i}_{x,y}$, can simply be projected by multiplying with proper rotation matrices. We expand this block to 3×3 by appending the information for the z axis which is 0. The transformation reads

$$\mathcal{I}(\mathcal{L}_i)^{\mathcal{W}}_{x,y,z} = {}^{\mathcal{W}}\mathbf{R}_{\mathcal{L}_i} \begin{bmatrix} \mathcal{I}(\mathcal{L}_i)^{\mathcal{L}_i}_{x,y} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} {}^{\mathcal{L}_i}\mathbf{R}_{\mathcal{W}}. \qquad (18)$$

Angular uncertainties however are not easy to project finding a solution for which is beyond the concerns of this work. Therefore we approximate $\mathcal{I}^{\mathcal{W}}_{\psi,\psi}$ as

$$\mathcal{I}(\mathcal{L}_i)^{\mathcal{W}}_{\psi,\psi} = |{}^{\mathcal{W}}\mathbf{R}(3,3)_{\mathcal{L}_i}| * \mathcal{I}(\mathcal{L}_i)^{\mathcal{L}_i}_{\psi,\psi} \qquad (19)$$

where ${}^{\mathcal{W}}\mathbf{R}(3,3)_{\mathcal{L}_i}$ is a measure of how much the lidar is tilted. The off-diagonal elements for the ψ information are ignored which does not affect the performance of the estimator significantly.

The uncertainty due to the range-based estimator is then written as

$$\Sigma(\mathcal{L})_{x,y,z} = \left[\mathcal{I}(\mathcal{L}_1)^{\mathcal{W}}_{x,y,z} + \mathcal{I}(\mathcal{L}_2)^{\mathcal{W}}_{x,y,z}\right]^{-1} \qquad (20)$$

$$\Sigma(\mathcal{L})_{\psi} = \left[\mathcal{I}(\mathcal{L}_1)^{\mathcal{W}}_{\psi,\psi} + \mathcal{I}(\mathcal{L}_2)^{\mathcal{W}}_{\psi,\psi}\right]^{-1}. \qquad (21)$$

IV. EXPERIMENTS

A. *6-DOF Pose Estimation*

The experiment environments are penstocks of Carters Dam, GA and Glen Canyon Dam, AZ. Results we present here include estimation results with ground truth comparisons when possible (Figure 11). At Glen Canyon Dam, we marked the walls with spray paint with 2 meters separation visible from the camera which we use as the ground truth source. We collected this dataset while the robot was flying along the horizontal section as can be seen in Figure 10. The Carters Dam dataset was collected along the inclination which was hard and dangerous to climb. For this reason, we can rate the odometry success by observing scratches on the walls.

Figure 13 shows the Carters Dam results. We manually detect loop closure by following significant scratches at the start and end of the flight. The drift for the ≈ 40 m. flight is less than 1 m which corresponds to $< 2.5\%$ error.

Figure 11 shows the results for VO in Glen Canyon Dam. By manually inspecting the markers painted on the walls, we record ground truth displacements and compare with our estimation results. The time axis refers to instances the painted markers are seen at the center of the right camera.

B. *Panoramic Image Generation*

Figure 12 shows a panoramic image and textured map reconstruction from Glen Canyon Dam. On the left is the 360 degree panoramic image generated by back-projecting the pixels from the four cameras onto $\mathcal{M}$ and then projecting onto a hypothetical omni-directional cylindrical camera. The images on the right show the textured reconstruction plotted inside the RViz environment.

Videos related to the experiments can be found at http://mrsl.grasp.upenn.edu/tolga/iros2016/

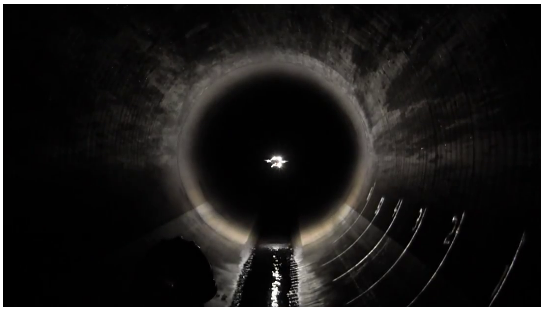
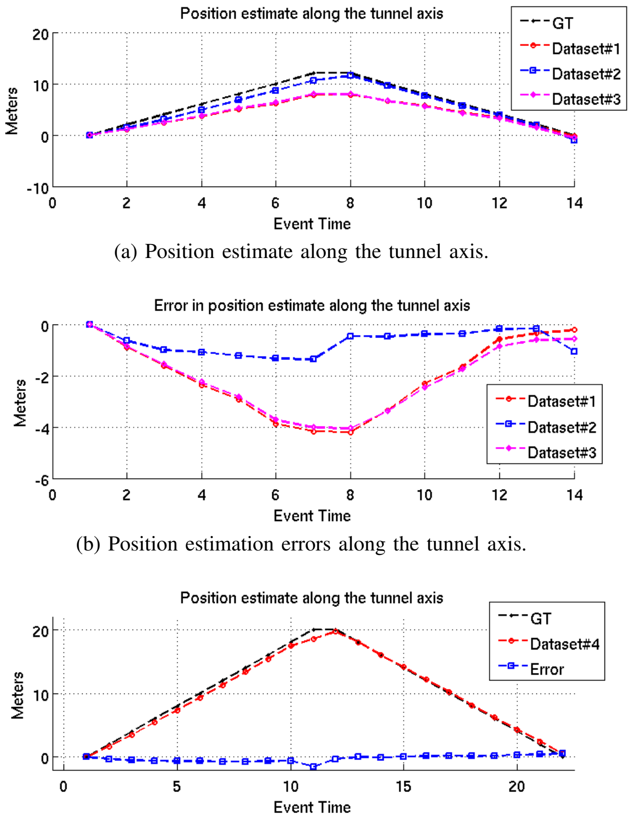


Fig. 10: A snapshot from the experiments inside the Glen Canyon Dam penstock. The robot is flying fully autonomously using on-board illumination. Also in Figure 12 we show the local 3D reconstruction and the 360 degree panoramic image generated using the images from the four cameras.



(a) Position estimate along the tunnel axis.

(b) Position estimation errors along the tunnel axis.

(c) Position estimate and error along the tunnel axis.

Fig. 11: These plots compare the VO results with ground truth data for datasets collected in Glen Canyon Dam penstock. The x-axis counts the instances that the markers on the walls are seen at the center of the right camera.

V. CONCLUSION & FUTURE WORK

The focus of this work is robust robot pose estimation to attain autonomy in dark, featureless, symmetric and GPS-denied environments like penstocks. This work demonstrates our methods and results with ground truth comparisons for autonomous inspection in Carters Dam, GA and Glen

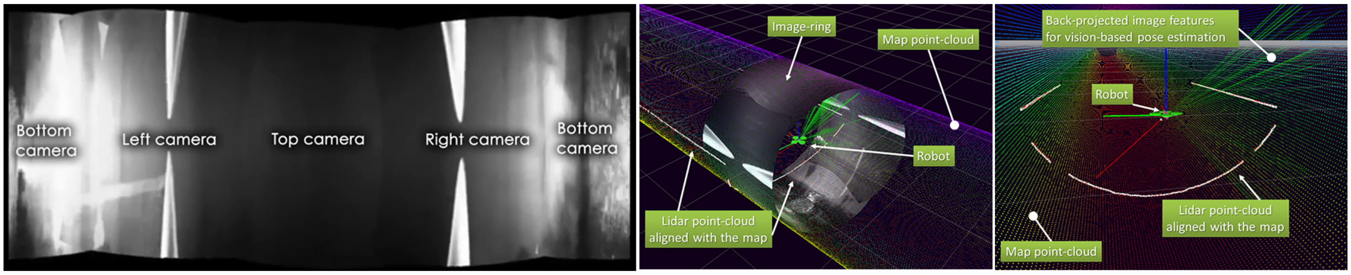

Bottom camera
Left camera
Top camera
Right camera
Bottom camera
Image-ring
Map point-cloud
Robot
Lidar point-cloud aligned with the map
Back-projected image features for vision-based pose estimation
Robot
Lidar point-cloud aligned with the map
Map point-cloud

Fig. 12: This figure shows the 360 degrees panoramic image reconstruction obtained using images from the four cameras and the 3D visualization in the RViz environment. On the left image the water drainage, propellers and the ceiling are clearly seen. Images at the right show a similar panoramic image wrapped around $\mathcal{M}$.

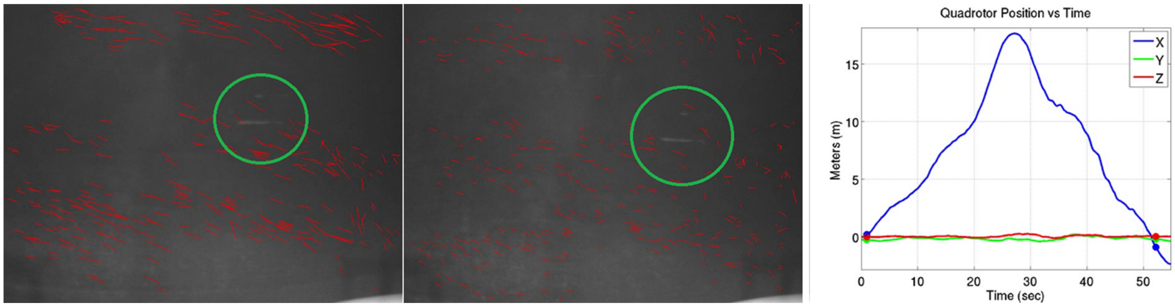


Fig. 13: This figure presents the VO results on the dataset collected in Carters Dam penstock, inclined section. The left and middle images show the flow field laid on the camera view. Green circles focus on the scratches on the wall which we use for manual loop closure. The plot shows the position estimate on which the two instances are showed with dots. The drift along the 40 m flight is <1 m.

Canyon Dam, AZ. We achieve safe autonomy in this challenging environment by fusing information from multiple lidars and VGA resolution cameras while the control is shared with a moderately skilled operator providing high-level commands using an RC interface. We present 360 degrees panoramic images and 3D textured reconstruction which offer a solution for convenient penstock inspection. In the future, we plan to do pose estimation and navigation in tunnels with obtuse angle bends, branch-off and merges.

References

[1] D. Mellinger and Alex Kushleyev, "The KHex Hex-Rotor Platform," KMel Robotics, Philadelphia, PA.

[2] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *Springer Tracts Adv. Robot.*, vol. 79, no. 5, pp. 361–373, 2014.

[3] M. Hehn and R. D'Andrea, "Quadrocopter trajectory generation and control," in *IFAC World Congr.*, vol. 18, no. 1, 2011, pp. 1485–1491.

[4] C. Richter, A. Bry, and N. Roy, "Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments," *Isrr*, no. Isrr, pp. 1–16, 2013.

[5] M. Pivtoraiko, D. Mellinger, and V. Kumar, "Incremental micro-UAV motion replanning for exploring unknown environments," in *Proc. - IEEE Int. Conf. Robot. Autom.* IEEE, 2013, pp. 2452–2458.

[6] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained MAV," in *Proc. - IEEE Int. Conf. Robot. Autom.* IEEE, 2011, pp. 20–25.

[7] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. - IEEE Int. Conf. Robot. Autom.*, 2014, pp. 15–22.

[8] N. Michael, S. Shen, K. Mohta, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno, E. Takeuchi, and S. Tadokoro, "Collaborative mapping of an earthquake damaged building via ground and aerial robots," *Springer Tracts Adv. Robot.*, vol. 92, no. 5, pp. 33–47, 2014.

[9] J. Das, G. Cross, C. Qu, and A. Makineni, "Devices, Systems, and Methods for Automated Monitoring enabling Precision Agriculture," *CASE*, 2015.

[10] G. Morgenthal and N. Hallermann, "Quality Assessment of Unmanned Aerial Vehicle (UAV) Based Visual Inspection of Structures," *Adv. Struct. Eng.*, vol. 17, no. 3, pp. 289–302, 2014.

[11] P. Hansen, H. Alismail, P. Rander, and B. Browning, "Visual mapping for natural gas pipe inspection," *Int. J. Rob. Res.*, vol. 34, no. 4-5, pp. 532–558, 2014.

[12] T. Özaslan, S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Field and Service Robotics," in *Field and Service Robotics 2013*, ser. Springer Tracts in Advanced Robotics, vol. 105, 2015, pp. 123–136.

[13] E. E. B. Olson, "Real-time correlative scan matching," in *2009 IEEE Int. Conf. Robot. Autom.* IEEE, 2009, pp. 4387–4393.

[14] A. Censi, "An ICP variant using a point-to-line metric," in *Proc. - IEEE Int. Conf. Robot. Autom.* IEEE, 2008, pp. 19–25.

[15] N. Naikal, J. Kua, G. Chen, and A. Zakhor, "Image augmented laser scan matching for indoor dead reckoning," University of California, Berkeley, Tech. Rep. UCB/EECS-2009-35, 2009.

[16] T. Liu, M. Carlberg, G. Chen, J. Chen, J. Kua, and A. Zakhor, "Indoor localization and visualization using a human-operated backpack system," in *2010 Int. Conf. Indoor Position. Indoor Navig. IPIN 2010 - Conf. Proc.* IEEE, 2010, pp. 1–10.

[17] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. Third Int. Conf. 3-D Digit. Imaging Model.* IEEE Comput. Soc, 2001, pp. 145–152.

[18] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *Comput. Vision ECCV*, 2006.

[19] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: a survey," *Trends® Comput. Graph.*, 2008.

[20] J. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel Corp.*, 2001.

[21] A. Censi, "On achievable accuracy for range-finder localization," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 4170–4175, 2007.