# Robust Localization and Localizability Estimation with a Rotating Laser Scanner

Weikun Zhen, Sam Zeng and Sebastian Scherer

*Abstract*— **This paper presents a robust localization approach that fuses measurements from inertial measurement unit (IMU) and a rotating laser scanner. An Error State Kalman Filter (ESKF) is used for sensor fusion and is combined with a Gaussian Particle Filter (GPF) for measurements update. We experimentally demonstrated the robustness of this implementation in various challenging situations such as kidnapped robot situation, laser range reduction and various environment scales and characteristics. Additionally, we propose a new method to evaluate localizability of a given 3D map and show that the computed localizability can precisely predict localization errors, thus helps to find safe routes during flight.**

## I. INTRODUCTION

There is a fast growing demand of small unmanned aerial vehicles (UAVs) in industry for the purpose of autonomous exploration and inspection. UAV's compact form-factor, ease of control and high mobility make them well suited for many tasks that are difficult for humans due to limited space or potential danger. However, this also requires the UAV system to be robust enough to handle multiple tasks in challenging situations such as lowlight, GPS-denied, cluttered or geometrically under-constrained environments.

To achieve robustness, state estimation algorithms must produce high quality results in challenging situations. A common solution to this problem is increasing the redundancy of the sensing system. A diverse set of sensors tend to capture more useful information, especially if they are different modalities. However, sensor redundancy creates new problems of its own such as synchronization issues and payload constraints. Therefore sensing system design must be a trade-off between redundancy and payload cost. Our robot localization system (see Figure 5) includes an IMU and a rotating 2D laser scanner. IMU is able to capture fast motion and the laser scanner could provide a global pose reference. Combination of them makes a compact, lightweight system that is robust to aggressive motion and guarantees low drift.

Popular solutions for robot localization can be divided into two categories: filtering based and optimization based approaches. Filtering based approaches (e.g. Bayesian filtering) infer the most likely state from available measurements and uncertainties, while optimization based approaches try to minimize reprojection error to find the optimal states. Selection of these two methods depends on the type of problem. Our goal is to develop an online algorithm capable of accurately localizing a UAV with limited onboard computing power. Thus the filtering approach is preferred since it is usually faster than iterative optimization procedures. Additionally, since our system deploys a rotating 2D laser scanner to estimate robot pose in 3D, it is generally impossible to find match between two consecutive scans. Although people [1] [2] has proposed methods accumulating a sweep (laser rotates for 180 degrees) of scans and then find match of two sweep maps, it is still not robust enough to fast motion which introduces large distortion in the sweep map.

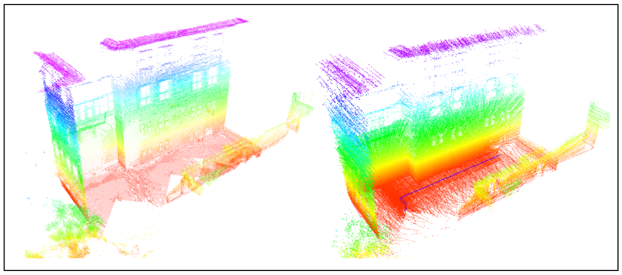

Fig. 1. An out door localization test. Original map is shown on the left and plots on the right visualizes the reconstruction and estimated path.

This work has three main contributions. First, it presents an implementation of ESKF-based localization algorithm which is suitable for different sensing systems, e.g., cameras or 3D LiDAR can be easily integrated into this framework. Second, this paper proposes a novel method for real time localizability estimation in 3D. This model is demonstrated by observing a correlation between predicted localizability and real pose estimation errors. Third, extensive tests are conducted based on real data or simulation to evaluate the robustness of ESKF localization algorithm and the correctness of localizability model.

The rest of this paper is organized as follows: Section II reviews prior work related to laser-based localization and localizability estimation. Section III details the the ESKF filtering procedures. Section IV introduces the localizability estimation algorithm. And experimental results and demonstrations regarding the introduced algorithms are presented in section V. Finally, section VI concludes this paper and examines possible future work.

## II. PRIOR WORK

There are plenty of research and applications on laser based localization. Here we include the most recent work on this topic. For optimization based approaches, iterative

Weikun Zhen is with the Robotics Institute and the Department of Mechanical Engineering of Carnegie Mellon University, weikunz@andrew.cmu.edu

Sam Zeng is with the Robotics Institute of Carnegie Mellon University, slz@andrew.cmu.edu

Sebastian Scherer is with the Robotics Institute of Carnegie Mellon University, basti@andrew.cmu.edu

closest point (ICP) and its variants are the most popular methods. For instance, Zhang [1] deploys a geometric feature extraction before solving ICP thus substantially improved the speed and accuracy. Extensions are achieved by fusing monocular vision in [3]. [2] proposes a graph based approach to account for distortion in a local map. Although both the methods in [1] [2] build high quality point cloud map, as stated in the last section, they are not reliable to aggressive motion. In the Bayesian filtering setting, laser data is typically transformed into pose measurements by Monte Carlo methods, such as [4], which is then used to update the measurement. Our work differs from [4] in that we are manipulating error states in both the prediction and correction steps. Also, IMU bias are included in the system states and estimated together with velocity and pose.

As for localizability estimation, there are multiple threads of research on this topic with similar objective but differing perspectives. Coastal navigation [5] is the earliest attempt to model information content, i.e. information gain, in the environment such that a planner can request a path maximizing accumulated information gain. Localizability has also been evaluated as a metric for planning by Merali et al. [6] for vehicles using landmark based localization and is simply calculated by counting the number of visible landmarks. Censi et al. [7] proposed an information matrix approach to evaluate the achievable accuracy using Cramér-Rao lower bound theorem which does not rely on feature matching or landmarks. Specifically, the eigenvalues and corresponding eigenvectors of the information matrix at a certain point indicates how well it is constrained along each direction with given measurements. Liu et al. [8] applied this approach in a 2D case and their planner aimed at maximizing the determinant of computed information matrix. [9] proposed a learning based approach to predict covariance matrix of a sensing system based on available informative features. Our paper differs from those prior work in that it develops a localizability model for 3D range sensors.

## III. ESTIMATION FORMULATION

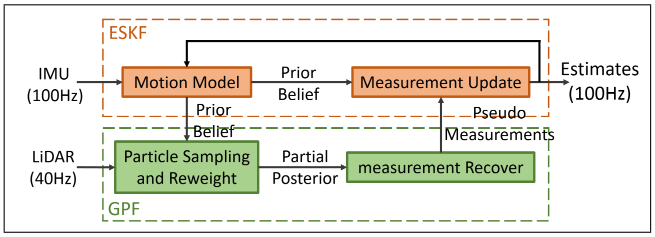

Fig. 2. Localization algorithm overview.

The problem setting is as follows: given a pre-bulit map and the initial pose of the robot, its current pose, velocity and IMU biases are estimated immediately when a new set of range data is available. The localization is achieved by combining an ESKF with a GPF as shown in Figure 2. This section gives a description of the propagation and update procedures and more details can be found in [4] [10].

An error state representation, compared to a nominal state representation, has several benefits [11]. First of all, error states are always close to zero, thus making it valid to approximate $R(\delta\theta)$ as $I + [\delta\theta]_\times$, where $[\delta\theta]_\times$ is the skew-symmetric operator. This approximation makes the derivatives of an exponential map easy to compute. Second, in an error state, the rotation error is represented as a 3D vector, which is more intuitive than other types of rotation representations such as matrix or quaternion. Besides, a 3D vector is straight forward to be put in a state vector, while a rotation matrix does not fit and a quaternion requires additional efforts to propagate its uncertainties. Finally, as the rotation error is always close to zero, it is far from a singular configuration.

### A. Error States

The state vector of the system contains

$v$: velocity in global frame
$p$: position in global frame
$R$: rotation matrix
$a_b$: accelerometer bias
$\omega_b$: gyroscope bias

The true state, predicted state and error state are represented as $x$, $\hat{x}$ and $\delta x$ respectively and satisfy

$$x = \hat{x} \oplus \delta x \tag{1}$$

Here $\oplus$ indicates a generic composition. Also note that in error state, angle vector $\delta\theta$ is represented by a $3 \times 1$ vector as a minimal representation of rotation error.

### B. Error Dynamics

The system error dynamics are derived from nominal state dynamics.

$$\dot{\hat{x}} = \begin{bmatrix} \dot{\hat{v}} \\ \dot{\hat{p}} \\ \dot{\hat{R}} \\ \dot{\hat{a}}_b \\ \dot{\hat{\omega}}_b \end{bmatrix} = \begin{bmatrix} \hat{R}(a_m - \hat{a}_b) + g \\ \hat{v} \\ \hat{R}[\omega_m - \hat{\omega}_b]_\times \\ 0 \\ 0 \end{bmatrix} \tag{2}$$

$$\delta x = \begin{bmatrix} \delta\dot{v} \\ \delta\dot{p} \\ \delta\dot{\theta} \\ \delta\dot{a}_b \\ \delta\dot{\omega}_b \end{bmatrix} = \begin{bmatrix} -\hat{R}[a_m - \hat{a}_b]_\times \delta\theta - \hat{R}\delta a_b - \hat{R} a_n \\ \delta v \\ -[\omega_m - \hat{\omega}_b]_\times \delta\theta - \delta\omega_b - \omega_n \\ a_w \\ \omega_w \end{bmatrix} \tag{3}$$

where $a_m, \omega_m$ is the acceleration and angular velocity measurements, $a_n, \omega_n$ denote accelerometer and gyroscope noise, and $a_w, \omega_w$ is the Gaussian random walk noise of biases.

### C. Propagation

The propagation step contains the estimate state propagation and the error covariance propagation. The estimate state is propagated through a direct Euler integration of (2). And the error covariance is propagated by linearizing the error state dynamics. Discrete propagation rule is shown in Eqn. (4) and (5) respectively.

$$\begin{bmatrix} \hat{v}_{t+1} \\ \hat{p}_{t+1} \\ \hat{R}_{t+1} \\ \hat{a}_{b(t+1)} \\ \hat{\omega}_{b(t+1)} \end{bmatrix} = \begin{bmatrix} \hat{v}_t + [\hat{R}_t(a_m - \hat{a}_{b(t)}) + g]\Delta t \\ \hat{p}_t + \hat{v}_t\Delta t + \frac{1}{2}[\hat{R}_t(a_m - \hat{a}_{b(t)}) + g]\Delta t^2 \\ \hat{R}_t R\left\{(\omega_m - \hat{\omega}_{b(t)})\Delta t\right\} \\ \hat{a}_{b(t)} \\ \hat{\omega}_{b(t)} \end{bmatrix} \tag{4}$$

$$\bar{\Sigma}_{t+1} = F_x \Sigma_t F_x^T + F_n Q_n F_n^T \tag{5}$$

where

$$F_x = \begin{bmatrix} I_3 & 0 & -\hat{R}_t[a_m - \hat{a}_{b(t)}]_\times \Delta t & -\hat{R}_t\Delta t & 0 \\ I_3\Delta t & I_3 & 0 & 0 & 0 \\ 0 & 0 & R^T\left\{(\omega_m - \hat{\omega}_{b(t)})\Delta t\right\} & 0 & -I_3\Delta t \\ 0 & 0 & 0 & I_3 & 0 \\ 0 & 0 & 0 & 0 & I_3 \end{bmatrix}$$

$$F_n = \begin{bmatrix} \hat{R}_t & 0 \\ 0 & 0 \\ 0 & I_9 \end{bmatrix}$$

$$Q_n = \begin{bmatrix} (\sigma_{a_n}\Delta t)^2 I_3 & 0 & 0 & 0 \\ 0 & (\sigma_{\omega_n}\Delta t)^2 I_3 & 0 & 0 \\ 0 & 0 & (\sigma_{a_w}\Delta t)^2 I_3 & 0 \\ 0 & 0 & 0 & (\sigma_{\omega_w}\Delta t)^2 I_3 \end{bmatrix}$$

### *D. Measurement Update*

In this step, a pseudo pose error measurement $\delta y \in \mathbb{R}^6$ is used to update full error state vector $\delta x \in \mathbb{R}^{15}$ in a normal KF fashion. $\delta y$ is called pseudo measurement since it is not acquired from sensors directly but recovered using a GPF.

*1) Observation Model:* With error state representation, the observation model is simply linear.

$$\delta y = H\delta x = \begin{bmatrix} 0 & I_3 & 0 & 0 & 0 \\ 0 & 0 & I_3 & 0 & 0 \end{bmatrix} \delta x \tag{6}$$

*2) Recover Pseudo Measurement:* Intuitively, the pseudo measurement $\delta y$ can be thought as measured by an imagery sensor. Actually, it is computed by following steps: First, based on pose priors $\delta \bar{x}^m_{t+1} \in \mathbb{R}^6, \bar{\Sigma}^m_{t+1} \in \mathbb{R}^{6\times 6}$, particles are drawn and weighted using the likelihood field model [12]. Second, the pose posterior $\delta x^m_{t+1} \in \mathbb{R}^6, \Sigma^m_{t+1} \in \mathbb{R}^{6\times 6}$ are computed as the weighted mean and covariance of the particles. Third, a pseudo measurement $\delta y^m_{t+1}$ and pseudo noise $C^m_{t+1}$ is recovered by inversing the KF measurement update

$$C^m_{t+1} = ({\Sigma^m_{t+1}}^{-1} - (\bar{\Sigma}^m_{t+1})^{-1})^{-1} \tag{7}$$

$$\delta y^m_{t+1} = (K^m)^{-1}(\delta x^m_{t+1} - \delta \bar{x}^m_{t+1}) + \delta \bar{x}^m_{t+1} \tag{8}$$

where $K^m = \bar{\Sigma}_{t+1} H^{mT}(H^m \bar{\Sigma}_{t+1} H^{mT} + C^m_{t+1})^{-1}$ is the pseudo Kalman gain. We refer readers to [4] for more details on inversing KF update.

*3) Correction:* Once the pseudo measurements are computed, it is used to update the full error states by a normal KF update. The Kalman gain is

$$K = \bar{\Sigma}_{t+1} H^T (H \bar{\Sigma}_{t+1} H^T + C_{t+1}) \tag{9}$$

Note $K \in \mathbb{R}^{15\times 6}$ is different to $K^m \in \mathbb{R}^{6\times 6}$. And full error state posterior and covariance are updated as

$$\delta x_{t+1} = K(\delta y_{t+1} - H\delta \bar{x}_{t+1}) \tag{10}$$

$$\Sigma_{t+1} = (I_{15} - KH)\bar{\Sigma}_{t+1} \tag{11}$$

### *E. Reset Nominal States*

The updated errors are integrated into the normal state by adding the error state to the estimate state.

$$\begin{bmatrix} \hat{v}_{t+1} \\ \hat{p}_{t+1} \\ \hat{R}_{t+1} \\ \hat{a}_{b(t+1)} \\ \hat{\omega}_{b(t+1)} \end{bmatrix} = \begin{bmatrix} \hat{v}_t + \delta v_{t+1} \\ \hat{p}_t + \delta p_{t+1} \\ \hat{R}_t \cdot R(\delta\theta_{t+1}) \\ \hat{a}_{bt} + \delta a_{b(t+1)} \\ \hat{\omega}_{bt} + \delta\omega_{b(t+1)} \end{bmatrix} \tag{12}$$

It is important to note that before the next iteration, the error states are set to zero.

## IV. LOCALIZABILITY

Given a map of the environment, in the form of a point-cloud, we would like to determine if the localization will consistently produce accurate results if the robot was in a certain configuration. To do this, we estimate the localizability of a given pose in a map to predict the localization performance. Localizability is a measure of a map's geometric constraints available to a range sensor from a given pose. Regions of high localizability should correspond to low state estimation errors and regions of low localizability should correspond to higher state estimation errors. In order to calculate this, we first make a key assumption that for a given point measurement, we can determine the corresponding surface normal based on the map. Next, we estimate surface normals for every point in the map. Then a set of visible points from the given pose is determined and finally, we accumulate the normals and analyze the constraints in each direction.

### *A. Position Constraints*

Each valid measurement from the sensor provides a constraint on the robot's pose. Specifically, by approximating surfaces as a plane locally, a measurement point $p_i$ lying on the plane is constrained by

$$n_i^T(p_i - p_{i,0}) = 0 \tag{13}$$

where $n_i$ is the surface normal, and $p_{i,0}$ is a point on the plane. Additionally, the sensor measurement provides the offset between the robot's position $x$ and $p_i$ as $x + r_i = p_i$, where $r_i$ is ray vector of this measurement. By substitution we have

$$n_i^T(x + r_i - p_{i,0}) = 0 \Rightarrow n_i^T x = n_i^T(p_{i,0} - r_i) = d_i \tag{14}$$

where $d_i$ is a constant vector. When combining all the constraints imposed by a set of measurement points, we have

$$\begin{bmatrix} n_{1_x} & n_{1_y} & n_{1_z} \\ n_{2_x} & n_{2_y} & n_{2_z} \\ \vdots & \vdots & \vdots \\ n_{k_x} & n_{k_y} & n_{k_z} \end{bmatrix} x = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \end{bmatrix} \Rightarrow Nx = D \tag{15}$$

### B. Evaluating Localizability

In order to accurately localize the robot, the sensor needs to be able to adequately constrain its pose in the three translational dimensions. The matrix $N$ describes the set of observable constraints from the given pose. Preforming a principal component analysis (PCA) on the row vectors of $N$, provides an orthonormal basis spanning the space described by the constraints from the surface normals. Furthermore, we can examine the singular values of $N$ with SVD as $U\Sigma V^T$. Here $\Sigma$ describes the cumulative strength of the constraints form each corresponding basis vector. Theoretically, we should be able to localize as long as all three of the singular values are non-zero. However, this proves to be unreliable in practice so we calculated localizability as the minimum singular value of $N$. More specifically: $L = \min(\text{diag}(\Sigma))$. This sets localizability equal to the strength of the constraints in the minimally constrained direction. Furthermore, this analysis also allows us to determine the minimally constrained direction as the singular vector corresponding to the minimal singular value.

### C. Examples

The following figures depict several results of this localizability algorithm near the outside of a building and around a bridge. In the case of the building, the sensor range was artificially reduced to 5m and 10m respectively. In these figures, blue regions correspond to areas with low localizability while red regions correspond to higher localizability.

*1) Building:* The left diagram in Figure 3 shows the localizability with the range limited to just a 5m radius. It becomes very difficult to localize outside of the spots nearest to the corners. Increasing the range to 10m allows the robot to localize in significantly larger area. However, the localization is expected to fail if the robot was to enter the open space in the middle section since the walls at either end are both well out of the 10m range.

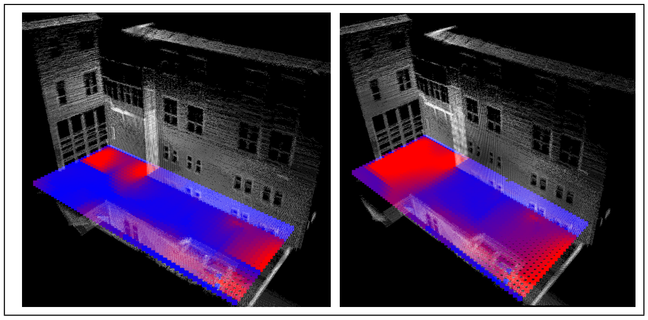

Fig. 3. The localizability of the area surrounding the outside of a building calculated across a 0.5m grid for both a 5m and 10m sensor ranges. In these pictures, the robot is also oriented facing the building

*2) Bridge:* The localizability around the bridge, depicted in Figure 4, is much more complex due to its 3D structure and large scale. Here localizability is estimated for sensors full 30m radius. In this case, localization is expected to perform poorly in the open space on top of the deck of the bridge since all of the surfaces in the map are parallel to the roadway.

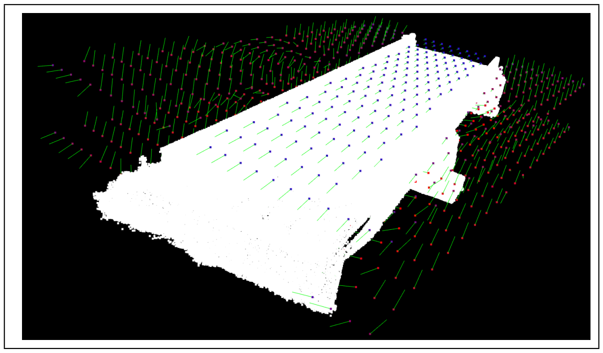

Fig. 4. The figure above demonstrates the localizability in 3D around a bridge. Each of the green arrows point in the minimally constrained direction which is given by the singular vector corresponding to the minimum singular value of $N$.

## V. EXPERIMENTS

### A. System Overview

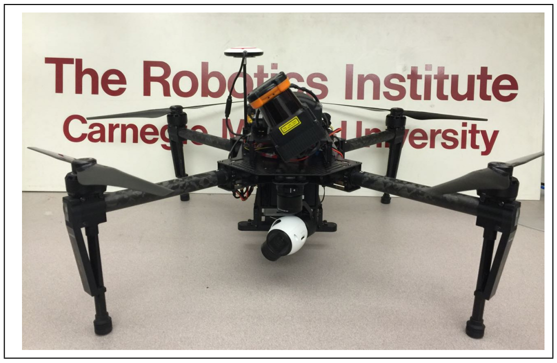

Fig. 5. The customized robot platform with on board computation and sensing system (rotating laser scanner and IMU). The front camera is for inspection thus is not used for state estimation.

A customized quadrotor, as shown in Figure 5, was used to test our localization algorithms. The quadrotor carries an onboard computer (Quad-core, 2GB RAM) which is responsible for all of the computations, an IMU (100 Hz), and a Hokuyo 2D laser scanner (270°FOV, 0.25° angle resolution, 40Hz) mounted on a continuously rotating motor (about 30 RPM). A motor encoder is used to project laser range data into body frame ($z$-axis downward, $x$-axis forward), which is defined to coincide with IMU frame.

### B. Indoor and Outdoor Tests

Figure 6 shows an indoor test conducted around an office area. Prior map is acquired by a LiDAR SLAM algorithm [1] and then converted into an occupancy map with Octomap [13]. The occupancy map has a scale of approximately $15 \times 20 \times 5$m, with 0.05m resolution. The robot is hand-held passing through corridors, a conference room, coffee lounge and eventually returns to the start point. The maximum translational velocity is about 1.2 m/s. In Figure 6, the reconstructed map aligns nicely with the original map, which implies the estimated poses are accurate.

Outdoor spaces are wider open and pre-built maps have larger grid size (0.1m) than indoor ones, resulting in fewer

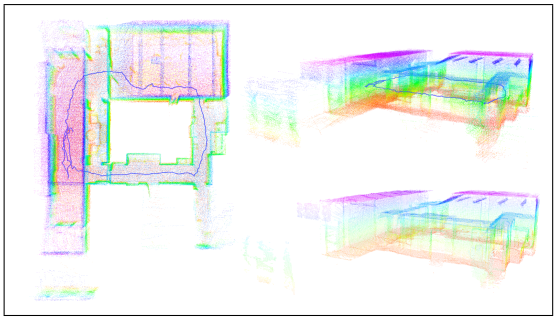

Fig. 6. Localization results in office area. Point clouds on the left and upper right are reconstructed with estimated pose trajectory (blue lines). Original reference map is shown at the lower right for comparison with reconstructed map.

details of structure to be captured for localization. This will cause an increase in estimation noise and uncertainties. Figure 1 visualizes the localization results in an inter-building area. The map is of size $15 \times 28 \times 9$m. The robot is controlled flying laterally with a maximum speed at about $0.8$ m/s while yawing between $-45°$ and $45°$ before returning to the origin (as indicated by Figure 10, upper plots). By observation, one can tell that overall estimation is quite accurate except some reconstructed building edges are slightly ‘fuzzy’ due to the estimation noise.

### C. Robustness Tests

The robustness of localization algorithm is measured by its capacity to recover from errors. It is tested in various settings using simulated or real data and also compared to an implementation of nominal state EKF. The two algorithms are sharing identical parameters such as sensor noise, map resolution and so on. We show that the ESKF outperforms the EKF in terms of robustness.

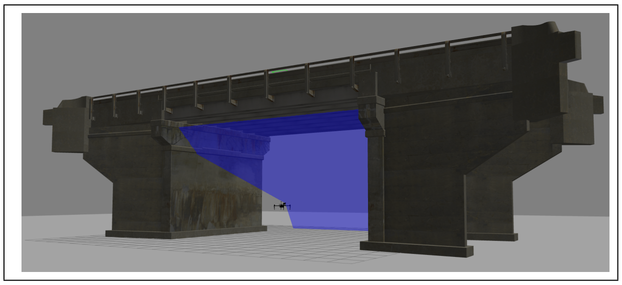

Fig. 7. A simulation of robot localization under a bridge model.

The first robustness experiment is a kidnapped robot test in simulation (see Figure 7). In this test, the algorithm assumes robot starts at the origin, however actual initial height is set with different values (from $0.5$m to the maximum error that still converges to zero). As shown in Figure 8, the ESKF implementation has an error tolerance of about 4 meters while the EKF implementation can only handle error no larger than 0.6 meter. Additionally, ESKF converges faster than the EKF. For instance, it takes about 2 seconds for the ESKF algorithm to recover from an error of 0.5 meter, while the EKF algorithm needs more than 8 seconds to converge.

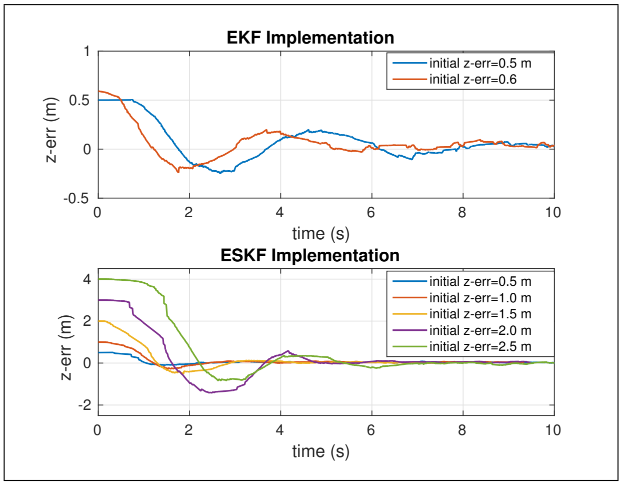

Fig. 8. Kidnapped robot test. The initial error tolerance is found by increasing the initial error until the algorithm diverges. Note that diverging cases are not visualized here.

The second experiment is designed to compare the localization algorithms on real world data. In this experiment, the robot orientation is kept pointing to the wall, while moving laterally towards right and then left. The effective range of laser is artificially reduced from 30m to 10m, which eliminates some useful structures for localization and degrades the state estimate. The 30 meter state estimates are then used as ground truth. Figure 9 shows the localization error of the two algorithms. We can observe that the ESKF algorithm recovers from some small errors (about 0.7m at maximum), while the EKF algorithm diverges at the end.

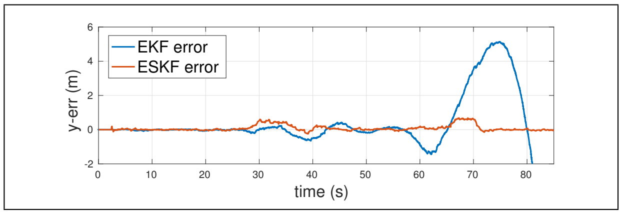

Fig. 9. Localization error in $y$-axis with reduced range laser measurements.

### D. Localizability Model Validation

To evaluate the proposed localizability model, we make use of the same outdoor dataset in Figure 1 and reduced the effective laser range to 10 meters.

When moving from left to right, the robot orientation is fixed. Thus at some places most of the captured data points lie on the wall and ground (as shown by the range plot in Figure 10), which weakens the constraints along the direction of motion. This results in large pose errors (blue lines of lower plots in Figure 10).

When moving from right back to left, the robot pointed at nearby structures. Since the robot has a rotating 2D laser, its measurements are densest along the axis of rotation. (denoted as thick line in Figure 10). The second path is predicted to have higher localizability which is followed by a decrease in positioning error as can be seen in Figure 10. Furthermore,

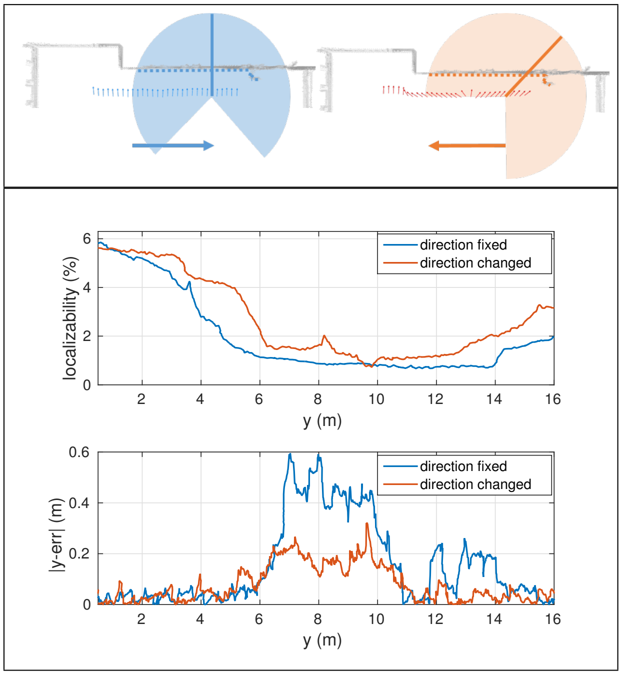

Fig. 10. Range plot (upper) shows the robot trajectories and sensor ranges. Correlation plot (lower) shows the correlation between localizability and positioning errors of two moving trajectories. Only error on $y$-axis are visualized here since errors on the other two axis are close to zero.

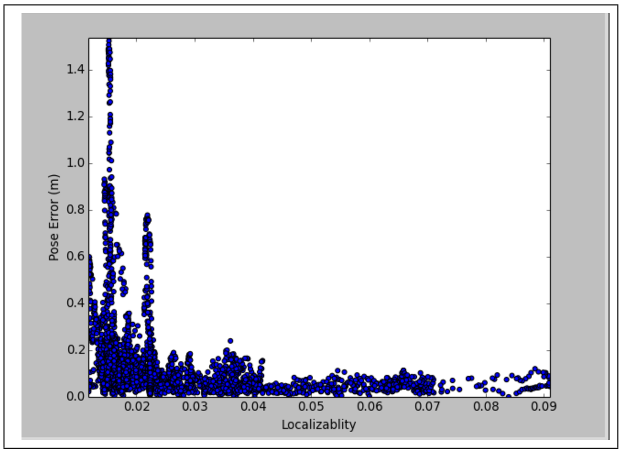

Fig. 11. Localization Error vs. Localizability at 10m range.

we collected additional data to provide further verification of the localizability estimation. Figure 11 compares the localization errors to the localizability estimates as the robot moved around the colored region shown in Figure 1. Full sensor range at 30m was used for ground truth pose and then we artificially reduced the range to 10m for pose estimates and the resulting error was compared to the predicted localizability at that pose. From the plot, we can tell that the proposed localizability estimation method provides a valid prediction of actual localization performance.

## VI. CONCLUSIONS

This paper presents a robust localization approach fusing IMU and laser range data into an ESKF framework. The algorithm is tested in an online setting and it is robust in various environments with different characteristics and scale. Additionally we provide a new method for estimating localizability to predict localization performance in 3D based on map geometry. Experimental results have shown its accurately predicts areas where localization can be unreliable.

In future work, we plan to develp a localizability based planner that would allow a robot to avoid unlocalizable areas or plan localzability optimal routes. Overall, this should allow for significantly more robust behavior resulting in safer autonomous UAV operations.

## ACKNOWLEDGMENT

This work is funded by the National Science Foundation under grant IIS-1328930. The authors would like to thank Geetesh Dubey for his help with building the robot system and solving hardware issues, and Ji Zhang for building indoor maps in localization tests.

## REFERENCES

[1] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems Conference (RSS)*, 2014, pp. 109–111.

[2] L. Kaul, R. Zlot, and M. Bosse, "Continuous-time three-dimensional mapping for micro aerial vehicles with a passively actuated rotating laser scanner," *Journal of Field Robotics*, vol. 33, no. 1, pp. 103–132, 2016.

[3] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2174–2181.

[4] A. Bry, A. Bachrach, and N. Roy, "State estimation for aggressive flight in gps-denied environments using onboard sensing," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1–8.

[5] N. Roy, W. Burgard, D. Fox, and S. Thrun, "Coastal navigation-mobile robot navigation with uncertainty in dynamic environments," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 1. IEEE, 1999, pp. 35–40.

[6] R. S. Merali, C. Tong, J. Gammell, J. Bakambu, E. Dupuis, and T. D. Barfoot, "3d surface mapping using a semiautonomous rover: A planetary analog field experiment," in *Proc. of the 2012 Int. Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*. Citeseer, 2012.

[7] A. Censi, "On achievable accuracy for range-finder localization," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 4170–4175.

[8] Z. Liu, W. Chen, Y. Wang, and J. Wang, "Localizability estimation for mobile robots based on probabilistic grid map and its applications to localization," in *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on*. IEEE, 2012, pp. 46–51.

[9] W. Vega-Brown, A. Bachrach, A. Bry, J. Kelly, and N. Roy, "Cello: A fast algorithm for covariance estimation," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3160–3167.

[10] J. Sola, "Quaternion kinematics for the error-state kf," *Laboratoire dAnalyse et dArchitecture des Systemes-Centre national de la recherche scientifique (LAAS-CNRS), Toulouse, France, Tech. Rep*, 2012.

[11] V. K. Madyastha, V. C. Ravindra, S. Mallikarjunan, and A. Goyal, "Extended kalman filter vs. error state kalman filter for aircraft attitude estimation," in *AIAA GNC*, 2011.

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[13] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013, software available at http://octomap.github.com. [Online]. Available: http://octomap.github.com