Visual-Inertial Localization With Prior LiDAR Map Constraints

Xingxing Zuo, Patrick Geneva, Yulin Yang, Wenlong Ye, Yong Liu, and Guoquan Huang

***Abstract*—In this letter, we develop a low-cost stereo visual-inertial localization system, which leverages efficient multi-state constraint Kalman filter (MSCKF)-based visual-inertial odometry (VIO) while utilizing an *a priori* LiDAR map to provide bounded-error three-dimensional navigation. Besides the standard sparse visual feature measurements used in VIO, the global registrations of visual semi-dense clouds to the prior LiDAR map are also exploited in a tightly-coupled MSCKF update, thus correcting accumulated drift. This cross-modality constraint between visual and LiDAR pointclouds is particularly addressed. The proposed approach is validated on both Monte Carlo simulations and real-world experiments, showing that LiDAR map constraints between clouds created through different sensing modalities greatly improve the standard VIO and provide bounded-error performance.**

***Index Terms*—Sensor fusion, localization, SLAM, visual-based navigation.**

I. Introduction

THE ability to perform high-precision localization is essential for autonomous vehicles. Over the past decades, a variety of sensors have been employed for localization in different environments [1]–[6]. GPS is widely used to provide absolute positioning, and suffers from unavailable measurements in radio-shadowed areas, unreliable signals due to blocked line-of-sight paths to external reference stations, or multi-pathing errors due to reflections of signals off nearby structures. As such, GPS often fails to give a reliable localization in urban and indoors areas. In particular, visual navigation with the aid of IMUs (i.e., visual-inertial navigation) is among the most popular approaches to provide 6DOF localization [1], [2], [5]–[7], which can be cheap and does not rely on the external signals like GPS. On the other hand, LiDAR-based localization and mapping has shown to have better performance [3], [4], [8] than visual-based solutions, due to its accurate range measurements, while also being robust to illumination variation, extending its application domains. However, the high cost of 3D LiDAR sensors largely hinders its *wide* deployment. The low-cost visual-inertial sensors for localization are desirable but are unable to achieve the same level of accuracy and robustness. The cost-effective fusion of these two sensing modalities to improve the estimation accuracy of a *pure* visual-inertial system, provided that a single accurate prior LiDAR map can be provided by a third party or built *a priori* with a LiDAR, is what we propose.

In this work, we propose a tightly-coupled visual-inertial state estimator that is able to utilize a LiDAR pointcloud map built *a priori*. For computational efficiency, we leverage the lightweight multi-state-constraint Kalman filter (MSCKF) [1] for online localization, which contains only a constant-size sliding window of IMU poses in the state vector, without keeping features. At the same time, we perform semi-dense mapping and produce visual pointclouds that can be registered with the prior LiDAR map. The registration results are used as global measurements of the camera poses and fused with visual sparse features' and inertial measurements in a tightly coupled manner within the MSCKF update, allowing for the correction of accumulated drift of the visual-inertial trajectory. As a result, the proposed visual-inertial system is efficient and provides 6DOF pose estimates in real time.

We note that prior visual feature maps are often used to aid online visual localization by matching the descriptors of visual feature [10], [11]. However, visual features with (local) descriptors are highly related to the appearance, which is easily changeable, highly affected by illumination, and can change over time. Compared with visual feature maps, range-based maps allow for the capturing of structural and geometric features of the environment which are less likely to vary over time, and thus do not require the re-mapping of areas unless major changes have occurred (i.e. construction or road changes), reducing the required prior map cost. Moreover, regardless of the lighting conditions, prior LiDAR maps of the environment can be reconstructed by LiDAR SLAM [3], [12] or static 3D laser scans, motivating us to leverage these LiDAR maps that can be easily created, updated, and contain large amounts of prevalent structural information.

To the best of our knowledge, this is the first time prior LiDAR map constraints have been *tightly* fused into computationally-efficient MSCKF-based visual-inertial estimation to provide real

Manuscript received February 24, 2019; accepted June 17, 2019. Date of publication July 5, 2019; date of current version July 19, 2019. This letter was recommended for publication by Associate Editor T. Peynot and Editor E. Marchand upon evaluation of the reviewers' comments. This work was supported in part by the National Key R&D Program of China under Grant 2017YFB1302003, in part by the University of Delaware College of Engineering, and in part by the Google. The work of P. Geneva was supported by the Delaware Space Grant College and Fellowship Program NASA under Grant NNX15AI19H. *(Xingxing Zuo and Patrick Geneva contributed equally to this work.) (Corresponding author: Yong Liu.)*

X. Zuo, W. Ye, and Y. Liu are with the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China (e-mail: xingxingzuo@zju.edu.cn; wenlong@zju.edu.cn; yongliu@iipc.zju.edu.cn).

P. Geneva is with the Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716 USA (e-mail: pgeneva@udel.edu).

Y. Yang and G. Huang are with the Department of Mechanical Engineering, University of Delaware, Newark, DE 19716 USA (e-mail: yuyang@udel.edu; ghuang@udel.edu).

Digital Object Identifier 10.1109/LRA.2019.2927123

time localization of bounded errors. In particular, the main contributions of this letter include the following:

- We design a *tightly-coupled* state estimator for visual-inertial localization which can efficiently utilize the prior LiDAR map constraints (of different sensing modality to live measurements).[1] As compared to expensive LiDAR-based counterpart, this is a low-cost solution providing 6DOF pose estimates of bounded error in real time.
- Global measurement constraints of the prior LiDAR map are constructed through visual semi-dense reconstruction and normal distribution transform (NDT)-based registration. These measurements are used in the MSCKF update along with the conventional sparse visual feature measurements and correct accumulated drift, for which we have also derived the analytical measurement Jacobians.
- The proposed visual-inertial localization system is computational efficient running *only* on a single multi-threaded CPU and is validated both in Monte Carlo simulations and real-world experiments.

II. Related Work

While map-based visual localization has been an active field of research in recent years [13], [14], using multi-modal sensing data in vision-based localization holds potential in the improvement of localization robustness and accuracy. Vision sensors can capture the appearance of the environment, while LiDARs are able to perceive structure more accurately. Over the past few years, there have been surging research efforts on visual localization with prior LiDAR maps. In particular, Lu *et al.* [15] proposed a monocular vision localization system for urban environments. In their work, the road markings in the LiDAR map, including solid and broken lines, are manually extracted and represented as a set of sparse points, after which Chamfer matching is used to register the detected road markings in the image against those in the prior map. Lu *et al.* [16] further extended to monocular localization aided by prior inputs, which leveraged the planar structure extracted from both vision and prior LiDAR data as anchoring information to fuse the heterogeneous maps. Coplanarity constraints were introduced to the bundle adjustment and showed improved visual odometry performance. Park *et al.* [17] proposed to combine constraints from LiDAR and visual features, and validated loop closure candidates with sequential observations to provide high quality loop closure detection. Compared to this work, we propose to leverage the whole LiDAR pointcloud, not only extracted planes, through registration to constrain our visual-inertial odometry (VIO) within an efficient EKF-based framework.

In [18]–[24] visual localization used the appearance of the prior map. In particular, in [19], a prior LiDAR map with reflectance information was used to render several synthetic views from different poses, to which live images captured by the camera were matched by normalized mutual information. This method can only use a single monocular camera for 2D localization. In [20], a prior LiDAR pointcloud was appended with illumination invariant appearance information allowing for registration in the illumination invariant space using Normalized Information Distance (NID) to measure the discrepancy of appearances. Pascoe *et al.* [21], Pascoe, Maddern, and Newman [22] achieved accurate localization by minimizing the NID between live images and those generated from the prior map. Wong *et al.* [23] proposed a method to determine a camera's pose that used area of edge regions shared between rendered views of a voxel occupancy map and in-vehicle camera images. The Monte Carlo approach was used in [24] to localize a panoramic camera by minimizing mutual projections of the gradient extracted from both synthesized depth and visual images. However, due to the high computational cost of obtaining synthetic appearance images from 3D prior maps, most of these methods require GPU acceleration.

There are also efforts focusing on matching pointclouds generated from cameras to those from LiDAR sensors to obtain relative poses. In [25] a registration based monocular localization algorithm was proposed, where a set of sparse 3D image keypoints were continuously matched with a prior LiDAR map for 6DOF pose estimation at approximate 10 Hz. A structure-based vision-laser matching framework was introduced in [26], where three types of structural descriptors were extracted to find point correspondences between the sensors. Kim, Jeong, and Kim [27] recently proposed a method of direct image alignment of synthetically generated LiDAR depth and stereo depth images to recover pose estimates. In [28], a probabilistic data association policy was proposed to improve pointcloud registration. Unlike standard ICP, each point in the source pointcloud was associated with a set of points in the target pointcloud and weighted based on a probabilistic distribution. Different from the above methods, our proposed approach is a low-cost light-weight MSCKF-based visual-inertial localization system, which is able to use a prior LiDAR map for bounding navigation errors. Our system is able to provide 6DOF pose estimates at high rate (attributed to the high frequency of IMU) while requiring *only* a multi-threaded CPU. Additionally, as a useful byproduct, a semi-dense map can be built online, which could be utilized to support high-level tasks such as obstacle avoidance and semantic segmentation.

III. Visual-Inertial State Estimation

In this section, within the standard MSCKF framework [1], we present the proposed visual-inertial estimator which tightly fuses visual and inertial measurements as well as prior LiDAR map constraints in order to bound localization errors.

A. State Vector

Our navigation state is given by:

$$\mathbf{x}_k = \begin{bmatrix} {}^{I_k}_{G}\bar{q}^\top & \mathbf{b}_\omega^\top & {}^{G}\mathbf{v}_{I_k}^\top & \mathbf{b}_a^\top & {}^{G}\mathbf{p}_{I_k}^\top & {}^{M}_{G}\bar{q}^\top & {}^{G}\mathbf{p}_M^\top & \mathbf{x}_C^\top \end{bmatrix}^\top \tag{1}$$

where ${}^{I_k}_{G}\bar{q}$ is the JPL unit quaternion [29] associated with the rotation matrix, ${}^{I_k}_{G}\mathbf{R}$, which rotates vectors from the global

[1]While in this work we particularly consider the LiDAR pointcloud map due to its commonness in practice, the proposed approach in principle can utilize any prior pointcloud map with correct scale.

frame of reference $\{G\}$ into the local frame $\{I_k\}$ of the IMU at timestep k, $\mathbf{b}_\omega$ and $\mathbf{b}_a$ are the gyroscope and accelerometer biases which corrupt the IMU measurements, ${}^G\mathbf{p}_{I_k}$ is the IMU position expressed in the global frame, and ${}^G\mathbf{v}_{I_k}$ is the corresponding velocity. We additionally estimate the rotation ${}^M_G\bar{q}$ and translation ${}^G\mathbf{p}_M$ between the LiDAR "map" frame $\{M\}$ and the global inertial frame $\{G\}$.

Following the standard MSCKF, we maintain a sliding window of IMU clones at the past m imaging times which do not evolve over time and are used during feature update:

$$\mathbf{x}_C = \begin{bmatrix} {}^{I_{k-1}}_{G}\bar{q}^\top & {}^G\mathbf{p}^\top_{I_{k-1}} & \cdots & {}^{I_{k-m}}_{G}\bar{q}^\top & {}^G\mathbf{p}^\top_{I_{k-m}} \end{bmatrix}^\top \tag{2}$$

The corresponding total error state is:

$$\delta\mathbf{x}_k = \begin{bmatrix} {}^{I_k}\delta\boldsymbol{\theta}^\top_G & \delta\mathbf{b}^\top_\omega & {}^G\delta\mathbf{v}^\top_{I_k} & \delta\mathbf{b}^\top_a & {}^G\delta\mathbf{p}^\top_{I_k} & {}^M\delta\boldsymbol{\theta}^\top_G & {}^G\delta\mathbf{p}^\top_M & \delta\mathbf{x}^\top_C \end{bmatrix}^\top \tag{3}$$

We define that the true value of the state, $\mathbf{x}_k$, estimated value $\hat{\mathbf{x}}_k$, and corresponding error state $\delta\mathbf{x}_k$, is related by the following generalized update operation:

$$\mathbf{x}_k = \hat{\mathbf{x}}_k \boxplus \delta\mathbf{x}_k \tag{4}$$

where for vector quantities, $\mathbf{v}$, this operation is simply addition, i.e., $\mathbf{v} = \hat{\mathbf{v}} + \delta\mathbf{v}$, and for quaternions we have:

$$\bar{q} \simeq \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta} \\ 1 \end{bmatrix} \otimes \hat{\bar{q}} \tag{5}$$

where $\otimes$ denotes quaternion multiplication [29].

B. State Propagation

The above state and corresponding covariance are propagated over time by integrating the incoming IMU measurements of linear accelerations ($\mathbf{a}_m$) and angular velocities ($\boldsymbol{\omega}_m$) based on the following generic nonlinear IMU kinematics [30]:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{a}_m - \mathbf{n}_a, \boldsymbol{\omega}_m - \mathbf{n}_\omega) \tag{6}$$

where $\mathbf{n}_a$ and $\mathbf{n}_\omega$ are the zero-mean white Gaussian noise of the IMU measurements. Note that the transform between the map and global inertial frame $\{{}^M_G\bar{q},\ {}^G\mathbf{p}_M\}$ and clone states $\mathbf{x}_C$ do not evolve over the propagation period. We linearize this nonlinear model at the current estimate, and then propagate the state estimate and covariance matrix using the standard EKF (e.g., see [1]).

C. State Update

We now look at how we can incorporate global registration measurements of the current camera in the map, alongside conventional sparse feature tracks. Explained in more detail in the next section, we will receive a registration pose with covariance by registering the semi-dense pointcloud with the *a priori* LiDAR map.

1) LiDAR Map Constraints: Consider receiving a measurement of $\{{}^{C_k}_M\mathbf{R}, {}^M\mathbf{p}_{C_k}\}$ which denotes the current left camera pose at timestep k in the map frame of reference. We can write this measurement as a function of the state:

$${}^{C_k}_M\mathbf{R} = {}^C_I\mathbf{R}\,{}^{I_k}_G\mathbf{R}\,{}^G_M\mathbf{R} \tag{7}$$

$${}^M\mathbf{p}_{C_k} = {}^C_M\mathbf{R}^\top({}^G\mathbf{p}_{I_k} - {}^G\mathbf{p}_M - {}^{I_k}_G\mathbf{R}^\top\,{}^C_I\mathbf{R}^\top\,{}^C\mathbf{p}_I) \tag{8}$$

where ${}^C_I\mathbf{R}$ and ${}^C\mathbf{p}_I$ are the extrinsic calibration transform between the IMU and the left camera frame. We can write the above measurement function and the linearization of it about the current state estimate $\hat{\mathbf{x}}_k$ as follows:

$$\mathbf{z} = \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k \tag{9}$$

$$\simeq \mathbf{h}(\hat{\mathbf{x}}_k) + \mathbf{H}_x\tilde{\mathbf{x}}_k + \mathbf{n}_k \tag{10}$$

where $\mathbf{n}_k$ is the white Gaussian noise with covariance $\mathbf{R}_k$ and $\mathbf{H}_x$ is the measurement Jacobian with respect to all state elements. The non-zero Jacobians are computed as:

$$\frac{\partial^{C_k}\delta\boldsymbol{\theta}_M}{\partial^{I_k}\delta\boldsymbol{\theta}_G} = {}^C_I\mathbf{R} \tag{11}$$

$$\frac{\partial^{C_k}\delta\boldsymbol{\theta}_M}{\partial^{M}\delta\boldsymbol{\theta}_G} = {}^C_I\mathbf{R}\,{}^{I_k}_G\mathbf{R} \tag{12}$$

$$\frac{\partial^{M}\delta\mathbf{p}_{C_k}}{\partial^{I_k}\delta\boldsymbol{\theta}_G} = {}^G_M\mathbf{R}^\top\,{}^{I_k}_G\mathbf{R}^\top\lfloor\,{}^C_I\mathbf{R}^\top\,{}^C\mathbf{p}_I\rfloor \tag{13}$$

$$\frac{\partial^{M}\delta\mathbf{p}_{C_k}}{\partial^{M}\delta\boldsymbol{\theta}_G} = -{}^G_M\mathbf{R}^\top\lfloor\,{}^G\mathbf{p}_{I_k} - {}^G\mathbf{p}_M - {}^{I_k}_G\mathbf{R}^\top\,{}^C_I\mathbf{R}^\top\,{}^C\mathbf{p}_I\rfloor \tag{14}$$

$$\frac{\partial^{M}\delta\mathbf{p}_{C_k}}{\partial^{G}\delta\mathbf{p}_{I_k}} = {}^G_M\mathbf{R}^\top \tag{15}$$

$$\frac{\partial^{M}\delta\mathbf{p}_{C_k}}{\partial^{G}\delta\mathbf{p}_{M}} = -{}^G_M\mathbf{R}^\top \tag{16}$$

where $\lfloor\cdot\rfloor$ denotes the skew symmetric matrix. We can directly update the state using the pose measurement as in the standard EKF update [31]. Note that while we are estimating the transform between the map frame and the global inertial frame, an initial guess of this transform is needed in practice.

2) Visual Feature Measurements: As in the standard VIO, we track a set of sparse features across the sliding window of scholastically cloned poses $\mathbf{x}_C$. When these features have reached maximum track length or have lost track, they are first triangulated in 3D and then further refined using bundle adjustment (BA). Successfully optimized features are passed through a Mahalanobis-distance test and used in the standard MSCKF update in which their measurements are projected onto the nullspace of the feature measurement Jacobian, preventing the need to include the positions of the features in the state vector [1]. These sparse features allow for short-term localization, while the prior map constraint pose update prevents long-term drift.

IV. Visual Processing

The proposed visual-inertial localization system architecture is illustrated in Figure 2, whose visual processing module is composed of two main parts: i) sparse feature tracking and ii) semi-dense visual to LiDAR map registration. Specifically,

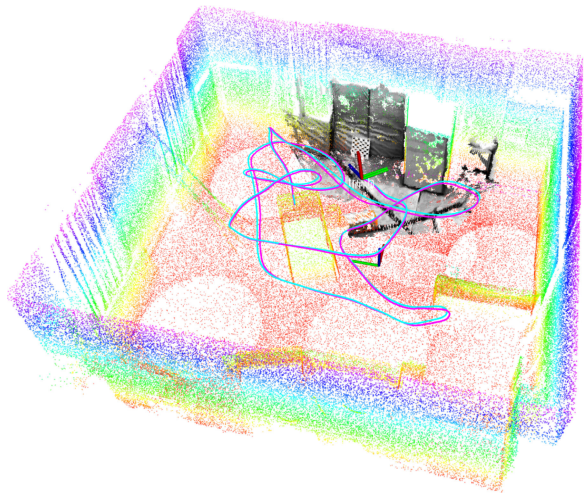

Fig. 1. The proposed visual-inertial localization system with the prior LiDAR map constraints runs on the EurocMav dataset [9]. The prior LiDAR map is colored by height, while the groundtruth and estimated trajectory are plotted in cyan and pink, respectively. A semi-dense pointcloud reconstructed from a series of keyframes is also shown in black.

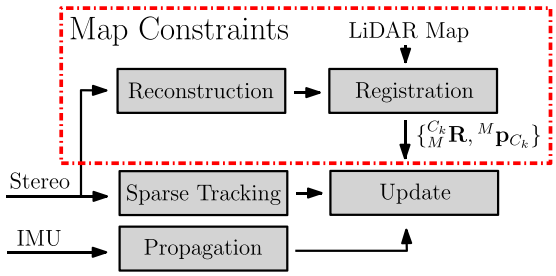

Fig. 2. Data flow of the proposed localization system. Incoming stereo and inertial measurements can be seen on the left, while the proposed map constraint sub-system has been highlighted with a red box.

incoming stereo images are processed in two separate ways: i) conventional sparse feature tracks are triangulated and used in the MSCKF update (Section III-C2), and ii) stereo pairs are used to construct a semi-dense pointcloud that is then registered with a prior LiDAR map to provide a LiDAR map constraint (Section III-C1). In this section, we detail how we reconstruct a semi-dense visual cloud and then register it to the prior LiDAR map to obtain the constraints that are tightly fused in the MSCKF update.

A. Semi-Dense Reconstruction

As compared to the sparse features extracted and tracked through KLT [32], a semi-dense cloud captures the 3D structure of the environment and creates a high-density pointcloud that is suitable for registration. We note that due to the different modalities of the prior LiDAR pointcloud, it is expected that the prior pointcloud contains structural surfaces such as planes, while a sparse visual cloud typically contains points that have high intensity gradients corresponding to edges and corners in the environment. This motivates us to leverage semi-dense cloud reconstruction for registration with the prior LiDAR pointcloud.

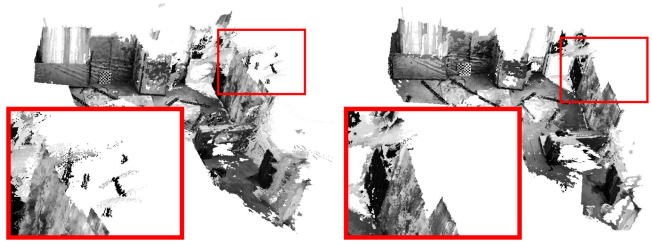

Fig. 3. Visual semi-dense reconstruction without depth refinement (left) has high levels of noise on walls of the room. By contrast, the visual semi-dense reconstruction *with* depth refinement (right) exhibits lower amounts of noise along the boundary (see picture inset). These pointclouds are a small subset of the EurocMav V1_02_medium sequence [36].

To remain computationally efficient, we reconstruct the semi-dense cloud for a subset of incoming images, which we denote as "keyframes" in the rest of the letter. Keyframes are selected based on distance and orientation thresholds to ensure that they cover the largest spacial area with minimal overlap. Note that while the current keyframe selection strategy appears to be simple and ad hoc, it does prove the concept of our keyframe-based semi-dense reconstruction, nevertheless more sophisticated methods will be explored in the future. Due to the nature of pointcloud registration problem, a window of keyframes is desired as it increases the spacial volume and ensures that the registration problem is well constrained. The larger baseline between consecutive keyframes increases the overall reconstruction quality of the semi-dense cloud. While in this work we leverage stereo depth map computation to expedite semi-dense cloud reconstruction, one could construct the scaled depth map through recent developments in neural networks [33]. Specifically, for a new incoming keyframe we first compute its depth-map using stereo block matching by minimizing the sum of absolute distances (SAD) error over patches in the image. Conventional stereo block matching, as compared to other more accurate methods [34], [35], is both computationally efficient and has acceptable depth reconstruction that we will refine later using additional keyframes.

B. Depth Correspondence Matching

After computing each keyframe's depth map in the window, the depth is refined through correspondence matching. An example of the improved semi-dense cloud quality resulting from the combination of multiple keyframes and depth refinement can be seen in Figure 3, in which the overall noise in the cloud is shown to be reduced. Since the pose estimate of each keyframe is known from the MSCKF estimator, we project each keyframe's depth map into the other image planes to calculate common correspondences. Denoting the frame we are projecting into as kf_j, we use the estimated transforms to project the 3D points contained in *each* keyframe kf_i as follows:

$$ {}^{C_j}\mathbf{p}_f = {}^{C_j}_{G}\mathbf{R}({}^{G}\mathbf{p}_{C_i} - {}^{G}\mathbf{p}_{C_j}) + {}^{C_j}_{C_i}\mathbf{R}\,{}^{C_i}\mathbf{p}_f \tag{17} $$

$$ \mathbf{u}' = \Pi({}^{C_j}\mathbf{p}_f) \tag{18} $$

where $\Pi(\cdot)$ is the camera projection function and $\mathbf{u}'$ is the corresponding pixel coordinate in the j-th keyframe.

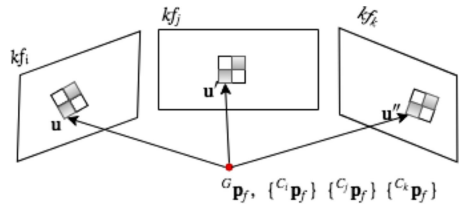

Fig. 4. Illustration of depth correspondence matching, find multiple observations among different keyframes. The neighboring pixels in the 2×2 path are also possible observations.

Figure 4 shows the depth correspondence matching process for depth refinement. Having projected all points from all other keyframes into the j-th keyframe we need to determine what projections match the points in the j-th frame. We consider the projection of a pixel $\mathbf{u}'$, and look to determine if it matches the pixel at the projected location $\mathbf{u}$. If it does match the pixel at the projected location, we add both the point and its depth to an "observation" set $\mathcal{P}_j(\mathbf{u})$ for the given pixel in the j-th keyframe. Specifically, for a given point, $\mathbf{u}'$, that projects to the pixel $\mathbf{u}$ in the j-th keyframe we perform the following compatibility test:

i) Difference between the intensities is smaller than a given threshold: $|I_i(\mathbf{u}') - I_j(\mathbf{u})| \leq \Phi_I$
ii) Difference between the image gradient of is smaller than a given threshold: $|G_i(\mathbf{u}') - G_j(\mathbf{u})| \leq \Phi_G$
iii) Depth value between the transformed point ${}^{C_j}\mathbf{p}'_f$ and ${}^{C_j}\mathbf{p}_f$ should not be over a given threshold: $|{}^{C_j}\mathbf{p}'_f(z) - {}^{C_j}\mathbf{p}_f(z)| \leq \Phi_D$

where $I_i(\cdot)$, $I_j(\cdot)$ and $G_i(\cdot)$, $G_j(\cdot)$ return the intensity and gradient for the i-th and j-th keyframe, respectively. Due to numerical evaluation, when we project a pixel from the i-th to the j-th keyframe we have to discretize the projected pixel location. To account for this discretization, we say that the bounding "neighboring" pixels in the j-th keyframe could also correspond to the projected point and its depth. Thus, we perform the compatibility test for a projected pixel in respect to the 2×2 patch of pixels around the projection point. We empirically found that only checking the 2×2 neighborhood, as compared to a larger area, provided adequate rejection and allowed for a reduction in the amount of computation required per-projection. If a neighboring pixel $\mathbf{u}_n$ in the 2×2 patch passes the test, the projected point and its depth is added to the corresponding location in the $\mathcal{P}_j(\mathbf{u}_n)$ set.

We repeat this projection process for each keyframe in the window. After creation of the observation set $\mathcal{P}_j$ for all keyframes, we use this correspondence and observation information to both refine the pointcloud of each keyframe, while simultaneously rejecting outliers. We check the amount of observations for each pixel, if the number of observations is below a certain threshold, we consider this point as an outlier and is removed. If a pixel has many observations, we refine the depth by taking the mean of all projected depth observations for that pixel. We found that this gives high quality pointclouds with reduced noise levels due to the fusion of multiple depth observations.

C. Pointcloud Assembly

After refinement of each keyframe's depth estimates, we projected each cloud into the newest keyframe's frame which we denote as the "reference" keyframe. We note that this reconstruction process can be computationally expensive due to the large amount of semi-dense points that can be recovered. To allow for real-time computation, we parallelize this process using a secondary thread that asynchronously provides prior LiDAR map constraints. We found that a small window of three keyframes spread over half a meter with at least 30 degrees of orientation change allowed for enough density to constrain the assembled pointcloud during NDT while also balancing the total computational cost. Note that other keyframe policies are possible as long as they ensure that there are view overlaps between sequential keyframes for a smooth and uniform reconstruction.

D. NDT Pointcloud Registration

Having reconstructed a semi-dense pointcloud in our reference keyframe, we now look to register it to our LiDAR prior map. We selected NDT for both its shown accuracy and speed [37], [38], along with the possibility to quantify the uncertainty of the registration result. NDT leverages representing the pointcloud as a combination of normal distributions [39]. Consider a set $\mathcal{L} = \{\mathbf{p}_i \mid i \in \{1 \dots m\}\}$ of $|\mathcal{L}| = m$ point samples that have been drawn from a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where the mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ can be obtained as follows:

$$\boldsymbol{\mu} = \frac{1}{|\mathcal{L}|} \sum_{i=1}^{|\mathcal{L}|} \mathbf{p}_i \tag{19}$$

$$\mathbf{n} = \left[\, (\mathbf{p}_1 - \boldsymbol{\mu})^\top \cdots (\mathbf{p}_{|\mathcal{L}|} - \boldsymbol{\mu})^\top \, \right]^\top \tag{20}$$

$$\boldsymbol{\Sigma} = \frac{1}{|\mathcal{L}| - 1} \mathbf{n}\mathbf{n}^\top \tag{21}$$

Since all points are represented as Gaussians, NDT is insensitive to uneven sample distributions commonly found in LiDAR mapping applications. We use point-to-distribution (P2D) variant of NDT [39], which formulates the registration of a source cloud, $\mathcal{L}_s$, to a target pointcloud, $\mathcal{L}_t$, as a problem of fitting the source points to the target's distribution. In the P2D variant of NDT, the best pose $\{{}^t_s\mathbf{R}, {}^t\mathbf{p}_s\}$, is found by optimizing the following objective function:

$$\mathrm{c}(\mathcal{L}_s, \mathcal{M}_t, {}^t_s\mathbf{R}, {}^t\mathbf{p}_s) = \sum_{i=1}^{|\mathcal{L}_s|} -d_1 \exp\Big(-\frac{d_2}{2} \bar{\mathbf{p}}_{si}^\top \, {\Sigma_{xi}}^{-1} \, \bar{\mathbf{p}}_{si} \Big) \tag{22}$$

with:

$$d_1 = -\log(c_1 + c_2) + d_3 \tag{23}$$

$$d_2 = -2\log((-\log(c_1 \exp(-1/2) + c_2) - d_3)/d_1) \tag{24}$$

$$d_3 = -\log(c_2) \tag{25}$$

$$\bar{\mathbf{p}}_{si} = {}^t_s\mathbf{R}\mathbf{p}_{si} + {}^t\mathbf{p}_s - \boldsymbol{\mu}_{xi} \tag{26}$$

where c_1, c_2 are design constants and $\boldsymbol{\mu}_{xi}, \Sigma_{xi}$ are the mean and Gaussian distribution of a NDT cell in the target pointcloud that

Fig. 5. The bird's-eye view of the synthetic Gazebo dataset. The 836 meter long groundtruth trajectory of the robot is shown in red. The maximum velocity of the robot was set to be 2.5 m/s.

the source point $\mathbf{p}_{si}$ resides in. Using this objective function, we can derive the Hessian matrix whose inverse is an approximate covariance of the registration result [39, p. 61]. Special care is taken to transform the calculated measurement covariance into the correct measurement error state, see (7)–(8), as the P2D NDT implementation within the PCL library [40] uses Euler angles to represent orientation, thus requiring a covariance propagation to transform the orientation error state into that of our quaternion parameterization [41].

To evaluate the performance of the NDT registration and reject outliers when the problem is under constrained or noise levels are too great, we employ the following rejection criteria to ensure that only healthy measurements will be processed by the filter:

i) After calculation of the NDT measurement Hessian matrix [39], we compute the minimum eigenvalue, λ_h, of its negative and ensure that it is larger then a threshold.
ii) We ensure that the negative summed cost (22), is small and below a certain threshold.
iii) The inlier ratio between the final set of NDT inliers and initial source cloud reflects the quality of the registration result and should ideally be near one.
iv) The final prior LiDAR map constraint measurement [see (7) and (8)], is processed through a Mahalanobis-distance test.

V. Experimental Validations

A. Monte Carlo Simulations

We first evaluated the proposed visual-inertial localization (termed *map-aided MSCKF*) within the Gazebo simulator [42]. A mobile Pioneer 3-DX [43] with a stereo camera, LiDAR, and IMU was simulated moving through a constructed town (see Figure 5). Using the groundtruth poses, the prior LiDAR pointcloud was generated by transforming each scan into the starting frame. The entire cloud was downsampled with a voxel grid filter of 0.2 meters to increase its sparsity. The groundtruth IMU readings were corrupted with white noise and random walk biases, while the synthetic images were corrupted with white noise distributed with an intensity distribution. All other key simulation parameters are specified in Table I. Due to the large-scale nature of the environment (far away objects) for the small-scale robot considered, we especially found that using multiple keyframes (three in the keyframe window) was necessary to ensure that enough environmental structure was available to constrain the NDT registration.

As shown in Figure 6, the proposed method achieved an overall lower root mean square error (RMSE) [44] in both orientation and position estimates. It is interesting to test the performance of the proposed method with different levels of prior map noises. We injected different white noise standard deviations σ to all points in the simulated prior map. The average RMSE of position and orientation errors for 5 Monte Carlo simulations are shown in Table II. We found that our system is robust to the quality of the prior map and outperforms the standard

TABLE I
Monte Carlo Simulation Parameters

Parameter	Value	Parameter	Value
IMU Freq. (Hz)	200	Camera Freq. (Hz)	20
Gyro noise σ (rad/s)	2.6968e-04	Pixel intensity σ (pixel)	4
Gyro bias σ (rad/s)	2.9393e-06	NDT Cell Resolution (m)	0.7
Acc. noise σ (m/s^2)	4.00e-3	Pointcloud noise σ (m)	Table II
Acc. bias σ (m/s^2)	4.00e-4	Trajectory length (m)	836

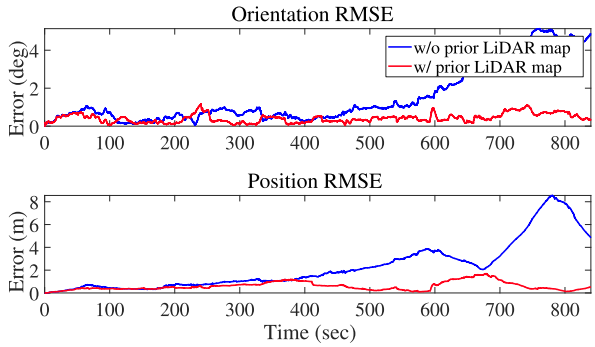

Fig. 6. Simulation results of the orientation and position RMSE for the standard MSCKF and map-aided MSCKF with a prior LiDAR map under noises of $\sigma = 0.03$ m).

TABLE II
RMSE With Different Levels of Prior Map Noises

RMSE	MSCKF	MSCKF w/ Map $\sigma = 0.03m$	MSCKF w/ Map $\sigma = 0.30m$	MSCKF w/ Map $\sigma = 0.40m$	MSCKF w/ Map $\sigma = 0.50m$
Position (m)	3.19	1.26	2.33	3.08	3.24
Orientation (deg)	2.77	1.11	1.87	2.22	2.94

TABLE III
Relative Pose Error for Different Segment Lengths

Segment Length	MSCKF	MSCKF w/ Map	VINS-Mono (odom) [6]	VINS-Mono (loop) [6]
7m	**0.136**	0.143	0.162	0.156
14m	**0.148**	0.154	0.180	0.160
21m	0.194	**0.184**	0.233	0.208
28m	0.202	**0.175**	0.246	0.223
35m	0.237	**0.191**	0.273	0.260

TABLE IV
AVERAGE ATE [45] FOR 5 RUNS OF THE MAP-AIDED MSCKF, STANDARD MSCKF, AND VINS-MONO VARIANTS [6] (LEFT). TIMING INFORMATION FOR THE TWO SYSTEM THREADS: (I) SPARSE VISUAL-INERTIAL ODOMETRY AND (II) PRIOR MAP CONSTRAINT (RIGHT)

Dataset	MSCKF	MSCKF w/ Map	VINS-Mono (odom) [6]	VINS-Mono (loop) [6]	Timing Sparse VIO (s)	Timing Map Constraints (s)
V1_01_easy	0.072	0.056	0.077	**0.044**	0.0267	0.8546
V1_02_medium	0.073	0.055	0.095	**0.054**	0.0256	0.7859
V1_03_difficult	0.108	**0.087**	0.158	0.209	0.0277	0.7734
V2_01_easy	0.079	0.069	0.069	**0.062**	0.0297	0.8273
V2_02_medium	0.093	**0.089**	0.132	0.114	0.0248	0.9412
V2_03_difficult	0.203	**0.149**	0.253	**0.149**	0.0246	0.6609

MSCKF even with the prior map deteriorated to 0.4 meter noise levels.

B. Real-World Experiments

To further validate, we compared the proposed map-aided MSCKF with the standard MSCKF [1] and VINS-Mono [6] on the EurocMav datasets [36]. The EurocMav datasets provide stereo greyscale images at 20 Hz along with a 200 Hz ADIS16448 IMU and groundtruth room scan (prior LiDAR map). Each dataset has a dynamic aerial trajectory, of average length of 70 meters, that each exhibit varying degrees of motion blur and textureless regions. To our knowledge, there is no cross-modality algorithm that leverages LiDAR prior maps within the visual-inertial architecture, and thus we compare to the start-of-the-art VINS-Mono which can leverage loop-closures to limit long-term drift. On startup, the prior LiDAR map is loaded into memory and the position of the filter initialized in the map frame with a perturbation of the provided groundtruth transform (in our experiments, we found that NDT could easily recover from perturbations of 3 cm and 5 degrees in the initial guess). In practice one would need to solve the "kidnapped robot" problem to initialize the unknown transform between the initialized frame and that of the map, which is not trivial. An example trajectory and reconstructed semi-dense pointcloud that has been registered to the global prior LiDAR map can be seen in Figure 1.

To evaluate the accuracy of the compared localization algorithms, we compute the absolute trajectory error (ATE) [45] and are shown in Table IV (units are in meters). Note that these ATE results are averaged over 5 runs in order to account for the randomness inherent in the feature tracking frontends. It is clear from Table IV that the proposed map constrained method in general outperforms the standard MSCKF and obtains similar performance to that of VINS-Mono with loop closures. This is expected as our odometry system is able to leverage the loop closure information provided by the prior map and should perform with similar accuracy to methods that leverage other forms of loop closure information. Shown in Table III (units are in meters) and Figure 7, we have additionally calculated the Relative Pose Error (RPE) [46] over all trajectories to provide insight into how the error of each algorithm grows with the trajectory length. We can see that as the length of the trajectory segment grows the larger impact the map constraint has on the estimate. The poor performance in the shorter segments lengths

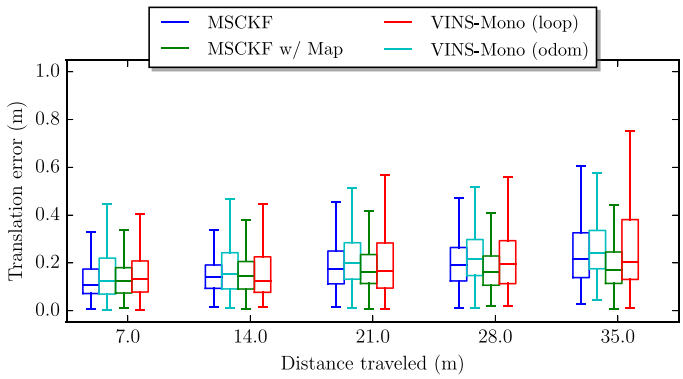

Fig. 7. Boxplot of the relative trajectory error statistics. The middle box spans the first and third quartiles, while the whiskers are the upper and lower limits. Plot best seen in color.

are likely due to the correction "jumps" caused after update when using the prior map constraint.

Timing averages of the major threads in the system can be seen in Table IV. The standard MSCKF does not use the secondary map constraint thread and thus, with sparse feature tracking only, can operate on the upwards of 30 Hz. The proposed system has only the overhead of a secondary thread that performs the semi-dense reconstruction and NDT with the LiDAR prior map which operates at a lower 1.25 Hz and updates the state as soon as NDT registrations become available.[2] We note that the pose estimate is still updated on the upwards of 30 Hz with the standard sparse feature tracks. We found that the computation within the secondary NDT thread is split evenly between semi-dense reconstruction, NDT pointcloud registration, and covariance calculation, while the main sparse visual-inertial odometry thread is dominated by the sparse feature tracking.

VI. CONCLUSIONS AND FUTURE WORK

In this letter, we have developed a tightly-coupled state estimation algorithm for visual-inertial localization with prior LiDAR map constraints. Within the efficient MSCKF framework, the proposed approach is able to provide real-time 6DOF pose estimates. In particular, in order to leverage an accurate prior map of a different sensing modality to bound localization errors,

[2]We ran on an Intel(R) Xeon(R) E3-1505Mv6 @ 3.00 GHz CPU.

we perform NDT to register visual semi-dense map (point-clouds) to the LiDAR prior map, whose results are then tightly fused in the MSCKF update along with the sparse visual feature measurements. It should be noted that, as the cameras and IMUs are becoming ubiquitous in part due to their complementary sensing capabilities as well as decreasing cost and size, the proposed low-cost light-weight global localization holds great implications in a wide range of practical applications such as autonomous driving. In the future, we will investigate how to efficiently take into account the prior map uncertainty into our tightly-coupled estimation framework, as well as how the visual semi-dense map can be used to update the prior map.

REFERENCES

[1] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 10–14, 2007, pp. 3565–3572.

[2] M. Li and A. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *Int. J. Robot. Res.*, vol. 32, no. 6, pp. 690–711, 2013.

[3] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Proc. Robot., Sci. Syst.*, 2014, vol. 2, pp. 1–9.

[4] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 2174–2181.

[5] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular slam with map reuse," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 796–803, Apr. 2017.

[6] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[7] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.

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

[9] M. Burri *et al.*, "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.

[10] W. Zhang and J. Kosecka, "Image based localization in urban environments," in *Proc. 3rd Int. Symp. 3-D Data Process., Visualization, Transm.*, 2006, pp. 33–40.

[11] H. Kim, D. Lee, T. Oh, H.-T. Choi, and H. Myung, "A probabilistic feature map-based localization system using a monocular camera," *Sensors*, vol. 15, no. 9, pp. 21636–21659, 2015.

[12] D. Droeschel and S. Behnke, "Efficient continuous-time slam for 3d lidar-based online mapping," in *Proc. IEEE Int. Conf. Robot. Autom*, 2018, pp. 1–9.

[13] N. Piasco, D. Sidibé, C. Demonceaux, and V. Gouet-Brunet, "A survey on visual-based localization: On the benefit of heterogeneous data," *Pattern Recognit.*, vol. 74, pp. 90–109, 2018.

[14] A. Sujiwo, E. Takeuchi, L. Y. Morales, N. Akai, Y. Ninomiya, and M. Edahiro, "Localization based on multiple visual-metric maps," in *Proc. IEEE Int. Conf. Multisensor Fusion Integration Intell. Syst.*, 2017, pp. 212–219.

[15] Y. Lu, J. Huang, Y.-T. Chen, and B. Heisele, "Monocular localization in urban environments using road markings," in *Proc. IEEE Intell. Vehicles Symp*. 2017, pp. 468–474.

[16] Y. Lu, J. Lee, S.-H. Yeh, H.-M. Cheng, B. Chen, and D. Song, "Sharing heterogeneous spatial knowledge: Map fusion between synchronous monocular vision and lidar or other prior inputs," in *The International Symposium on Robotics Research (ISRR), Puerto Varas, Chile*, vol. 158, 2017.

[17] C. Park, S. Kim, P. Moghadam, J. Guo, S. Sridharan, and C. Fookes, "Robust photogeometric localization over time for map-centric loop closure," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1768–1775, Jan. 2019.

[18] A. D. Stewart and P. Newman, "Laps-localisation using appearance of prior structure: 6-DoF monocular camera localisation using prior pointclouds," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 2625–2632.

[19] R. W. Wolcott and R. M. Eustice, "Visual localization within lidar maps for automated urban driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 176–183.

[20] W. Maddern, A. D. Stewart, and P. Newman, ''LAPS-II: 6-DoF day and night visual localisation with prior 3d structure for autonomous road vehicles," in *Proc. IEEE Intell. Vehicles Symp.*, 2014, pp. 330–337.

[21] G. Pascoe, W. Maddern, A. D. Stewart, and P. Newman, "Farlap: Fast robust localisation using appearance priors," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 6366–6373.

[22] G. Pascoe, W. Maddern, and P. Newman, "Direct visual localisation and calibration for road vehicles in changing city environments," in *Proc. IEEE Int. Conf. Comput. Vision Workshops*, 2015, pp. 9–16.

[23] D. Wong, Y. Kawanishi, D. Deguchi, I. Ide, and H. Murase, "Monocular localization within sparse voxel maps," in *Proc. IEEE Intell. Vehicles Symp.*, 2017, pp. 499–504.

[24] P. Neubert, S. Schubert, and P. Protzel, "Sampling-based methods for visual navigation in 3-d maps by synthesizing depth images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 2492–2498.

[25] T. Caselitz, B. Steder, M. Ruhnke, and W. Burgard, "Monocular camera localization in 3-d lidar maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 1926–1931.

[26] A. Gawel, T. Cieslewski, R. Dub, M. Bosse, R. Siegwart, and J. Nieto, "Structure-based vision-laser matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2016, pp. 182–188.

[27] Y. Kim, J. Jeong, and A. Kim, "Stereo camera localization in 3-D LiDAR maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–9.

[28] G. Agamennoni, S. Fontana, R. Y. Siegwart, and D. G. Sorrenti, "Point clouds registration with probabilistic data association," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 4092–4098.

[29] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3-D attitude estimation," Univ. Minnesota, Dept. Comput. Sci. Eng., Tech. Rep., 2005-002, Mar. 2005.

[30] A. B. Chatfield, *Fundamentals of High Accuracy Inertial Navigation*. Reston, VA, USA: Amer. Inst. Aeronaut. Astronaut., Inc., 1997.

[31] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. London, U.K.: Academic, 1979, vol. 1.

[32] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Seattle, WA, USA, Jun. 21–23, 1994, pp. 593–600.

[33] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 270–279.

[34] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

[35] F. Cheng, H. Zhang, M. Sun, and D. Yuan, "Cross-trees, edge and superpixel priors-based cost aggregation for stereo matching," *Pattern Recognit.*, vol. 48, no. 7, pp. 2269–2278, 2015.

[36] M. Burri *et al.*, "The euroc micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.

[37] B. Huhle, M. Magnusson, W. Straßer, and A. J. Lilienthal, "Registration of colored 3d point clouds with a kernel-based extension to the normal distributions transform," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 4025–4030.

[38] M. Magnusson, A. Nuchter, C. Lorken, A. J. Lilienthal, and J. Hertzberg, "Evaluation of 3d registration reliability and speed-a comparison of ICP and NDT," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 3907–3912.

[39] M. Magnusson, "The three-dimensional normal-distributions transform: An efficient representation for registration, surface analysis, and loop detection," Ph.D. dissertation, Örebro universitet, Örebro, Sweden, 2009.

[40] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1–4.

[41] N. Trawny and S. Roumeliotis, "Jacobian for conversion from euler angles to quaternions," Dept. Comput. Sci. Eng., Univ. Minnesota, Tech. Rep. 2005-004, Nov. 2005.

[42] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2004, vol. 3, pp. 2149–2154.

[43] Feb. 24, 2019. [Online]. Available: https://robots.ros.org/pioneer-3-dx/

[44] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. New York, NY, USA: Academic, 1988.

[45] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D slam systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 573–580.

[46] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 7244–7251.