Received: 25 December 2018 | Revised: 27 October 2019 | Accepted: 19 December 2019

DOI: 10.1002/rob.21936

FIELD REPORT

WILEY

Multimodal localization: Stereo over LiDAR map

Xingxing Zuo[1] | Wenlong Ye[1] | Yulin Yang[2] | Renjie Zheng[1] | Teresa Vidal-Calleja[3] | Guoquan Huang[2] | Yong Liu[1]

[1]Institute of Cyber-Systems and Control, College of Control Science and Engineering, Zhejiang University, Hangzhou, China

[2]Department of Mechanical Engineering, University of Delaware, Newark, Delaware

[3]Centre for Autonomous Systems, School of Mechanical and Mechatronic Engineering, University of Technology Sydney, Sydney, New South Wales, Australia

Correspondence
Yong Liu, Institute of Cyber-Systems and Control, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China
Email: yongliu@iipc.zju.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Numbers: 61836015, U1509210

Abstract

In this paper, we present a real-time high-precision visual localization system for an autonomous vehicle which employs only low-cost stereo cameras to localize the vehicle with a priori map built using a more expensive 3D LiDAR sensor. To this end, we construct two different visual maps: a sparse feature visual map for visual odometry (VO) based motion tracking, and a semidense visual map for registration with the prior LiDAR map. To register two point clouds sourced from different modalities (i.e., cameras and LiDAR), we leverage probabilistic weighted normal distributions transformation (ProW-NDT), by particularly taking into account the uncertainty of source point clouds. The registration results are then fused via pose graph optimization to correct the VO drift. Moreover, surfels extracted from the prior LiDAR map are used to refine the sparse 3D visual features that will further improve VO-based motion estimation. The proposed system has been tested extensively in both simulated and real-world experiments, showing that robust, high-precision, real-time localization can be achieved.

KEYWORDS
computer vision, localization, perception, position estimation, sensors

1 | INTRODUCTION

It is essential for autonomous vehicles to be able to perform low-latency, high-precision, and robust localization, which has attracted significant research efforts over the past decades. Different sensors providing either global or local measurements have been employed for this purpose. For example, as a common practice, the global positioning system (GPS) or real-time kinematic (RTK)-GPS is widely used outdoor for global position estimates. However, in many GPS-denied environments, such as near tall buildings, indoor, and underground, such global positioning measurements are not reliable or unavailable, and sensors of local measurements such as cameras and LiDARs have to be resorted (Cvišić, Ćesić, Marković, & Petrović, 2018; Mur-Artal & Tardós, 2017a,b; Qin & Shen, 2017; J. Zhang & Singh, 2014, 2015).

Vision-based and LiDAR-based localization approaches are among the two most popular solutions in GPS-denied environments. In particular, monocular camera based visual odometry (VO) is cost-effective but only provides up-to-scale ego-motion estimates. Although stereo VO is able to recover motion scale with considerable accuracy, it is still an open-loop odometry system with localization errors growing over time. To reduce VO drift and bound navigation error, a prior visual feature-based map is typically leveraged during online localization, by matching the descriptors of visual features detected in the current images to those in the map (Kim, Lee, Oh, Choi, & Myung, 2015; Mur-Artal & Tardós, 2017b; W. Zhang & Kosecka, 2006). Once feature correspondences are established, the 6DOF pose (relative transformation from the current camera frame to the map frame) can be efficiently computed by using, for example, iterative closest point (ICP; Besl & McKay, 1992) or perspective-n-point (PnP; Lepetit, Moreno-Noguer, & Fua, 2009) algorithms. It is known that descriptor-based visual features are highly related to scene appearance and can be easily affected by lighting conditions, especially outdoor. Thus, substantial research efforts have focused on building visual feature-based maps for long-term operation by detecting distinct features from multiple runs to capture visual variances, which clearly requires significant efforts to collect and manage data

(Churchill & Newman, 2013; Mühlfellner et al., 2016). On the other hand, LiDAR-based localization solutions (J. Zhang & Singh, 2014, 2016) can achieve higher accuracy even with illumination variation, while the high cost of 3D LiDAR sensors greatly hinders their widespread adoptions. Nevertheless, in some application field, such as autonomous driving, it is well worth building an environmental map by LiDAR sensors, primarily because an ideal map should be as accurate as possible and as stable as possible (with minimum updates at later times) regardless of the dynamic changes into the environment so that it can be reused at later times and/or by other vehicles. With this in mind, a priori LiDAR map may be maintained and updated only when nonnegligible environmental changes occur (e.g., new constructions, route changes). Once it is built, regardless of lighting variation, the rich structural information captured by the LiDAR map can be used to enhance online visual localization. Furthermore, a prior visual map may carry little information for online visual localization when there is a large path deviation between the route of online visual localization and that of the prior mapping, in part because of the camera's smaller field of view (often less than 180°) as compared to the full 360° coverage of the LiDAR sensor (e.g., Velodyne LiDARs; Velodyne VLP-16, 2018).

In this study, we propose to perform keyframe-based stereo vision-based localization with a prior LiDAR map to provide bounded-error 6DOF pose estimates in real-time. To this end, we particularly build and maintain two different visual maps aiming to utilize the information available in images effectively: (a) The first is the sparse visual feature-based map that includes visual features and encodes the covisibility relationship between keyframes as in Mur-Artal, Montiel, and Tardos (2015); (b) The second is the semidense visual map that consists of point clouds reconstructed from local keyframes and is used for registration with the prior LiDAR map (PLM). As the sparse visual feature map is unable to capture sufficient structural information of the environment and it is difficult to obtain accurate registration by matching the visual feature map to the prior LiDAR map, we primarily leverage the visual point cloud for this purpose, while the prior LiDAR map is also used to refine the visual feature map and thus improve the VO. It should be noted that, due to the different sensing modality used in mapping (LiDAR) and localization (camera), the point clouds generated from the stereo camera and LiDAR are of significant discrepancy (see Figure 1), posing significant challenges on registration of these two types of point clouds. For example, the LiDAR point clouds cover wider areas with relatively accurate metric information but low elevation resolution, while the dense visual point clouds from stereo images offers a better interpretation of surroundings but with poor depth measurement. To take up the challenge arising in this multimodal point-cloud registration, we advocate the probabilistic weighted normal distributions transformation (ProW-NDT) by explicitly modeling and considering the uncertainty of every single point in the semidense visual map. Due to the efficiency of this registration, the proposed approach is lightweight and can run real-time on a multicore CPU, while the state-of-the-art approaches on visual localization with LiDAR maps (Neubert, Schubert, & Protzel, 2017; Maddern, Stewart, & Newman, 2014; Pascoe, Maddern, & Newman, 2015; Pascoe, Maddern, Stewart, & Newman, 2015; Stewart & Newman, 2012; Wolcott & Eustice, 2014; Wong, Kawanishi, Deguchi, Ide, & Murase, 2017) that often rely on the support of powerful GPU to render synthetic images from LiDAR maps.

Specifically, the main contributions of this paper are the following:

- We develop a low-cost, real-time visual localization system by utilizing a prior LiDAR map. As compared to expensive LiDAR-based localization, this is a low-cost solution. The proposed method is able

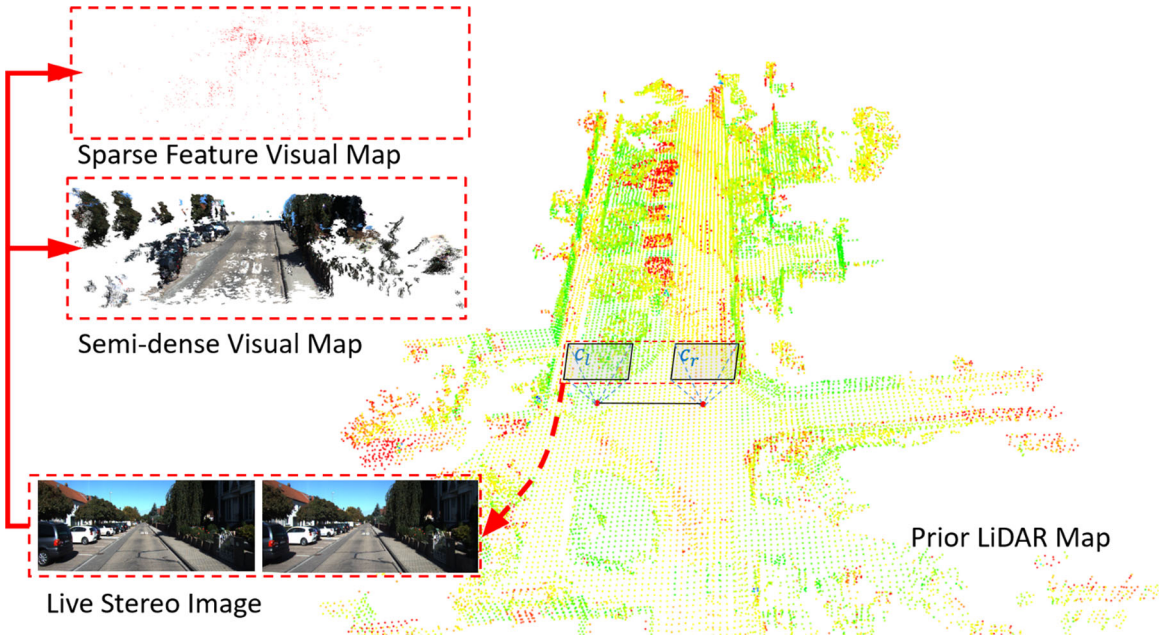

FIGURE 1 We propose a stereo visual localization system aided by a prior LiDAR map. In this system, both the sparse feature visual map and semidense visual map are constructed in order for robust fusion with the prior LiDAR map to provide low-cost, high-precision localization solutions [Color figure can be viewed at wileyonlinelibrary.com]

to provide full 6DOF poses of the stereo camera with centimeter-level accuracy, and suitable for real-time application (with no need of GPU).
- We leverage the ProW-NDT to effectively register the semidense visual map to the prior LiDAR map, by considering the uncertainty of point clouds in the visual map. We also develop a novel nonrigid refinement method for the 3D sparse visual features. The refinement is based on surfels extracted from the PLM and shows to improve VO.
- The proposed method is extensively validated on the simulated data set, the publicly available KITTI (Geiger, Lenz, & Urtasun, 2012) and KAIST urban data sets (Jeong, Cho, Shin, Roh, & Kim, 2019), as well as our own data. The results demonstrate that our algorithm is able to improve visual-only localization. Additionally, three different map reuse systems: the proposed visual localization over LiDAR map, visual localization over visual map, LiDAR localization over LiDAR map, are compared to further validate the proposed system.

The rest of the paper is organized as follows: After reviewing related literature in the next section, we describe the overview of the proposed visual localization system in Section 3. In Sections 4.1 and 4.2, we present in detail the local semidense visual map reconstructed from keyframes and its registration to the prior LiDAR map via the ProW-NDT. In Section 4.3, we enforce structure constraints to refine the sparse visual features with the prior LiDAR map. The pose graph optimization is presented in Section 4.4. In Section 5, we perform extensive validations of the proposed approach on both simulations and experiments. Finally, we conclude the paper with possible future research directions in Section 6.

2 | RELATED WORK

There is a rich body of literature on robot localization and mapping (e.g., Cvišić et al., 2018; Mur-Artal & Tardós, 2017a,b; Qin & Shen, 2017; J. Zhang & Singh, 2014, 2015), while in this section we only review the work that is closely related to the proposed visual localization with LiDAR maps.

2.1 | Visual localization and SLAM

Map-based visual localization using cameras is an active research topic in recent years (Piasco, Sidibé, Demonceaux, & Gouet-Brunet, 2018; Sujiwo et al., 2017). Various measurement information can be exploited in vision-based localization such as semantic clues and geometric features, which essentially determines the specifications of vision-based localization systems (Piasco et al., 2018). A combination of different types of information is often leveraged to overcome the limits of image-only based methods. Moreover, different robust visual features such as SIFT (Ng & Henikoff, 2003) and BRIEF (Calonder, Lepetit, Strecha, & Fua, 2010), which describe the key points detected in images to make them distinguishable from each other, are often used for visual localization, by matching features extracted from live image streams to the features in prior 3D feature database (Kim et al., 2014, 2015; Lu, Ly, Shen, Kolagunda, & Kambhamettu, 2013; W. Zhang & Kosecka, 2006). For example, W. Zhang and Kosecka (2006) prototype an image-based localization system in urban environments using SIFT features to select the closest views in the database. Cummins and Newman (2008) propose FAB-Map, a probabilistic localization method in the space of appearance, which aims to address the problem of visual aliasing for large-scale navigation. Kim et al. (2015) consider the uncertainty of features when performing 3D-to-2D correspondence to estimate 6DOF pose. Specifically, after performing PnP (Lepetit et al., 2009) with these correspondences, a batch least-squares optimization problem is formulated to further improve the localization accuracy.

Visual feature-based SLAM jointly estimates the camera pose and features in the scene by minimizing the reprojection errors and has attracted significant attention (Klein & Murray, 2007; Mur-Artal & Tardós, 2017a; Mur-Artal et al., 2015). In particular, performing loop closures in SLAM is essential, which can be achieved by relying on matching currently observed features with those in the map (Klein & Murray, 2007; Mur-Artal & Tardós, 2014; Williams et al., 2009). However, such feature-based methods highly rely on the availability of visual features and the accuracy of the prior feature map, both of which are often not reliable in outdoor environments. Hence, research efforts have focused on building feature maps for long-term navigation (Churchill & Newman, 2013; McManus, Churchill, Maddern, Stewart, & Newman, 2014; Mühlfellner et al., 2016; Sujiwo et al., 2017). For example, McManus et al. (2014) propose a method that runs two localization threads in parallel: one using images in the RGB color space and the other using images in an illumination-invariant color space. This method is able to reduce visual localization failure rates when dealing with severe lighting changes. Churchill and Newman (2013) incrementally learn a model of the environment, whose complexity varies naturally due to the variations of scene appearance but which is of sufficient richness of the instances to allow for reliable localization despite various operation conditions. Mühlfellner et al. (2016) build a summary map of visual features for long-term localization, which is constructed by accumulating distinct features when a robot repeatedly traverses its workspace, and implicitly represents the scene variations of changing lighting conditions and different weathers. Sujiwo et al. (2017) build several visual feature maps over multiple runs based on LiDAR-based localization, which are reused during online visual localization, shown to be robust to changing environments.

2.2 | LiDAR map-based visual localization

Cameras can capture the appearance of the environment, while LiDAR is able to perceive metric structural information. To take advantage of both sensors, research efforts have been devoted to leveraging the LiDAR maps for visual localization. For example, in Maddern et al. (2014), Neubert et al. (2017), Pascoe, Maddern, and Newman (2015), Pascoe, Maddern, Stewart, and Newman (2015), Stewart and Newman (2012), Wolcott and Eustice (2014), and Wong et al. (2017), visual localization with synthetic images rendered from prior LiDAR maps, is performed. In particular, in

Wolcott and Eustice (2014), a prior LiDAR map augmented with surface reflectivities is used to render several synthetic views from different poses. The live images captured by the camera are matched with these synthetic images based on the normalized mutual information, which, however, can only use a monocular camera for 2D localization. In Maddern et al. (2014), both the prior LiDAR point clouds and live images are attached with illumination-invariant appearance, and registration between the two are conducted in the corresponding illumination-invariant space. During the registration, the normalized information distance (NID) is used to measure the discrepancy of the appearance. In Pascoe, Maddern, and Newman (2015) and Pascoe, Maddern, Stewart, and Newman (2015) better localization is achieved by minimizing the NID between a live image and an image generated from the prior colored 3D map. Recently, Wong et al. (2017) propose a method that uses a monocular camera to localize a vehicle, which uses the edge regions shared between rendered views from a voxel occupancy map and live images to determine the camera pose. Neubert et al. (2017) develop a Monte-Carlo based approach to localize a panoramic camera by minimizing mutual projections of the gradients, extracted from synthesized depth images and live visual images. However, since it is computationally expensive to obtain the synthetic images by rendering the prior map from different camera poses, all these methods are computationally intensive and need the GPU support. It should be noted that the very recent work (Kim, Jeong, & Kim, 2018) estimates 6DOF camera poses based on the minimization of depth residual, which is able to run on CPUs but unable to achieve real-time performance. Note also that Lu, Huang, Chen, and Heisele (2017) propose a monocular localization system in urban environment, where the road markings in the LiDAR map, including solid lines and broken lines, are manually extracted and represented as a set of sparse points and the Chamfer matching (Barrow, Tenenbaum, Bolles, & Wolf, 1977) is used to register the detected road markings in an image against those in the prior map. As an extension of this study, the planar structure extracted from both visual and prior LiDAR data is used as the anchoring information to fuse the heterogeneous maps (Lu, Lee, et al., 2017). This approach exploits the coplanarity constraints in its bundle adjustment (BA), while our proposed method uses the whole LiDAR point clouds (not only the planar structure).

There are also research efforts focusing on matching the point clouds generated from cameras with those from LiDAR sensor to obtain the transformation best aligning the two point clouds. In particular, in Caselitz, Steder, Ruhnke, and Burgard (2016), the algorithm of continuous registration of a set of sparse 3D visual features to the prior LiDAR map for 6DOF camera poses, is proposed, which is tested on one modified sequence of KITTI data sets (Geiger et al., 2012) and their own data set. A structure-based vision-LiDAR matching framework is introduced in Gawel et al. (2016), where three types of structural descriptors are designed to find point correspondences and evaluated on several different data sets. In Agamennoni, Fontana, Siegwart, and Sorrenti (2016), the probabilistic data association is proposed to improve the registration between the sparse and dense point clouds. Different from the standard ICP, each point in the source point cloud is associated with a set of points in the target point cloud, and each association is weighted according to a probabilistic distribution. Acceptable performance can be obtained when the algorithm converges. In contrast, the proposed system uses two types of visual map and both visual maps are fused with the prior LiDAR map, thus improving the accuracy and robustness.

3 | SYSTEM DESIGN

In this section, we present the overall architecture of the proposed real-time visual localization system, which includes three main parallel threads: visual tracking, local mapping, and prior LiDAR map-based visual localization, as shown in Figure 2. The main steps of the proposed approach are outlined in Algorithm 1.

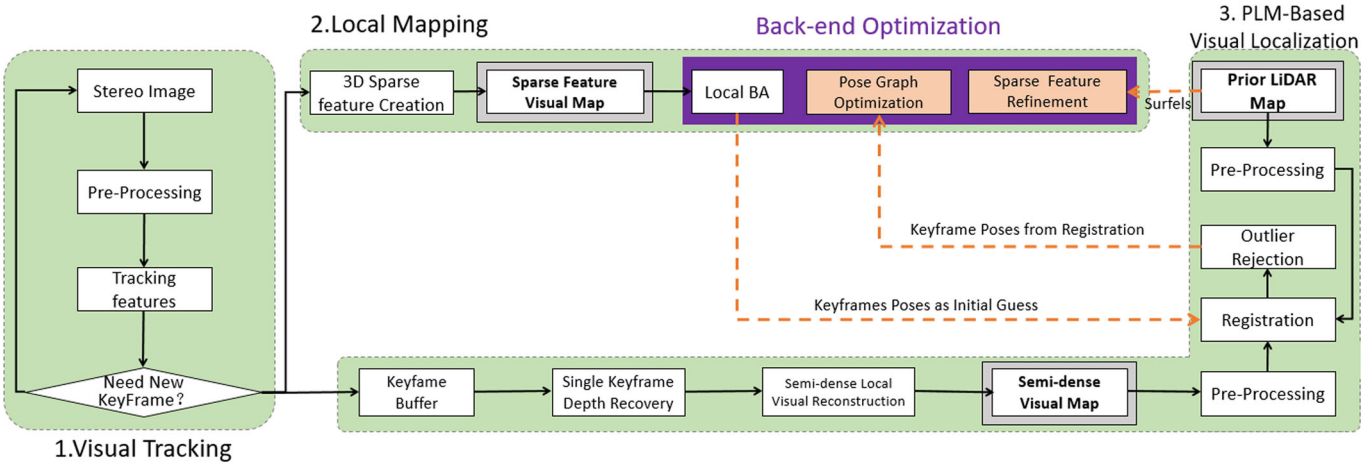

FIGURE 2 The proposed visual localization system with the PLM. The three main threads are visual tracking, local mapping, and PLM-based visual localization. Two types of visual maps are maintained: sparse feature visual map used for visual tracking and semidense visual map for registration. PLM, prior LiDAR map [Color figure can be viewed at wileyonlinelibrary.com]

Algorithm 1: Multi-modal Localization

Input: A set of stereo images $\mathcal{I}$, PLM, and initial pose $\mathbf{T}_{M,W}$

Output: 6DOF pose at every imaging time $\hat{\mathbf{T}}_{W,k}$

Thread 1: // Visual tracking thread

for *image i in* $\mathcal{I}$ **do**

- Pre-processing
- Feature extractions
- Feature tracking across images
- Feature tracking with the sparse feature map
- $\hat{\mathbf{T}}_{W,i}$ = PnP(tracked features, features in map)
- **if** *isKeyframe(i)* **then**
  - Inert_into(Thread 2)
  - Inert_into(Thread 3)

Thread 2: // Local mapping thread

Initialize new features to sparse feature map

Update the covisible graph between keyframes

/* $\check{\mathbf{T}}_{W,r}$ is the registration from Thread 3. */

Posegraph_optimization($\hat{\mathbf{T}}_{W,l}|_{l\in locaKeyframes}, \check{\mathbf{T}}_{W,r}$)

Local bundle adjustment with sparse feature refinement by PLM

Provide initial pose for the registration in Thread 3

Thread 3: // PLM-based visual localization thread

Insert into keyframe buffer

Semi-dense reconstruction from a sliding window of keyframes

Pre-processing the semi-dense point cloud

$\check{\mathbf{T}}_{M,r}$ = ProW-NDT_Registration()

if *isGoodRegistration()* **then**

- /* W, M denote the world frame of VO and PLM frame, respectively. */
- **if** $\neg$ *isAligned(W, M)* **then**
  - Align W, M based on the registration result; // Transform the two frame into the same frame.
- set_prior_pose($\check{\mathbf{T}}_{W,r}$)

3.1 | Visual tracking

As in the state-of-the-art feature-based VO algorithms (Cvišić et al., 2018; Mur-Artal & Tardós, 2017a; Qin & Shen, 2017), once live images are available from the stereo camera, we extract sparse features and describe them with distinct descriptors. Features are tracked in the latest frame or the local visual feature map for real-time motion estimation. With a set of 3D to 2D correspondences, the camera pose is computed based on PnP (Lepetit et al., 2009) with RANSAC-based outlier rejection.

3.2 | Local mapping

In this thread, we build a local sparse feature-based visual map that consists of the distinct sparse visual features, which is used by visual tracking. If the stereo match of a feature is found, the feature will be initialized immediately; if not, it will be triangulated later using subsequent images. When a new keyframe is determined, it will be inserted into both the local mapping thread and the PLM-based visual localization thread. The visual features within this new keyframe are initialized and associated with other features that already exist in the feature map. A covisibility graph encoding the data associations of sparse features is also maintained in the feature map. From the PLM-based visual localization thread, we will have the 6DOF pose estimates of some selected keyframes by registration. Pose graph optimization will adjust the poses of local keyframes by fusing the VO and registration results. Note that we refer to the pipeline without using the PLM as VO, which includes the visual tracking thread and local mapping thread without structure constraints. Since the PLM is typically more accurate in metric space, we refine the local visual feature map by a local portion of the LiDAR map to enforce local planar structure constraints.

3.3 | Prior LiDAR map-based visual localization

There are two main tasks in the PLM-based visual localization thread: (a) semidense visual map reconstruction and (b) registration with the PLM. For real-time capacity, we propose to perform a fast semidense

map reconstruction from keyframes only. The keyframes, inserted into the PLM-based visual localization thread, are kept in a keyframe buffer (see Figure 3), which only stores a fixed number of latest keyframes, while the oldest keyframe is dropped. The recovered map from one single keyframe usually contains only few structures, whereas a map with enough structures is required to benefit the registration. Therefore, we collect several keyframes into a sliding window, and after recovering the point cloud of each keyframe, we aggregate the point clouds from these keyframes to form an integrated local map as a semidense visual map. Among all the keyframes in the sliding window, a reference keyframe is determined; we select the second latest as the reference keyframe in our implementation. During the aggregation, point cloud from every single keyframe will be transformed into this reference keyframe by relative-pose estimations from VO. And the 6DOF pose of the reference keyframe with respect to PLM will be obtained from registration and set as the prior in the pose graph. There are several reasons for selecting the second latest keyframe as the reference keyframe. First, the pose estimation of the newest keyframe is not optimized, which is not able to give a good initial guess to the registration if it is selected as the reference. Second, although the oldest keyframe has been optimized several times and has an accurate pose estimation, it is the farthest from the current frame. The pose correction to the oldest keyframe after registration is not able to correct the pose of the latest frame in time. If the drift of the current frame is not corrected in time, the subsequent registrations may be not able to converge to good values.

The number of keyframes n, in the sliding window, will fluctuate in a small range (such as in our experiments, $n \in \{6, 10\}$ as we will show later), according to the number of covisible sparse features among consecutive keyframes. The key insights behind this can be described as follows. We try to include as many keyframes into the local semidense visual map, which is able to capture sufficient structures of the scene and usually benefit the registration. Moreover, a small number of covisible sparse features from consecutive keyframes usually lead to poor relative-pose estimation. This often occurs when sudden changes happen to the robot's orientation or illumination conditions of the environment. In such cases, we will reduce the size of the sliding window to suppress the errors of the local semidense visual map, which are raised from the poor relative-pose estimates. If the number of covisible features between any two consecutive keyframes in the sliding window is smaller than a given threshold, we will set a smaller size to the "sliding window." After the reconstruction, a local semidense visual map in the reference keyframe will be matched with the PLM. Sometimes, this thread cannot process all the keyframes in time, because of the fast keyframe insertion or the time-consuming registration. Some keyframes will be ignored, and will not be reconstructed. If the number of new-coming keyframes inserted into the keyframe buffer, during the reconstruction and registration, exceeds the "sliding window" size limit, the exceeding keyframes will be truncated and will not be used for the local semidense visual map reconstruction. Although the exceeding keyframes are ignored in this thread, the keyframes in the sliding window for reconstruction and registration are always consecutive in time, thus leading to a consistent and even local reconstruction. Furthermore, in this thread, the semidense visual map is reconstructed with uncertainty under Gaussian distribution, which will be taken into consideration in the registration.

In the registration, the semidense visual map is treated as source point cloud, while the PLM as target point cloud. Both PLM and semidense visual map will be preprocessed before registration: semidense visual map should be downsampled for fast registration and filtered for noises repression, while prior LiDAR map will be divided into multiple cells in preparation for registration. During the down-sample of the semidense visual map, all points inside a cube will be replaced by the mean of those points. In our experiments, the length of the cube is set as 0.25 m.

Although not all keyframes will be assigned with prior poses from the registration, it is enough to suppress the drift of online VO. Since the proposed approach is not designed for global localization, a rough initial guess of the camera pose in the prior LiDAR map, when starting the system, should be provided (e.g., by GPS). If a successful registration is performed, we can get the camera pose with respect to prior LiDAR map, and thus be able to

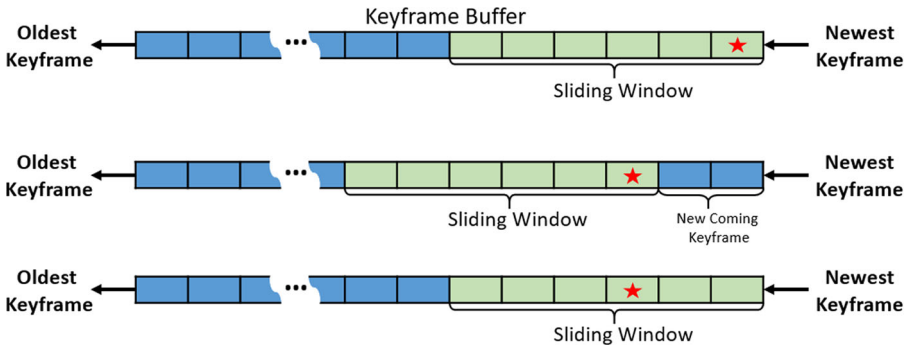

FIGURE 3 Illustration of keyframe buffer used to store the inserted keyframes. The keyframes in the sliding window are used for local semidense reconstruction. A specific number (6, in this figure) of new keyframes in the sliding window are used for reconstruction; during this reconstruction and registration, another two new keyframes are inserted to the buffer, shown in the middle figure; in the bottom, the window slides to the front of the keyframe buffer after finishing registration [Color figure can be viewed at wileyonlinelibrary.com]

know the rigid transformation between the frame of VO and the frame of prior LiDAR map. For simplicity, we transform the VO into the frame of LiDAR map, and thus they share the same frame. From now on, we set the frame of prior LiDAR map as the global frame. The pose of reference keyframe obtained from VO, will serve as the initial guess to the registration. A refined pose estimation, will be obtained from the registration, and passed into a pose graph in local mapping thread to adjust the poses of keyframes (see Figure 2). After the registration and pose graph optimization, the local visual feature map will be aligned with the LiDAR map, and we will refine the positions of the sparse 3D visual features by their surrounding LiDAR point cloud.

4 | VISUAL PROCESSING

In this section, we explain in detail the four main visual processing modules in the proposed visual localization system, including semidense reconstruction, feature refinement, multimodal registration, and optimization.

4.1 | Semidense visual reconstruction

As real-time localization is the primary goal of this study, instead of attempting to reconstruct the environment perfectly, we develop a simple but effective semidense visual reconstruction algorithm that involves two main steps: (a) estimate depth and point cloud of every keyframe in the sliding window, and (b) aggregate these keyframes' point clouds based on the VO results to form an integrated local 3D semidense map.

As a common practice, we perform stereo block matching to estimate the depth of each keyframe, which minimizes the sum of squared distance error over image patches to compute the disparities. Specifically, we minimize the photometric error between the left and right image with respect to the disparity d for point $\mathbf{p} = \{x, y, z\} \in \mathbb{R}^3$:

$$\arg\min_{d}\left[\frac{1}{\sigma_{\text{int}}^2}\| r_{\mathbf{I}} \|^2\right], \quad \text{where } r_{\mathbf{I}} = \mathbf{I}_L(u, v) - \mathbf{I}_R(u - d, v_R), \tag{1}$$

where $\mathbf{u} = (u, v)$ and $\mathbf{u}_R = (u_R, v_R)$ represents the measurement pairs of point $\mathbf{p}$ in the left and right image, $u_R = u - d$, and $\mathbf{I}(\cdot)$ represents the image intensity with variance σ_{int}^2. Thus, the variance of the disparity d can be computed based on covariance propagation:

$$\sigma_d^2 = \left(\left(\frac{\partial r_{\mathbf{I}}}{\partial d}\right)^T \frac{1}{\sigma_{\text{int}}^2}\left(\frac{\partial r_{\mathbf{I}}}{\partial d}\right)\right)^{-1} = \frac{2\sigma_{\text{int}}^2}{\mathbf{g}_{\mathbf{I}_R}^2}, \tag{2}$$

where $\mathbf{g}_{\mathbf{I}_R}$ represents the image intensity gradient in the right image. As a result, the covariance of $[u, v, d]^T$ is given by $\mathbf{\Sigma}_{\mathbf{I}} = \begin{bmatrix} \sigma_{\text{pixel}}^2 & 0 & 0 \\ 0 & \sigma_{\text{pixel}}^2 & 0 \\ 0 & 0 & \sigma_d^2 \end{bmatrix}$, where σ_{pixel}^2 is the known variance of image pixel noise. After computing the disparity, we recover the 3D point $\mathbf{p}$ with the camera intrinsic parameters (such as focal length f, optical centers (c_u, c_v)) and stereo baseline b as follows:

$$\begin{cases} x = \dfrac{(u - c_u)b}{d}, \\ y = \dfrac{(v - c_v)b}{d}, \\ z = \dfrac{fb}{d}. \end{cases} \tag{3}$$

We also propagate the covariance of $(u, v, d)^T$ to obtain the covariance of the 3D point $\mathbf{p}$:

$$\mathbf{\Sigma}_{\mathbf{p}} = \mathbf{J}\mathbf{\Sigma}_{\mathbf{I}}\mathbf{J}^T, \quad \mathbf{J} = \frac{b}{d^2}\begin{bmatrix} d & 0 & -(u - c_u) \\ 0 & d & -(v - c_v) \\ 0 & 0 & -f \end{bmatrix}. \tag{4}$$

Since the depth estimate of $\mathbf{p}$ is uncertain, as in Mur-Artal and Tardós (2015), we perform the depth outlier removal and smoothing to suppress the noise by its neighbor pixels.

After we reconstruct the point cloud for every single keyframe in the sliding window, we now aggregate them into a consistent local 3D point-cloud map. By choosing one keyframe in the window as the reference frame, we transform other keyframes' point clouds into this reference frame and accordingly propagate the covariance. Specifically, $\mathcal{F} = \{F_a, F_{a+1}, ..., F_r, ..., F_{b-1}, F_b\}, r \in [a, b]$ represents the set of keyframes in the sliding window, and F_r is the reference frame. Let $\mathbf{T}_{r,a}(\theta_{r,a}) = \begin{bmatrix} \mathbf{C}_{r,a} & \mathbf{t}_{r,a} \\ \mathbf{0} & 1 \end{bmatrix} \in SE(3)$ represent the rigid transformation from F_a to F_r, $\mathbf{C}_{r,a}$ is the rotation matrix, and $\theta_{r,a} = [\boldsymbol{\omega}_{r,a}^T \ \ \mathbf{t}_{r,a}^T]^T \in se(3) \in \mathbb{R}^6$ is the Lie algebra associated with $SE(3)$, and $\boldsymbol{\omega}$ and $\mathbf{t}$ represent the rotation and translation 3D vector, respectively. We then directly apply this rigid transformation to transform a point ${}^{\{a\}}\mathbf{p}$ with covariance $\text{cov}({}^{\{a\}}\mathbf{p})$ in keyframe F_a to reference frame F_r, that is $\begin{bmatrix} {}^{\{r\}}\mathbf{p} \\ 1 \end{bmatrix} = \mathbf{T}_{r,a}(\theta_{r,a})\begin{bmatrix} {}^{\{a\}}\mathbf{p} \\ 1 \end{bmatrix}$.

To compute the covariance of the aggregated point cloud, we first denote the error states of ${}^{\{r\}}\mathbf{p}$, ${}^{\{a\}}\mathbf{p}$ and $\boldsymbol{\theta}$ by ${}^{\{r\}}\delta\mathbf{p}$, ${}^{\{a\}}\delta\mathbf{p}$, and $\delta\boldsymbol{\theta}$, respectively. During update, the 6DOF pose estimate is updated by $\mathbf{T}_{r,a}^{+} = \exp(\delta\theta_{r,a}{}^{\wedge})\mathbf{T}_{r,a}^{-}$, where $\mathbf{T}_{r,a}^{-}$ and $\mathbf{T}_{r,a}^{+}$ represents the pose before and after update, where the wedge operation $\wedge$ is given by Barfoot (2017): $\delta\theta_{r,a}^{\wedge} = \begin{bmatrix} \delta\boldsymbol{\omega}_{r,a} \\ \delta\mathbf{t}_{r,a} \end{bmatrix}^{\wedge} = \begin{bmatrix} \lfloor\delta\boldsymbol{\omega}_{r,a}\rfloor_{\times} & \delta\mathbf{t}_{r,\mathbf{a}} \\ \mathbf{0}^T & 0 \end{bmatrix} \in \mathbb{R}^{4\times4}$, and $\lfloor . \rfloor_{\times}$ denotes the skew-symmetric matrix of a 3D vector. The update operation for point is simply additive: ${}^{\{r\}}\mathbf{p}^{+} = {}^{\{r\}}\delta\mathbf{p} + {}^{\{r\}}\mathbf{p}^{-}$ and the covariance of ${}^{\{r\}}\mathbf{p}$ is updated as follows:

$$\text{cov}({}^{\{r\}}\delta\mathbf{p}) = \frac{\partial^{\{r\}}\delta\mathbf{p}}{\partial^{\{a\}}\delta\mathbf{p}}\text{cov}({}^{\{a\}}\mathbf{p})\left(\frac{\partial^{\{r\}}\delta\mathbf{p}}{\partial^{\{a\}}\delta\mathbf{p}}\right)^T + \frac{\partial^{\{r\}}\delta\mathbf{p}}{\partial\delta\theta_{r,a}}\text{cov}(\theta_{r,a})\left(\frac{\partial^{\{r\}}\delta\mathbf{p}}{\partial\delta\theta_{r,a}}\right)^T, \tag{5}$$

where

$$\frac{\partial^{\{r\}}\delta\mathbf{p}}{\partial^{\{a\}}\delta\mathbf{p}} = \mathbf{C}_{r,a}, \tag{6}$$

$$\frac{\partial^{\{r\}}\mathbf{p}}{\partial\delta\theta_{r,a}} = \left[-\lfloor{}^{\{r\}}\delta\mathbf{p}\rfloor_{\times}\mathbf{I}_{3\times3}\right]. \tag{7}$$

4.2 | Multimodal point cloud registration

We now register the aggregated point cloud (i.e., semidense visual map) with the PLM to provide accurate localization solution with bounded navigation errors. The workflow of registration is shown in Figure 4. For multimodal point cloud registration, we adapt the standard NDT method to compensate for the two different modalities of point clouds to be registered.

In the standard NDT (Magnusson, 2009), instead of directly matching individual points in the point cloud, the point cloud is represented by a set of Gaussian distributions with sample mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$:

$$\boldsymbol{\mu} = \frac{1}{m}\sum_{k=1}^{m}\mathbf{p}_i, \tag{8}$$

$$\mathbf{n} = [\mathbf{p}_1 - \boldsymbol{\mu} \ldots \mathbf{p}_m - \boldsymbol{\mu}], \tag{9}$$

$$\boldsymbol{\Sigma} = \frac{1}{m-1}\mathbf{n}\mathbf{n}^{\top}. \tag{10}$$

In this way, NDT is expected to be insensitive to uneven sample distributions and reduce the memory requirements for large-scale maps by efficiently compresses the original point cloud data. Also, as there is no explicit nearest neighbor search in the NDT, it is computationally efficient and can converge faster from a wider range of initial pose estimates than the ICP (Huhle, Magnusson, Straßer, & Lilienthal, 2008; Magnusson, Nuchter, Lorken, Lilienthal, & Hertzberg, 2009). Two variants of NDT are widely used: (a) point-to-distribution (P2D) variant of NDT (Magnusson, 2009), which formulates the registration of a new scan as a problem of fitting the points to the distribution in the target point cloud; and (b) distribution-to-distribution (D2D) NDT (Stoyanov, Magnusson, Andreasson, & Lilienthal, 2012), which matches the distributions in the source and target and is significantly faster but less robust than P2D variant (Magnusson, Vaskevicius, Stoyanov, Pathak, & Birk, 2015). For this reason, we employ the P2D as the baseline NDT for our multimodal point cloud registration approach, which is experimentally found to perform better than the D2D NDT.

Specifically, given the Gaussian pdf $p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{3/2}(\det(\boldsymbol{\Sigma}))^{1/2}}\exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{T}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}))$, in the P2D-NDT, the optimal pose estimate $\mathbf{T}(\theta)$, is found by maximizing the following likelihood function:

$$\Psi(\mathcal{P}_{src}, \mathcal{M}_{tar}, \theta) = \prod_{i=1}^{n_{src}} p_i(\mathbf{p}_{src\_i}), \tag{11}$$

$$p_i(\mathbf{p}_{src\_i}) = (c_i p(\mathbf{T}(\theta\mathbf{p}_{src\_i}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) + c_{io}p_{io}), \tag{12}$$

where $\mathcal{P}_{src} = \{\mathbf{p}_{src\_i}\}$, $i \in \{1, 2, \ldots, n_{src}\}$, is the set of points in source point cloud; $\mathcal{M}_{tar}$ is the set of Gaussian distributions created from the target point cloud; $\mathcal{N}(\boldsymbol{\mu}_j, \Sigma_j)$ is the closest Gaussian component to point $\mathbf{p}_{src\_i}$ among all Gaussian components in $\mathcal{M}_{tar}$. p_{io} represents the expected ratio of outliers. The normalization constants c_i, c_{io} can be determined by requiring that the probability mass of $_{pi}$ $(\mathbf{p}_{src\_i})$ equals one within the space spanned by a cell.

Uncertainty is leveraged when registering the noisy visual point cloud to the LiDAR point cloud. Intuitively, a point close to the reference keyframe in the source point cloud has smaller uncertainty, and we expect a better alignment with the LiDAR map than those far away from the reference, resulting in better pose estimates of the reference keyframes than treating all points equally. Unlike the conventional P2D-NDT, which assigns simply the same outlier ratio to every point, we derive the expression of this outlier ratio of every point in a probabilistic way, and this outlier ratio can be regarded as a weight when performing registration. We thus term this method by *Probabilistically Weighted NDT (ProW-NDT)*. In particular, the probability that a point is truly matched with a cell should be proportional to the probability mass of the Gaussian distribution of this point within this cell. We approximate the outlier ratio of point $\mathbf{p}_{src\_i}$ as follows:

$$\mathbf{v}_{\sigma} = \begin{bmatrix} v_{\sigma 1} \\ v_{\sigma 2} \\ v_{\sigma 3} \end{bmatrix} = \mathrm{Diag}(\boldsymbol{\Sigma}_i)^{-1/2}\mathbf{v}, \tag{13a}$$

$$a_i = 1.0 - \int_D p(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)d\mathbf{x} \simeq 1.0 - \prod_{k=1}^{3}\mathrm{erf}\left(\frac{\sqrt{2}\,v_{\sigma k}}{2}\right), \tag{13b}$$

$$p_{io} = \begin{cases} 0.35 & (a_i \le 0.35), \\ a_i & (0.35 < a_i < 0.9), \\ 0.9 & (0.9 \le a_i), \end{cases} \tag{13c}$$

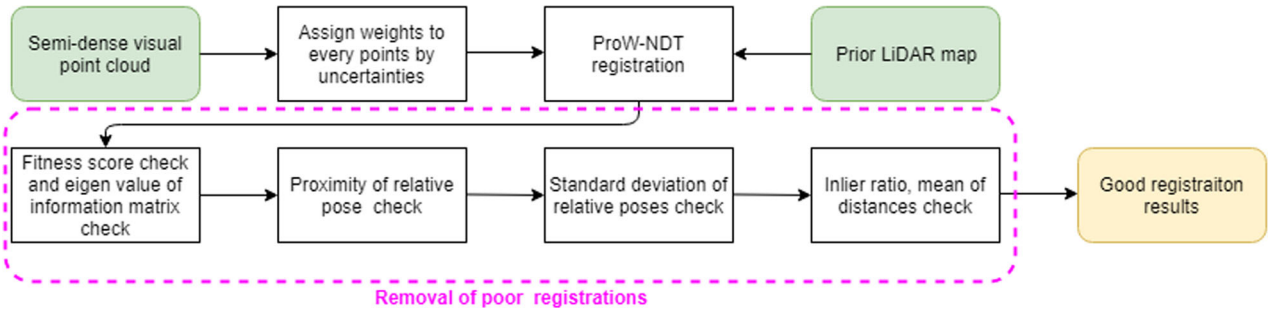

FIGURE 4 Flow chart of registration. Points in the semidense visual point cloud are assigned with different weights by uncertainties (see Equation (13)); Then the point cloud registered with the prior LiDAR map; The poor registration results will be removed by designed criteria described in Section 4.2.1. Only the good registration results will be used in the later pose graph optimization [Color figure can be viewed at wileyonlinelibrary.com]

where $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ represents the uncertainty of $\mathbf{p}_{src\_i}$ in the visual point cloud generated in the process of stereo semidense reconstruction. erf(·) denotes the Gauss error function. In (13a), Diag (.) returns the matrix only keeping the diagonal items of the original matrix. We omit the off-diagonal items of the covariance matrix for simplicity in the integral of multivariate Gaussian. The domain of integral D is $[\boldsymbol{\mu}_i - \mathbf{v}, \boldsymbol{\mu}_i + \mathbf{v}]$, which spans the cubic cell centered at $\boldsymbol{\mu}_i$ and with size of $2\mathbf{v}$. The piecewise function for p_{io} in (13c) accounts for the unmodeled noise. Applying a Gaussian $\tilde{\mathbf{p}}_i(\mathbf{p}_{src\_i})$ to approximate the log-likelihood of original Gaussian ${}_{pi}(\mathbf{p}_{src\_i})$, the final score function of NDT is given by Magnusson (2009):

$$f(\mathcal{P}_{src}, \mathcal{M}_{tar}, \theta) = \sum_{i=1}^{n_{src}} d_1 \exp\left(-\frac{d_2}{2}(\mathbf{T}(\theta)\mathbf{p}_{src\_i} - \boldsymbol{\mu}_j)^T(\boldsymbol{\Sigma}_j)^{-1}(\mathbf{T}(\theta)\mathbf{p}_{src\_i} - \boldsymbol{\mu}_j)\right), \quad (14)$$

where the constants d_1 and d_2, are computed in the approximation by:

$$\begin{aligned} d_1 &= -\log(c_i + c_{io}) + \log(c_{io}), \\ d_2 &= -2\log((-\log(c_i \exp(-1/2) + c_{io}) + \log(c_{io}))/d_1), \end{aligned} \quad (15)$$

The optimal pose estimate $\boldsymbol{\theta}$ can be obtained by minimizing the above score function in Equation (14).

4.2.1 | Remove poor registration results

As poor registration is inevitable in practice, we advocate the following criteria to filter out bad results:

(i) We evaluate the fitness score and information matrix of ProW-NDT registration. The Hessian matrix $\mathbf{H}_{reg}$ of score function (14) can be computed at the final pose estimation $\boldsymbol{\theta}$. This Hessian matrix can be used as an estimation of the information matrix. We set h_λ to be proportional to the minimum eigenvalue of negative Hessian matrix $\mathbf{H}_{reg}$,

$$h_\lambda \propto \min\{Eigen(-\boldsymbol{H}_{reg})\}, \quad (16)$$

h_λ is usually large when a correct registration is obtained from ProW-NDT, and vice versa, a bad registration always corresponds to a small value of h_λ. As for the fitness score, a very large fitness score usually implies a poor registration. However, this clue often fails to reflect the true situation, and an accurate pose estimation might correspond to a high score (Magnusson, 2009). Only when the score stays within a threshold, and h_λ is bigger than a certain value, we will consider to add the registration result to the pose graph.

(ii) We are able to get the relative-pose between two keyframes from VO or NDT registration only. The relative pose estimated by VO between two nearby keyframes is usually with considerable accuracy. If the pose estimation from registration is reasonable, the relative pose obtained from it should be close to that obtained form VO. Therefore, we will reject the registration result of the keyframe when its relative pose with respect to the last registered keyframe has a large difference with that got from VO.

(iii) VO occasionally fails to provide a good pose estimation due to aggressive motion or great illumination change, while ProW-NDT relying on structural information can give a good pose estimation. However, the criterion (ii) probably rejects the good pose estimation from ProW-NDT. So we adopt another strategy to avoid these cases. If the standard deviation of several relative poses from ProW-NDT is smaller than the one obtained from VO, the pose estimation from ProW-NDT will be added to the pose graph by force. This is reasonable, because in a short period of time, relative poses will not strongly fluctuate with a large standard deviation. As the trajectory of robot is always smooth, the estimation from ProW-NDT with mild fluctuation is acceptable in most cases.

(iv) Like most registration methods, the inlier ratio of the source point cloud, and the mean distances of the inlier correspondences can also reflect the circumstance of the registration result. We also take these criteria into consideration.

For example, in Figure 5, the norm of absolute translation error of the initial guess from VO and the registration result are shown, when we run the proposed method on sequence 05 of KITTI data set (Geiger et al., 2012). It is obvious that the registration result is with smaller errors than the initial guess in most cases. In this run, 873 keyframes are created totally, 331 ProW-NDT are performed, and three registration results are rejected by the above criteria.

4.3 | Structure-constrained visual feature refinement

While the proposed ProW-NDT is robust to some extent, it still can be affected by the initial guess provided by VO. To improve the VO and thus the ProW-NDT, we propose to refine the sparse visual feature map by leveraging the structural constraints inferred from the prior LiDAR map. To this end, we formulate the following maximum a posteriori (MAP) estimation problem:

$$\max p(\mathcal{P}|\mathcal{U}, \Theta, Q) \propto p(\mathcal{U}|\mathcal{P}, \Theta, Q)p(\mathcal{P}|\Theta, Q) = \prod_{k=1}^{|\mathcal{P}|} p(\mathcal{U}_k|\mathbf{p}_k, \Theta, Q)p(\mathbf{p}_k|Q), \quad (17)$$

where $\mathcal{U} = \{\mathcal{U}_1 \cup \mathcal{U}_2 \cup \cdots \cup \mathcal{U}_p\}$ is the set of visual measurements, and $\mathcal{U}_k = \{\mathbf{u}_{k,1}, \mathbf{u}_{k,2}, ..., \mathbf{u}_k, |\mathcal{U}_k|\}$ is the observation set of one 3D feature $\mathbf{p}_k$, since a feature may be observed in different images; $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_{|\mathcal{P}|}\}$ is the set of 3D sparse visual features that are initially estimated in the local mapping thread; $Q = \{\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_{|Q|}\}$ is the set of points in the prior LiDAR map; $\Theta = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_{|\Theta|}\}$ is the set of camera poses $\mathbf{T}_i(\boldsymbol{\theta}_i) \in SE(3)$ expressed in the global frame. Note that as both sparse visual features and the prior map are expressed in the same global frame, we have $p(\mathcal{P}|\Theta, Q) = p(\mathcal{P}|Q)$. The common assumption that visual measurements are independent leads to

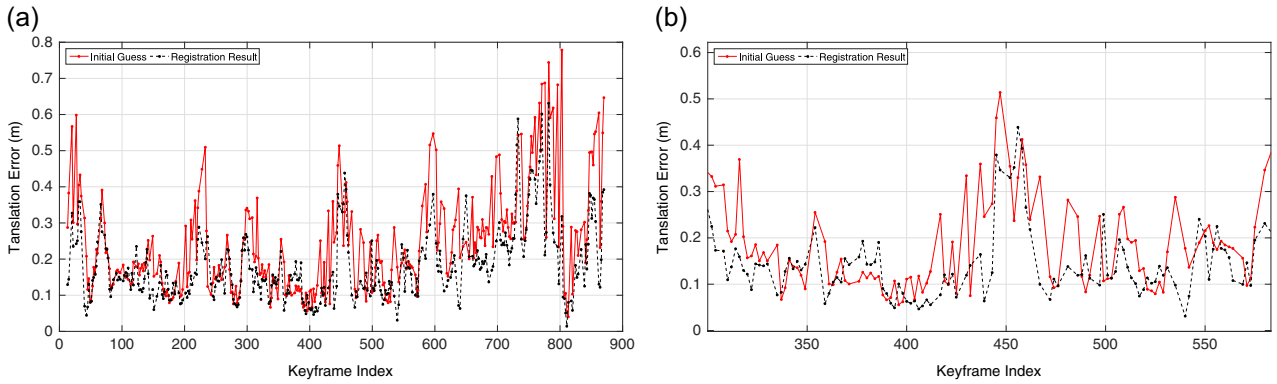

FIGURE 5 An example of the norm of absolute translation error before and after registration, which is performed on Sequence 05 of KITTI data set. (a) Norm of translation error; (b) A close view [Color figure can be viewed at wileyonlinelibrary.com]

$p(\mathcal{U}_k|\mathbf{p}_k,\Theta,Q)=\prod_{i=1}^{|\mathcal{U}_k|}p(\mathbf{u}_{k,i}|\mathbf{p}_k,\theta_i,Q)$, where $p(\mathbf{u}_{k,i}|\mathbf{p}_k,\theta_i,Q):=\mathcal{N}(\mathbf{u}_{k,i};\pi(\mathbf{T}_i(\theta_i)\mathbf{p}_k),\mathbf{\Sigma}_{k,i})$ with the covariance of observation $\mathbf{u}_{k,i}$ denoted by $\mathbf{\Sigma}_{k,i}$, which is related to the image pyramid level where it is observed. With that, as a common practice, maximizing the measurement likelihood $p(\mathcal{U}_k|\mathbf{p}_k,\Theta,Q)$ is equivalent to minimize the reprojection errors:

$$\max p(\mathcal{U}_k|\mathbf{p}_k,\Theta,Q)\Leftrightarrow \tag{18}$$

$$\min \mathbf{e}_{proj\_k}=\sum_{i=1}^{|\mathcal{U}_k|}\rho\left(\mathbf{r}_{proj\_k,i}^T\mathbf{\Sigma}_{k,i}^{-1}\mathbf{r}_{proj\_k,i}\right),\quad \text{with } \mathbf{r}_{proj\_k,i}$$
$$=\mathbf{u}_{k,i}-\pi(\mathbf{T}_i(\theta_i)\mathbf{p}_k). \tag{19}$$

Note that in our implementation, the robust Huber kernel (Z. Zhang, 1995) is used to compensate for possible measurement outliers.

Ideally, we would like to have a sparse visual feature $\mathbf{p}_k$ exactly matched to the point $\mathbf{q}_k$ in the prior LiDAR map, which is clearly not the case in practice primarily due to the sparseness of LiDAR point clouds; even if it is the case, it is yet impossible to find the correct correspondence because the LiDAR map only contains 3D positions of the points. To address this issue, we enforce structure constraint that penalizes the misalignment between the 3D visual feature and the LiDAR point cloud. Specifically, the proposed approach creates a surfel $\mathbf{y}_l$ ($\mathbf{y}_l\in\{\mathbf{y}_1,\mathbf{y}_2,...,\mathbf{y}_M\}\subseteq Q$) to summarize the LiDAR point cloud around the visual feature $\mathbf{q}_k$ by its sample mean $\boldsymbol{\mu}_{yl}$ and covariance $\mathbf{\Sigma}_{yl}$, from which we build the structural error term:

$$p(\mathbf{p}_k|Q)=p(\mathbf{p}_k|\mathbf{y}_l)=\mathcal{N}(\mathbf{p}_k;\boldsymbol{\mu}_{yl},\mathbf{\Sigma}_{yl})\Rightarrow \tag{20}$$

$$-\log(p(\mathbf{p}_k|\mathbf{y}_l))=\eta+\frac{1}{2}(\mathbf{p}_k-\boldsymbol{\mu}_{yl})^T\mathbf{\Sigma}_{yl}^{-1}(\mathbf{p}_k-\boldsymbol{\mu}_{yl})\Rightarrow \tag{21}$$

$$\mathbf{e}_{struct\_k}=\mathbf{r}_{struct\_k}^T\mathbf{\Sigma}_{yl}^{-1}\mathbf{r}_{struct\_k},\quad \text{with } \mathbf{r}_{struct\_k}=\mathbf{p}_k-\boldsymbol{\mu}_{yl}, \tag{22}$$

which is leveraged to refine the visual features. When adjusting the visual features, the corresponding surfel of $\mathbf{p}_k$ will not change; even if it is a false positive association between $\mathbf{p}_k$ and $\mathbf{y}_l$, robust kernel and outlier rejection will be employed. Many factors may deteriorate this data association, such as registration error, poor 3D visual feature map, inherited error from an imperfect LiDAR map. As data association is determined before optimization, the structural error should be robust to noisy data association. To tradeoff the computational cost and accuracy, inspired by the fast global registration (Zhou, Park, & Koltun, 2016), we use the Black-Rangarajan duality between robust estimation and line process (Black & Anandan, 1996; Black & Rangarajan, 1996) to deal with bad data associations. That is, assuming $c_{k,l}$ is a line process over the correspondence of $\mathbf{p}_k$ and $\mathbf{y}_l$, we jointly optimize $\mathbf{p}_k$ and $c_{k,l}$ by instead using the following structural cost function (see (22)):

$$\mathbf{e}_{struct\_k}=c_{k,l}\mathbf{r}_{struct\_k}^T\mathbf{\Sigma}_{yl}^{-1}\mathbf{r}_{struct\_k}+\Phi(c_{k,l}), \tag{23}$$

$$\Phi(c_{k,l})=\alpha(\sqrt{c_{k,l}}-1)^2, \tag{24}$$

where $c_{k,l}$ can be considered as a soft data association, ranging over [0, 1] and modeling the validity of the data association; $\Phi(c_{k,l})$ encodes the belief of a genuine correspondence, and minimizing this error term ($c_{k,l}\to 1$) believes this correspondence as the true positive. If it is a false positive correspondence that will introduce a spurious constraint corrupting the optimization, $\Phi(c_{k,l})$ is able to devitalize this correspondence by pushing ($c_{k,l}\to 0$). It is important to note that if one eigenvalue of the covariance of the surfel $\mathbf{\Sigma}_{yl}$ is near zero, the structural error becomes the distance of point to plane. Thus, when Σ_{yl} is nearly singular, we will modify it by enlarging its small eigenvalues as mentioned in Magnusson (2009). The scalar α balances the strengths of alignment term and prior term. In our implementation, we begin with a large α, and then decrease it gradually during the optimization.

At this point, we are ready to leverage the prior LiDAR map to refine the visual features by minimizing both the reprojection error and the structural error:

$$\{\mathcal{P}^*,\Theta^*\}=\arg\min_{\mathcal{P},\Theta}\sum_{k=1}^{|\mathcal{P}|}(\mathbf{e}_{reproj\_k}+w\mathbf{e}_{struct\_k}), \tag{25}$$

where w denoting the weight parameter to balance between structural error and reprojection error varies for different correspondences:

$$w\propto\frac{1}{\sqrt{\lambda_{l1}}}, \tag{26}$$

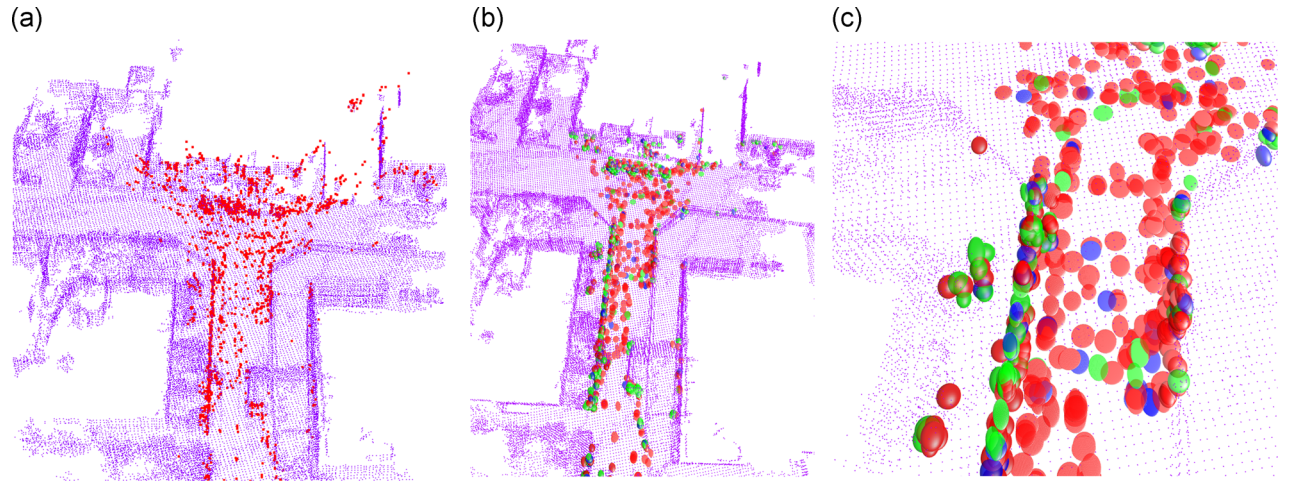

FIGURE 6 Surfels are extracted around sparse visual features from the prior LiDAR map. They are used to correct the positions of the 3D Visual features. Extracted surfels are color-coded by normal directions (a) Sparse visual features in LiDAR map; (b) Surfels from the prior LiDAR map; (c) A close view of the extracted surfels [Color figure can be viewed at wileyonlinelibrary.com]

where λ_{l1} is the smallest eigenvalue of Σ_{yl} and used to penalize nonplane surfel. As an example shown in Figure 6, we refine the sparse feature by it is surrounding surfel, the surfel summarizes the distribution of the point cloud in a spherical field, centered at the sparse feature point with a radius of 0.5 meters. If less than four points in the spherical field are founded, we will not create a surfel to refine this sparse feature. As the planar surfel is usually with less noises, so we set a bigger weight to it.

4.4 | Pose graph optimization

We construct a pose graph shown in Figure 7 to fuse the registration and VO results, in which the registration results are added as the prior poses of reference keyframes. Since we have maintained the covisible graph in the local mapping thread, local keyframes are found out by checking the number of covisible visual features at the current keyframe, and a keyframe with sufficient covisible features is classified as a local keyframe. The keyframe set in the pose graph is denoted by $\mathcal{F} = \{F_i, F_{i+1}, ..., F_{j-1}, F_j\}$, with pose estimates $\hat{\mathbf{T}}_{W,i}, \hat{\mathbf{T}}_{W,i+1}, ..., \hat{\mathbf{T}}_{W,j-1}, \hat{\mathbf{T}}_{W,j}$. The relative pose between two keyframes F_k, F_l, k, $l \in \{i, ..., j\}$, $k \neq l$ follows a Gaussian distribution with mean $\hat{\mathbf{T}}_{k,l} = \hat{\mathbf{T}}_{W,k}^{-1}\hat{\mathbf{T}}_{W,l}$ and covariance $\mathbf{\Sigma}_{k,l}$. The registration result $\breve{\mathbf{T}}_{W,r}$ serves as the prior of the reference keyframe, F_r. The residual of prior constraint $\mathbf{r}_r$ is also Gaussian with covariance $\mathbf{\Sigma}_r$, which is computed as the inverse of the Hessian matrix of NDT score function at final pose estimate (Magnusson, 2009). Therefore, the cost function can be formulated as follows:

$$e = \sum_r \rho_a(\mathbf{r}_r^T \mathbf{\Sigma}_r^{-1} \mathbf{r}_r) + \sum_{k<l, k\neq l} \rho_b\left(\mathbf{r}_{k,l}^T \mathbf{\Sigma}_{k,l}^{-1} \mathbf{r}_{k,l}\right), \tag{27a}$$

$$\text{with } \mathbf{r}_r = \log\left(\breve{\mathbf{T}}_{W,r}^{-1}\mathbf{T}_{W,r}\right), \quad \text{and} \quad \mathbf{r}_{k,l} = \log\left(\hat{\mathbf{T}}_{k,l}^{-1}\mathbf{T}_{k,l}\right), \tag{27b}$$

where ρ_a and ρ_b are the robust kernels. As our criteria for rejecting poor registration may occasionally fail, we further employ the dynamic covariance scaling (DCS) approach (Agarwal, Tipaldi, Spinello, Stachniss, & Burgard, 2013) to robustify our pose graph optimization, which is solved iteratively.

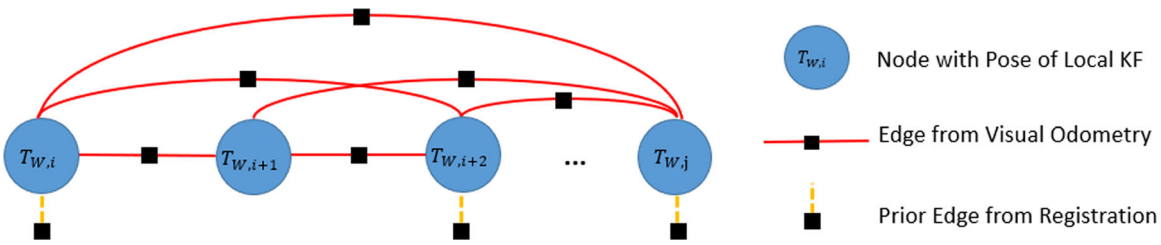

FIGURE 7 We construct the pose graph to fuse the registration and VO results. The blue circles represent the pose of the local keyframes. The red solid edges represent the relative-pose constraints from VO. The yellow dash edges denote the prior constraints obtained from ProW-NDT. VO, visual odometry [Color figure can be viewed at wileyonlinelibrary.com]

FIGURE 8 A bird's-eye view of the synthetic environment in Gazebo, the blue line marks the trajectory of the robot when collecting data [Color figure can be viewed at wileyonlinelibrary.com]

5 | EXPERIMENTAL VALIDATION

To validate the proposed real-time visual localization system, we test the proposed method on both simulated and real-world data sets. The simulated data set is collected in the ROS/Gazebo simulation engine (Ros, 2018). The real-world data sets include: (a) the well-known KITTI odometry benchmark (Geiger et al., 2012), which has data sequences collected under various road conditions such as rural, urban, and highway, (b) and the KAIST Urban data set (Jeong et al., 2019) collected in complex urban scenarios with abundant moving objects, and (c) the data collected using our own robot/sensors in the Zhejiang University (ZJU) campus. We implement the proposed visual localization system based on the state-of-the-art feature-based ORB-SLAM2 (Mur-Artal & Tardós, 2017a), and primarily adapt its visual tracking and local mapping thread, while we do not perform loop closing and global bundle adjustment. This is primarily due to the fact that loop closures are computationally intensive as shown in the timing consuming results in (Mur-Artal & Tardós, 2017a), and we have a prior map (although of a different modality) that is able to constrain localization drift. The performance metrics used in our evaluation include the averaged root mean square error (RMSE) of absolute trajectory error (ATE; Sturm, Engelhard, Endres, Burgard, & Cremers, 2012), the mean of absolute translation and rotation errors (as in the vein of Kim et al., 2018) to evaluate the localization accuracy and CPU runtime to evaluate the computational cost. All tests run on an Intel Core i7-7700k desktop computer with 16 GB RAM without GPU support. Videos fragments showing the performance of the proposed system on different data sets can be found on https://youtu.be/VnfVCu80TAc.

5.1 | Gazebo simulation

We create a synthetic environment in Gazebo, which includes houses, trees, vehicles, and pedestrians. Figure 8 shows a bird's-eye view of such a simulated environment. We use a Pioneer 3-DX robot (Pioneer, 2018) to collect data set in this synthetic environment, and the robot has a maximum speed of 2.5 m/s and is equipped with multiple sensors, such as a Velodyne LiDAR (Velodyne VLP-16, 2018), a stereo camera with a baseline of 0.4 m. The range measurements of the virtual LiDAR are corrupted by Gaussian noise with a standard deviation of 0.01 m, and the images obtained from the stereo camera are also injected Gaussian noise with a standard deviation of 0.01. The details related to the noise of the virtual sensor can be found in Gazebo Sensor Noise Model (2018). In this synthetic data set, we have the perfect parameters of all the sensors, such as the intrinsics and baseline of the stereo camera, and the extrinsics between the stereo camera and Velodyne, which facilitates our evaluations at different noise levels. As in real-world experiments, we may not have true calibration parameters of the stereo camera,

In particular, in this Gazebo simulation, we add 5, 10, and 15 pixel × m to the product of baseline and focal length, $f \times b$. These disturbances will cause different degrees of damage to the semidense reconstruction, and pose challenges to feature-based VO. Three different disturbances are applied to ground truth to generate three sequences. We run the proposed method six times on each sequence and compute the RMSE of ATE for evaluations, whose results are shown in Table 1. In this table, *Mean* represents the mean of the RMSE of ATE in six runs; *Mean/Traj* represents the *Mean* divided by the length of the trajectory; *Max* and *Min* represent the maximum and minimum RMSE of ATE in six runs. Better results are highlighted in bold. For fair comparisons, the proposed method and ORB-SLAM2 use exactly the same parameter settings for all the tests. As evident, the average, maximum and minimum ATE of the proposed method are all smaller than ORB-SLAM2 under the three different noise levels. When the deviation is small, the proposed method is only slightly better than ORB-SLAM2; while if the deviation is relatively large, our system performs significantly better, though the errors of both methods become larger. Figure 9 depicts the trajectories estimated by the proposed method and ORB-SLAM2 under different disturbances to calibration parameters. Clearly, in the case that we do not have accurate calibration parameters or good 3D reconstruction of the scene, the proposed visual localization with prior LiDAR map also outperforms the visual SLAM, which shows that our method is robust to the calibration disturbance.

TABLE 1 RMSE on Gazebo data set

	Proposed				ORB-SLAM2			
Noise level	Mean (m)	Mean/Traj (%)	Max (m)	Min (m)	Mean (m)	Mean/Traj (%)	Max (m)	Min (m)
5	**0.3962**	**0.1345**	**0.4150**	**0.3751**	0.6916	0.2348	0.7409	0.6636
10	**0.8338**	**0.2831**	**0.8649**	**0.7508**	1.8472	0.6272	4.5553	1.3713
15	**1.2249**	**0.4159**	**1.3265**	**1.1019**	2.0637	0.7007	2.0721	2.0519

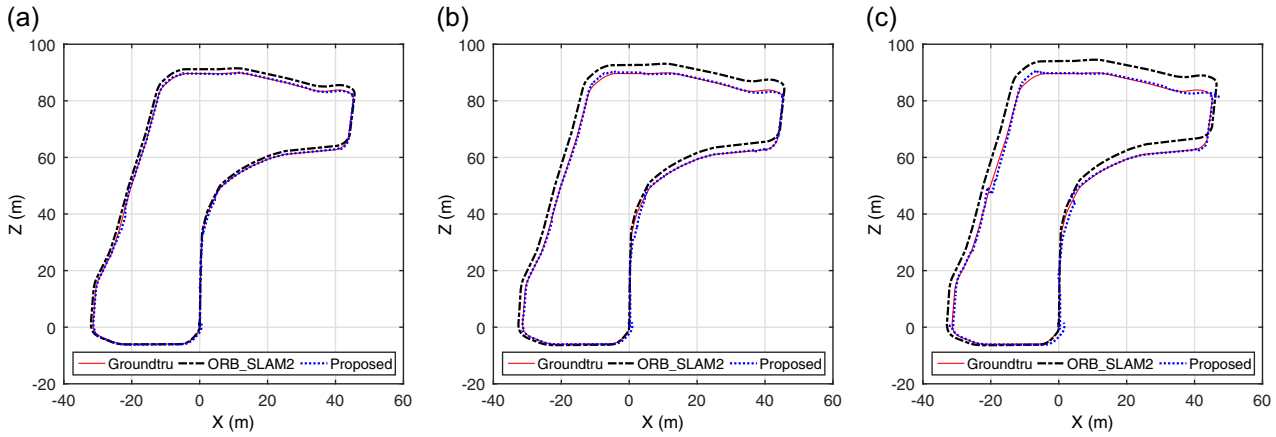

FIGURE 9 The trajectory estimated by the posed method and ORB-SLAM2 on Gazebo simulated data set under different disturbances. The red solid line indicates the trajectory of ground truth. Black dot-dash is estimated by ORB-SLAM2, and the blue dash is got from the proposed method. (a) On sequence with noise 5; (b) On sequence with noise 10; (c) On sequence with noise 15 [Color figure can be viewed at wileyonlinelibrary.com]

5.2 | Experimental evaluation on KITTI data set

The KITTI data set (Geiger et al., 2012) is recorded on a vehicle equipped with a variety of sensor modalities such as high-resolution color and grayscale stereo cameras, a Velodyne LiDAR and a high-precision GPS/IMU inertial navigation system. The odometry benchmarks are widely used for evaluating the accuracy of the localization algorithm, which is captured by driving around a midsize city, in rural areas and on highways. The ground truth poses are obtained from the GPS/IMU localization unit, and we use the provided ground truth to build up the prior LiDAR map by aggregating all LiDAR scans together. It should be noted that, while sequence 00–10 provide ground truth poses, the provided ground truth is not perfect in some sequences and inconsistencies appear in some loop closure areas, mainly in the vertical direction, which may introduce inaccuracies to our prior LiDAR map. In particular, the inconsistency on sequence 08 is large, negatively degrading the performance evaluation of our system. Nevertheless, for completeness, we test the proposed system on all the sequences while using the same parameter setup.

The average ATE over six runs by using the color images are shown in Table 2. From the table, we can see that with the prior LiDAR map, the position error is greatly reduced on most sequences. Note that Sequence 01 is collected on a highway, and few 3D structures are captured by the LiDAR and the stereo camera. The main structures in this scene are road surfaces. In this case, the LiDAR map fails to provide substantial help for online visual localization. It should be noted that there are some structureless scenarios where registration fails to provide a valid result, and thus the prior LiDAR map, is not able to provide useful information for online visual localization. In Figure 10, we plot one typical estimated trajectory of six runs. A global semidense visual map is shown in Figure 11, which is obtained by aggregating all the local maps together.

TABLE 2 RMSE on KITTI data set (color stereo)

	Proposed				ORB-SLAM2			
	Mean (m)	Mean/Traj (%)	Max (m)	Min (m)	Mean (m)	Mean/Traj (%)	Max (m)	Min (m)
Sequence 00	**0.5030**	**0.01351**	**0.5315**	**0.4798**	2.2118	0.05939	2.2459	2.1754
Sequence 01	5.7062	0.23260	7.2113	3.3990	**4.5987**	**0.1875**	**5.5239**	**3.2422**
Sequence 02	**0.4388**	**0.00866**	**0.5035**	**0.3608**	3.9894	0.07873	4.2795	3.6798
Sequence 03	**0.5492**	**0.09791**	**0.6406**	**0.4695**	3.2926	0.58702	3.3234	3.2346
Sequence 04	**0.3691**	**0.09377**	**0.4337**	**0.2505**	0.6641	0.1687	0.7549	0.5652
Sequence 05	**0.3315**	**0.01503**	**0.3450**	**0.3144**	1.0445	0.04736	1.0749	1.0121
Sequence 06	**0.5269**	**0.04274**	**0.5814**	**0.4635**	1.5535	0.1260	1.6774	1.4067
Sequence 07	**0.1901**	**0.02736**	**0.2019**	**0.1750**	0.5999	0.08635	0.7269	0.5334
Sequence 08	**2.9526**	**0.09162**	**3.0843**	**2.8029**	3.1963	0.09918	3.3789	3.0835
Sequence 09	**0.2100**	**0.01231**	**0.2247**	**0.1938**	2.7872	0.1635	4.0569	1.6237
Sequence 10	**0.1838**	**0.01999**	**0.1952**	**0.1720**	1.7234	0.1874	1.9307	1.6547

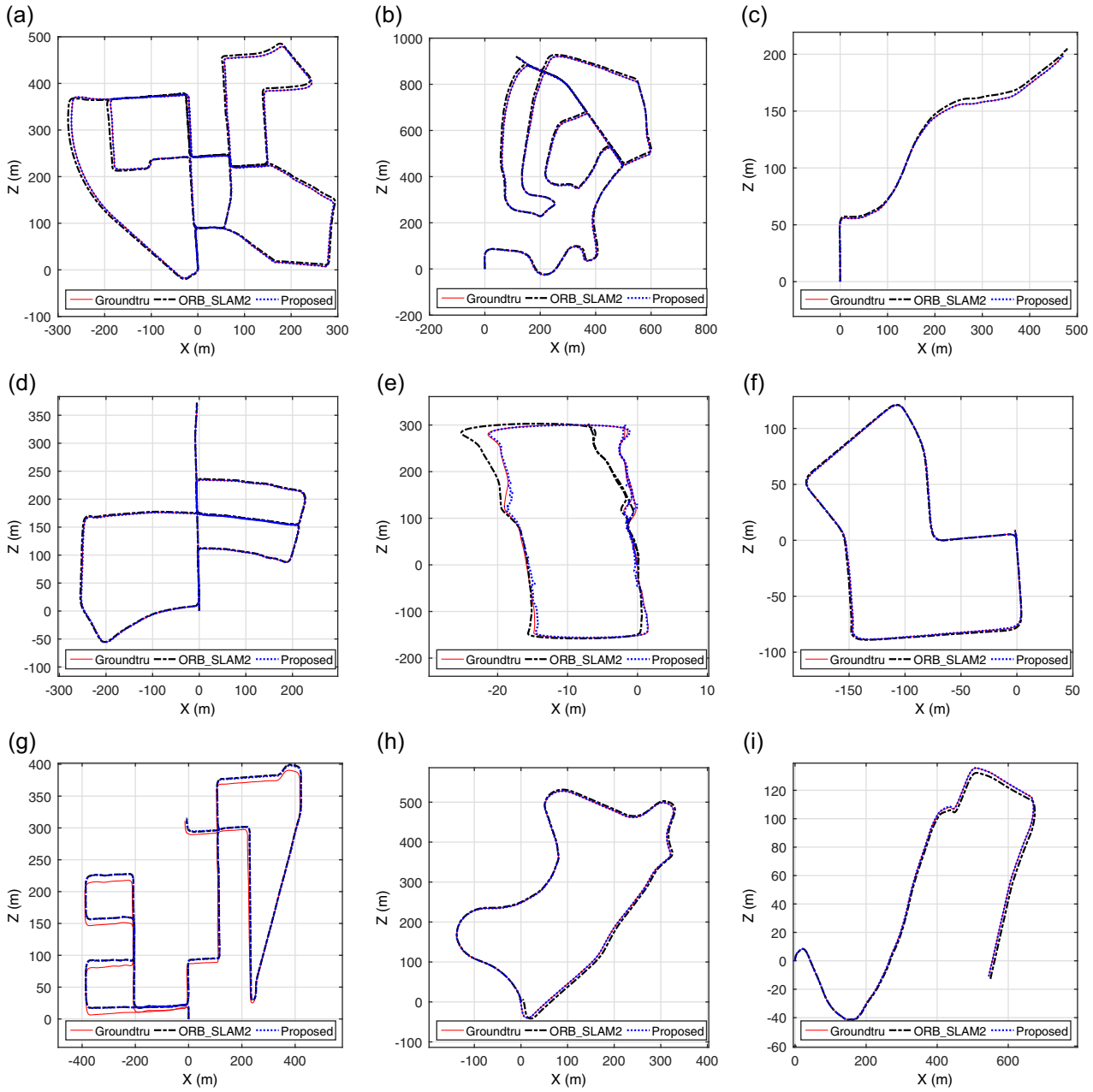

FIGURE 10 The estimated trajectories on some sequences of KITTI data set (Geiger et al., 2012). Red solid line indicates the trajectory of ground truth. Black dot-dash is estimated by ORB-SLAM2, and blue dash is got from our proposed method. (a) On sequence 00; (b) On sequence 02; (c) On sequence 03; (d) On sequence 05; (e) On sequence 06; (f) On sequence 07; (g) On sequence 08; (h) On sequence 09; and (i) On sequence 10 [Color figure can be viewed at wileyonlinelibrary.com]

As no open-source algorithm using the same information as the proposed system is available, we compare the proposed with Kim's method (Kim et al., 2018) which uses the same inputs as the proposed approach. It should be noted that the results of our method are from real-time performance, while the results reported in Kim et al. (2018) are not. As in the vein of Kim et al. (2018), Table 3 shows the averaged (in six runs) mean of absolute translation and rotation errors using grayscale stereo images on KITTI data set, which are different from the averaged RMSE shown in Table 2. Note that in Kim et al. (2018) they did not show the performance on sequence 01. From these results, we can see that the proposed method has comparable accuracy with the non-real-time method (Kim et al., 2018).

5.3 | Experimental evaluation on KAIST urban data set

KAIST urban data set (Jeong et al., 2019) was collected in complex urban driving scenarios. We further tested the proposed method on the Urban 28 Sequence with a total length of 11.47 km. Figure 12 shows some sample images in this sequence. These complex driving

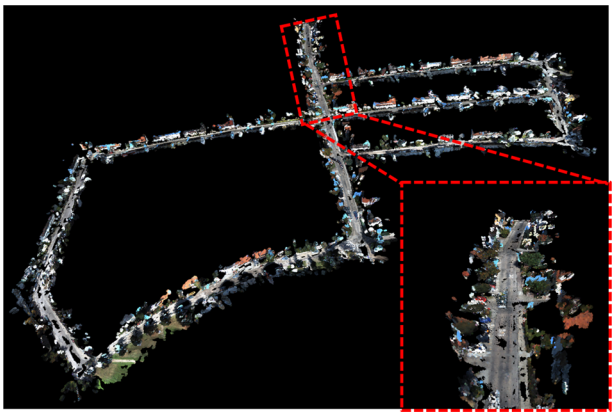

FIGURE 11 A global semidense visual map on sequence 05 of KITTI data set [Color figure can be viewed at wileyonlinelibrary.com]

scenarios are challenging for localization tasks due to significant structural variation, wide roads, high speed, and lots of dynamic objects. To fully utilize the sensor measurements and reserve enough time for back end optimization, the data in this sequence are played at half of the original recording rates for all tests. Since there are lots of dynamic moving vehicles in this sequence, which poses significant challenges for registering the reconstructed point cloud with the prior map, the proposed method fails to go through the whole trajectory without restarts due to consecutive failures of registration. Thus we perform two tests: (i) without restarts, we test the proposed system in a portion of the Urban 28 Sequence with 730 s along a trajectory of 4.1 km; (ii) with restarts, the system is tested on the whole 11.47 km. Benefiting from the failure detection of registration in Section 4.2.1, the system will restart automatically after 20 consecutive failures for the registration are detected. It should be noted that we aim to provide a vision only localization method, even with restarts, the experiments demonstrate the feasibility of our algorithm in a real complex urban scenario. And the initial position guesses in the map where restart the system can be easily obtained by relocalization or GPS as we start the system from the beginning.

For the test (i) without restarts, we perform visual localization not only over the LiDAR map constructed on the same data (Urban 28 sequence, collected on December 12, 2018), but also the LiDAR map constructed on other data (Urban 38, 39 Sequences, collected on May 30, 2019). The mean of translation and rotation errors are shown in Table 4, where "PROPOSED 1" are the results of the proposed approach using the (*in-run*) LiDAR map, and "PROPOSED 2" are the results of the proposed method using the LiDAR map constructed on (*out-run*) data. The LiDAR maps are constructed by using the ground truth poses provided officially. Since the points on dynamic objects are removed, there are hollows in one single LiDAR map. We combine the LiDAR maps on Urban 38 and Urban 39 by transforming them into the same reference of frame. From Table 4, we can find the proposed method without loop closures is significantly better than the ORB-SLAM2 with loop closures. For "PROPOSED 2," although the images used for online localization have a time gap of about 6 months with the PLM, the out-run LiDAR

TABLE 3 Mean of translation and rotation errors on KITTI data set (grayscale stereo)

	Proposed						ORB-SLAM2						Kim's	
	Translation				Rotation		Translation				Rotation		Translation	Rotation
	Mean (m)	Mean/ Traj (%)	Max (m)	Min (m)	Mean (deg)	Max (deg)	Min (deg)	Mean (m)	Mean/ Traj (%)	Max (m)	Min (m)	Mean (deg)	Max (deg)	Min (deg)	Mean (m)	Mean (deg)
Sequence 00	**0.4677**	**0.01256**	**0.4936**	**0.4502**	**0.8698**	**0.9112**	**0.8386**	1.1543	0.03099	1.1843	1.1356	1.4759	1.5661	1.3961	0.1325	0.3221
Sequence 01	**10.0072**	**0.4079**	11.6966	**8.4670**	2.7833	3.5993	2.1788	10.1073	0.4120	**10.9960**	9.6058	**1.7800**	**2.1127**	**1.5211**	-	-
Sequence 02	**0.3726**	**0.00735**	**0.4551**	**0.2786**	**0.7280**	**0.7424**	**0.7131**	4.9751	0.09818	5.8315	3.9967	1.5179	1.8216	1.3872	0.2205	0.3262
Sequence 03	**0.2819**	**0.05026**	**0.3840**	**0.1965**	0.5666	0.6218	0.5288	0.5823	0.1038	0.6932	0.4481	**0.4193**	**0.4793**	**0.3434**	0.2368	0.4133
Sequence 04	**0.2335**	**0.05932**	**0.2658**	**0.1951**	0.7159	0.8647	0.5542	0.2471	0.06277	0.3089	0.2093	**0.1839**	**0.2347**	**0.1427**	0.4496	0.8758
Sequence 05	**0.2573**	**0.01167**	**0.2761**	**0.2451**	**0.4468**	**0.4598**	**0.4334**	0.7084	0.03212	0.7469	0.6754	0.5492	0.5837	0.4940	0.1462	0.3402
Sequence 06	**0.3269**	**0.02652**	**0.3780**	**0.2807**	**0.5049**	**0.5454**	**0.4099**	0.7830	0.06351	0.8863	0.7175	1.0244	1.2347	0.8783	0.3753	0.8485
Sequence 07	**0.1573**	**0.02265**	**0.1817**	**0.1473**	**0.4571**	**0.4923**	**0.4249**	0.5017	0.07222	0.5225	0.4942	0.4625	0.5004	0.4167	0.1305	0.4872
Sequence 08	**3.2348**	**0.1004**	**3.3110**	3.1730	**1.5046**	**1.6720**	**1.2293**	3.2387	0.1005	3.4906	**3.1327**	1.6170	1.7085	1.5554	0.1440	0.3279
Sequence 09	**0.2297**	**0.01347**	**0.2424**	**0.2172**	**0.5171**	**0.5482**	**0.4855**	2.4043	0.1410	2.6969	1.4209	1.0023	1.136	0.9049	0.1799	0.3375
Sequence 10	**0.1352**	**0.01470**	**0.1391**	**0.1326**	**0.5149**	**0.5527**	**0.4917**	1.0304	0.1121	1.1344	0.9331	1.8300	2.0894	1.7209	0.2398	0.4934

FIGURE 12 Sample images of the KAIST Urban 28 sequence (Jeong et al., 2019) [Color figure can be viewed at wileyonlinelibrary.com]

map can still provide substantial help for online visualization. "PROPOSED 2" has slightly lower errors than "PROPOSED 1" due to its higher map quality. The rotation estimates of the proposed system are slightly worse than the baseline due to the jumps while correcting the drift by registration. The trajectory estimates and the ground truth are depicted in Figure 13.

For the test (ii) with restarts, the proposed system restarts four times (0.349 restarts per km) along the whole 11.47 km of Urban 28 sequence. And the mean of translation error and rotation error are 4.31 m (0.0376%), 2.25 deg. In Kim et al. (2018), the authors also tested their method on their own collected sequence (with a total length of 15.7 km) over almost the same route of Urban 28 sequence. Kim's method needs 28 restarts (1.783 restarts per km) for traversing the whole 15.7 km. Comparing the performance with Kim's method, we can see that the proposed method is more robust in the complex urban driving scenario. The estimated trajectory aligned with ground truth and four restart positions of our method can be seen in Figure 14.

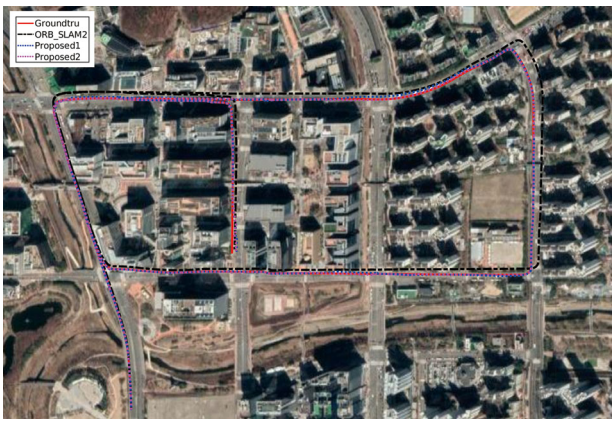

FIGURE 13 Trajectory estimates overlaid on the satellite map, which are performed on a portion of Urban 28 sequence (4.1 km) without restarts. The "Proposed 1" is the proposed visual localization using in-run PLM, and "Proposed 2" is the proposed visual localization using out-run PLM with a time gap of about 6 months; That is, the two LiDAR maps are built about 6 months apart. PLM, prior LiDAR map [Color figure can be viewed at wileyonlinelibrary.com]

5.4 | Experimental evaluation on ZJU data set

We used a wheeled robot to collect data sets in ZJU campus. As shown in Figures 15 and 16, our mobile robot is equipped with multiple sensors, such as Velodyne (Velodyne VLP-16, 2018), Xsens IMU (Xsens MTi-300 AHRS; Xsens, 2018), and two synchronized industrial cameras (MV-GE231GC-T Industrial Camera; Mind Vision Technology, 2018). Both the synchronized stereo cameras and LiDAR runs at 10 Hz, and the IMU is at 400 Hz. We calibrate the LiDAR and stereo camera by Dhall, Chelani, Radhakrishnan, and Krishna (2017). The timestamps of each sensor are synchronized to the same time axis by hardware. As the widely used RTK-GPS fails to give valid measurements in most places in our campus due to the tall buildings and dense trees, we use a LiDAR-IMU SLAM system to provide the

TABLE 4 Mean of translation and rotation errors on KAIST urban data set (4.1 km)

	Translation				Rotation		
	Mean m)	Mean/Traj (%)	Max (m)	Min (m)	Mean (deg)	Max (deg)	Min (deg)
ORB-SLAM2	7.5925	0.18467	7.8441	7.3017	**1.4274**	**1.7704**	**1.2348**
PROPOSED 1	3.8296	0.09315	5.4923	2.1727	1.9840	2.1727	1.6441
PROPOSED 2	**2.6312**	**0.06400**	**3.9114**	**1.8144**	1.8325	2.3829	1.5549

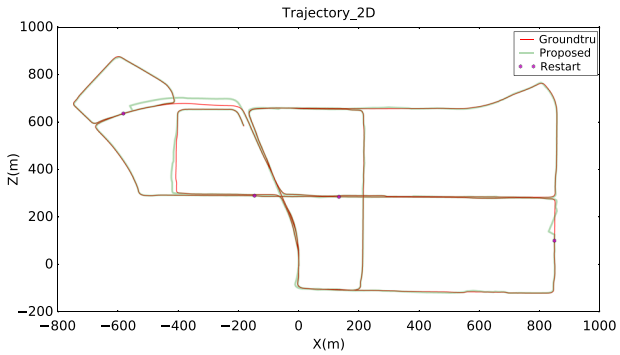

FIGURE 14 The trajectory estimates (in green) on the whole Urban 28 sequence (11.47 km) and the ground truth (in red). There are in total four restarts along the whole trajectory. The restart positions are marked by magenta hexagon [Color figure can be viewed at wileyonlinelibrary.com]

ground truth. In this SLAM system, the keyframe strategy is used, and only a portion of scans during the whole trajectory is used for mapping. A local LiDAR point cloud map by assembling several scans is maintained. Keyframe scans are registered with this local LiDAR point cloud map by P2D-NDT. The keyframe poses are added into the pose graph optimization. We use IMU for the motion compensation in one single LiDAR scan and also IMU preintegration (Forster, Carlone, Dellaert, & Scaramuzza, 2016) to gather all the IMU measurements between two keyframes, which are then used as the initial guesses of the registrations. In addition, we also use images for loop-closure detection for this LiDAR-IMU SLAM system, which is performed by using DBoW2 (Gálvez-López & Tardos, 2012). Once the loop closure is detected, we will use the generalized-ICP (Segal, Haehnel, & Thrun, 2009) to register the current local LiDAR map to the loop-keyframe, and the resulting pose by the generalized-ICP is added to the pose graph as a loop-closure edge.

We collected six sequences: Sequence 1, 2, 3, and 5 are collected in scene A, and sequence 4 and 6 are collected in scene B. These data sets are collected at different times during the day, and in various weather, such as sunny, cloudy, after rain. See Figure 17 and Table 5 for the details. It is clear that significant changes occur among different sequences, such as light conditions, cars, pedestrians, and weathers. To test whether the proposed method works under large path deviations, we collect data sets while moving in different directions. Sequence 1, 3, and 5 are collected in counterclockwise routes seen from a bird's-eye view, while sequence 2, 4, and 6 are clockwise.

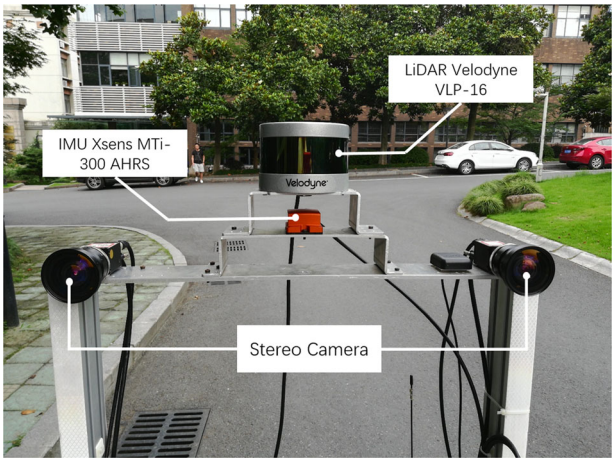

FIGURE 15 Sensors setup on our wheeled robot [Color figure can be viewed at wileyonlinelibrary.com]

5.4.1 | Localization over in-run prior LiDAR map

We first run the proposed system with images and PLM reconstructed on the same sequence, which means that the environments, while we construct the LiDAR map and perform the online visual localization, are the same. Six runs on each sequence are performed as the preceding experiments, and the experimental results are shown in Table 6. From the table, we can see that in each sequence, our method has the smaller errors than the ORB-SLAM2. In sequences 3, 4, and 6, which are more difficult as there are plenty of dynamic objects such as pedestrians and vehicles, our algorithm shows a greater advantage than ORB-SLAM2. In addition, under poor illumination conditions, the performance of ORB-SLAM2 degrades significantly. Sequences 3 and 4 are collected under poor illumination conditions, and both of them contain many dynamic objects. On these two sequences, our proposed algorithm performs much better than the ORB-SLAM2. Figure 18 shows the estimated trajectories on our own data sets with the aid of prior LiDAR map constructed on the same sequence.

5.4.2 | Different prior map reuse comparisons over out-run LiDAR map

We also test the system using the LiDAR map built on other sequences collected in the same scene, to evaluate whether a prior LiDAR map is still useful for the online visual localization after a period of time and with changes to the environment. We randomly select sequence 5 to construct a prior map for sequence 1, 2, 3, and 5, in scene A and select sequence 6 to construct a prior map for sequence 4 and 6, in scene B. Figure 16 shows the prior LiDAR map constructed on sequence 5 and sequence 6 in scene A and B, respectively.

Besides the proposed method, we will also test another two map reuse system: online visual localization aided by a prior visual feature map, and online LiDAR localization over a PLM. To reuse the sparse visual feature map, we modify the original ORB-SLAM2 to add the capabilities of map saving and loading. After running ORB-SLAM2 online, we save the constructed visual feature map in a binary file, which contains the following data: the poses of keyframes, ORB features with positions and descriptors, and the observation relationships between the visual features and

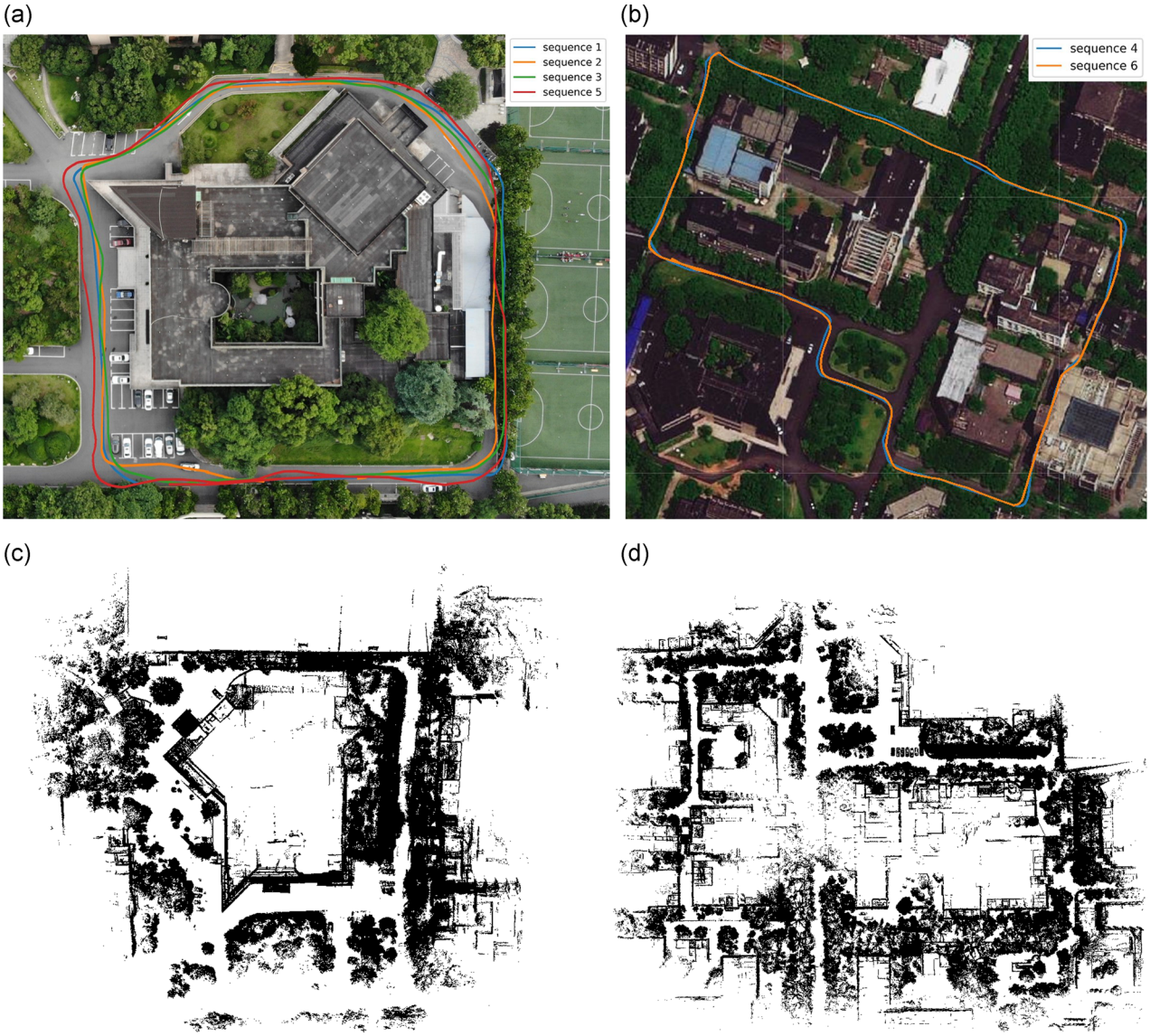

FIGURE 16 Bird's-eye views of our own data sets, including scene A and B, and the trajectories are also displayed. The prior LiDAR maps of both scenes are constructed on sequences 5 and 6. For clarity, we remove the ground points in both LiDAR maps. (a) A bird's-eye view of scene A; (b) A bird's-eye view of scene B; (c) Prior LiDAR map of scene A; and (d) Prior LiDAR map of scene B [Color figure can be viewed at wileyonlinelibrary.com]

keyframes. To construct a good visual feature map, we perform a final great adjustment including a pose graph optimization that corrects all the keyframe poses with the ground truth, and the sequent global bundle adjustment. After the final adjustment, the poses of keyframes are nearly identical to the ground truth, hence we regard this adjusted visual feature map to be accurate enough. Same as the prior LiDAR map, we also use a prior visual map created by sequence 5 for scene A for online localization, and a prior visual feature map created by sequence 6 for scene B. In online real-time visual localization, after loading the visual feature map into ORB-SLAM2, we enable the relocation mode of the system. Under this model, the system only estimates the poses of keyframes, but local mapping for feature map reconstruction is prohibited. The sizes of prior visual feature map (binary file) for scene A and B are 119.7 MB and 411.0 MB, respectively. And the sizes of prior LiDAR point cloud map (pcd file) for scene A and B are 6.7 MB and 14.2 MB, respectively. It should be noted that we do not need a dense PLM, and the PLM used in our system has been downsampled by a voxel grid filter with a resolution of 0.2 m. The voxel grid filter is from PCL (Point Cloud Library, 2018). As the LiDAR map only stores the 3D positions of the points, its size is much smaller than the visual feature map. We also perform online LiDAR localization in the LiDAR map. Online LiDAR localization is carried out by matching online LiDAR scans to map by the standard P2D-NDT, and the LiDAR map is the same map used in aided online visual localization.

The experimental results of different types of map reuse systems are shown in Table 7. The proposed system performs better than the online visual localization with the aid of a prior visual feature map. As the bird's-eye view shown in Figure 16, the routes when the robot

FIGURE 17 Snapshots of our own data sets collected in two different scenes. Different sequences are collected at different time and under various weather conditions [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 5 Details about our own data sets

Sequence	Date	Time	Scene	Illumination	Weather	Dynamic objects	Direction	Length (m)
1	June 1	9:18	A	Day	Sunny	Few	Counterclockwise	341.990
2	June 1	9:32	A	Day	Sunny	Few	Clockwise	325.425
3	June 5	18:16	A	Dusk	After rain	Many	Counterclockwise	361.765
4	June 5	18:38	B	Dusk	After rain	Many	Clockwise	777.568
5	June 12	17:48	A	Dusk	Cloudy	Several	Counterclockwise	406.877
6	June 12	18:09	B	Dusk	Cloudy	Many	Clockwise	742.766

builds up the prior map have obvious deviations from the routes of online visual localization. The PLM can still improve the accuracy of the online visual localization. As for visual localization aided by accurate visual feature map, online visual localization in the clockwise sequence 2 fails to reuse the visual feature map created by the counterclockwise sequence 5. Meanwhile, the LiDAR map is able to provide substantial help in this case of the reverse directions. Furthermore, we can see that improvements occur on sequences 1, 3, 4, 5, 6 with the aid of prior visual feature map on sequence 5, 6. However, the accuracy is only improved slightly, even on sequences 5, 6, both of which use the prior visual feature map built on their own.

It is worth noting that the visual localization on one sequence with the prior LiDAR map created by the same sequence (in Table 6) and with the prior LiDAR map created by another

TABLE 6 RMSE on ZJU data set (over In-run prior LiDAR map)

	Proposed				ORB-SLAM2			
	Mean (m)	Mean/Traj (%)	Max (m)	Min (m)	Mean (m)	Mean/Traj (%)	Max (m)	Min (m)
Sequence 1	**0.5111**	**0.1494**	**0.5314**	**0.4777**	0.9130	0.2670	0.9804	0.8474
Sequence 2	**0.5238**	**0.1610**	**0.5466**	**0.5011**	1.3245	0.4070	1.3936	1.2136
Sequence 3	**0.2513**	**0.06946**	**0.2782**	**0.2314**	1.3525	0.3739	1.3898	1.3304
Sequence 4	**0.8181**	**0.1052**	**1.2998**	**0.5278**	2.5547	0.3286	2.6900	2.4391
Sequence 5	**0.2771**	**0.06811**	**0.3218**	**0.1994**	0.7044	0.1731	0.7170	0.6916
Sequence 6	**0.8093**	**0.1090**	**0.8989**	**0.5128**	1.8210	0.2452	1.8903	1.7390

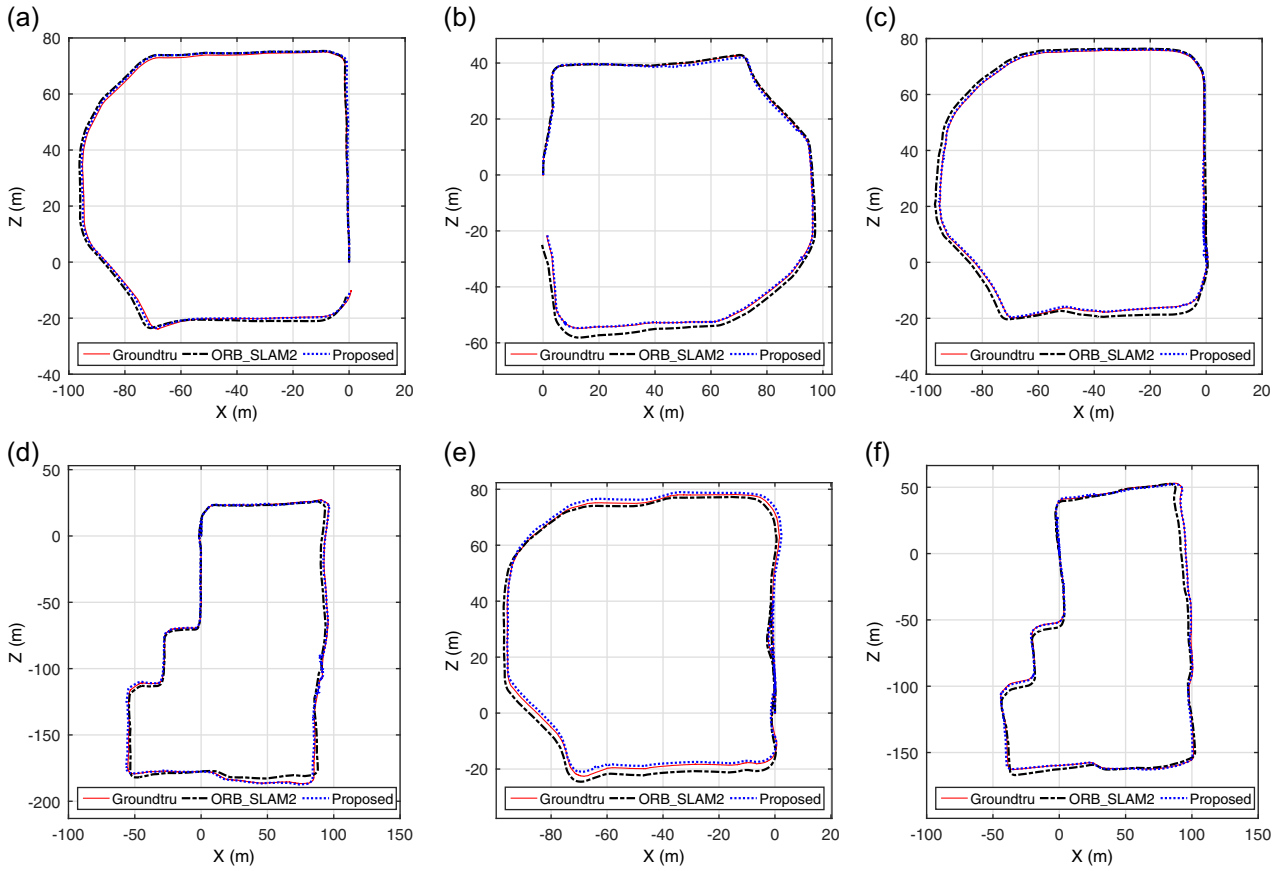

FIGURE 18 Trajectories estimated on our own data sets, shown in the camera frame. The proposed method is with the aid of in-run prior LiDAR map. Red solid line indicates the ground truth. Black dot-dash is estimated by ORB-SLAM2, and the blue dash is got from our proposed method. (a) On sequence 1; (b) On sequence 2; (c) On sequence 3; (d) On sequence 4; (e) On sequence 5; and (f) On sequence 6 [Color figure can be viewed at wileyonlinelibrary.com]

sequence (in Table 7) have similar performances. This implies that the visual localization aided by the PLM is robust and suitable for somehow long-term usage. In Table 7, *LiDAR NDT* denotes the online LiDAR localization with the aid of prior LiDAR map by the conventional P2D-NDT, and it has a better and much more consistent performance than our proposed visual online localization, due to the accurate metric measurements of LiDAR. ATE in one typical run of the 6 runs, during the whole trajectory, is shown in Figure 19.

5.5 | Evaluation of ProW-NDT and visual feature refinement

We will demonstrate the advantages of the proposed ProW-NDT and the sparse visual feature refinement by the prior LiDAR map. The ablation experiments are performed on both the Gazebo simulated data set and the KITTI data set, both of which provide fairly accurate ground truth. We set up three different configurations for the proposed system as shown in Table 8 and Table 9. In

TABLE 7 Three map reuse systems: RMSE on ZJU data set (over out-run prior LiDAR map)

	Proposed				ORB-SLAM2				LiDAR NDT			
	Mean (m)	Mean/ Traj (%)	Max (m)	Min (m)	Mean (m)	Mean/ Traj (%)	Max (m)	Min (m)	Mean (m)	Mean/ Traj (%)	Max (m)	Min (m)
Sequence 1	**0.4746**	**0.1388**	**0.4855**	**0.4561**	0.6515	0.1905	0.6529	0.6505	0.3861	0.1129	0.3866	0.3859
Sequence 2	**0.6234**	**0.1916**	**0.6531**	**0.5919**	FAIL				0.5450	0.1675	0.5463	0.5442
Sequence 3	**0.7620**	**0.2106**	**0.7978**	**0.7086**	1.1698	0.3234	1.1701	1.1696	0.6719	0.1857	0.6724	0.6713
Sequence 4	**1.4874**	**0.1913**	2.0561	**0.6578**	1.8183	0.2338	**1.9289**	1.7449	0.4264	0.05483	0.4270	0.4253
Sequence 5	**0.2771**	**0.06811**	**0.3218**	**0.1994**	0.6880	0.1691	0.6899	0.6808	0.1617	0.03975	0.1622	0.1614
Sequence 6	**0.8093**	**0.1090**	**0.8988**	**0.5128**	1.6937	0.2280	1.7016	1.6897	0.6388	0.08600	0.6392	0.637

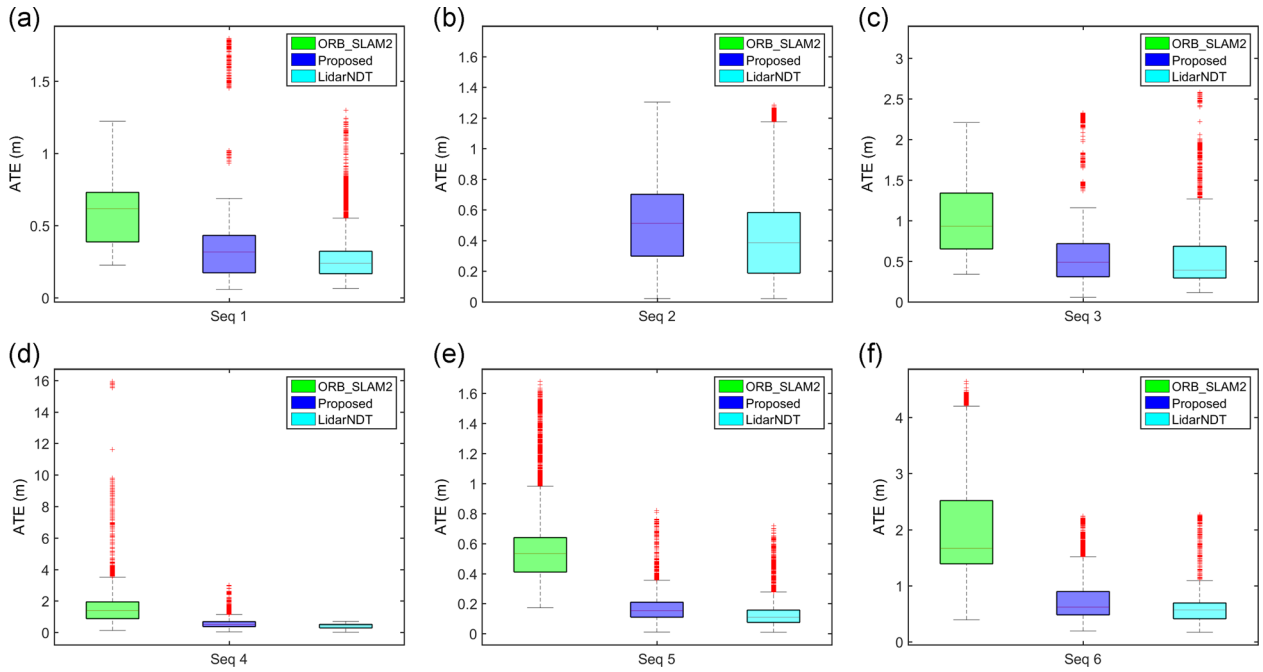

FIGURE 19 Norm of ATE, during the whole trajectory of one typical run, got from three types of map reuse system on our data set: the proposed online visual localization aided by prior LiDAR map (*Proposed*), the online visual localization aided by prior visual feature map (*ORB_SLAM2*), the online LiDAR map aided by prior LiDAR map (*Lidar NDT*). As online visual localization aided by prior visual feature map fails on sequence 2, it's set blank. (a) On sequence 1; (b) On sequence 2; (c) On sequence 3; (d) On sequence 4; (e) On sequence 5; and (f) On sequence 6 [Color figure can be viewed at wileyonlinelibrary.com]

both tables, *Normal NDT Only* denotes that the registration method used in our framework is the conventional P2D-NDT, which sets the same outlier ratio 0.6 to every point in the source point cloud, and the surfel based sparse feature refinement is abandoned; *ProW-NDT Only* denotes that ProW-NDT (see Section 4.2) is used for registration, while the surfel based feature refinement is still abandoned; *Full System* represents that ProW-NDT is used for registration and the surfel based sparse feature refinement is conducted. On the Gazebo simulated data set, we find that when the disturbance is relatively small, the ProW-NDT shows a clear advantage over the conventional P2D-NDT. When the disturbance becomes large, surfel based refinement can significantly improve the performance of our system. On the KITTI data set, we find that the effectiveness of surfel based refinement is significant on some sequences. It can be seen that our localization system becomes more stable with the surfel based refinement for the visual features.

5.6 | Computational cost

Lastly, we show the computational cost of the proposed real-time multimodal visual localization system, by counting the averaged CPU runtime spent on the main stages of our approach in Table 10, when running the proposed approach on all sequences (00–10) KITTI data set. In this table, *Visual Tracking* denotes the time consumption for feature extraction and tracking in the visual tracking thread. *Recovery of KF Depth* is the time consumption of both getting the depth and recovering 3D point cloud from one keyframe with a stereo image pair; *Visual Semi-dense Reconstruction* is the time spent on assembling point cloud reconstructed from multiple keyframes into one local semidense visual point cloud. *ProW-NDT* is the time spent on registering the local visual point cloud to the prior LiDAR map. *Local BA and Surfel Refinement* is the time spent for local bundle adjustment and 3D visual feature refinement by surfels; *Pose Graph Optimization* is the time spent on

TABLE 8 Ablation experiments on Gazebo data set

Noise level	Normal NDT only				ProW-NDT only				Full system			
	Mean (m)	Mean/ Traj (%)	Max (m)	Min (m)	Mean (m)	Mean/ Traj (%)	Max (m)	Min (m)	Mean (m)	Mean/ Traj (%)	Max (m)	Min (m)
5	0.4411	0.1498	0.4606	0.4226	0.4102	0.1393	0.4675	**0.3641**	**0.3962**	**0.1345**	**0.4150**	0.3751
10	0.8447	0.2868	0.9129	0.7711	0.8505	0.2888	0.8870	0.7865	**0.8338**	**0.2831**	**0.8649**	**0.7508**
15	1.3389	0.4546	1.4172	1.2492	1.3160	0.4468	1.3765	1.2343	**1.2249**	**0.4159**	**1.3265**	**1.1019**

TABLE 9 Ablation experiments on KITTI data set (color stereo)

	Normal NDT only				ProW-NDT only				Full system			
	Mean (m)	Mean/ Traj (%)	Max (m)	Min (m)	Mean (m)	Mean/ Traj (%)	Max (m)	Min (m)	Mean (m)	Mean/ Traj (%)	Max (m)	Min (m)
Sequence 00	**0.4898**	**0.01315**	0.5268	**0.4618**	0.5027	0.01350	**0.5126**	0.4956	0.5030	0.01351	0.5315	0.4798
Sequence 01	6.5878	0.26854	7.4510	5.5660	6.6353	0.27048	10.2103	**3.0155**	**5.7062**	**0.23260**	**7.2113**	3.3990
Sequence 02	0.6672	0.01317	0.6740	0.6611	0.5959	0.01176	0.6600	**0.3161**	**0.4388**	**0.00866**	**0.5035**	0.3608
Sequence 03	1.4154	0.25234	3.2518	0.4935	**0.5330**	**0.09503**	0.6452	**0.4451**	0.5492	0.09791	**0.6406**	0.4695
Sequence 04	0.4322	0.10979	0.6171	0.3311	0.3767	0.09571	0.4964	0.2538	**0.3691**	**0.09377**	**0.4337**	**0.2505**
Sequence 05	0.2675	0.01213	0.3218	0.2476	**0.2580**	**0.01170**	**0.2638**	**0.2466**	0.3315	0.01503	0.3450	0.3144
Sequence 06	0.5588	0.04532	0.6697	0.4905	0.5464	0.04432	0.6059	**0.4110**	**0.5269**	**0.04274**	**0.5814**	0.4635
Sequence 07	0.2210	0.03182	0.2438	0.1841	0.19446	0.02800	0.2029	0.1867	**0.1901**	**0.02736**	**0.2019**	**0.1750**
Sequence 08	3.4411	0.10677	3.8263	3.1362	3.1972	0.09921	3.6531	2.8571	**2.9526**	**0.09162**	**3.0843**	**2.8029**
Sequence 09	0.2418	0.01418	0.2800	0.2111	**0.2090**	**0.01226**	**0.2330**	**0.1924**	0.2100	0.01231	0.2247	0.1938
Sequence 10	0.2003	0.02179	0.2428	0.1811	0.1905	0.02072	0.1963	0.1836	**0.1838**	**0.01999**	**0.1952**	**0.1720**

TABLE 10 Run-time of the proposed method (sec)

Procession	Mean of time spent	Std of time spent
Visual tracking	0.0447	0.0083
Recovery of KF depth	0.0325	0.0013
Visual semidense reconstruction	0.0131	0.0031
ProW-NDT	0.1043	0.0783
Local BA and surfel refinement	0.1655	0.0629
Pose graph optimization	0.0060	0.0097

fusing the registration result by a pose graph optimization. It should be noted that since the proposed system is in a multithread framework, the visual tracking processes the raw image measurements in real-time at a high frequency, and the other operations are only needed to be executed at a lower frequency in the other threads. This mechanism achieves a balance between accuracy and time consumption.

6 | CONCLUSIONS AND FUTURE WORK

We have developed a real-time stereo visual localization system that can efficiently utilize the prior LiDAR point cloud map. In particular, the proposed approach constructs both a sparse visual feature map and a semidense visual point cloud map. The former is used for visual tracking and will be refined based on structure constraints enforced by surfels extracted from the LiDAR map; the latter is registered to the LiDAR map with the ProW-NDT approach that weighs each point in the source point cloud based on its uncertainty. Pose graph optimization is used to fuse the registration and VO results. We have extensively validated the proposed system on both simulated and real-world data sets and shown that the proposed approach is able to provide accurate 6D pose estimates in real-time without the support of GPU. In the future, we will investigate improving this system by fusing IMU, GPS, and wheel odometer measurements.

ACKNOWLEDGMENTS

This study is partially supported by the National Natural Science Foundation of China under Grant U1509210 and 61836015. We would like to thank Jinhong Xu and Zheng Zhang for maintaining the robot platform used to collect our own data set.

ORCID

Xingxing Zuo http://orcid.org/0000-0003-4158-3153

REFERENCES

Agamennoni, G., Fontana, S., Siegwart, R. Y., & Sorrenti, D. G. (2016). Point clouds registration with probabilistic data association. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (pp. 4092–4098). IEEE

Agarwal, P., Tipaldi, G. D., Spinello, L., Stachniss, C., & Burgard, W. (2013). Robust map optimization using dynamic covariance scaling. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 62–69). IEEE.

Barfoot, T. D. (2017). *State estimation for robotics* (pp. 217–219). New York, NY: Cambridge University Press.

Barrow, H. G., Tenenbaum, J. M., Bolles, R. C., & Wolf, H. C. (1977). *Parametric correspondence and chamfer matching: Two new techniques for image matching*. Technical report, SRI International Menlo Park CA Artificial Intelligence Center.

Besl, P. J., & McKay, N. D. (1992). Method for registration of 3-d shapes. *Proceedings of Sensor Fusion IV: Control Paradigms and Data Structures* (Vol. 1611, pp. 586–607). International Society for Optics and Photonics.

Black, M. J., & Anandan, P. (1996). The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, *63*(1), 75–104.

Black, M. J., & Rangarajan, A. (1996). On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, *19*(1), 57–91.

Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). Brief: Binary robust independent elementary features. *Proceedings of the European conference on computer vision*, (pp. 778–792). Springer.

Caselitz, T., Steder, B., Ruhnke, M., & Burgard, W. (2016). Monocular camera localization in 3d lidar maps. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (pp. 1926–1931). IEEE.

Churchill, W., & Newman, P. (2013). Experience-based navigation for long-term localisation. *The International Journal of Robotics Research*, *32*(14), 1645–1661.

Cummins, M., & Newman, P. (2008). Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, *27*(6), 647–665.

Cvišić, I., Ćesić, J., Marković, I., & Petrović, I. (2018). Soft-slam: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles. *Journal of Field Robotics*, *35*(4), 578–595.

Dhall, A., Chelani, K., Radhakrishnan, V., & Krishna, K. M. (2017). Lidar-camera calibration using 3d-3d point correspondences. arXiv preprint arXiv:1705.09785.

Forster, C., Carlone, L., Dellaert, F., & Scaramuzza, D. (2016). On-manifold preintegration for real-time visual-inertial odometry. *IEEE Transactions on Robotics*, *33*(1), 1–21.

Gálvez-López, D., & Tardos, J. D. (2012). Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, *28*(5), 1188–1197.

Gawel, A., Cieslewski, T., Dubé, R., Bosse, M., Siegwart, R., & Nieto, J. (2016). Structure-based vision-laser matching. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (pp. 182–188). IEEE.

Gazebo Sensor Noise Model. (2018). Sensor noise model. Retrieved from http://gazebosim.org/tutorials?tut=sensor_noise

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The kitti vision benchmark suite. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 3354–3361). IEEE

Huhle, B., Magnusson, M., Straßer, W., & Lilienthal, A. J. (2008). Registration of colored 3d point clouds with a kernel-based extension to the normal distributions transform. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 4025–4030). IEEE

Jeong, J., Cho, Y., Shin, Y.-S., Roh, H., & Kim, A. (2019). Complex urban dataset with multi-level sensors from highly diverse urban environments. *The International Journal of Robotics Research*, *38*(6), 642–657.

Kim, H., Lee, D., Oh, T., Choi, H.-T., & Myung, H. (2015). A probabilistic feature map-based localization system using a monocular camera. *Sensors*, *15*(9), 21636–21659.

Kim, H., Lee, D., Oh, T., Lee, S. W., Choe, Y., & Myung, H. (2014). Feature-based 6-dof camera localization using prior point cloud and images. In J. H. Kim, E. Matson, H. Myung, P. Xu & F. Karray (Eds.), *Robot intelligence technology and applications 2* (pp. 3–11). Cham: Springer.

Kim, Y., Jeong, J., & Kim, A. (2018). Stereo camera localization in 3d lidar maps. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (pp. 1–9). IEEE.

Klein, G., & Murray, D. (2007). Parallel tracking and mapping for small ar workspaces. *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, (pp. 225–234). IEEE.

Lepetit, V., Moreno-Noguer, F., & Fua, P. (2009). Epnp: An accurate o (n) solution to the pnp problem. *International Journal of Computer Vision*, *81*(2), 155.

Lu, G., Ly, V., Shen, H., Kolagunda, A., & Kambhamettu, C. (2013). Improving image-based localization through increasing correct feature correspondences. *Proceedings of the International Symposium on Visual Computing*, (pp. 312–321). Springer.

Lu, Y., Huang, J., Chen, Y.-T., & Heisele, B. (2017). Monocular localization in urban environments using road markings. *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, (pp. 468–474). IEEE.

Lu, Y., Lee, J., Yeh, S.-H., Cheng, H.-M., Chen, B., & Song, D. (2017). Sharing heterogeneous spatial knowledge: Map fusion between asynchronous monocular vision and lidar or other prior inputs. *Proceedings of the International Symposium on Robotics Research (ISRR)*.

Maddern, W., Stewart, A. D., & Newman, P. (2014). Laps-II: 6-dof day and night visual localisation with prior 3d structure for autonomous road vehicles. *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, (pp. 330–337). IEEE.

Magnusson, M. (2009).*The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection* (PhD Thesis). Örebro Universitet.

Magnusson, M., Nuchter, A., Lorken, C., Lilienthal, A. J., & Hertzberg, J. (2009). Evaluation of 3d registration reliability and speed-a comparison of icp and ndt. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 3907–3912). IEEE.

Magnusson, M., Vaskevicius, N., Stoyanov, T., Pathak, K., & Birk, A. (2015). Beyond points: Evaluating recent 3d scan-matching algorithms. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 3631–3637). IEEE.

McManus, C., Churchill, W., Maddern, W., Stewart, A. D., & Newman, P. (2014). Shady dealings: Robust, long-term visual localisation using illumination invariance. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 901–906). IEEE.

Mind Vision Technology. (2018). MV-GE231GC-T industrial camera. Retrieved from http://www.mindvision.com.cn/cpzx/info_62.aspx?itemid=1393&lcid=100/

Mühlfellner, P., Bürki, M., Bosse, M., Derendarz, W., Philippsen, R., & Furgale, P. (2016). Summary maps for lifelong visual localization. *Journal of Field Robotics*, *33*(5), 561–590.

Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, *31*(5), 1147–1163.

Mur-Artal, R., & Tardós, J. D. (2014). Fast relocalisation and loop closing in keyframe-based slam. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 846–853). IEEE.

Mur-Artal, R., & Tardós, J. D. (2015). Probabilistic semi-dense mapping from highly accurate feature-based monocular slam. *Proceedings of Robotics: Science and Systems*, (Vol. 2015)

Mur-Artal, R., & Tardós, J. D. (2017a). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, *33*(5), 1255–1262.

Mur-Artal, R., & Tardós, J. D. (2017b). Visual-inertial monocular slam with map reuse. *IEEE Robotics and Automation Letters*, *2*(2), 796–803.

Neubert, P., Schubert, S., & Protzel, P. (2017). Sampling-based methods for visual navigation in 3d maps by synthesizing depth images. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (pp. 2492–2498). IEEE.

Ng, P. C., & Henikoff, S. (2003). Sift: Predicting amino acid changes that affect protein function. *Nucleic Acids Research*, *31*(13), 3812–3814.

Pascoe, G., Maddern, W., & Newman, P. (2015). Direct visual localisation and calibration for road vehicles in changing city environments. *Proceedings of the IEEE International Conference on Computer Vision Workshops* (pp. 9–16).

Pascoe, G., Maddern, W., Stewart, A. D., & Newman, P. (2015). Farlap: fast robust localisation using appearance priors. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 6366–6373). IEEE

Piasco, N., Sidibé, D., Demonceaux, C., & Gouet-Brunet, V. (2018). A survey on visual-based localization: On the benefit of heterogeneous data. *Pattern Recognition*, *74*, 90–109.

Pioneer. (2018). Pioneer 3-DX. Retrieved from https://robots.ros.org/pioneer-3-dx/

Point Cloud Library. (2018). PCL. Retrieved from http://pointclouds.org

Qin, T., & Shen, S. (2017). Robust initialization of monocular visual-inertial estimation on aerial robots. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (pp. 4225–4232). IEEE.

Ros. (2018). ROS. Retrieved from http://www.ros.org/

Segal, A., Haehnel, D., & Thrun, S. (2009). Generalized-ICP. *Robotics: Science and Systems*, *2*, 435.

Stewart, A. D., & Newman, P. (2012). Laps-localisation using appearance of prior structure: 6-dof monocular camera localisation using prior pointclouds. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 2625–2632). IEEE.

Stoyanov, T., Magnusson, M., Andreasson, H., & Lilienthal, A. J. (2012). Fast and accurate scan registration through minimization of the distance between compact 3d ndt representations. *The International Journal of Robotics Research*, *31*(12), 1377–1393.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012). A benchmark for the evaluation of rgb-d slam systems. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (pp. 573–580). IEEE.

Sujiwo, A., Takeuchi, E., Morales, L. Y., Akai, N., Ninomiya, Y., & Edahiro, M. (2017). Localization based on multiple visual-metric maps. *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, (pp. 212–219). IEEE.

Velodyne VLP-16. (2018). Velodyne. Retrieved from http://velodynelidar.com/vlp-16.html

Williams, B., Cummins, M., Neira, J., Newman, P., Reid, I., & Tardós, J. (2009). A comparison of loop closing techniques in monocular slam. *Robotics and Autonomous Systems*, *57*(12), 1188–1197.

Wolcott, R. W., & Eustice, R. M. (2014). Visual localization within lidar maps for automated urban driving. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (pp. 176–183). IEEE.

Wong, D., Kawanishi, Y., Deguchi, D., Ide, I., & Murase, H. (2017). Monocular localization within sparse voxel maps. *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, (pp. 499–504). IEEE.

Xsens. (2018). Xsens MTi-300 AHRS. Retrieved from https://www.xsens.com/products/mti-100-series/

Zhang, Z. (1995). *Parameter estimation techniques: A tutorial with application to conic fitting* (PhD Thesis). INRIA.

Zhang, W., & Kosecka, J. (2006). Image based localization in urban environments. *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission*, (pp. 33–40). IEEE.

Zhang, J., & Singh, S. (2014). Loam: Lidar odometry and mapping in real-time. *Proceedings of Robotics: Science and Systems*, *2*, 9.

Zhang, J., & Singh, S. (2015). Visual-lidar odometry and mapping: Low-drift, robust, and fast. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 2174–2181). IEEE.

Zhang, J., & Singh, S. (2016). Low-drift and real-time lidar odometry and mapping. *Autonomous Robots*, *41*, 401–416.

Zhou, Q.-Y., Park, J., & Koltun, V. (2016). Fast global registration. *Proceedings of European Conference on Computer Vision (ECCV)* (pp. 766–782). Springer.

How to cite this article: Zuo X, Ye W, Yang Y, et al. Multimodal localization: Stereo over LiDAR map. *J Field Robotics*. 2020;37:1003–1026. https://doi.org/10.1002/rob.21936