

Accurate Prior-centric Monocular Positioning with Offline LiDAR Fusion

Jinhao He^{1,*}, Huaiyang Huang^{2,*}, Shuyang Zhang³, Jianhao Jiao^{3,4}, Chengju Liu⁵ and Ming Liu^{1,6}

Abstract—Unmanned vehicles usually rely on Global Positioning System (GPS) and Light Detection and Ranging (LiDAR) sensors to achieve high-precision localization results for navigation purpose. However, this combination with their associated costs and infrastructure demands, poses challenges for widespread adoption in mass-market applications. In this paper, we aim to use only a monocular camera to achieve comparable onboard localization performance by tracking deep-learning visual features on a LiDAR-enhanced visual prior map. Experiments show that the proposed algorithm can provide centimeter-level global positioning results with scale, which is effortlessly integrated and favorable for low-cost robot system deployment in real-world applications.

Index Terms—Localization, Robotics in Under-Resourced Settings, Sensor Fusion

I. INTRODUCTION

A localization module that estimates the robot systems' states, e.g. position and orientation, is fundamental to downstream tasks like navigation, obstacle avoidance, decision-making, etc. From a systematic perspective, the state estimation results need to be represented under a global reference frame to effectively collaborate with perception information or prior annotations like the position of traffic lights and drivable areas. Since the Global Positioning System (GPS) is denied occasionally in places like urban canyons, mainstream solutions usually rely on an accurate global offline map generated before the autonomous operation [1].

In the domain of autonomous driving, usually, a centimeter-level localization precision is preferred. Achieving this level of localization accuracy demands innovative approaches and the fusion of multiple data sources. LiDAR-based methods [2]–[5] stand out benefiting from the capability of LiDAR to provide high-precision depth measurements. However, LiDAR sensor remains relatively expensive, need higher power consumption, and is more fragile to physical

¹ The authors are with the Thrust of Robotics and Autonomous Systems, The Hong Kong University of Science and Technology (Guangzhou), Nansha District, Guangzhou, China. ² The author is with the Baidu autonomous driving technology department (ADT). ³ The authors are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. ⁴ The author is with the Department of Computer Science, University College London, Gower Street, WC1E 6BT, London, UK. ⁵ The author is with the School of Electronics and Information Engineering, Tongji University, Shanghai, China. ⁶ The author is with the HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute, Shenzhen, China. *The first two authors are equally contributed. Corresponding author: Jianhao Jiao ucacjji@ucl.ac.uk.

This work was supported by the National Natural Science Foundation of China under Grants (62333017, 62173248), Guangdong Basic and Applied Basic Research Foundation (No. 2021B1515120032), Guangzhou-HKUST(GZ) Joint Funding Program (No. 2024A03J0618), and Project Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone(HZQB-KCZYB-2020083), awarded to Prof. Ming Liu.

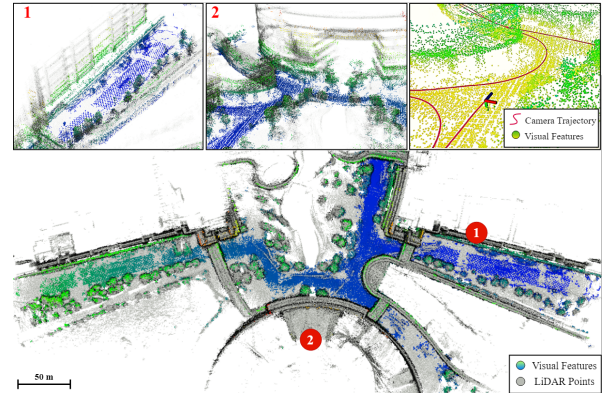


Fig. 1: Image tracking in the visual feature map generated by the proposed method (top-right). The feature map can well capture both the visual detail and the precise geometry structure of the environment (top-left, top-mid, bottom).

shock when compared to cameras, posing challenges for mass-market deployment.

Cameras on the other hand recover the depth information in an indirect way, which serves as a significant alternative for robot localization. Taking visual map construction as the first step, prevalent camera-centric methods extract features such as keypoints, edges, and textures and match them in the image sequence. The camera poses together with the 3D coordinates of the features are estimated by bundle adjustment (BA), this process is also known as structure from motion (SfM) [6]. A stereo setup or extra inertial measurement unit (IMU) is needed to help recover the absolute scale of the scene [7]–[9]. The recovered visual map can be used as a reference for further localization to ensure a fixed global frame and reduce drift [10]. Along this track, maplab [11] proposed a multi-session visual-inertial mapping framework to improve the efficiency of large-scale visual mapping. Lynen *et al.* [12], [13] employed the map and descriptor compression scheme to resolve the online computation burden for large-scale, real-time tracking on the visual map. However, it is still challenging to maintain the computational efficiency and accuracy of both SfM and localizing against a SfM model, especially compared to LiDAR-based methods. For example, without precise depth measurements, visual localization systems generally exhibit a lower level of precision than LiDAR. However, the rich low-level texture and high-level semantic information captured in the image still make the camera irreplaceable in the robotic system, especially localization.

To narrow the gap between visual-based and LiDAR-based localization methods, a recent branch of research [14]–[22] focuses on leveraging the high-precision LiDAR point cloud

in the visual state estimation process. Caselitz *et al.* [14] tracked the 7 degrees of freedom (DoF) camera pose with scale by reconstructing 3D points from image features and continuously matching them against a given 3D LiDAR map using Iterative Closest Point (ICP) scheme. Kim *et al.* [15] used a stereo camera setup and tracked the stereo disparity map against the 3D LiDAR map via minimizing the depth residual. Zuo *et al.* [17] proposed to fuse the prior LiDAR map constraints into a MSCKF stereo-visual inertial framework by performing normal distribution transform (NDT)-based registration between the visual semi-dense reconstruction and prior map. Yu *et al.* [16] extracted line features from both image and LiDAR map and the camera pose is recovered using 2D-3D line correspondences. Huang *et al.* [18] modeled the prior map by the Gaussian Mixture Model (GMM). The camera poses and visual structure are bundle adjusted considering hybrid structure constraint. Zhang *et al.* [19] aligned the point clouds from vision and a semantic LiDAR map by a point-to-plane ICP algorithm utilizing semantic consistency. Maplab 2.0 [20] extended the multi-session visual-inertial mapping framework [11] into a general multi-modal mapping system. Inter-session LiDAR registration can be used for loop-closure detection to align reference frames across sessions and mitigate drift. Oishi *et al.* [21], [22] generated rendered image from the prior map, and the localization is performed in a simultaneous tracking and rendering manner.

Different from the above-mentioned methods, the proposed method doesn't require a known initial pose during localization, and LiDAR fusion is only performed at an offline mapping stage, keeping the online localization lightweight, easy to deploy but still accurate. We outline the key challenges for this problem as follows:

1) *Cross-Modality*: Establishing correspondence between LiDAR point cloud and images can be challenging due to the difference in representation, sensing range, and density.

2) *Relocalization Ability*: Relocalization ability is key to a positioning system, as the initial pose for tracking in a prior map is hard to retrieve in practice, especially in a GPS-denied scenario. This ability is also essential for a positioning system to recover from localization failure [23].

3) *Map Representation*: Operating on the raw point cloud is inefficient when searching for correspondences from the prior map. Map representation requires careful design based on the usage scenario and also the data association strategy.

4) *Long-term Operation*: For long-term localization, the system may suffer from environment changes due to factors like lighting conditions, seasonal variations, construction etc., which may affect the data association reliability.

5) *Real-time Performance*: Extra operations like depth estimation, data association, etc. limit the performance of pose estimation for large-scale outdoor applications, especially for embedded platforms with strict resource limits.

In this paper, we proposed a novel system using only a monocular camera to achieve comparable onboard localization performance by tracking deep-learning visual features on a LiDAR-enhanced visual prior map. Addressing the key

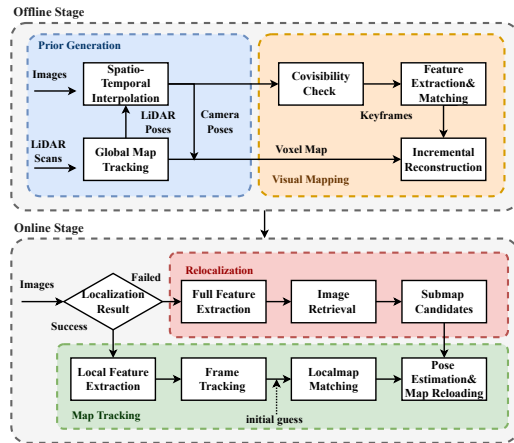


Fig. 2: System pipeline of the proposed method.

challenges mentioned above, the proposed method has the following features:

- Heavy computations are kept in an offline mapping stage and only lightweight tracking is performed during online localization to ensure real-time performance.
- The keyframe poses with scale and a high-precision visual feature map are jointly recovered and optimized in a geometry-aware bundle adjustment problem, considering both visual and geometry constraints.
- Volumetric representation enables efficient offline ray tracing for visual-LiDAR data association, while the co-visibility graph of keyframes facilitates quick online visual submap retrieval.
- A hierarchical localization system suitable for long-term, real-time applications, offering accurate and robust online re-localization and feature tracking.

The proposed localization system is not only evaluated on a benchmark dataset but is also deployed in a real-world autonomous driving platform, demonstrating the effectiveness of the proposed method.

II. PROPOSED METHOD

A. Notation

In the following part of this paper, we use bold uppercase to denote matrix, and bold lowercase for vector, e.g. $\mathbf{T} \in \text{SE}(3)$ for rigid body transformation and $\mathbf{t} \in \mathbb{R}^3$ for translation. Rotation is defined by the rotation matrix $\mathbf{R} \in \text{SO}(3)$ and quaternion $\mathbf{q} \in \mathbb{H}$. $\mathbf{p} = [x, y, z]^T \in \mathbb{R}^3$ denotes 3D point coordinate, $\mathbf{x} = [u, v]^T \in \mathbb{R}^2$ for 2D pixel coordinate. The coordinates and frame index are clarified at the superscript and subscript, e.g. the camera pose under the world frame at time i is denoted as \mathbf{T}_i^{wc} . The brace $\{\cdot\}$ is introduced to represent a set of data, e.g. $\{\mathbf{x}_j\}$ denotes a set of image pixels. Other symbols and variables will be introduced and defined within their respective sections.

B. System Overview

The architecture of the proposed method is illustrated in Fig. 2, where dashed rounded rectangle with different color represents different submodule of the system. The whole system includes an offline mapping stage and an online

localization stage. During the offline mapping stage, our goal is to build a prior map, which is further used for localization purpose. The offline stage is separated into two steps, the prior generation step which will be discussed in Sec.II-C, and the visual mapping step which will be discussed in Sec.II-D. During the online localization stage, the camera pose is estimated by efficiently tracking the image sequence on the prior map, which will be discussed in Sec.II-E.

C. Prior Generation

Prior generation takes LiDAR scans and monocular image sequence as input, and is responsible for providing pose priors of each image within the reference global point cloud map \mathcal{M}^{ref} . Meanwhile, it produces a voxel map of the operation area generated from the LiDAR scans, serving as geometry priors in the visual mapping stage.

\mathcal{M}^{ref} is first converted into K-D tree and point-wise covariance matrix $\mathbf{C}^{\text{ref}} = \{\mathbf{C}_i^{\text{ref}}\}$. Then for each incoming LiDAR scan $\mathcal{P}_k = \{\mathbf{p}_{k,i}\}$, we perform similar preprocessing step and get \mathbf{C}^k . The transformation \mathbf{T}_k^{wl} of each LiDAR scan under global map frame can be solved by scan-to-scan (eq. (1)) and scan-to-map (eq. (2)) two-stage GICP (Generalized Iterative Closest Point) registration:

$$\hat{\mathbf{T}}_k^{(k-1)} = \arg \min_{\mathbf{T}_k^{(k-1)}} \sum_i^N \mathbf{d}_i^\top \left(\mathbf{C}_i^{(k-1)} + \right. \quad (1)$$

$$\left. \mathbf{T}_k^{(k-1)} \mathbf{C}_i^k \mathbf{T}_k^{(k-1)\top} \right)^{-1} \mathbf{d}_i, \quad (1)$$

$$\hat{\mathbf{T}}_k = \arg \min_{\mathbf{T}_k} \sum_j^M \mathbf{d}_j^\top \left(\mathbf{C}_j^{\text{ref}} + \mathbf{T}_k \mathbf{C}_j^k \mathbf{T}_k^\top \right)^{-1} \mathbf{d}_j. \quad (2)$$

where $\mathbf{d}_i = \mathbf{p}_i^{k-1} - \mathbf{T}_k^{(k-1)} \mathbf{p}_i^k$ and $\mathbf{d}_j = \mathbf{p}_j^{\text{ref}} - \mathbf{T}_k \mathbf{p}_j^k$ indicate the point-wise distance error, and \mathbf{T}_k with $\mathbf{T}_{(k-1)} \mathbf{T}_k^{(k-1)}$ as initial guess. A detailed description of the GICP matching process can be referred to [5].

When the LiDAR sensor and camera is not hard synchronized, to get the prior pose of the mapping images, the interpolated pose $\mathbf{T}_k = (\mathbf{R}_k, \mathbf{t}_k)$ at timestamp k can be calculated by the nearest two neighbor frames at time l and r using the linear interpolation $\mathbf{t}_k = \mathbf{t}_l + s \cdot (\mathbf{t}_r - \mathbf{t}_l)$ for translation and spherical linear interpolation (slerp) $\mathbf{R}_k = \text{slerp}(\mathbf{R}_l, \mathbf{R}_r, s)$ for rotation, with interpolation ratio $s = (k - l)/(r - l)$. The final interpolated LiDAR poses are further transformed to the camera frame using the sensor extrinsics \mathbf{T}^{cl} .

Besides the camera trajectories, we also use the LiDAR sequence to build a reference UFOMap [24] of the task area. This step is for two reasons, firstly rebuilding the reference map can make sure the prior data is up-to-date and consistent with the information from the mapping image sequence. Also, this volumetric representation is more compact compared to raw point cloud and is superior in data loading and traversal.

D. Geometry-Aware Visual Mapping

In this step, the mapping images are fed into a SfM pipeline, along with the prior camera poses and the reference voxel map to recover the 3D geometry of the scene using visual features.

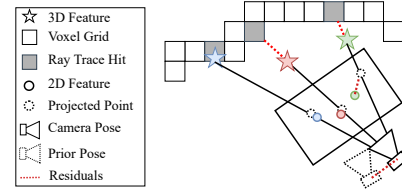


Fig. 3: Illustration of the geometry-aware bundle adjustment.

1) *Incremental Reconstruction*: We first sample keyframe images on the reference trajectory, ensuring a well-rounded understanding of the scene and also reducing data redundancy. If the relative translation and rotation between two keyframes fall below a specific threshold, they are considered to have co-observation, prompting visual matching for subsequent reconstruction. The SfM pipeline then leverages learning-based feature extractors SuperPoint [25] to detect keypoints with local descriptors and NetVLAD [26] to extract global image descriptors for similarity retrieval. Later, SuperGlue [27] is used for local feature matching of connected neighbors, enabling reconstruction based on the established visual feature correspondences.

We then follow an incremental SfM scheme [28]–[30] to reconstruct the visual map. New images are progressively added to the problem by matching against the existing images followed by a BA step to refine the camera poses and 3D feature positions of an active image subset. The full reconstruction would be refined periodically by global optimization to further improve the reconstruction quality.

2) *Geometry-aware Bundle Adjustment (GABA)*: Since visual reconstruction using only monocular images has ambiguity in scale and may drift over time, in this paper, we propose to take full advantage of the prior information generated at Sec.II-C by GABA to help recover scale and also improve the geometry consistency of the feature map against real-world structure. With \mathcal{T} and \mathcal{M}^{vis} be the set of camera poses and visual map point positions to be optimized, a GABA process can be modeled by an optimization problem:

$$\hat{\mathcal{T}}, \hat{\mathcal{M}}^{\text{vis}} = \arg \min_{\mathcal{T}, \mathcal{M}^{\text{vis}}} \sum_{i=1}^{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{M}^{\text{vis}}|} \rho \left(\underbrace{\left\| \mathbf{x}_j^i - \pi(\mathbf{T}_i, \mathbf{p}_j) \right\|_{\Sigma}}_{\text{visual factor}} \right) + \quad (3)$$

$$\underbrace{\rho \left(\left\| \mathbf{p}_j - \mathbf{r}(\hat{\mathcal{M}}^{\text{ref}}, \mathbf{T}_i, \mathbf{p}_j) \right\|_{\Sigma} \right)}_{\text{structure factor}} + \underbrace{\rho \left(\left\| \mathbf{e}(\mathbf{T}_i, \bar{\mathbf{T}}_i) \right\|_{\Sigma} \right)}_{\text{prior factor}}.$$

where $\rho(\cdot)$ is the Huber norm to reduce the influence of outliers in the optimization, and $\|x - \mu\|_{\Sigma} = (x - \mu)^\top \Sigma^{-1} (x - \mu)$ is the Mahalanobis norm to encounter the covariance of the residuals. For each image I_i , \mathcal{M}_i denotes the set of visual map points visible at frame i .

The visual factor measures the disparity between observed position x_j^i of \mathbf{p}_j at image i and the corresponding projected position using projection function $\pi(\mathbf{T}_i, \mathbf{p}_j) = \mathbf{K}(\mathbf{T}_i^{-1} \cdot [\mathbf{p}_j^\top, 1]^\top)$ with \mathbf{K} denotes the intrinsic matrix of the camera. The structure factor measures the geometric difference between the estimated visual map \mathcal{M}^{vis} and the prior LiDAR voxel map $\hat{\mathcal{M}}^{\text{ref}}$. The data association between the two maps can be found by an indirect or direct manner.

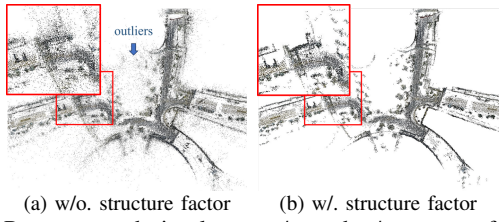


Fig. 4: Reconstructed visual map w/o. and w/. structure factor. Introducing geometry constraints in visual mapping greatly improves the overall reconstruction quality.

In an indirect setting, map points are projected to the image plane according to the projection matrix. However, unlike LiDAR, cameras do not have a specified maximum sensing range, it is hard to cull the visible subarea of the prior map to process, making it inappropriate for large-scale outdoor scenes. In a direct setting, for each feature point \mathbf{p}_j , the corresponding reference structure is obtained by a ray tracing operation that simulates the path of a light ray starting from the camera center to the feature point and intersects with the reference map \mathcal{M}^{ref} . The euclidean distance between the map point position \mathbf{p}_j and the ray tracing hit point $\bar{\mathbf{p}}_j = \mathbf{r}(\mathcal{M}^{\text{ref}}, \mathbf{T}_i, \mathbf{p}_j)$ is added as residual. The ray tracing operation would be repeated after each optimization round to update data association. After several rounds of optimization, the reconstructed map would be geometrically consistent with the LiDAR map structure.

We also use the prior camera poses discussed in Sec.II-C as anchor points to prevent the optimized pose going too far away from the initial guess. The prior pose residual can be modeled directly by the absolute difference between the estimated pose and the prior pose:

$$\mathbf{e}(\mathbf{T}_i, \bar{\mathbf{T}}_i) = \begin{bmatrix} \bar{\mathbf{t}} - \mathbf{t} \\ 2 \cdot \mathbf{q}^{-1} \bar{\mathbf{q}} \end{bmatrix} \in \mathbb{R}^6. \quad (4)$$

Following the resolution of the GABA, the 3D positions of the inlier deep learning keypoints are subsequently refined. This optimization process ensures alignment of the reconstructed visual map \mathcal{M}^{vis} with the global reference frame and also maintains both scale and geometric coherence with the prior map $\bar{\mathcal{M}}^{\text{ref}}$. Fig. 3 gives an intuitive illustration of the GABA problem and a qualitative comparison of the visual mapping result is presented in Fig. 4.

E. Visual Localization

After the above visual mapping process, in this section, we will describe how to leverage the reconstructed visual prior map to get high-precision localization results. As shown in Fig. 2, during the online stage, the workflow goes into two branches, relocalization and map tracking, according to the current localization state.

The relocalization branch tries to provide pose estimation results when no initial guess is provided. Following the hierarchical localization scheme [31], the relocalization branch starts with extracting the global and local features $\mathcal{F}_i^g, \mathcal{F}_i^l = \{(\mathbf{x}_j^i, \mathbf{f}_j^i)\}$ of the input image I_i . The global features are used to retrieve k -nearest candidate keyframes and the corresponding connected clusters from the database.

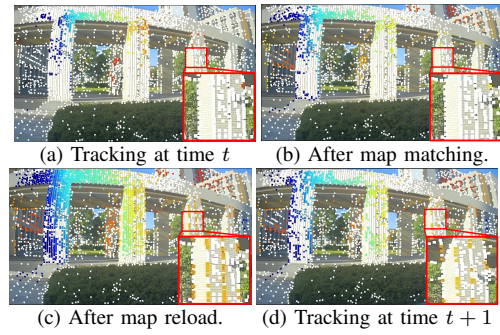


Fig. 5: Illustration of the feature tracking process. The tracked features are colored based on depth, free keypoints are colored by white. After the map reloading step, more map points that are not tracked at the map matching step are kept in the next frame.

These candidate keyframes and connected cluster indicates the most possible positions of the input image. After that, the local features are used to perform keypoint matching between the input image and the candidate clusters. Since the 3D locations of the database keypoints are precomputed in the offline mapping stage, the final camera pose \mathbf{T}_i can then be estimated using the PnP algorithm together with RANSAC outlier rejection. After the PnP-RANSAC optimization step, inlier matches are updated with the map point positions from the database and we get $\mathcal{F}_i^l = \{(\mathbf{x}_j^i, \mathbf{f}_j^i, \mathbf{p}_j)\}$.

Successful relocalization leads to the camera tracking branch, where the input image’s pose is recovered following a coarse-to-fine pose estimation scheme:

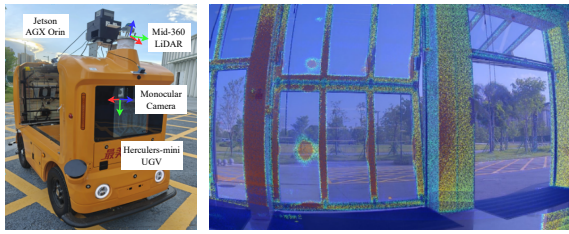
1) *Frame Tracking*: As shown in Fig. 5, we first perform LK optical flow [32] estimation to track the pixel locations of local features $\{(\mathbf{x}_j^{i-1}, \mathbf{p}_j)\}$ from last frame into the current image plane $\{(\bar{\mathbf{x}}_j^i, \mathbf{p}_j)\}$. Then we use these coarse map point associations to recover the camera pose $\bar{\mathbf{T}}_i$. This coarse estimation is then used as the prior pose of the following local map-matching step.

2) *Local Map Matching*: The local map $\mathcal{M}^{i-1} = \{(\mathbf{f}_j, \mathbf{p}_j)\}$ consists of all the map points visible in the connected clusters \mathcal{C}_{i-1} of last frame. We then need to establish data association between the latest local map \mathcal{M}^{i-1} and current frames’ local features $\mathcal{F}_i^l = \{(\mathbf{x}_k^i, \mathbf{f}_k^i)\}$. Each associated map point-keypoint pair should satisfy two criteria: first, the keypoint pixel \mathbf{x}_k^i should be near the reprojected pixel $\bar{\mathbf{x}}_j^i = \pi(\bar{\mathbf{T}}_i, \mathbf{p}_j)$, and second, the keypoint descriptor \mathbf{f}_k^i should be similar to the map point descriptor \mathbf{f}_j , this can be modeled into a bi-parted graph matching problem. For each map point \mathbf{p}_j we first use the reprojected pixel to search for features within a search radius r , and then associate it to the keypoint with the smallest descriptor difference.

3) *Pose Optimization and Map Point Reloading*:

$$\hat{\mathbf{T}}_i = \arg \min_{\mathbf{T}_i} \sum_{j=1}^{\|\mathcal{M}^{i-1}\|} \underbrace{\rho \left(\left\| \mathbf{x}_j^i - \pi(\mathbf{T}_i, \mathbf{p}_j) \right\|_{\Sigma} \right)}_{\text{visual factor}} + \underbrace{\rho \left(\left\| \mathbf{e}(\mathbf{T}_i, \bar{\mathbf{T}}_i) \right\|_{\Sigma} \right)}_{\text{prior factor}}. \quad (5)$$

After matching against the map frame, the camera pose of the current frame is refined by solving eq. (5), which is quite



(a) Sensor configuration. (b) Calibration result by [33].

Fig. 6: Illustration of the experiment platform. The LiDAR is mounted 15° front-facing to cover the camera’s field of view.

similar to eq. (3), but only camera pose is optimized. The prior pose we use in eq. (5) can directly come from the frame tracking step.

After the pose optimization, the outlier map point-keypoint matches would be removed $\mathcal{F}_i^{-l} = \{(\mathbf{x}_k^i, \mathbf{f}_k^i, null)\}$, and for the inlier matches $\mathcal{F}_i^{+l} = \{(\mathbf{x}_k^i, \mathbf{f}_k^i, \mathbf{p}_j)\}$, we update the map point descriptor in the database $\mathcal{M}^{i-1} = \{(\mathbf{f}_j, \mathbf{p}_j)\}$ by descriptor from current observation $\hat{\mathcal{M}}^{i-1} = \{(\hat{\mathbf{f}}_k^i, \mathbf{p}_j)\}$ to ensure inter-frame continuity. Subsequently, as shown in Fig. 5, a refined local map matching operation with a very small search radius is performed to augment map point-keypoint candidates, enhancing localization by tracking more map points and ensuring system stability.

III. EXPERIMENTAL RESULTS

In this section, the proposed system will be assessed thoroughly in real-world captured and benchmark datasets to evaluate its performance and effectiveness in real-world applications.

1) *Campus Dataset*: As shown in Fig. 6, a Hercules mini [34] carrying a Livox Mid-360 LiDAR and a monocular camera is used as the data acquisition platform. The LiDAR is ingeniously affixed to a detachable base, which can be unmounted from the system for mass and low-cost localization deployment once the prior map is built.

The campus dataset contains 23 sequences, each capturing 10 Hz 1280×720 RGB images and 10 Hz LiDAR scans. With a maximum driving speed of 15 km/h and covering an approximate total mileage of 25 km across 5 distinct unmanned delivery operation routes, this dataset is specifically designed to facilitate navigation testing for robotic systems.

2) *KITTI Dataset*: To better evaluate the localization performance of our method, we use KITTI Odometry dataset [35], [36] for benchmarking. The KITTI odometry dataset is captured by driving around the mid-size city of Karlsruhe, covering common outdoor automatic driving scenarios. The dataset contains time-synchronized undistorted stereo RGB images and LiDAR scans.

The evo library [37] is used for trajectory evaluation. Before each assessment, we perform SE(3) Umeyama alignment between the reference and estimated trajectories.

A. Evaluation of All-day Localization Ability

Since visual features may behave inconsistently under different lighting conditions, we assess the all-day localization performance of the proposed method by evaluating its



Fig. 7: Illustration of the all-day localization dataset.

TABLE I: APE (m) for Different Combinations of Mapping and Localization Time (AM/PM)

$t_{\text{map}} \backslash t_{\text{loc}}$	10	12	15	16	18	19	20
10	0.066	0.064	0.063	0.067	0.065	0.069	1.687
12	0.063	0.054	0.055	0.057	0.055	0.059	0.070
15	0.067	0.057	0.051	0.054	0.054	0.057	-
16	0.061	0.055	0.051	0.053	0.053	0.058	-
18	0.062	0.057	0.052	0.052	0.053	0.052	4.055
19	0.057	0.054	0.050	0.053	0.050	0.048	0.048
20	4.163	0.056	0.053	1.933	0.384	0.050	0.047

precision using prior maps captured at different hours within a day. As illustrated in Fig. 7, several sequences on the same route at different moments are captured, remaining similar scene structures but showing different illumination. We went through different combinations of mapping and localization sequences to prove that the proposed method has a certain adaptability to scene illumination.

The mean absolute pose error (APE) results are presented in Table.I. For better illustration, we organize the pairwise mean APE and also the average tracked features per frame in matrix form and visualize them by false color rendering (Fig. 8). The proposed method shows consistent localization ability on mapping sequences collected in most of the daytime unless it gets completely dark (around 20 PM in our case, e.g. in test 20–10 and test 15–20 large pose error or tracking lost is observed, the average features tracked per frame is also low, possibly due to bad data associations).

B. Evaluation on Campus dataset

In this section, we evaluate the localization performance on four different scenes in our campus dataset. The reference LiDAR map of the campus is pre-generated using a hand-held data acquisition platform [38] and the localization accuracy is evaluated with respect to this global map. For each scene, we pick two mapping sequences and generate two visual maps. All sequences are used for localization on both maps if not chosen for mapping. LiDAR trajectories in the global map generated by Sec.II-C serve as prior and ground truth for mapping and localization, respectively. The mean APE is calculated based on the experiments conducted for each scene. We compare the proposed method with the hloc [31] toolkit, which contains similar global image retrieval and local feature matching settings but is designed for general visual localization purpose. The experiment result is presented in Table. II. The original hloc pipeline failed to generate reasonable reconstruction results in scene 04 and observed severe drift in scene 01–03. By combining hloc

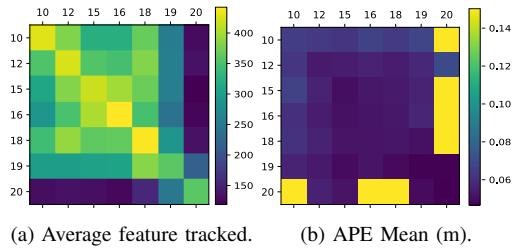


Fig. 8: False color rendering of all-day localization results with mapping time on the vertical axis and localization time on the horizontal axis.

TABLE II: Localization Comparison on Campus Dataset

Scene	hloc [31]		hloc + our map		Proposed	
	APE(m)	RPE(m)	APE(m)	RPE(m)	APE(m)	RPE(m)
01	97.983	1.952	0.100	0.107	0.113	0.035
02	138.669	2.100	0.161	0.117	0.115	0.053
03	75.198	1.793	0.099	0.101	0.106	0.045
04	-	-	0.095	0.098	0.099	0.036

with the feature map generated by the proposed method (Sec. II-D), the estimated absolute pose is reliable. Furthermore, the proposed hierarchical and lightweight localization method can achieve the same level of accuracy without using a learning-based feature matcher (e.g. Superglue [27], run only at around 2 Hz for top-10 retrieved candidates of each query) and significantly reduce the mean relative pose error (RPE) at the same time. This indicates our system is suitable for robotic navigation tasks, which require high localization accuracy while restricting computational resources.

C. Evaluation on KITTI Dataset

In this experiment, we first use the SemanticKITTI API [39] to merge all the LiDAR scans into the reference point cloud map using loop-closed LiDAR trajectories. For workflow consistency, a similar LiDAR tracking procedure is done to generate the reference LiDAR trajectories for visual mapping purpose. Since the proposed method is mainly suitable for localization in repeated operation routes, the left color image would be used to generate the visual feature map and the right image would be used for localization test. Localization results would be compared with loop-closed LiDAR trajectories. We compare our method against [15], which also focuses on outdoor camera tracking using prior map and evaluated on the KITTI dataset except highway sequence 01 due to the limited amount of features in the scene. The comparison result is presented in Table.III, and we reuse the result presented in [15]. We can see that the proposed method can achieve centimeter-level accuracy on the KITTI dataset. It's worth pointing out that, our localization result is not compared against the provided ground-truth pose, because the prior point cloud map generated using ground-truth poses is of low quality. However, this experiment is sufficient to show that the localization result provided by our method has strong consistency against the prior map, which is more desirable in many autonomous driving applications as described in Sec. I.

TABLE III: Localization Comparison on KITTI Dataset

Sequence	Proposed		Kim <i>et al.</i> [15]	
	APE(m)	APE(°)	APE(m)	APE(°)
00	0.048	0.013	0.133	0.322
02	0.149	0.016	0.221	0.326
03	0.031	0.008	0.237	0.413
04	0.041	0.045	0.450	0.876
05	0.035	0.007	0.146	0.340
06	0.042	0.008	0.375	0.849
07	0.040	0.006	0.131	0.487
08	0.049	0.007	0.144	0.328
09	0.040	0.007	0.180	0.338
10	0.039	0.006	0.240	0.493

TABLE IV: Localization Time Cost (ms)

Steps	Desktop	Orin
Global Feature Extraction	645	1180
Relocalization	330	450
Local Feature Extraction	15	35
Tracking	31	37
Localization	45	72

D. Localization Runtime Analysis

The localization runtime performance of the proposed system is measured in two different platforms: an Intel Core i9-12900KF desktop with GeForce RTX 3080Ti and a Nvidia Jetson AGX Orin (32G). The feature extractors are speeded up using the Torch-TensorRT engine, and we use half-precision for Orin platform. The time cost result is presented in Table.IV, in which the localization step is the sum of local feature extraction and tracking steps. The localization takes only 72 ms on the Orin platform, which is sufficient for real-time localization of a 10 Hz image input and is suitable for onboard applications.

E. Unmanned Delivery Deployment

The proposed localization system is adapted to an autonomous delivery system to track a preset delivery route. The proposed localization method can well cooperate with downstream modules like path planning and control, demonstrating that the proposed system can serve as a low-cost substitution for a LiDAR-based localization module. The result is presented in the attached media.

IV. CONCLUSIONS

In this paper, a localization system using only a monocular camera is proposed. A high-quality visual prior map that is geometrically consistent with real-world structure is built by considering LiDAR structure constraint, prior pose constraint, and visual feature constraint in the proposed geometry-aware bundle adjustment problem. By tracking deep-learning features on this prior map, high-precision localization results are provided with a real-time guarantee, which is favorable for low-cost robot system deployment in real-world applications.

In future work, the proposed method will be further optimized by integrating IMU in a tightly coupled configuration and also adapting the current pipeline to multi-camera systems or panoramic cameras to improve the robustness in challenging featureless scenarios.

REFERENCES

- [1] Claudine Badue, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius B Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago M Paixao, Filipe Mutz, et al. Self-driving cars: A survey. *Expert Systems with Applications*, 165:113816, 2021.
- [2] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and systems*, volume 2, pages 1–9. Berkeley, CA, 2014.
- [3] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765. IEEE, 2018.
- [4] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 38(4):2053–2073, 2022.
- [5] Kenny Chen, Brett T Lopez, Ali-akbar Agha-mohammadi, and Ankur Mehta. Direct lidar odometry: Fast localization with dense point clouds. *IEEE Robotics and Automation Letters*, 7(2):2000–2007, 2022.
- [6] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 203(1153):405–426, 1979.
- [7] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.
- [8] Jakob Engel, Jörg Stückler, and Daniel Cremers. Large-scale direct slam with stereo cameras. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 1935–1942. IEEE, 2015.
- [9] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.
- [10] Helen Oleynikova, Michael Burri, Simon Lynen, and Roland Siegwart. Real-time visual-inertial localization for aerial and ground robots. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3079–3085. IEEE, 2015.
- [11] T. Schneider, M. T. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart. maplab: An Open Framework for Research in Visual-inertial Mapping and Localization. *IEEE Robotics and Automation Letters*, 3(3):1418–1425, 2018.
- [12] Simon Lynen, Torsten Sattler, Michael Bosse, Joel A Hesch, Marc Pollefeys, and Roland Siegwart. Get out of my lab: Large-scale, real-time visual-inertial localization. In *Robotics: Science and Systems*, volume 1, page 1, 2015.
- [13] Simon Lynen, Bernhard Zeisl, Dror Aiger, Michael Bosse, Joel Hesch, Marc Pollefeys, Roland Siegwart, and Torsten Sattler. Large-scale, real-time visual-inertial localization revisited. *The International Journal of Robotics Research*, 39(9):1061–1084, 2020.
- [14] Tim Caselitz, Bastian Steder, Michael Ruhnke, and Wolfram Burgard. Monocular camera localization in 3d lidar maps. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1926–1931. IEEE, 2016.
- [15] Youngji Kim, Jinyong Jeong, and Ayoung Kim. Stereo camera localization in 3d lidar maps. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9. IEEE, 2018.
- [16] Huai Yu, Weikun Zhen, Wen Yang, Ji Zhang, and Sebastian Scherer. Monocular camera localization in prior lidar maps with 2d-3d line correspondences. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4588–4594, 2020.
- [17] Xingxing Zuo, Patrick Geneva, Yulin Yang, Wenlong Ye, Yong Liu, and Guoquan Huang. Visual-inertial localization with prior lidar map constraints. *IEEE Robotics and Automation Letters*, 4(4):3394–3401, 2019.
- [18] Huaiyang Huang, Haoyang Ye, Yuxiang Sun, and Ming Liu. Gmmloc: Structure consistent visual localization with gaussian mixture models. *IEEE Robotics and Automation Letters*, 5(4):5043–5050, 2020.
- [19] Chi Zhang, Hengwang Zhao, Chunxiang Wang, Xuanlai Tang, and Ming Yang. Cross-modal monocular localization in prior lidar maps utilizing semantic consistency. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4004–4010, 2023.
- [20] A. Cramariuc, L. Bernreiter, F. Tschopp, M. Fehr, V. Reijgwart, J. Nieto, R. Siegwart, and C. Cadena. maplab 2.0 – A Modular and Multi-Modal Mapping Framework. *IEEE Robotics and Automation Letters*, 8(2):520–527, 2023.
- [21] Shuji Oishi, Yasunori Kawamata, Masashi Yokozuka, Kenji Koide, Atsuhiko Banno, and Jun Miura. C*: Cross-modal simultaneous tracking and rendering for 6-dof monocular camera localization beyond modalities. *IEEE Robotics and Automation Letters*, 5(4):5229–5236, 2020.
- [22] Shuji Oishi, Kenji Koide, Masashi Yokozuka, and Atsuhiko Banno. Lc*: Visual-inertial loose coupling for resilient and lightweight direct visual localization. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4033–4039. IEEE, 2023.
- [23] Konstantinos A. Tsintotas, Loukas Bampis, and Antonios Gasteratos. The revisiting problem in simultaneous localization and mapping: A survey on visual loop closure detection. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):19929–19953, 2022.
- [24] Daniel Duberg and Patric Jensfelt. UFOMap: An efficient probabilistic 3D mapping framework that embraces the unknown. *IEEE Robotics and Automation Letters*, 5(4):6411–6418, 2020.
- [25] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.
- [26] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.
- [27] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.
- [28] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [29] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.
- [30] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 127–134. IEEE, 2013.
- [31] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019.
- [32] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, volume 2, pages 674–679, 1981.
- [33] Chongjian Yuan, Xiyuan Liu, Xiaoping Hong, and Fu Zhang. Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments. *IEEE Robotics and Automation Letters*, 6(4):7517–7524, 2021.
- [34] Tianyu Liu, Qing hai Liao, Lu Gan, Fulong Ma, Jie Cheng, Xupeng Xie, Zhe Wang, Yingbing Chen, Yilong Zhu, Shuyang Zhang, et al. The role of the hercules autonomous vehicle during the covid-19 pandemic: An autonomous logistic vehicle for contactless goods transportation. *IEEE Robotics & Automation Magazine*, 28(1):48–58, 2021.
- [35] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [36] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.
- [37] Michael Grupp. evo: Python package for the evaluation of odometry and slam. <https://github.com/MichaelGrupp/evo>, 2017.
- [38] Jianhao Jiao, Hexiang Wei, Tianshuai Hu, Xiangcheng Hu, Yilong Zhu, Zhijian He, Jin Wu, Jingwen Yu, Xupeng Xie, Huaiyang Huang, Ruoyu Geng, Lujia Wang, and Ming Liu. Fusionportable: A multi-sensor campus-scene dataset for evaluation of localization and mapping accuracy on diverse platforms, 2022.
- [39] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.