Visual-LiDAR Odometry and Mapping with Monocular Scale Correction and Visual Bootstrapping

Hanyu Cai[1], Ni Ou[1] and Junzheng Wang[1,*]

Abstract— **This paper presents a novel visual-LiDAR odometry and mapping method with low-drift characteristics. The proposed method is based on two popular approaches, ORB-SLAM and A-LOAM, with monocular scale correction and visual-bootstrapped LiDAR poses initialization modifications. The scale corrector calculates the proportion between the depth of image keypoints recovered by triangulation and that provided by LiDAR, using an outlier rejection process for accuracy improvement. Concerning LiDAR poses initialization, the visual odometry approach gives the initial guesses of LiDAR motions for better performance. This methodology is not only applicable to high-resolution LiDAR but can also adapt to low-resolution LiDAR. To evaluate the proposed SLAM system's robustness and accuracy, we conducted experiments on the KITTI Odometry and S3E datasets. Experimental results illustrate that our method significantly outperforms standalone ORB-SLAM2 and A-LOAM. Furthermore, regarding the accuracy of visual odometry with scale correction, our method performs similarly to the stereo-mode ORB-SLAM2.**

I. Introduction

Simultaneous Localization and Mapping (SLAM) is an irreplaceable technique for mobile robots and autonomous vehicles, providing reliable surrounding environment information and real-time positions. According to the use of sensors, this technique can be divided into two categories: visual-based and LiDAR-based. Over the past two decades, visual SLAM has made significant strides, resulting in commercially available frameworks. Modern visual SLAM algorithms develop into two branches: feature-based and direct methods. Feature-based methods [1], [2] reduce the reprojection error of matched feature points (keypoints) through bundle adjustment (BA) [3]. On the other hand, direct methods normally optimize the photometric error of sparse keypoints without corresponding matchings [4], [5]. The advantage of visual SLAM is rich semantic information, low cost and small size, which is an indispensable part of the field of automatic driving and AR.

In most cases, LiDAR SLAM usually outperforms visual SLAM. Most recent pure LiDAR SLAM methods are developed based on LOAM [6], a milestone LiDAR SLAM framework combined with SCAN-to-SCAN and SCAN-to-MAP registration modes. These LOAM-based techniques yield superior performance compared to the baseline LOAM, with improvements in efficiency [7], robust registration [8], motion compensation [9] and local optimization [10]. In addition, LiDAR-based loop closure detection techniques [11], [12] have been widely employed for place recognition and graph optimization to reduce the accumulated error of LiDAR SLAM further.

Nonetheless, standalone visual or LiDAR SLAM either has intractable drawbacks. Visual SLAM systems are prone to localization failure [1] in fast motions. On the other hand, for LiDAR SLAM, motion distortion [6], [9] is still a tricky problem for spinning LiDAR, and its loop closure detection is more complicated and challenging due to lack of stable features [12]. It has been a noticeable trend to fuse visual and LiDAR SLAM to enhance the overall performance.

According to the fusion techniques, LiDAR-camera fused SLAM can be divided into three categories: LiDAR-assisted visual SLAM, vision-assisted LiDAR SLAM, and vision-LiDAR coupled SLAM. The first two means rely mainly on LiDAR or camera, and the other sensor takes the assistance role. Moreover, the last type generally utilizes both visual and LiDAR odometry in the system.

The first category tends to focus on image depth enhancement [13] or combines with direct methods without estimating the depth of feature points [14]. The second category has few related studies, and it often uses visual information to help LiDAR SLAM perform loop closure detection or render map texture [15]–[17]. Since this category is not the research content of this paper, we will not describe it in detail. The third category is the hot field of current research, which can be subdivided into loosely coupling and tightly coupling. Loosely coupling is to cascade the two or filter the results of the two [18], [19]. Tightly coupling [20] focuses on constructing a joint optimization problem, including vision and LiDAR factors for state estimation.

Our work is deeply related to depth enhancement. Whereas the error of depth enhancement is significant when the point cloud is sparse, and the feature points with enhanced depth may not be successfully tracked. Directly tracking projected points with high gradients is a solution, but such points cannot be tracked accurately and stably. In this study, we combine the powerful tracking ability of the feature-based method with optical flow and propose a novel scale correction method to address the monocular scale drift problem. Moreover, considering the LOAM algorithm depends on the constant velocity model, it is prone to failure in scenes with excessive acceleration or degradation. Using the results of the visual odometry to initialize the LiDAR odometry's pose can increase the LOAM performance.

The contributions of this paper are as follows:

*This work was supported by State Key Laboratory of Intelligent Control and Decision of Complex Systems

[1] Hanyu Cai, Ni Ou, and Junzheng Wang are with School of Automation, Beijing Institute of Technology, Beijing, China. Hanyu: caihanyu4258@163.com, Ni: 3120205431@bit.edu.cn

* Corresponding author: wangjz@bit.edu.cn

1) A visual-LiDAR loosely coupled odometry. Solve the problem that LOAM fails in degradative scenarios, and increase the performance.
2) A novel scale correction algorithm is proposed that does not need to enhance the depth of the visual feature point. It guarantees that the output of the visual odometry will not have a significant drift.
3) Implement our system on a large-scale dataset and verify its effectiveness.

This paper is organized as follows. Section II presents studies related to our work. Section III introduces the proposed loosely coupled system and our scale correction algorithm. Section IV shows the experimental datasets and results. Finally, Section V demonstrates our conclusion and possible extensions to our work.

II. Related Work

LiDAR-camera SLAM can be broadly classified into three categories: LiDAR-assisted visual SLAM, vision-assisted LiDAR SLAM, and vision-LiDAR coupled SLAM. Note that vision-assisted LiDAR SLAM systems [16] are not comprehensively reviewed in this paper because this system usually hinges on semantic information, which requires knowledge of image recognition that is out of our scope.

A. LiDAR-assisted Visual SLAM

LiDAR-assisted visual SLAM generally aims to utilize LiDAR's point cloud data to obtain more accurate depth information for image feature points. A typical method in this category is LIMO, where LiDAR data is directly applied to estimate the depth of feature points [13]. Yuewen et al. proposed CamVox, an RGBD SLAM system combined with Livox LiDAR [21]. The performance of outdoor RGBD cameras is improved by depth enhancement, and the depth of many enhanced feature points reaches 100 meters. Another approach to using LiDAR data in visual SLAM is by projecting point clouds onto images and performing the direct method on projected points [14]. Reproject the projected points to the next frame image and then minimize the photometric error to solve the pose. This method does not have the error caused by depth enhancement, but it requires accurate extrinsic parameters between the camera and LiDAR. LiDAR points are too sparse compared to image pixels, and the above methods can obtain the depth of a small number of points. In order to increase the number of pixels with depth, Varuna et al. used the Gaussian process regression on the projected points from LiDAR to the image to improve the depth estimation [22]. Within a local image patch, they use the enhanced depth pixels as a priori to predict the depth of the remaining pixels in the image patch. In addition to depth enhancement, LiDAR can also improve the robustness of visual SLAM to illumination, which is also reflected in CamVox [21]. Jiawei Mo et al. proposed a method that uses LiDAR's descriptor to address the issue that visual loop closure detection is heavily affected by illumination changes [23]. They calculate the LiDAR point cloud into three descriptors and store them. The stereo SLAM map is also calculated as three descriptors and matched with the LiDAR descriptors. This method only relies on three-dimensional points to complete visual loop closure detection.

To summarize, depth enhancement is the most popular technique in LiDAR-assisted visual SLAM. In this paper, we propose a novel approach that can apply to the low-resolution LiDAR case, where the density of LiDAR point clouds is much lower than that of the camera images.

B. Vision-LiDAR Coupled SLAM

In contrast to LiDAR-assisted visual SLAM, vision-LiDAR coupled SLAM integrates both visual and LiDAR odometry modules to enhance the system's accuracy. V-LOAM is a loosely coupled system that combines visual and LiDAR odometry modules [24]. In this study, visual odometry recovers the depth of feature points from surrounding projected LiDAR points, while LiDAR odometry leverages high-frequency camera poses to mitigate drift. However, V-LOAM still faces two significant issues: ineffective depth enhancement and non-negligible drift error on the z-axis (also remains in its baseline [6]). Zikang Yuan et al. proposed SDV-LOAM [19]. It tracks the high-gradient projected LiDAR points as visual odometry and employs an adaptive scan-to-map optimization method to constrain pose in all six dimensions well. By contrast, TVL-SLAM [20] does not enhance the visual odometry's depth estimation nor utilize the motion estimation from visual odometry as the LiDAR odometry's initial guess. Instead, it establishes a joint optimization problem of visual and LiDAR features, thereby establishing a tightly coupled system.

The advantage of loose coupling is that the system structure is simple and the precision is high, but the robustness is not strong due to the influence of each module. Tight coupling is generally more robust due to joint state estimation but requires more computation.

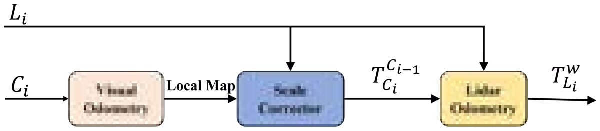


Fig. 1. Overview of our method.

III. Methodology

A. System Overview

The overview figure of our method is shown in Fig. 1, and the definitions of primary notations are present in Table I. Our system synchronizes the camera and LiDAR data at 10Hz. During the first stage, a local vision map is generated using the mono camera initialization or tracking. Subsequently, we utilize LiDAR data to estimate the monocular scale factor that represents the ratio between the corresponding vision local map and laser scan. However, due to the scale drift of the monocular odometry, we correct the scale factor periodically during the trajectory using the proposed scale corrector. Following scale correction, the LiDAR odometry

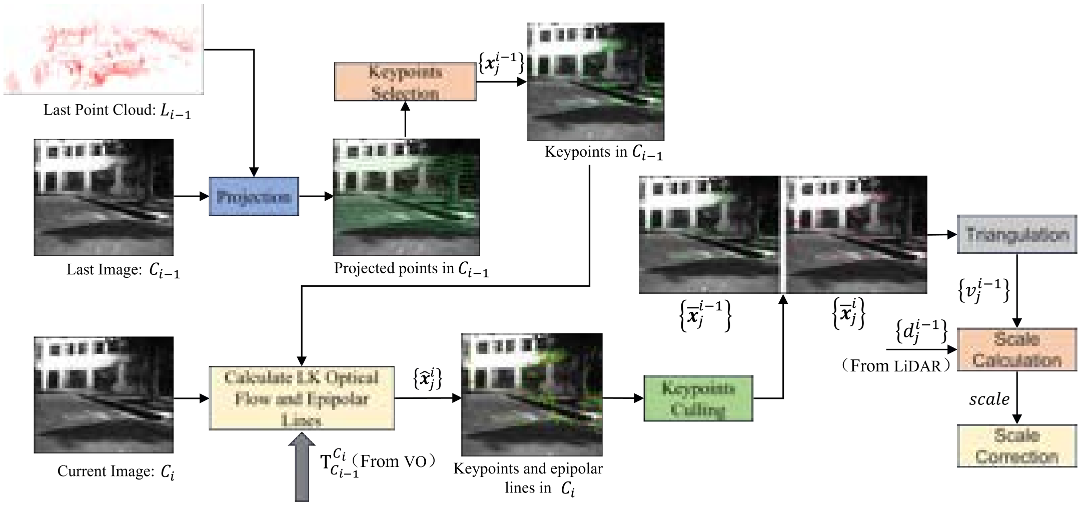


Fig. 2. Pipeline of the Scale Corrector.

TABLE I
NOMENCLATURE

Notations	Description
C_i	i^{th} frame of image keyframe
L_i	i^{th} frame of point cloud
$\{\boldsymbol{x}_j^i\}$	keypoints projected from L_i onto C_i
$\{\widehat{\boldsymbol{x}}_j^i\}$	reserved keypoints of $\{\boldsymbol{x}_j^i\}$ after optical flow tracking
$\{\overline{\boldsymbol{x}}_j^i\}$	reserved keypoints of$\{\widehat{\boldsymbol{x}}_j^i\}$ after keypoint culling
$\mathbf{T}_A^B$	transformation of A with respect to B
$\mathbf{P}_w^i$	coordinates of i^{th} map point with respect to the world
$\mathbf{K}$	camera intrinsic matrix
d_j^i	measured depth of the j^{th} projected point onto C_i
v_j^i	visual depth of the j^{th} projected point onto C_i
$\boldsymbol{p}_j^i$	LiDAR point corresponding to $\boldsymbol{x}_j^i$

generates the final pose with the initial guess from the visual odometry (we call it visual bootstrapping), resulting in a final localization frequency of 10 Hz. The visual odometry and LiDAR odometry are implemented based on ORB-SLAM2 [1] and A-LOAM [6], respectively, so we focus on performance comparison with the two baselines in our experiments part (Section IV).

The remaining parts (Section III-B and Section III-C) jointly introduce the implementation of the proposed scale corrector. The pipeline of our scale corrector is displayed in Fig. 2. To start with, we project the last frame of point cloud L_{i-1} onto the corresponding image C_{i-1} and select keypoints $\{\boldsymbol{x}_j^{i-1}\}$ among the projected points. Subsequently, the optical flow algorithm is employed to track each $\boldsymbol{x}_j^{i-1}$ in the current image C_i and thus get $\{\widehat{\boldsymbol{x}}_j^{i-1}\}$ and $\{\widehat{\boldsymbol{x}}_j^i\}$ simultaneously. Moreover, to guarantee the accuracy of keypoint correspondence, we also design two criteria for keypoints culling based on epipolar lines, which are further introduced in (3) and (4). Based on this keypoint matching, we can conduct triangulation between matched $\{\overline{\boldsymbol{x}}_j^{i-1}\}$ and $\{\overline{\boldsymbol{x}}_j^i\}$ to recover their depth in the local map. Finally, scale correction is performed between the local map and the corresponding laser scan periodically throughout the trajectory.

B. Scale Corrector: Keypoint Extraction

1) Projection and Matching: As outlined in Section III-A, the content of this section includes the projection, matching and culling steps of keypoints. For clarity, we did not take image distortion into account. Then, the process of projection between C_{i-1} and L_{i-1} can be formulated in (1).

$$\boldsymbol{x}_j^{i-1} = \frac{1}{d_j^{i-1}}\mathbf{K}\mathbf{T}_L^C\boldsymbol{p}_j^{i-1} \tag{1}$$

where $\boldsymbol{p}_j^{i-1}$ is the j^{th} point of L_{i-1}, $\mathbf{T}_L^C$ is the extrinsic parameter between camera and LiDAR. Imprecise extrinsic will cause a significant error, and the corresponding calibration method is shown in our previous work [25].

Further, the following criteria are applied to filter out distinctive $\{\boldsymbol{x}_j^{i-1}\}$ through neighbouring image information.

a) $\boldsymbol{x}_j^{i-1}$ should meet the requirements of the FAST-9 [26] corner.

b) The image gradient at $\boldsymbol{x}_j^{i-1}$ should be large enough.

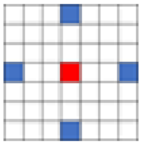
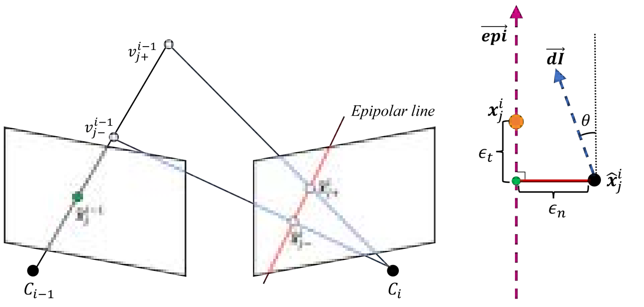


Fig. 3. FAST-12 pre-testing process. The red point is the keypoint to be tested. The blue points are domain image points.

However, the first criterion is not applicable to low-resolution LiDAR due to the scarcity of projected points. To resolve this issue, we lower the requirement to obtain $\{\boldsymbol{x}_j^{i-1}\}$ as shown in Fig. 3. We adopt the FAST-12 pre-testing process. Calculate the difference in the pixel values between the keypoint and the surrounding four points, and if more than three meet the threshold, our requirements are met. In addition, we employ non-maximum suppression to ensure a uniform distribution of keypoints.

Regarding keypoint matching, we employ the Lucas-Kanade [27] optical flow with the input of $\{\boldsymbol{x}_j^{i-1}\}$ from the last image to track corresponding points $\{\widehat{\boldsymbol{x}}_j^i\}$ in the current image.



(a) Epipolar line (b) Error directions

Fig. 4. (a): The projection keypoint will theoretically lie on the epipolar line. (b): The error between the tracked keypoint(black) and the theoretical point(orange) is divided into tangential and normal errors.

2) Culling: Since many keypoints are not Fast corners, tracking these keypoints by optical flow will cause significant uncertainty. To further improve the accuracy of keypoint matching, we conduct keypoint culling based on the epipolar lines. Fig. 4(a) shows the conception of the epipolar line. According to epipolar geometry, the keypoint $\widehat{x}_j^i$ should be located on the epipolar line. For one, it should be discarded when its distance to the epipolar line is too large. For another, in some special cases, the keypoint still should be culled when its pixel gradient is perpendicular to the epipolar line even though it is near the epipolar line. The reason is that other points distributed along the pixel gradient direction are also likely to be extracted to match this epipolar line, thereby increasing the uncertainty of its distance to the epipolar line. To cull the keypoints under the above circumstances, we propose two errors indicated in Fig. 4(b), denoted as normal error ϵ_n and tangential error ϵ_t. We will formulate them in the following parts of this section.

According to the theory of epipolar line, $\widehat{\boldsymbol{x}}_j^{i-1}$ and $\widehat{\boldsymbol{x}}_j^i$ can theoretically be constrained by (2).

$$(\widehat{\boldsymbol{x}}_j^i)^T\mathbf{K}^{-T}(\mathbf{t}_{C_{i-1}}^{C_i})_\times\mathbf{R}_{C_{i-1}}^{C_i}\mathbf{K}^{-1}\widehat{\boldsymbol{x}}_j^{i-1} = 0 \qquad (2)$$

where $\mathbf{R}_{C_{i-1}}^{C_i}$ and $\mathbf{t}_{C_{i-1}}^{C_i}$ are the rotation and translation parts of $\mathbf{T}_{C_{i-1}}^{C_i}$ respectively, and $(\cdot)_\times$ represents an antisymmetric matrix. More obviously, from the formula (2), the equation of the epipolar line can be obtained as $Ax + By + C = 0$.

Based on this definition, the quality of tracking points can be evaluated quantitatively. As displayed in Fig. 4(b), we propose two evaluation metrics of different directions. Intuitively, as formulated in (3), the **normal error** ϵ_n is evaluated through the distance between $\widehat{\boldsymbol{x}}_j^i$ and epipolar line.

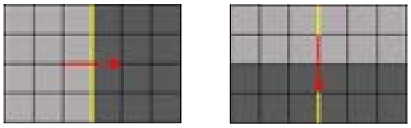


Fig. 5. Two extreme cases of pixel gradient and epipolar line directions. The yellow line is the epipolar line, and the red is the pixel gradient. **Left:** The two are perpendicular; many similar pixels are on the epipolar line. Thus, the matching uncertainty on the epipolar line is significant. **Right:** The two are parallel; the boundary pixels have a higher degree of discrimination than other pixels on the epipolar line. Thus, the matching uncertainty on the epipolar line is small.

We also set a threshold (0.5) to filter out fine points subject to this condition.

$$\epsilon_n = \frac{|A\widehat{\boldsymbol{x}}_{j.x}^i + B\widehat{\boldsymbol{x}}_{j.y}^i + C|}{\sqrt{A^2 + B^2}} < 0.5 \qquad (3)$$

where $\widehat{\boldsymbol{x}}_{j.x}^i$ & $\widehat{\boldsymbol{x}}_{j.y}^i$ are the x & y coordinates of $\widehat{\boldsymbol{x}}_j^i$, respectively.

Before explaining the tangential error, it is necessary to introduce optical flow again. Optical flow relies on pixel gradient to track the keypoint, usually using an image patch around the keypoint to increase accuracy. The same trick is used in the epipolar search [4]. Therefore, we can refer to the epipolar search to give a qualitative description of the tangential error. Inspired by [28], the angle between the epipolar line direction and the pixel gradient can be used to describe the matching uncertainty along the epipolar tangential direction. Fig. 5 shows two extreme cases. The larger the angle between the pixel gradient and the epipolar line, the more considerable the uncertainty along the epipolar tangential direction.

Consequently, for a keypoint $\widehat{\boldsymbol{x}}_j^i$ tracked by optical flow, we denote $\overrightarrow{\boldsymbol{epi}}$ and $\overrightarrow{\boldsymbol{dI}}$ as the epipolar line direction vector and pixel gradient vector, respectively, as shown in Fig. 4(b). Then, we can define the $|\cos\theta|$ and its threshold in (4).

$$|\cos\theta| = \left|\frac{\overrightarrow{\boldsymbol{epi}} \cdot \overrightarrow{\boldsymbol{dI}}}{\|\overrightarrow{\boldsymbol{epi}}\| \cdot \|\overrightarrow{\boldsymbol{dI}}\|}\right| > 0.5 \qquad (4)$$

Where θ is the angle between $\overrightarrow{\boldsymbol{epi}}$ and $\overrightarrow{\boldsymbol{dI}}$. The **tangential error** ϵ_t may be more significant if $|\cos\theta|$ is smaller than the threshold according to the matching uncertainty from the previous analysis.

At the end of keypoint culling, the points not subject to (3) and (4) are discarded, thereby reserving reliable matched points $\{\overline{\boldsymbol{x}}_j^i\}$ and $\{\overline{\boldsymbol{x}}_j^{i-1}\}$.

3) Scale Calculation: With matched keypoints $\{\overline{\boldsymbol{x}}_j^i\}$ and $\{\overline{\boldsymbol{x}}_j^{i-1}\}$, we can restore the depth of each point $\overline{\boldsymbol{x}}_j^{i-1}$ by triangulation and calculate the scale factor s_j^{i-1} through being dividing by the measured depth d_j^{i-1}, which is the distance of LiDAR point $\boldsymbol{p}_j^i$ previously projected to C_i in (1).

$$s_j^{i-1} = \frac{d_j^{i-1}}{v_j^{i-1}} \qquad (5)$$

Note that there are probably a considerable proportion of outliers among $\{s_j^{i-1}\}$, so we introduce RANSAC [29] for outlier rejection and output the mean of inliers as the final scale factor.

C. Scale Corrector: Scale Correction

In this section, we detail how to apply scale correction to the whole SLAM system. As mentioned in Section III-A, our visual odometry is implemented based on ORB-SLAM2 [1]. We remove the loop closing thread and employ scale correction during local mapping process. Without loop detection and closure, the scale of local map is unstable, and thus we periodically correct the scale of local map throughout the trajectory.

At the first stage, denote $\{\mathbf{T}_w^{C_0}, \mathbf{T}_w^{C_1}, \mathbf{T}_w^{C_2} \dots \mathbf{T}_w^{C_m}\}$ as the poses of keyframes in the local map and $\{\mathbf{P}_w^0, \mathbf{P}_w^1, \mathbf{P}_w^2 \dots \mathbf{P}_w^n\}$ as the constituent map points of the local map. Note that these values are all with respect to the world coordinate system. Therefore, we transform poses and map points to reference frame C_0 using $(\mathbf{T}_w^{C_0})^{-1}$. Subsequently, in the local map coordinate system, we can correct the scale of the local map after local bundle adjustment. Finally, the local map is transformed into the world coordinate system again for the sake of compatibility with ORB-SLAM2.

Notably, we do not frequently correct the scale, as this can interfere with the local mapping thread and cause a loss of efficiency. Instead, the scale correction is only triggered when $|scale - 1| \geq 2\%$, where $scale$ is the final scale factor calculated by the scale corrector.

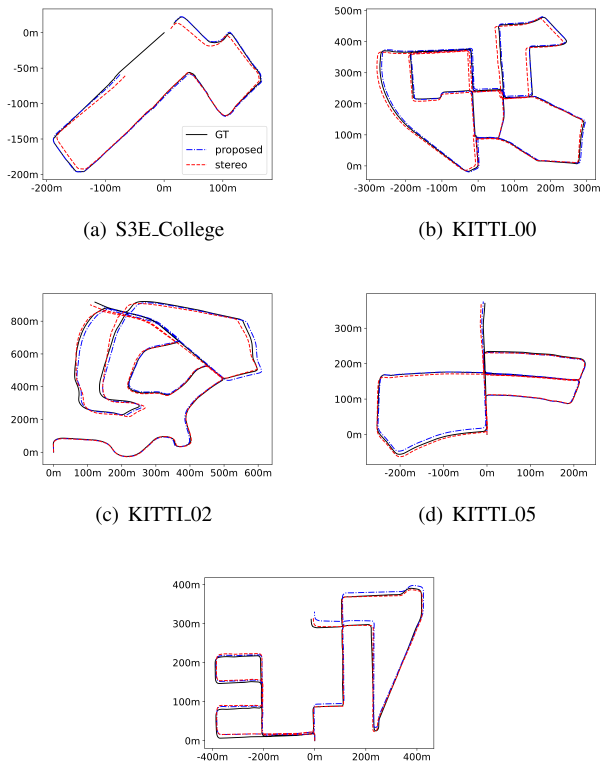
IV. Experiments

We evaluate the performance of the proposed system on KITTI Odometry and S3E datasets. They both incorporate data collected from visual and LiDAR sensors. Four challenging sequences with long distances are selected for evaluation. Regarding data setting, KITTI Odometry uses *HDL-64E* LiDAR and *FL2-14S3M-C* cameras, while S3E uses *VLP-16* LiDAR and *HikRobot MV-CS050-10GC* cameras, which is more challenging for scale correction due to the vertical sparsity of reprojected LiDAR points. Note that we have presented a solution to the sparsity issue in Section III-B.1.

Given that our method is developed based on ORB-SLAM2 [1] and A-LOAM [6], we focus on comparing the localization performance of our system to that of these two baselines. In addition, we also compared with SDV-LOAM [19], one of the state-of-the-art algorithms introduced in Section II-B. All SLAM systems are performed on a laptop with a single-core AMD 6800H @3.2GHz.

A. Effectiveness of Scale Corrector

To verify the effectiveness of the proposed scale corrector, we compare the absolute rotation and translation error (ATE & ARE) between our visual odometry and the stereo-mode ORB-SLAM2. The formulation and implementation of the two metrics can be found in *evo* [30] tool.



(a) S3E_College

(b) KITTI_00

(c) KITTI_02

(d) KITTI_05

(e) KITTI_08

Fig. 6. Trajectories estimated by visual odometry. Other legends are consistent with (a). In S3E, the pose of some frames cannot be estimated due to monocular initialization.

It should be noted that the ground-truth poses of the S3E dataset are provided by RTK without orientation (ARE is not evaluated for S3E), which worked at a much lower rate than the camera. In addition, the extrinsic calibration between RTK and camera (left) is not given. To solve these problems, we interpolate the trajectory of the visual odometry using timestamps to synchronize the predicted poses to the ground truth values using *evo* and meanwhile employ Umeyama [31] alignment between the predicted and ground-truth trajectories. Quantitative results on five representative sequences are shown in Table II while corresponding qualitative results are drawn in Fig 6. When loop closure is banned for both, our visual odometry yields better performance than stereo-mode ORB-SLAM2 in most cases, indicating the effectiveness of our scale correction module. Regarding underlying reasoning, we assume that our method is more capable of correcting the depth of distant keypoints due to the assistance of scale corrector, which is challenging for stereo vision as the parallax is not sufficient enough in this case. Moreover, we change the reference coordinate system during local optimization to the earliest keyframe in the local map, which reduce the value during optimization compared to the original solution and bring a slight performance improvement.

TABLE II

Trajectory Errors of SLAM Methods

Sequence / Length		Ours VO	ORB-SLAM2(Stereo)		Ours VLO	A-LOAM	SDV-LOAM
KITTI_00 / 3724m	ATE(m)	**5.631**	8.946	translationl RMSE(%)	1.182	1.655	**0.9836**
	ARE(deg)	**1.791**	1.920	rotational error(deg/m)	0.0061	0.0078	**0.0041**
KITTI_02 / 5067m	ATE(m)	**13.53**	17.20	translationl RMSE(%)	3.263	11.26	**0.8022**
	ARE(deg)	**1.821**	3.300	rotational error(deg/m)	0.0103	0.0307	**0.0024**
KITTI_05 / 2205m	ATE(m)	5.096	**4.460**	translationl RMSE(%)	1.4496	4.7189	**0.7036**
	ARE(deg)	**0.6319**	1.100	rotational error(deg/m)	0.0065	0.0155	**0.0030**
KITTI_08 / 3222m	ATE(m)	13.98	**12.47**	translationl RMSE(%)	1.895	5.100	**1.1031**
	ARE(deg)	**1.803**	1.824	rotational error(deg/m)	0.0075	0.0187	**0.0037**
[1]S3E_College / 920m	ATE(m)	**1.673**	5.374	ATE(m)	**3.097**	5.505	[2]Failed
	ARE(deg)	–	–	ARE(deg)	–	–	

[1] The ground truth of the S3E dataset has only the translation part, and the rotation part is the unit quaternion.

[2] SDV-LOAM fails on S3E_College.

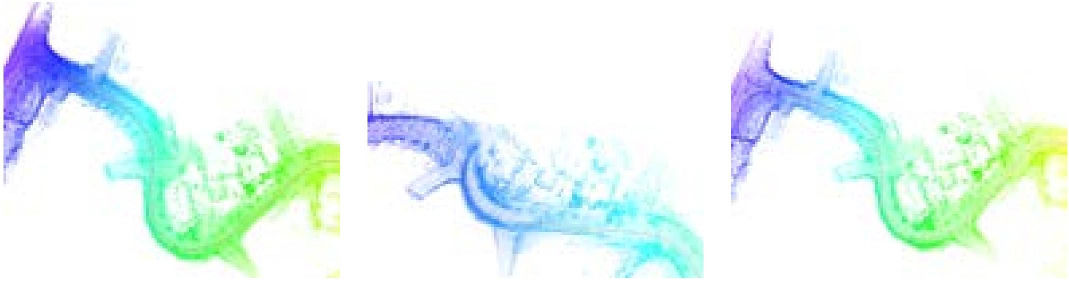

(a) GT (b) A-LOAM (c) Ours

Fig. 7. Performance of degraded scenes. On a big detour with a degraded scene, A-LOAM makes wrong pose estimates, while ours works well.

B. Effectiveness of Visual Bootstrapping

As for the verification of the effectiveness of Visual Bootstrapping for the LiDAR odometry, we compare it with the baseline A-LOAM [6] and SDV-LOAM [19] on the same datasets shown in Section IV-A. However, there is a slight difference in evaluation. For the KITTI dataset, we replace the evo tool with the official KITTI evaluation tool [32] for localization evaluation since it better demonstrates the drift degree in a long distance. Table II illustrates that our system achieves significantly lower translation drift and slightly lower rotation drift than the A-LOAM. In the KITTI dataset, our performance is not as good as SDV-LOAM, but SDV-LOAM does not adapt to the *VLP-16* LiDAR and thus fails on the S3E dataset.

For qualitative results, we present a partial view of LiDAR map in Fig 7, which is part of a curved road with only trees around. In this case, A-LOAM suffers degradation while our LiDAR odometry works well. Therefore, both qualitatively and quantitatively, our method outperforms A-LOAM. As for the reasons, A-LOAM lacks constraints on the z-axis, and the loss function easily falls into a minimum value in a degraded scene. Using the results of visual odometry to compensate for the initial value of A-LOAM can reduce the number of iterations and avoid the problem that the loss function falls into a minimum value due to the significant difference between the initial value and the actual value.

V. Conclusion and future work

In this study, we propose a loosely coupled monocular-LiDAR SLAM technique with a novel scale corrector. Its pose prediction derives from monocular odometry with scale correction and LiDAR odometry with visual bootstrapping. Concerning localization performance, our visual odometry achieves better performance than stereo-mode ORB-SLAM2 when loop closure for neither is available, while our LiDAR odometry significantly outperforms baseline A-LOAM [6]. It is illustrated by quantitative results that the whole system yields markedly lower translation drift and moderately lower rotation drift. Qualitative results also show that our system is more robust than A-LOAM [6] in degraded scenes. On the other hand, as for limitations, the proposed system relies heavily on the stability of visual odometry. In other words, a severe drift of visual odometry can cause a great loss of performance to our system, which deserves our deeper investigation.

In our future study, we are expected to refine the proposed framework, including enhancing the robustness of visual odometry through back-end optimization, adding trouble-detection and troubleshooting tragedies for visual odometry failure and involving LiDAR points in constructing visual map.

References

[1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “Orb-slam: a versatile and accurate monocular slam system,” *IEEE transactions on*

robotics, vol. 31, no. 5, pp. 1147–1163, 2015.

[2] I. Cvišić, I. Marković, and I. Petrović, "Soft2: Stereo visual odometry for road vehicles based on a point-to-epipolar-line metric," *IEEE Transactions on Robotics*, 2022.

[3] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski, "Bundle adjustment in the large," in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II 11*. Springer, 2010, pp. 29–42.

[4] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[5] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13*. Springer, 2014, pp. 834–849.

[6] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[7] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-loam: Fast lidar odometry and mapping," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4390–4396.

[8] P. Zhou, X. Guo, X. Pei, and C. Chen, "T-loam: truncated least squares lidar-only odometry and mapping in real time," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2021.

[9] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, "Ct-icp: Real-time elastic lidar odometry with loop closure," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 5580–5586.

[10] Z. Liu and F. Zhang, "Balm: Bundle adjustment for lidar mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3184–3191, 2021.

[11] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4802–4809.

[12] G. Kim, B. Park, and A. Kim, "1-day learning, 1-year localization: Long-term lidar localization using scan context image," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1948–1955, 2019.

[13] J. Graeter, A. Wilczynski, and M. Lauer, "Limo: Lidar-monocular visual odometry," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 7872–7879.

[14] Y.-S. Shin, Y. S. Park, and A. Kim, "Direct visual slam using sparse depth for camera-lidar system," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5144–5151.

[15] Z. Zhu, S. Yang, H. Dai, and F. Li, "Loop detection and correction of 3d laser-based slam with visual information," in *Proceedings of the 31st International Conference on Computer Animation and Social Agents*, 2018, pp. 53–58.

[16] X. Liang, H. Chen, Y. Li, and Y. Liu, "Visual laser-slam in large-scale indoor environments," in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2016, pp. 19–24.

[17] Z. Liu, Y. Hu, T. Fu, and M.-O. Pun, "Dense three-dimensional color reconstruction with data fusion and image-guided depth completion for large-scale outdoor scenes," in *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, 2022, pp. 3468–3471.

[18] J. Zhang and S. Singh, "Laser–visual–inertial odometry and mapping with high robustness and low drift," *Journal of field robotics*, vol. 35, no. 8, pp. 1242–1264, 2018.

[19] Z. Yuan, Q. Wang, K. Cheng, T. Hao, and X. Yang, "Sdv-loam: Semi-direct visual-lidar odometry and mapping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–18, 2023.

[20] C.-C. Chou and C.-F. Chou, "Efficient and accurate tightly-coupled visual-lidar slam," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14 509–14 523, 2021.

[21] Y. Zhu, C. Zheng, C. Yuan, X. Huang, and X. Hong, "Camvox: A low-cost and accurate lidar-assisted visual slam system," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5049–5055.

[22] V. De Silva, J. Roche, and A. Kondoz, "Fusion of lidar and camera sensor data for environment sensing in driverless vehicles," 2017.

[23] J. Mo and J. Sattar, "A fast and robust place recognition approach for stereo visual odometry using lidar descriptors," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5893–5900.

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

[25] N. Ou, H. Cai, J. Yang, and J. Wang, "Targetless extrinsic calibration of camera and low-resolution 3-d lidar," *IEEE Sensors Journal*, vol. 23, no. 10, pp. 10 889–10 899, 2023.

[26] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*. Springer, 2006, pp. 430–443.

[27] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI'81: 7th international joint conference on Artificial intelligence*, vol. 2, 1981, pp. 674–679.

[28] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1449–1456.

[29] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[30] M. Grupp, "evo: Python package for the evaluation of odometry and slam." https://github.com/MichaelGrupp/evo, 2017.

[31] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 13, no. 04, pp. 376–380, 1991.

[32] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.