2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Macau, China, November 4-8, 2019

Stereo Visual Inertial LiDAR Simultaneous Localization and Mapping

Weizhao Shao, Srinivasan Vijayarangan*, Cong Li*, and George Kantor

***Abstract*— Simultaneous Localization and Mapping (SLAM) is a fundamental task to mobile and aerial robotics. LiDAR based systems have proven to be superior compared to vision based systems due to its accuracy and robustness. In spite of its superiority, pure LiDAR based systems fail in certain degenerate cases like traveling through a tunnel. We propose Stereo Visual Inertial LiDAR (VIL) SLAM that performs better on these degenerate cases and has comparable performance on all other cases. VIL-SLAM accomplishes this by incorporating tightly-coupled stereo visual inertial odometry (VIO) with LiDAR mapping and LiDAR enhanced visual loop closure. The system generates loop-closure corrected 6-DOF LiDAR poses in real-time and 1cm voxel dense maps near real-time. VIL-SLAM demonstrates improved accuracy and robustness compared to state-of-the-art LiDAR methods.**

I. Introduction

SLAM solves the problem of mapping unknown environments while estimating robot state. Though SLAM is actively researched for the past few decades, Cadena et al. [1] note that there are still challenges in handling diverse environments and long-term continuous operations. SLAM systems operate on a wide range of sensor modalities each trying to exploit their benefits. In the past few years, LiDAR based SLAM systems have gained popularity over vision based systems due to their robustness to changes in the environment. However pure LiDAR based systems have their deficiencies. They fail in environments with repeating structures like tunnels or hallways. These environments are challenging to map and localize, and system which exploits the strengths of all the sensor modalities need to be deployed to succeed. We propose VIL-SLAM, which uses IMU, stereo cameras and LiDAR, and exploit their benefits collectively. Our experiments demonstrate that VIL-SLAM performs on par with pure LiDAR based systems in most cases and better on cases where pure LiDAR based systems simply fail. VIL-SLAM achieves this by integrating stereo VIO and LiDAR mapping with loop closure. To the best of our knowledge, this is the first work of this kind. In addition, we introduce a method to evaluate mapping results using a time-of-flight laser scanner (Faro). We also provide VIO validation results on the EuRoC MAV dataset.

VIL-SLAM uses a tightly-coupled stereo VIO that performs fixed-lag pose graph optimization, LiDAR mapping that uses sparse 3D features for map registration, and loop closure that integrates sparse point cloud alignment with visual loop detection. Loop closure optimizes a global pose graph using an incremental solver. VIL-SLAM is designed to operate long term and in different environments robustly. The high frequency IMU measurements produce estimates which are reasonable for the short interval but quickly drift. When constrained with stereo visual measurements, we can correct the biases and estimate accurate relative motion (referred to as VIO). The relative motion estimate is used to aid LiDAR scan matching which then accumulates the high-fidelity 3D point clouds to form an accurate map. The robot's state estimate accumulates drift during long traversals. Loop closure addresses this issue by recognizing the revisited sites using either visual or LiDAR methods. Visual methods involve using Bag-of-Words [2] to recognize the place and Perspective-n-Point (PnP) algorithm to estimate the pose correction. In LiDAR methods, the places are recognized using segment based algorithms like SegMatch [3], and pose correction is estimated using Iterative Closest Point (ICP) [4] algorithm. While the Bag-of-Words method is fast and versatile, it lacks the accuracy of the slow but robust LiDAR method which uses ICP. VIL-SLAM uses a hybrid approach where it first finds the loop closure candidate using Bag-of-Words technique, generates a rough estimate of the pose correction using Perspective-n-Point (PnP) algorithm, and then refines the rough estimate using ICP.

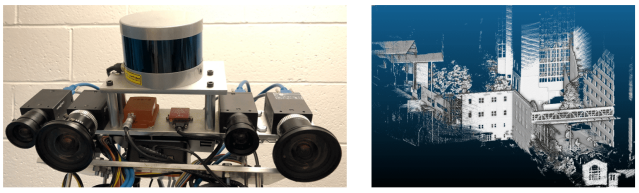


Fig. 1: (a) Experimental platform built. (b) Mapping result from an outdoor test. Streetlight is reconstructed clearly.

II. Related work

Current VIO literature introduces various formulations to integrate visual and inertial data. The literature characterizes different approaches into *tightly-coupled system* [5]–[7], in which visual information and inertial measurements are jointly optimized, or *loosely-coupled system* [8]–[11], in which IMU is a separate module and fused with a vision-only state estimator. The approaches could be further divided into either filtering-based [11]–[16] or graph-optimization based [5]–[7], [17], [18]. Tightly-coupled optimization-based approaches, taking the benefit of minimizing residuals iteratively, usually achieve better accuracy and robustness with a higher computation cost. In our work, we bound the

*These authors contributed equally to the paper

The authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA. {weizhaos, svijaya1, congl1, gkantor}@andrew.cmu.edu

This work is sponsored by Autel Robotics.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 02:56:18 UTC from IEEE Xplore. Restrictions apply.

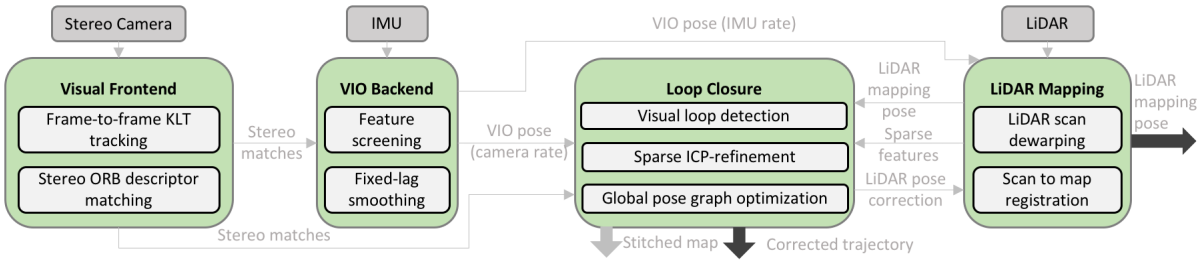


Fig. 2: The system diagram of VIL-SLAM. Sensors are in gray and modules are in green. Arrows indicate how messages flow within the system. The dark thick arrows indicate the system real-time output and the light thick arrow indicates the output generated in post-processing near real-time.

computation cost by forming landmarks in a structureless fashion and only optimizing for a fixed-size pose graph to achieve the real-time performance.

Current state-of-the-art SLAM systems using just laser scanner are [19]–[23], in which a motion model is required, either a constant velocity model or a Gaussian process. Approach in [24] combines stereo cameras and a laser scanner. It has motion estimation generated from a visual odometry (VO) and refined by matching laser scans. The differences to our system are that they use multi-resolution grid map representation and ours uses sparse point cloud to localize and outputs dense point cloud. Also, VIO is usually more robust and accurate compared to a VO [25]. VLOAM [26], which uses an IMU, a monocular camera, and a laser scanner is the most similar existing system to ours. One difference is that we use a tightly-coupled VIO as the motion model to initialize the LiDAR mapping algorithm whereas VLOAM uses loosely-coupled IMU and camera. Though our VIO is more robust, VLOAM has a more interactive system where information from both camera and LiDAR module could be used for IMU biases correction. One addition that VIL-SLAM has is the LiDAR enhanced loop closure.

III. System overview

The system has four modules as shown in Fig. 2. The visual frontend takes stereo pairs from the stereo cameras. It performs frame to frame tracking and stereo matching, and outputs stereo matches as visual measurements. The stereo VIO takes stereo matches and IMU measurements, performs IMU pre-integration and tightly-coupled fixed-lag smoothing over a pose graph. This module outputs VIO pose at IMU rate and camera rate. LiDAR mapping module uses the motion estimate from the VIO and performs LiDAR points dewarping and scan to map registration. The loop closure module conducts visual loop detection and initial loop constraint estimation, which is further refined by a sparse point cloud ICP alignment. A global pose graph constraining all LiDAR poses is optimized incrementally to obtain a globally corrected trajectory and a LiDAR pose correction in real-time. They are sent back to LiDAR mapping module for map update and re-localization. In post processing, we stitch the dewarped LiDAR scans with the best estimated LiDAR poses to have the dense mapping results (Fig. 5).

IV. Visual frontend

Visual frontend accepts a stereo pair, and performs frame to frame tracking and stereo matching for the generation of a set of stereo-matched sparse feature points, namely, stereo matches. A stereo match could either be one tracked from previous stereo pair, or a new one extracted in this pair. The frame to frame tracking performance directly affects the temporal constraints quality while the stereo matching helps constrain the scale. These two tasks are crucial for any stereo visual odometry. Direct methods show robust and efficient temporal tracking results in recent years [8], [27]. Thus, we use Kanade Lucas Tomasi (KLT) feature tracker [28] to track all feature points in the previous stereo matches, either in the left or right image. Only when they are both tracked, we have a tracked stereo match and it is pushed into the output. Large stereo baseline helps scale estimation and reduces degeneracy issues caused by distant features. We use feature-based methods which are better suited to handle large baselines than KLT. If the number of tracked stereo matches is below a threshold, we perform feature extraction using Shi-Tomashi Corner detector [29], followed by a feature elimination process in which features that have pixel coordinate distance to any existing features smaller than a threshold are deleted. ORB (Oriented FAST and Rotated BRIEF) [30] descriptors are then computed on all survived features, followed by a brute-force stereo matching to obtain new stereo matches. The system initializes by performing stereo matching on the first stereo pair.

V. Stereo visual inertial odometry

The goal of the stereo VIO is to provide real-time accurate state estimate at a relatively high frequency, serving as the motion model for the LiDAR mapping algorithm. A tightly-coupled fixed-lag smoother operating over a pose graph is a good trade-off between accuracy and efficiency. Optimization-based methods in general allow for multiple re-linearization to approach the global minimum. A fixed-lag pose graph optimizer further bounds the maximum number of variables, and hence the computation cost is bounded. Since bad visual measurements cause convergence issues, we enforce a strict outlier rejection mechanism on visual measurements. The system eliminates outliers by checking the average reprojection error, both stereo and temporal.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 02:56:18 UTC from IEEE Xplore. Restrictions apply.

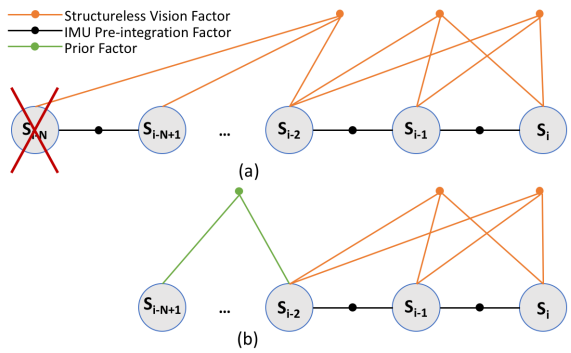


Fig. 3: Fixed-lag pose graph formulation in the VIO. State variables being optimized are circled, where i stands for the current state and N is the window size. (a) The state to be marginalized is crossed. (b) After marginalization, prior factors are added back on related variables.

The VIO proposed has *IMU Pre-integration Factor* and *Structureless Vision Factor* as constraints. The graph representation is shown in Fig. 3. Variables to be optimized are the states inside the window. Denote $\mathbf{S}_t$ as the state variable at the stereo frame time t. $\mathbf{S}_t$ contains the 6 Degrees of Freedom (DoF) system pose ξ_t (IMU frame), the associated linear velocity $\mathbf{v}_t$, accelerometer bias $\mathbf{b}_t^a$, and gyroscope bias $\mathbf{b}_t^g$. The window of state variables being estimated are of the most recent N stereo frames. Past state variables are marginalized, producing prior factors on related variables.

A. IMU pre-integration factor

We follow the IMU pre-integration method [31] [32] to generate relative IMU measurements between $\mathbf{S}_i$ and $\mathbf{S}_j$. Using the pre-integration technique, re-linearization could be performed efficiently during optimization. The residual represented by the IMU pre-integration factor is $\mathbf{r}_{ij}^I$, which consists of three terms: the residual of pose ($\mathbf{r}_{\Delta\xi ij}$), velocity ($\mathbf{r}_{\Delta\mathbf{v}ij}$), and biases ($\mathbf{r}_{\Delta\mathbf{b}ij}$).

B. Structureless vision factor

Visual measurements are modeled in a structureless fashion, similar to [31] [33] [34]. Consider a landmark p, whose position in global frame is $\mathbf{x}_p \in \mathbb{R}^3$, is observed by multiple states and denote the set of states observing p as $\{\mathbf{S}\}_p$. For any state $\mathbf{S}_k$ in $\{\mathbf{S}\}_p$, denote the residual formed by measuring p as in the left camera image as $\mathbf{r}^V_{\xi_{k,lc},p}$ ($\xi_{k,lc}$ is the left camera pose, obtained by applying a IMU-camera transformation to ξ_k):

$$\mathbf{r}^V_{\xi_{k,lc},p} = \mathbf{z}_{\xi_{k,lc},p} - h(\xi_{k,lc}, \mathbf{x}_p) \tag{1}$$

where $\mathbf{z}_{\xi_{k,lc},p}$ is the pixel measurement of p in the image and $h(\xi_{k,lc}, \mathbf{x}_p)$ encodes a perspective projection. Same formulation is derived for the right camera image. Iterative methods are adopted for optimizing the pose graph, and hence linearization of the above residual is required. Equation (2) shows the linearized residuals for landmark p.

$$\sum_{S_p} ||\mathbf{F}_{kp}\delta\xi_k + \mathbf{E}_{kp}\delta\mathbf{x}_p + \mathbf{b}_{kp}||^2 \tag{2}$$

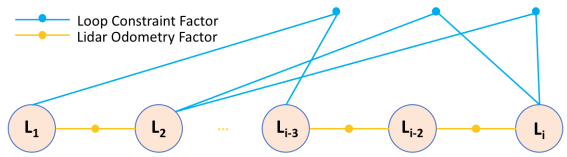


Fig. 4: The global pose graph consists of the *LiDAR Odometry Factor* and the *Loop Constraint Factor*. i stands for the current scan.

where the Jacobians $\mathbf{F}_{kp}$, $\mathbf{E}_{kp}$ and the residual error $\mathbf{b}_{kp}$ are results from the linearization and normalized by $\Sigma_c^{1/2}$, the visual measurement covariance. Stacking each individual component inside the sum into a matrix we have

$$||\mathbf{r}_p^V||^2_{\Sigma_C} = ||\mathbf{F}_p\delta\xi_k + \mathbf{E}_p\delta x_p + \mathbf{b}_p||^2 \tag{3}$$

To avoid optimizing over $\mathbf{x}_p$, we project the residual into the null space of $\mathbf{E}_p$: Premultiply each term by $\mathbf{Q}_p \doteq \mathbf{I} - \mathbf{E}_p(\mathbf{E}_p^\top\mathbf{E}_p)^{-1}\mathbf{E}_p^\top$, an orthogonal projector of $\mathbf{E}_p$ [31]. We thus have the *Structureless Vision Factor*, for landmark p as

$$||\mathbf{r}_p^V||^2_{\Sigma_C} = ||\mathbf{Q}_p\mathbf{F}_p\delta\xi_k + \mathbf{Q}_p\mathbf{b}_p||^2 \tag{4}$$

C. Optimization and marginalization

Given the residuals, the pose graph optimization is a *maximum a posteriori* (MAP) problem whose optimal solution is

$$\mathbf{S}_w^* = \arg\min_{S_w^*}(||\mathbf{r}_0||^2_{\Sigma_0} + \sum_{i\in w}||\mathbf{r}^I_{i(i+1)}||^2_{\Sigma_I} + \sum_p ||\mathbf{r}_p^V||^2_{\Sigma_C}) \tag{5}$$

where $\mathbf{S}_w^*$ is the set of state variables inside the window. $\mathbf{r}_0$ and Σ_0 are prior factors and their associated covariance. Σ_I is the covariance of the IMU measurements. We use the Levenberg-Marquart optimizer to solve this nonlinear optimization problem. The most recent N state variables are maintained inside the optimizer. Schur-Complement marginalization [35] is performed on state variables getting out of the window. Prior factors are then added to related variables inside the window as in Fig. 3(b).

VI. LiDAR Mapping

LiDAR mapping uses high frequency IMU rate VIO poses as the motion prior to perform LiDAR points dewarping and scan to map registration. Denote a scan χ as the point cloud obtained from one complete LiDAR rotation. Geometric features including points on sharp edges and planar surfaces are extracted from χ after dewarping [22], [26]. The registration is then based on feature points from current scan to the map (all previous feature points), solved as an optimization problem by minimizing Euclidean distance residuals formed by the feature points as in [22].

A. LiDAR scan dewarping

Dewarping is required as points from a LiDAR scan are timestamped differently. Denote any time within a scan as t_i. We dewarp all points to the time of end of scan t_{k+1} based

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 02:56:18 UTC from IEEE Xplore. Restrictions apply.

on IMU rate VIO poses. Denote a LiDAR point at t_i as $\mathbf{P}_i$ and the dewarped itself as $\tilde{\mathbf{P}}_i$, we have

$$\tilde{\mathbf{P}}_i = (\mathbf{T}^L_{k+1})^{-1}\mathbf{T}^L_i\mathbf{P}_i \tag{6}$$

where $\mathbf{T}^L_{k+1}$, $\mathbf{T}^L_i$ are LiDAR frame poses transformed from the closest IMU rate VIO poses.

B. Scan to map registration

Feature points from the dewarped scan $\tilde{\chi}$ are registered to the map, optimizing for the LiDAR mapping pose at t_{k+1} denoted as $\mathbf{L}_{k+1}$. Denote the initial estimate of $\mathbf{L}_{k+1}$ as $\mathbf{L}^*_{k+1}$, we have:

$$\mathbf{L}^*_{k+1} = \mathbf{L}_k\mathbf{T}^L_{trans} \tag{7}$$

where $\mathbf{L}_k$ is the optimized previous LiDAR mapping pose and $\mathbf{T}^L_{trans}$ is the relative transformation obtained based on IMU rate VIO poses. All dewarped feature points are then transformed to world coordinate system by $\mathbf{L}^*_{k+1}$ for registration.

The residual $\mathbf{r}_E$ of an edge feature point in the current scan, is the Euclidean distance between itself and the line formed by the two closest edge points in the map. The residual $\mathbf{r}_U$ of a surface point in the current scan is the distance between itself and the planar patch formed by the three closest surface points in the map. [22] Incorporating $\mathbf{L}^*_{k+1}$, we can rewrite the two residuals as:

$$f_E(E^L_{(c,i)}, \mathbf{L}^*_{k+1}) = \mathbf{r}_E \tag{8}$$

$$f_U(U^L_{(c,i)}, \mathbf{L}^*_{k+1}) = \mathbf{r}_U \tag{9}$$

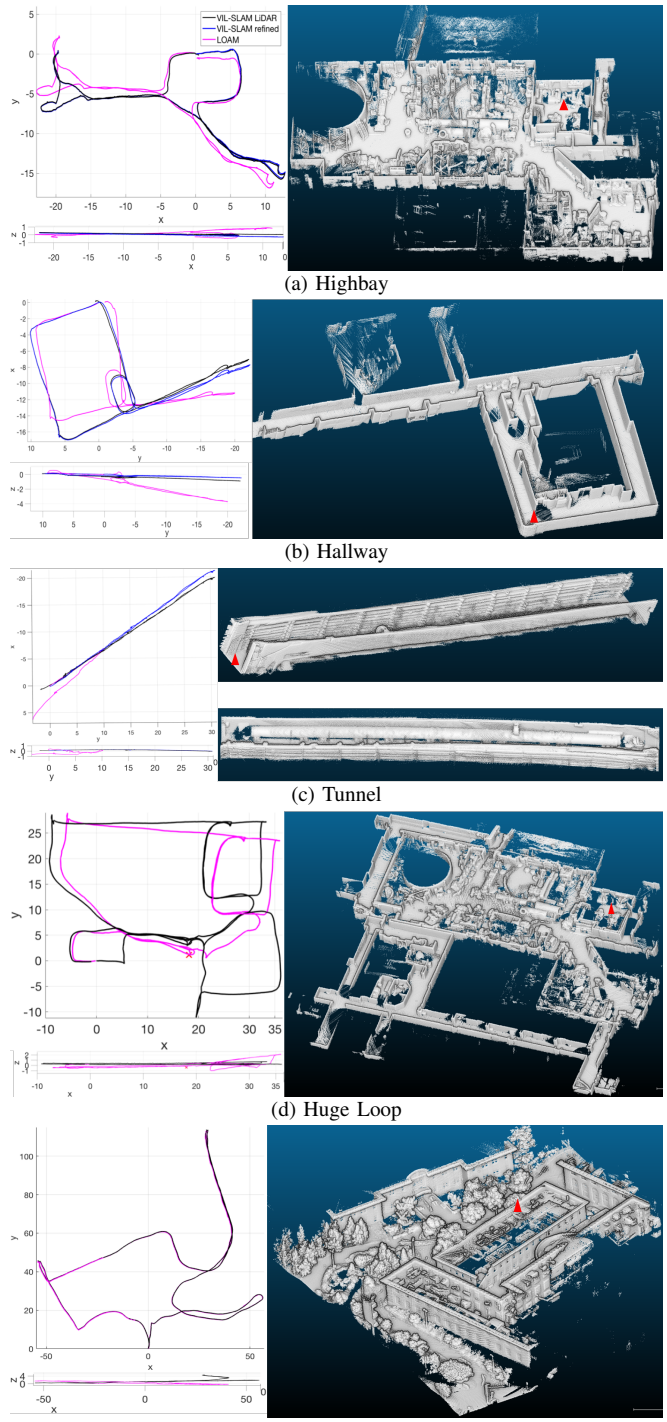
where $E^L_{(c,i)}$ and $U^L_{(c,i)}$ are the 3D position of the ith dewarped feature point in the LiDAR coordinate system. Levenberg-Marquardt optimizer is used to solve this nonlinear optimization problem, formed by stacking the cost functions for all feature points.

VII. LiDAR Enhanced Loop Closure

Loop closure is critical to any SLAM system as long term operation introduces drift. The objective of loop closure is to eliminate drift by performing a global pose graph optimization which incorporates loop constraints and relative transformation information from LiDAR mapping. To better assist LiDAR mapping, the corrected LiDAR pose is sent back in real-time so that feature points from new scans are registered to the revisited map. We propose adding ICP alignment in addition to visual Bag-of-Words [2] loop detection and PnP loop constraint formulation. The system uses iSAM2 [36], an incremental solver, to optimize the global pose graph, achieving real-time performance.

A. Loop detection

Stereo images and LiDAR scans are associated using their timestamps. Let us denote these as key images and key scans respectively. To prevent false loop detection we restrict candidates within a certain time threshold. Loop candidates are detected by testing the key images with the Bag-of-Words [2] database of previous key images. Furthermore, We match feature descriptors of the left key image with the loop candidates to filter out the false positives.



(a) Highbay

(b) Hallway

(c) Tunnel

(d) Huge Loop

(e) Outdoor

Fig. 5: Trajectories from VIL-SLAM and LOAM are shown on the left and maps generated by VIL-SLAM are shown on the right. Start(end) position is labeled with red triangle in the map and is the origin in the plot.

B. Loop constraint

The system first obtains visual loop constraint as an initial estimate. Since we use a structureless formulation for visual landmarks, triangulation on all the stereo matched features in the loop candidate is performed to obtain their 3D location. Their associations to current key images are given by de-

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 02:56:18 UTC from IEEE Xplore. Restrictions apply.

TABLE I: FDE (%) and MRE (m) TEST RESULTS

Test	Total Length	FDE		MRE	
		VIL-SLAM	LOAM	VIL-SLAM	LOAM
Highbay	118	**0.08**	0.56	**0.08**	0.22
Hallway	103	**0.61**	0.91	**0.10**	0.27
Tunnel	85	**1.86**	-[1]	×	×
Huge Loop	318	**0.01**	-	**0.22**	0.36
Outdoor	528	0.02	0.02	×	×

scriptor match. The visual loop constraint is then evaluated using EPNP [37]. To improve the accuracy of the visual loop constraint, we use ICP alignment on the feature points of the corresponding LiDAR key scans. With a bad initialization or a larger point count, ICP takes longer to converge and consumes more computation resources. However, the visual loop constraint provides a good initialization point and the ICP only uses sparse feature points (Section VI), which makes it converge faster.

C. Global pose graph optimization

The graph representation of the global pose graph is shown in Fig. 4. It contains all the available LiDAR mapping poses as variables, constrained by the *LiDAR Odometry Factor* and the *Loop Constraint Factor*, both are measurements of the relative transformation: $(\mathbf{L}_u)^{-1}\mathbf{L}_v$ where u and v stand for scan ID and $\mathbf{L}_u$, $\mathbf{L}_v$ are the associated poses. For the *LiDAR Odometry Factor*, u is the previous scan ID. For the *Loop Constraint Factor*, u is the key scan ID found as loop. For both cases, v is the current scan ID. Poses are expressed in 6 DoF minimum form in the optimization. To realize real-time performance, we use iSAM2 [36] to incrementally optimize the global pose graph.

D. Re-localization

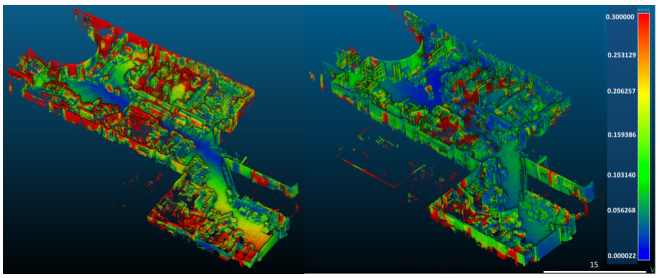
Once a true loop closure candidate is found, LiDAR mapping buffers the feature points (without registering them to the map) until it receives loop correction. The loop correction contains globally optimized trajectory. LiDAR mapping updates its map, adds the buffered feature points to the map and then resumes its operation. We can afford to update the map in real-time because (a) loop closure has a real-time performance (b) the sparse feature map does not take much memory, and (c) scan to map registration is fast enough to catch up the LiDAR data rate.

VIII. Experimental results

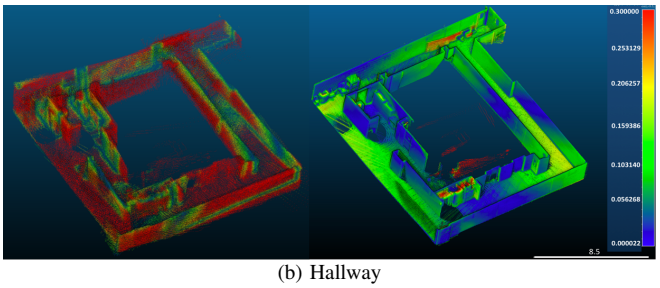
We evaluate VIL-SLAM and compare it with the best real-time LiDAR based system, LOAM[2] [22] on custom datasets. We did not use KITTI odometry dataset [38] because their evaluation sequences do not have inertial measurements which are needed for VIO. Also, most KITTI sequences are not challenging. So they do not evaluate the robustness of these systems which is the main focus of our experiments. We also evaluate the stereo VIO submodule (VIL-VIO) using the EuRoC MAV dataset [39].

[1]"-" indicates not finished. "×" indicates missing data.

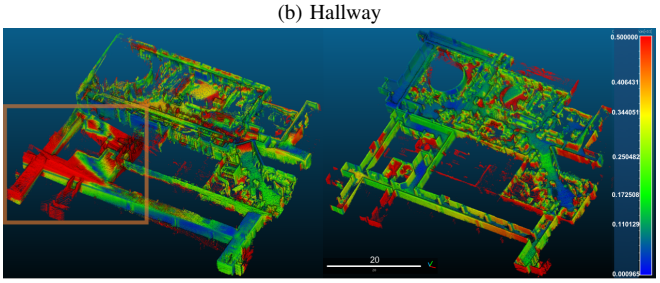
[2]This is the best implementation of LOAM we could find online https://github.com/laboshinl/loam_velodyne



(a) Highbay



(b) Hallway



(c) Huge Loop

Fig. 6: Map registration error of VIL-SLAM (right) and LOAM (left) comparing to the model. Errors above 0.3m are colored red for (a-b) and 0.5m for (c). Discontinuous red regions inside the blue and green are due to lack of the model caused by occlusions of the Faro scans.

A. Platform and software

We built a platform (Fig. 1(a)) with two megapixel cameras, a 16 scan-line LiDAR, an IMU (400Hz), and a 4GHz computer (with 4 physical cores). We built a custom microcontroller based time synchronization circuit that synchronizes the cameras, LiDAR, IMU and computer by simulating GPS time signals. The software pipeline is implemented in C++ with ROS communication interface. We use GTSAM library [40] to build the fixed-lag smoother in the VIO. For loop closure, we use ICP module from LibPointMatcher [41] to align point clouds, DBoW3 [42] to build the visual dictionary, and iSAM2 [36] implementation in GTSAM [40] to conduct global optimization.

B. Tests and results

We present results from five representative environments including featureless hallways, cluttered highbays, tunnels, and outdoor environments. The data collection started and ended at the same point for all these sequences. Odometry (LiDAR mapping pose) is evaluated based on the final drift error (FDE). Mapping results are evaluated in terms of mean registration error (MRE) using Faro scans as ground truth. We first align the map with the model (Faro scans), and then

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 02:56:18 UTC from IEEE Xplore. Restrictions apply.

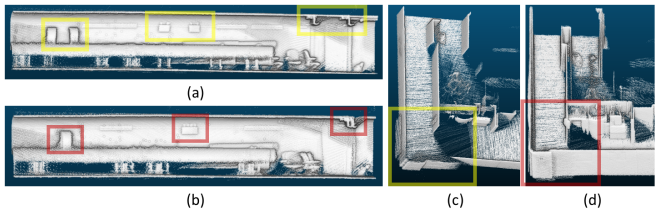


Fig. 7: (a) Map of the tunnel stitched using LIDAR mapping poses. (b) Map of the tunnel stitched using globally refined poses. Double image in (a) is mostly eliminated but not fully, because only one loop constraint is generated, not enough for a full correction. (c) Map of the hallway stitched using LIDAR mapping poses. (d) Map of the hallway stitched using globally refined poses. Double image in (c) is mostly eliminated. Walls are aligned with two loop constraints.

compute the Euclidean distance between a map point and its closest point in the model [43]. The odometry FDE and mapping results are shown in Table I with the better ones in bold. The trajectories and cross-sectioned maps are shown in Fig. 5. The map comparisons are shown in Fig. 6.

The *highbay* is an indoor warehouse which is open, structured, and rich in features. However, frequent structural occlusions could be a challenge for the visual frontend and the LiDAR feature extraction part. Both VIL-SLAM and LOAM handle this environment pretty well. For VIL-SLAM, LiDAR mapping module registers most of its scan to map, largely reducing the odometry error. Loop closure recognizes the starting position and closes the loop. The map is generated using the globally refined poses, with the majority of map errors below 0.15m.

The *hallway* and *tunnel* tests are challenging environments because of lack of visual features and the degeneracy issue along traversal direction for LiDAR. LOAM accumulates large error in the hallway, and fails the tunnel test mainly due to the degeneracy issue. Aided by the stereo VIO module (VIL-VIO), VIL-SLAM succeeds both tests. In the *hallway* test, the visual frontend returns fewer reliable measurements because of the featureless walls, under-constraining the VIO. This corrupts the map as observed by wall misalignment, which is later corrected by loop closure as shown in Fig. 7(c-d). Loop closure detects the loop twice when approaching the endpoint, lowering FDE to 0.05% and generating a refined map. In the *tunnel* test, because of the degeneracy issue, VIL-SLAM struggles as well and accumulates some error in the traversal direction. However, loop closure detects the loop at about 3m from the end point, lowering the FDE down to 0.08% and correcting the map as shown in Fig. 7(a-b).

The *huge loop* test features challenges from both *hallway* and *highbay* environments. In addition, we end the trajectory by re-entering the highbay after traversing along a long narrow corridor. LOAM fails this test after re-entering the highbay, at the place labeled by a red cross in Fig. 5(d). We think this is because it fails to register new scans to the original *highbay* map caused by a large error in z-direction accumulated in the corridor. VIL-SLAM succeeds in this test.

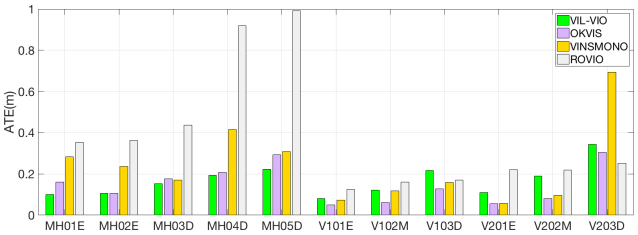


Fig. 8: Root mean square error of ATE for EoRoC Dataset.

Without loop closure being triggered, it achieves 0.01% FDE in odometry. VIL-SLAM is robust and achieves this result by successfully registering new scans to the original *highbay* map at re-entry. The map generated with the odometry estimate of VIL-SLAM is compared with the map generated with LOAM before its failure. The boxed region is where LOAM accumulates errors leading to its failure.

The *outdoor* test features an outdoor trajectory which is 546m long and includes a gentle slope. Pedestrians and cars were observed which served as potential outliers. VIL-SLAM and LOAM have comparable results along the xy-plane. However, LOAM fails to capture the changes in the z-direction. The inaccuracy in z of LOAM is also observed in the previous tests.

Overall, VIL-SLAM generates more accurate mapping results and lower FDE compare to LOAM when they both finish. Also, VIL-SLAM succeeds the more challenging environments where LOAM fails with qualitatively good mapping and odometry results.

C. EuRoC MAV Dataset test

VIL-VIO contributes to the robustness and accuracy of VIL-SLAM. We evaluate the VIO using the EuRoC MAV dataset [39] in terms of the absolute trajectory error (ATE) as in [44]. Fig. 8[3] shows the comparison results between VIL-VIO and three state-of-the-art methods. Results for VIL-VIO are deterministic, obtained in real-time on a desktop with 3.60GHz i7-4790 CPU. Results for the other methods are the better ones from experiments in [7] and [12]. VIL-VIO succeeds all sequences with accuracy comparable with the others, verifying its capability to handle aggressive motion, illumination changes, motion blur and textureless regions.

IX. Conclusions

VIL-SLAM is a state-of-the-art odometry and mapping system designed to robustly operate long term in different environments. Current framework loosely couples VIL-VIO and LiDAR mapping. We are extending it to a tightly-coupled framework such that refined pose estimate from LiDAR mapping could be used for IMU biases correction. In loop closure, ICP refinement operates on sparse feature points between scans. We suspect that we would obtain a better loop constraint by matching a scan to map.

[3]A sequence is named in the first four letters and the difficulty level is encoded in the last letter (E:easy, M:medium, D:difficult)

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 02:56:18 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, pp. 1309–1332, Dec 2016.

[2] D. G'alvez-L'opez and J. D. Tard'os, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, pp. 1188–1197, October 2012.

[3] R. Dub, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "Segmatch: Segment based place recognition in 3d point clouds," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5266–5272, May 2017.

[4] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, Feb 1992.

[5] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, pp. 1004–1020, Aug 2018.

[6] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," vol. 34, 02 2014.

[7] J. Hsiung, M. Hsiao, E. Westman, R. Valencia, and M. Kaess, "Information sparsification in visual-inertial odometry," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, (Madrid, Spain), Oct. 2018. To appear.

[8] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 611–625, March 2018.

[9] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225–234, Nov 2007.

[10] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to mav navigation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3923–3929, Nov 2013.

[11] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," in *2012 IEEE International Conference on Robotics and Automation*, pp. 957–964, May 2012.

[12] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 3, pp. 965–972, April 2018.

[13] M. Quan, S. Piao, M. Tan, and S.-S. Huang, "Map-based visual-inertial monocular slam using inertial assisted kalman filter," 09 2017.

[14] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," pp. 298–304, Sept 2015.

[15] K. Wu, A. Ahmed, G. A. Georgiou, and S. I. Roumeliotis, "A square root inverse filter for efficient vision-aided inertial navigation on mobile devices," 2015.

[16] J. Engel, J. Sturm, and D. Cremers, "Camera-based navigation of a low-cost quadrocopter," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2815–2821, Oct 2012.

[17] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, "Information fusion in navigation systems via factor graph based incremental smoothing," *Robotics and Autonomous Systems*, vol. 61, pp. 721–738, 2013.

[18] V. Usenko, J. Engel, J. Stckler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1885–1892, May 2016.

[19] S. Ceriani, C. Snchez, P. Taddei, E. Wolfart, and V. Sequeira, "Pose interpolation slam for large maps using moving 3d sensors," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 750–757, Sept 2015.

[20] M. Velas, M. Spanel, and A. Herout, "Collar line segments for fast odometry estimation from velodyne point clouds," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4486–4495, May 2016.

[21] C. Wei, T. Wu, and H. Fu, "Plain-to-plain scan registration based on geometric distributions of points," in *2015 IEEE International Conference on Information and Automation*, pp. 1194–1199, Aug 2015.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 02:56:18 UTC from IEEE Xplore. Restrictions apply.

[22] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," 07 2014.

[23] J.-E. Deschaud, "Imls-slam: scan-to-model matching based on 3d data," *CoRR*, vol. abs/1802.08633, 2018.

[24] D. Droeschel, J. Stckler, and S. Behnke, "Local multi-resolution representation for 6d motion estimation and mapping with a continuously rotating 3d laser scanner," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5221–5226, May 2014.

[25] J. A. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," 2018.

[26] J. Zhang and S. Singh, "Laser-visual-inertial odometry and mapping with high robustness and low drift," 08 2018.

[27] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15–22, May 2014.

[28] J.-Y. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker description of the algorithm," vol. 1, 01 2000.

[29] J. Shi and Tomasi, "Good features to track," in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, June 1994.

[30] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, pp. 2564–2571, Nov 2011.

[31] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *Robotics: Science and Systems*, 2015.

[32] L. Carlone, Z. Kira, C. Beall, V. Indelman, and F. Dellaert, "Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4290–4297, May 2014.

[33] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 3565–3572, April 2007.

[34] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1722–1729, May 2017.

[35] G. Sibley, L. Matthies, and G. Sukhatme, "Sliding window filter with application to planetary landing," *Journal of Field Robotics*, vol. 27, no. 5, pp. 587–608.

[36] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *2011 IEEE International Conference on Robotics and Automation*, pp. 3281–3288, May 2011.

[37] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o(n) solution to the pnp problem," *Int. J. Comput. Vision*, vol. 81, pp. 155–166, Feb. 2009.

[38] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[39] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[40] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," tech. rep., Georgia Tech, Sept. 2012.

[41] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP Variants on Real-World Data Sets," *Autonomous Robots*, vol. 34, pp. 133–148, Feb. 2013.

[42] "DBoW3 dbow3." `https://github.com/rmsalinas/DBoW2`, 2017.

[43] "Cloud-to-Cloud Distance cloudcompare." `https://www.cloudcompare.org/doc/wiki/index.php?title=Cloud-to-Cloud_Distance`, 2015.

[44] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 573–580, Oct 2012.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on September 11,2026 at 02:56:18 UTC from IEEE Xplore. Restrictions apply.