



ScienceDirect



Download

Robotics and Autonomous Systems

Volume 112, February 2019, Pages 201-210

Semi-direct monocular visual and visual-inertial SLAM with loop closure detection

Shao-peng Li ^{a, b}✉, Tao Zhang ^a✉, Xiang Gao ^c, Duo Wang ^a, Yong Xian ^b^a Department of Automation, Tsinghua University, Beijing, 100084, China^b High-Tech Institute of Xi'an, Xi'an, 710025, China^c Department of Computer Science, Technical University of Munich, Germany

Received 14 February 2018, Revised 7 October 2018, Accepted 13 November 2018, Available online 26 November 2018.



Check for updates

Show less ^



Outline



Share



Cite

<https://doi.org/10.1016/j.robot.2018.11.009>[Get rights and content](#)

Highlights

- A novel semi-direct monocular visual SLAM system is proposed.
- The SLAM system also fuses inertial data to form a visual-inertial SLAM.
- The method maintains the advantages of the direct and the feature-based method.
- Comparative results of semi-direct SLAM are given.

Abstract

A novel semi-direct monocular visual simultaneous [localization](#) and mapping (SLAM) system is proposed to maintain the fast performance of a direct method and the high precision and loop closure capability of a feature-based method. This system extracts and matches Oriented FAST and Rotated BRIEF features in a [keyframe](#) and tracks a non-keyframe via a direct method without the requirement of extracting and matching features. A keyframe is used for global or [local optimization](#) and loop closure, whereas a non-keyframe is used for fast tracking and localization, thereby combining the advantages of direct and feature-based methods. A monocular visual-inertial SLAM system that fuses [inertial measurement](#) data with visual SLAM is also proposed. This system successfully recovers the metric scale successfully. The evaluation on datasets shows that the proposed approach accomplishes loop closure detection successfully and requires less time to achieve accuracy comparable with that of feature-based method. The physical experiment demonstrates the feasibility and robustness of the proposed SLAM. The approach achieves good balance between speed and accuracy and provides valuable references for design and improvement of other SLAM methods.

[Previous](#)[Next](#)

Keywords

Robot vision; Simultaneous localization and mapping (SLAM); Loop closure detection

1. Introduction

Simultaneous [localization](#) and mapping (SLAM) has been a popular research topic in the field of emerging technology, such as autonomous navigation, [telepresence](#), and virtual and [augmented reality](#). These technologies can be used in [autonomous driving](#), service robots, and environmental conservation, particularly in a GPS-denied environment [1]. With the increasing demand for artificial intelligence and [human–computer interaction](#), SLAM will play an increasingly important role in the future of science and technology. The drift in long-term navigation and real-time performance are two key problems of this technology applied from theory to engineering practice.

A complete visual SLAM framework consists of four parts: tracking front end, optimization back end, loop closure detection, and map construction. Loop closure detection of which is essential for solving drift in long-term navigation and relocalization when tracking is lost. *Visual odometry* (VO) can be divided into two classes: *feature-based* and *direct methods*. Feature-based methods [2], [3], [4] extract salient image features in each image, match them in successive frames using invariant feature descriptors, robustly recover camera pose and

structure using epipolar geometry, and refine pose and structure by minimizing projection errors. The extracted features can also be used in loop closure and relocalization, which renders VO as a complete SLAM. However feature extraction and matching are time-consuming, thereby making feature-based methods slower than direct methods. Direct methods [5], [6], [7], [8] directly recover the camera pose and structure through photometric error without features extraction. However, the loop closure detection of direct SLAM remains an open topic. The drift in long-term navigation is one of the main problems encountered by direct methods.

A novel semi-direct approach is proposed in this study to maintain the fast performance of a direct method and the high precision and loop closure capability of a feature-based method. This approach extracts and matches Oriented FAST and Rotated BRIEF (ORB) features [9] in a [keyframe](#) and tracks a non-keyframe via sparse [image alignment](#) [10] without the requirement of extracting and matching features. A keyframe is used for global or [local optimization](#) and loop closure, whereas a non-keyframe is used for tracking and localization, which is time-saving. Accordingly, this system is called SVL¹ SLAM. In addition, a monocular visual-inertial (VI) SLAM system, which fuses [inertial measurement](#) data to SVL, is proposed and labeled as SVL-VI SLAM. Compared with a direct method, the proposed approaches exhibit the function of loop closure detection, which is necessary for long-term navigation. Furthermore, the results of the proposed monocular SLAM systems are compared with the results of state-of-the-art approaches as demonstrated on open datasets and in physical experiments.

The remainder of this paper is organized as follows. An overview of the current visual, visual-inertial SLAM and loop closure detection is provided in the next section. Sections [3 SVL SLAM system](#), [4 SVL-VI SLAM system](#) demonstrate the SVL and SVL-VI SLAM systems respectively. The results of the open datasets and physical experiments are presented in Section [5](#). Section [6](#) concludes the study.

2. Related works

2.1. Visual SLAM

Early visual SLAM was based on filtering. The nonlinear error model and large computations of this early version restricted its practical application. In recent years, most of the proposed visual SLAM methods are based on [nonlinear optimization](#) back end, which achieves considerably better accuracy than filtering-based methods.

Parallel tracking and mapping (PTAM) [9] is an early typical feature-based SLAM algorithm based on nonlinear optimization back end; It can be run in real time by paralleling motion estimation and mapping tasks and by relying on efficient keyframe-based Bundle Adjustment (BA) [10]. Thereafter, most feature-based methods were improved versions of PTAM, one of which is ORB-SLAM [11]. ORB-SLAM can be run in real time. It is robust to severe motion clutter, allows wide baseline loop closure and relocalization, and includes fully automatic [initialization](#), which is the best feature-based SLAM to our knowledge.

Large-scale direct SLAM (LSD-SLAM) [12] is a typical direct SLAM algorithm that can be run in real time without GPU acceleration. On the basis of LSD-SLAM, direct sparse odometry (DSO) [13] samples pixels evenly throughout the images and integrates a fully photometric calibration, which accounts for exposure time, lens vignetting, and nonlinear response functions. DSO outperforms all direct and feature-based methods in terms of both tracking accuracy and robustness. However, DSO also experiences the problem that all direct VO faces: no loop closure detection. *Semi-direct* visual odometry SVO [14], which lies between feature-based and direct methods also achieves impressive results. It is classified as a direct method because no feature matching is required.

A decision cannot be made on which is better between feature-based and direct methods. From the present results, however, the best direct method (i.e., DSO) exhibits better performance than the best feature-based method (i.e., ORB) in terms of accuracy and speed. However, DSO only performs VO without loop closure detection, which is a weakness of this method when working for a long period. Determining how to utilize the advantages of these two types of approaches is the starting point of this study.

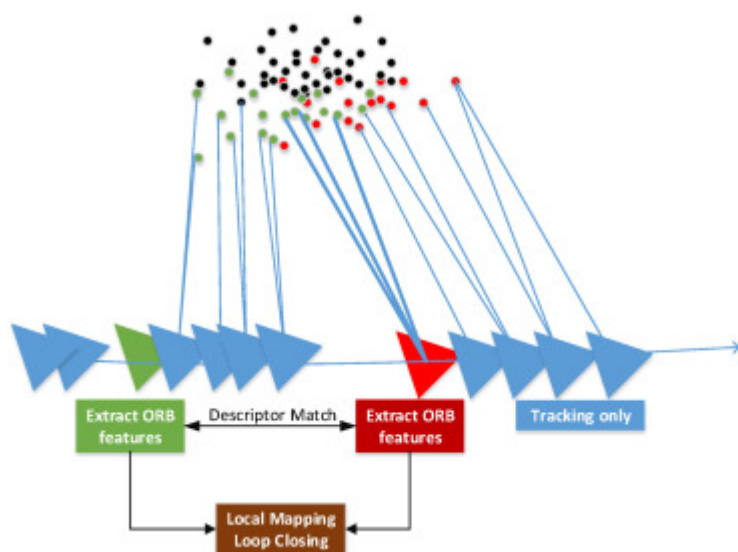
2.2. Visual-inertial SLAM

The fusion of visual and inertial is low-cost and complementary. It has been studied extensively and exhibits considerable potential. A camera is an exteroceptive sensor that enables collecting rich information from a textured environment with high performance under slow-motion condition. However, a monocular camera only cannot recover the metric scale. IMU is a proprioceptive sensor that renders metric-scale of monocular vision and gravity observable [15]. It also performs efficiently in a texture-less environment or under fast-motion condition. The data fusion method of *visual inertial odometry* (VIO) focuses on tightly-coupled, which simultaneously optimizes all sensors states. The optimization methods of VIO can be classified into filtering [16], [17], [18] and keyframe-based nonlinear optimization [19], [20], [21]. Neither method is significantly better than the other one. VIO methods can also be divided into feature-based method [22], [23] and direct method [21] methods according to the picture processing mode.

Multi-state constraint Kalman filter (MSCKF) [16] is the best VIO based on filter, it is faster than the approaches based on nonlinear optimization methods. OKVIS [24] is a typical VIO based on nonlinear optimization; it can be run with monocular and stereo cameras. Foster et al. [25] proposed and designed a real-time VIO that adopted the *preintegration* theory and the iSAM optimization method; it outperformed the MSCKF and OKVIS in terms of accuracy. Mur-Artal et al. [26] fused inertial data into ORB-SLAM, establishing a complete visual-inertial SLAM (VI-SLAM) system with high performance in accuracy. To our knowledge, it is the only VI-SLAM with loop closure detection. However, OKVIS is the only open-source product among these four approaches, which causes difficulty in method improvement and experimental comparison.

2.3. Loop closure detection

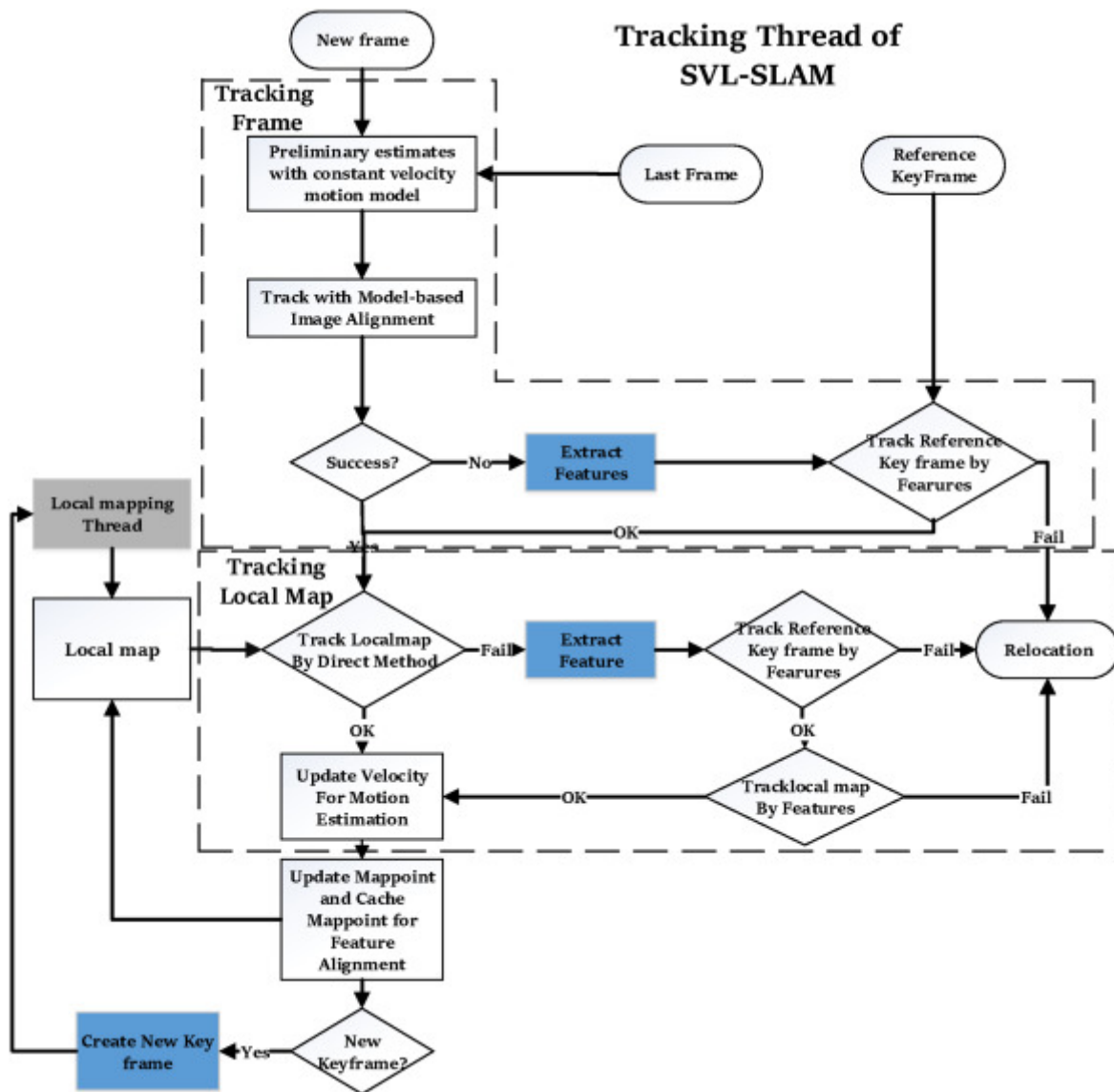
Loop closure detection is done by correctly judging that a robot has returned to its original place to effectively eliminate estimation errors between frames. The description and matching between pictures are the key technologies in loop closure detection. At present, the description of pictures uses *hand-crafted features*, which are designed through the process of artificial feature engineering. Bag-of-Visual-Words (BoVW) [27] descriptor which is based on local key point descriptors, is arguably the most successful image descriptor in loop closure. BoVW characterizes a picture as a histogram of visual words that are vector-quantized versions of the local **keypoint** descriptors. Therefore BoVW is suitable for the feature-based SLAM, i.e., ORB-SLAM, where local keypoint descriptors are extracted per frame. BoVW is also used in our SLAM methods.



[Download : Download high-res image \(267KB\)](#)

[Download : Download full-size image](#)

Fig. 1. SVL schematic. The green and red mappoints are extracted from the **keyframes** respectively. The blue non-key frames are used only for tracking, and pose optimization between keyframes (bold line) is achieved by feature matching. Local mapping and Loop closing threads are only related to keyframes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



[Download : Download high-res image \(556KB\)](#)

[Download : Download full-size image](#)

Fig. 2. Flow chart of SVL tracking thread. The frame adopts a feature-based method when passing through the blue module. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3. SVL SLAM system

Main objective of this study is to develop a semi-direct SLAM system with loop closure detection that can achieve an accuracy comparable with that state-of-the-art feature-based methods and that will enable the system to run as fast as possible, such that it can be adapted to lightweight aircraft for diverse tasks. As shown in Fig. 1, the system determines whether the frame is a **keyframe** based on the changes in scene, extracts features in the keyframe, and tracks these features in non-keyframes using a *direct method*. Therefore, extracting features and matching descriptors are no longer required in a non-keyframe. The system selects ORB as features, which are oriented multiscale FAST corners with an associated 256-bit descriptor.

The features are very fast to be extracted and matched with good **invariance** in viewpoint. The system incorporates three threads that run in parallel, namely, tracking, local mapping, and loop closing, similar to that in [28]. The tracking thread is in charge of localizing the camera pose with each frame and determining whether to insert a new keyframe (Fig. 2). The local mapping involves processing new keyframes and performing *local* BA to create and manage local maps. The loop closing searches for loops with each new keyframe and corrects drift error if the loop is detected.

3.1. Monocular initialization

The monocular **initialization** of SVL adopts the feature-based SLAM method for extracting and matching ORB features every frame, which refers to the initialization method of ORB-SLAM [28]. The local map and reference frame pose are solved with successful initialization and then tracked by the next frame.

3.2. Initial pose estimation from previous frame

After successful initialization, the tracking thread processes every captured image to estimate the pose of the current frame. If tracking is successful for the last frame, then the **constant velocity** motion model is adopted to predict the current camera pose. The 3D mappoints tracked in the reference frame are projected onto the current frame based on the estimated pose. When the ORB features are not extracted in the frame, the relative pose $\mathbf{T}_{cr} \in SE(3)$ between the current and reference frames is solved by minimizing photometric errors between image patches that correspond to the same 3D point. The residual pattern of which the patches is shown in Fig. 3.

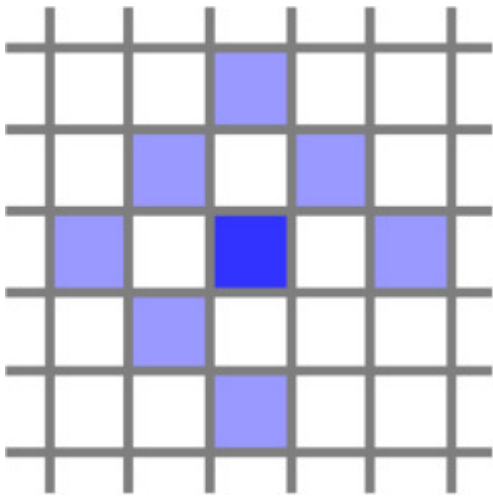
$$\mathbf{T}_{cr} = \arg \min_{\mathbf{T}} \int \int_{\mathcal{R}} \rho [\delta I (\mathbf{T}, \mathbf{u})] d\mathbf{u}, \quad (1)$$

where $\rho [\cdot] = \frac{1}{2} \|\cdot\|^2$, \mathbf{u} is the **pixel position** of the reference frame, \mathcal{R} is the image region, and the intensity residual δI is

$$\delta I (\mathbf{T}, \mathbf{u}) = I_c (\pi (\mathbf{T} \cdot \mathbf{p}_{ri})) - I_r (\mathbf{u}), \quad \forall \mathbf{u} \in \mathcal{R}, \quad (2)$$

where \mathbf{p}_{ri} is the position of the i th tracked 3D mappoint in the reference frame coordinate.

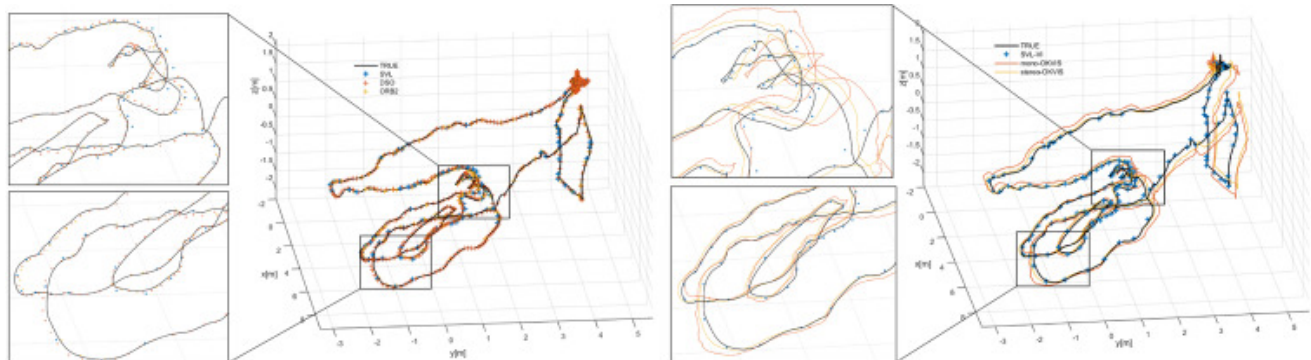
If tracking by direct method fails or if the reference frame velocity is unknown for a special situation, then the ORB features are extracted in the current frame and matched with last frame using the descriptors. If the method fails again, these features should be matched with the reference keyframe (see Fig. 2).



Download : [Download high-res image \(88KB\)](#)

Download : [Download full-size image](#)

Fig. 3. Pattern \mathcal{N} contains 8 pixels, with the bottom right pixel omitted to enable SSE-optimized processing. This pattern exhibits good balance between speed and accuracy, which is verified in [13].



Download : [Download high-res image \(628KB\)](#)

Download : [Download full-size image](#)

Fig. 4. Left chart shows a comparison among SVL, ORB, and DSO with the ground-truth. Right chart shows a comparison between SVL-VI and OKVIS with the ground-truth.

3.3. Tracking a local map using a direct method

In Step 3.2, the current frame pose is only obtained by matching features of the reference frame and its corresponding 3D points, which is insufficiently accurate. The tracking thread searches for *covisibility keyframes* [11] of the reference keyframe and their corresponding *covisibility mappoints* to obtain more matched mappoints that should be tracked and optimized. Although the ORB features are not extracted in the current frame, the pixels in current frame are matched with the local mappoints by pixel patch photometric errors. Local mappoints are projected onto the current frame based on the pose solved in Step 3.2. The corresponding 2D

feature positions \mathbf{u}' in the current frame are obtained individually by minimizing the patch photometric errors between the current frame patches and the reference patches in the covisibility keyframes, in which the corresponding 3D mappoints are extracted.

$$\mathbf{u}'_i = \arg \min_{\mathbf{u}'_i} \frac{1}{2} \sum_{j \in \mathcal{N}} \left\| I_c(\mathbf{u}'_i)_j - I_c(\pi(\mathbf{T}_{cw} \cdot \mathbf{p}_{wi})_j) \right\|^2, \quad \forall i, \quad (3)$$

where \mathcal{N} is the pattern in Fig. 3, \mathbf{p}_{wi} is the i th covisibility mappoint in world coordinates. After mappoints matching is completed, the relative pose \mathbf{T}_{cw} is optimized again by minimizing projection errors using *g2o* [29]:

$$\mathbf{T}_{cw} = \arg \min_{\mathbf{T}_{cw}} \frac{1}{2} \sum_i \left\| \mathbf{u}_i - \pi(\mathbf{T}_{cw} \cdot \mathbf{p}_{wi}) \right\|^2 \quad (4)$$

This problem is the *motion-only* BA [10]. The method for this step is analogous to SVO (not the same, Fig. 2). Refer to [14] for a detailed description of this method.

When tracking a local map fails, the ORB features are extracted in the current frame and matched with the local map using descriptors instead of the method in (3).

The mappoints of N (default=5) nearest covisibility keyframes are selected to maintain the local map. Projecting all the local mappoints onto the current frame is time-consuming, and thus *cache mechanism* is adopted to accelerate the mappoints search and matching. Cache mappoints (default number=200), which are successfully tracked in previous frames, are prioritized for tracking. More mappoints will be projected onto the current frame if mappoint projections in cache are insufficient.

3.4. Mappoints management and key frame process

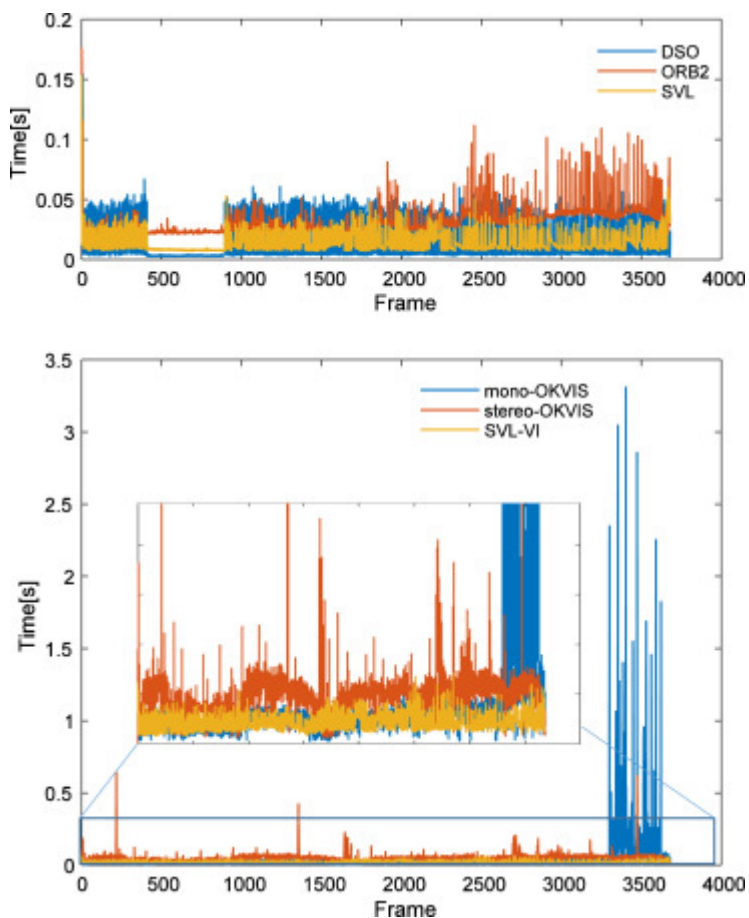
If the mappoints are tracked successfully in Step 3.3, then the mappoints are cached to be prioritized in the next frame. When the tracking is completed for the current frame, frame motion velocity and mappoints should be updated for next frame. Finally, whether the current frame should be spawned as a keyframe is decided, criteria of which refers [11]. If the frame is spawned as a keyframe, new ORB features are extracted and matched while the successfully tracked features are retained in this keyframe. The ORB features are extracted in keyframe; hence, the current keyframe pose and mappoints are optimized further via descriptor matching and projection error in local mapping thread.

If tracking via the direct method fails, then the ORB features are extracted in the current frame for *localization* by matching with the reference keyframe.

3.5. Local mapping and loop closing

Steps 3.1 to 3.4 are parts of the tracking thread, the local mapping and loop closing threads of SVL are related only to the keyframes in which features are extracted. The local mapping thread performs *local* BA [11] to optimize local mappoints and keyframes further via projection error minimization. The loop closing thread detects loops using the bag-of-words

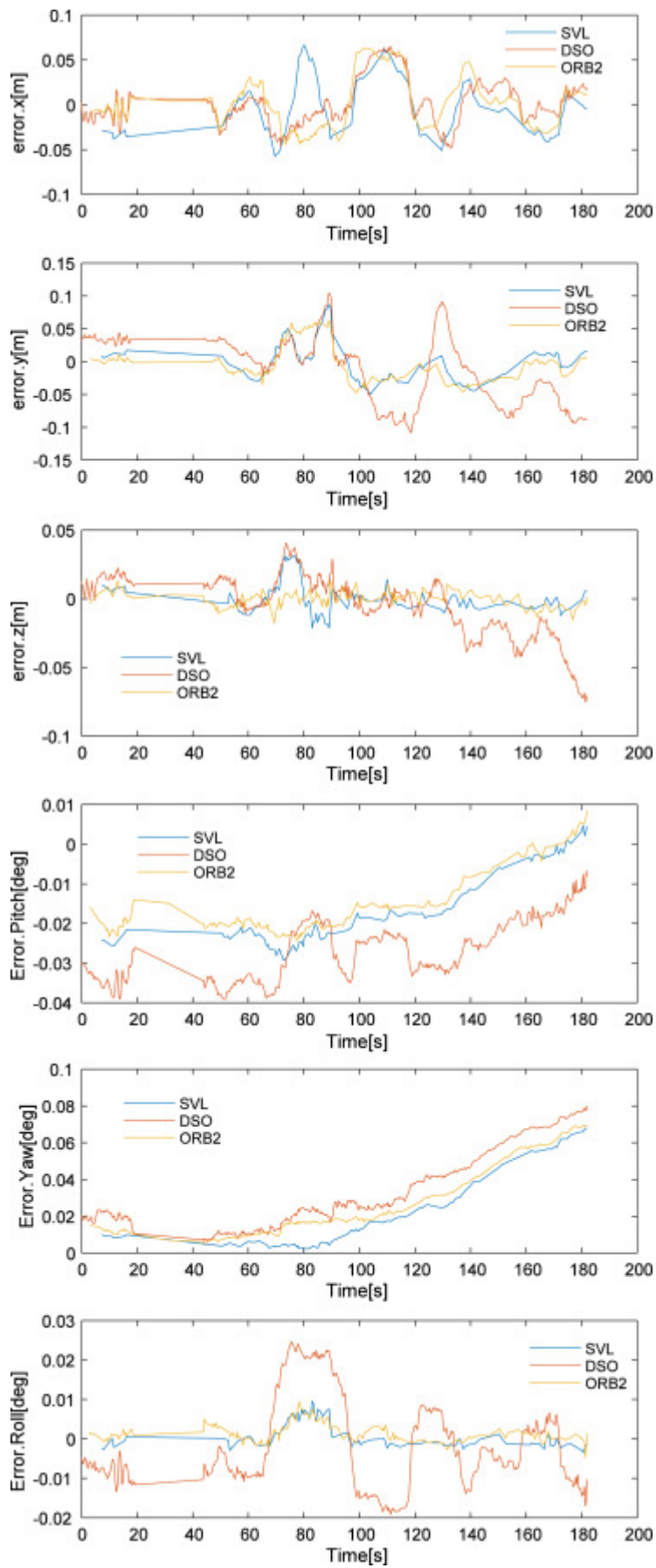
place [recognizer](#) built on DBoW2 with ORB [30]. The accumulated error is corrected via pose graph optimization [31], which distributes loop closing errors along the graph.



[Download](#) : [Download high-res image \(521KB\)](#)

[Download](#) : [Download full-size image](#)

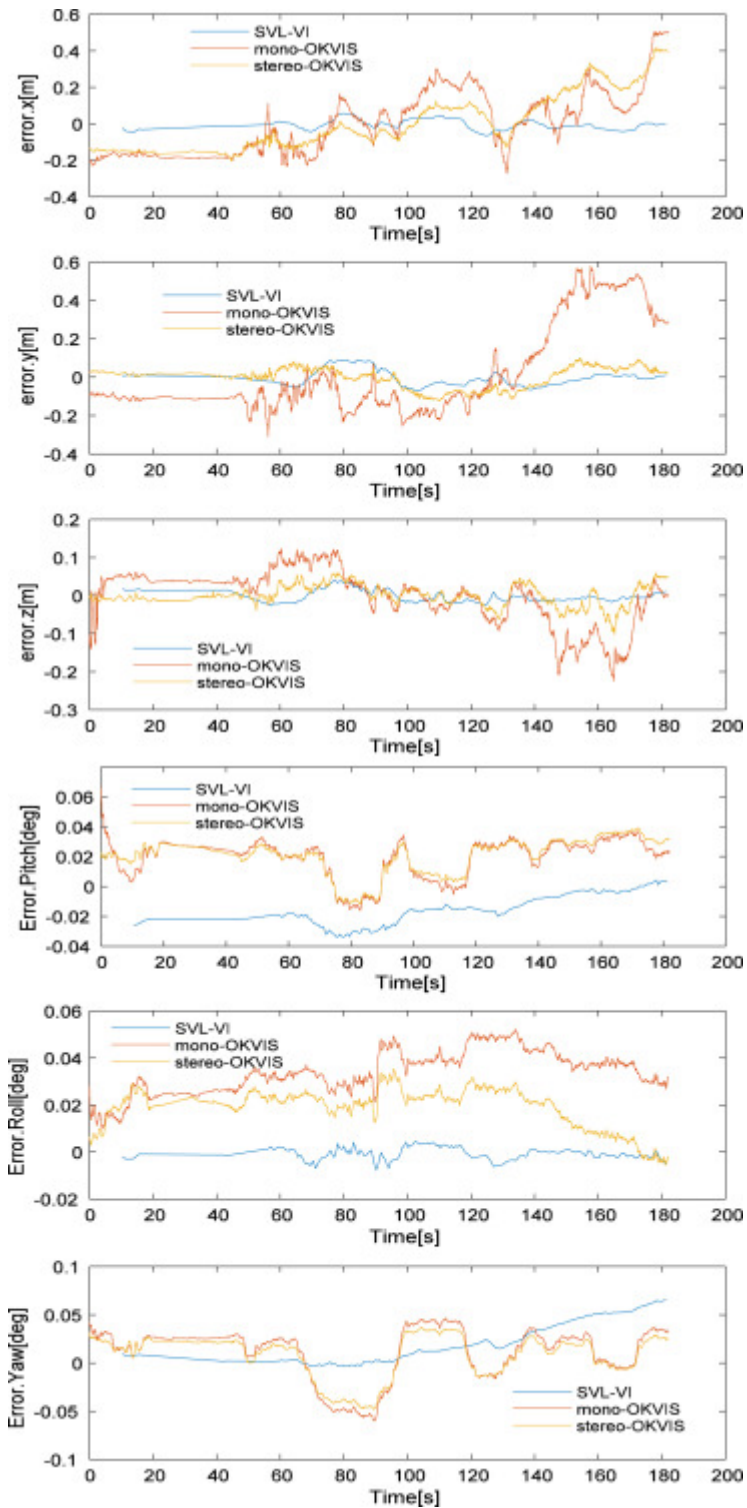
Fig. 5. Processing time per frame of *MH_01_easy*. The top chart shows a comparison of SVL with ORB and DSO. The bottom chart illustrates a comparison of SVL-VI with OKVIS.



[Download : Download high-res image \(852KB\)](#)

[Download : Download full-size image](#)

Fig. 6. Position and attitude drifts of SVL compared with those of ORB and DSO in *MH_01_easy*.



[Download : Download high-res image \(832KB\)](#)

[Download : Download full-size image](#)

Fig. 7. Position and attitude drifts of SVL-VI compared with those of OKVIS in *MH_01_easy*.

4. SVL-VI SLAM system

On the basis of the work in Section 3, a tightly coupled Visual-Inertial SLAM system is proposed, which can close loops and recover the metric scale. IMU data are processed via *preintegration* [25]. Similar to SVL, this system also has three parallel threads for tracking, local mapping and loop closing. Differences exist between these three thread and the visual-only SVL system due to the requirement of fusing the inertial navigation state. The basic flow of this system is presented in [26]. Given that the ORB features are not extracted in non-keyframe, the features matching is completed by minimizing photometric errors similar to that in (3).

4.1. Tracking

The tracking thread of SVL-VI is in charge of tracking pose, velocity, and IMU bias. The basic flow of this thread is shown in Fig. 2 (We did not exclusively draw the flowchart of SVL-VI exclusively. The tracking thread of SVL-VI is analogous, in which the initial pose is predicted by the IMU instead of the *constant velocity* motion mode in SVL, and the reference frame is the last *keyframe* or the last frame depending on whether the map is updated.). The initial pose is predicted by the IMU, which is more reliable than the constant velocity motion mode. The current frame is optimized further by minimizing the photometric error between the current and reference frames, similar to that in 3.2. After matching is completed using the method presented in (3), the current frame j is optimized by minimizing projection error similar to that in (4) with an IMU state error term. Optimization is different depending on whether the map is updated by another two threads.

If the map is updated, then the IMU term links the current frame j to the last keyframe i . Keyframe i is fixed, and the optimization variable is the state of the current frame:

$$\begin{aligned}\theta &= \left\{ \mathbf{R}_{wb}^j, {}_w\mathbf{p}_b^j, {}_w\mathbf{v}_b^j, \mathbf{b}_g^j, \mathbf{b}_a^j \right\}, \\ \theta^* &= \arg \min_{\theta} \left(\sum_k \mathbf{E}_p(k, j) + \mathbf{E}_I(i, j) \right),\end{aligned}\quad (5)$$

where the projection error \mathbf{E}_p for a given matched mappoint k , is defined as:

$$\begin{aligned}\mathbf{E}_p(k, j) &= \frac{1}{2} \left\| (\mathbf{u}_k - \pi({}_c\mathbf{p}_k))^T \sum_k (\mathbf{u}_k - \pi({}_c\mathbf{p}_k)) \right\|^2, \\ {}_c\mathbf{p}_k &= \mathbf{R}_{cb} \mathbf{R}_{bw}^j \left({}_w\mathbf{p}_k - {}_w\mathbf{t}_b^j \right) + {}_c\mathbf{t}_b,\end{aligned}\quad (6)$$

where ${}_w\mathbf{p}_k$ is the mappoint in world coordinates; \mathbf{R}_{cb} , \mathbf{R}_{bw}^j and ${}_w\mathbf{t}_b^j$, ${}_c\mathbf{t}_b$ are the rotation and translation matrices between different coordinates respectively; and \sum_k is the information matrix associated with the *keypoint* scale. If tracking local map via the direct method fails, ORB features are extracted in the current frame. When computing \mathbf{E}_p , 3D mappoints and keypoints in the image are matched using photometric errors similar to that in 3.3, or descriptors matched in [11], depending on whether the ORB features are extracted in the current frame j .

The IMU state error term is

$$\begin{aligned}
\mathbf{E}_I(i, j) &= \rho \left([\mathbf{e}_R^T \mathbf{e}_v^T \mathbf{e}_p^T] \sum_I ([\mathbf{e}_R^T \mathbf{e}_v^T \mathbf{e}_p^T])^T \right), \\
&+ \rho (\mathbf{e}_b^T \sum_R \mathbf{e}_b), \\
\mathbf{e}_R &= \log \left((\Delta \mathbf{R}_{ij} \exp(\mathbf{J}_{\Delta R}^g \mathbf{b}_g^j))^T \mathbf{R}_{bw}^i \mathbf{R}_{wb}^j \right), \\
\mathbf{e}_v &= \mathbf{R}_{bw}^i \left({}_w \mathbf{v}_b^j - {}_w \mathbf{v}_B^i - \mathbf{g}_w \Delta t_{ij} \right) \\
&- (\Delta \mathbf{v}_{ij} + \mathbf{J}_{\Delta v}^g \mathbf{b}_g^j + \mathbf{J}_{\Delta v}^a \mathbf{b}_a^j), \\
\mathbf{e}_p &= \mathbf{R}_{bw}^i \left({}_w \mathbf{p}_b^j - {}_w \mathbf{p}_B^i - {}_w \mathbf{v}_b^i \Delta t_{ij} - \frac{1}{2} \mathbf{g}_w \Delta t_{ij}^2 \right) \\
&- (\Delta \mathbf{p}_{ij} + \mathbf{J}_{\Delta p}^g \mathbf{b}_g^j + \mathbf{J}_{\Delta p}^a \mathbf{b}_a^j), \\
\mathbf{e}_b &= \mathbf{b}^j - \mathbf{b}^i,
\end{aligned} \tag{7}$$

where \sum_I and \sum_R are the information matrices of the preintegration and of the bias random walk, respectively; and ρ is the Huber robust cost function. The estimating result serves as a prior for the next optimization.

If no map is updated, then the next frame $j + 1$ will be optimized with a link to frame j . Both frames should be optimized as follow:

$$\begin{aligned}
\theta &= \left\{ \mathbf{R}_{wb}^j, {}_w \mathbf{p}_b^j, {}_w \mathbf{v}_b^j, \mathbf{b}_g^j, \mathbf{b}_a^j, \right. \\
&\quad \left. \mathbf{R}_{wb}^{j+1}, {}_w \mathbf{p}_b^{j+1}, {}_w \mathbf{v}_b^{j+1}, \mathbf{b}_g^{j+1}, \mathbf{b}_a^{j+1} \right\}, \\
\theta^* &= \arg \min_{\theta} (\sum_k \mathbf{E}_{proj}(k, j + 1) + \mathbf{E}_{IMU}(j, j + 1) \\
&\quad + \mathbf{E}_{prior}(j)),
\end{aligned} \tag{8}$$

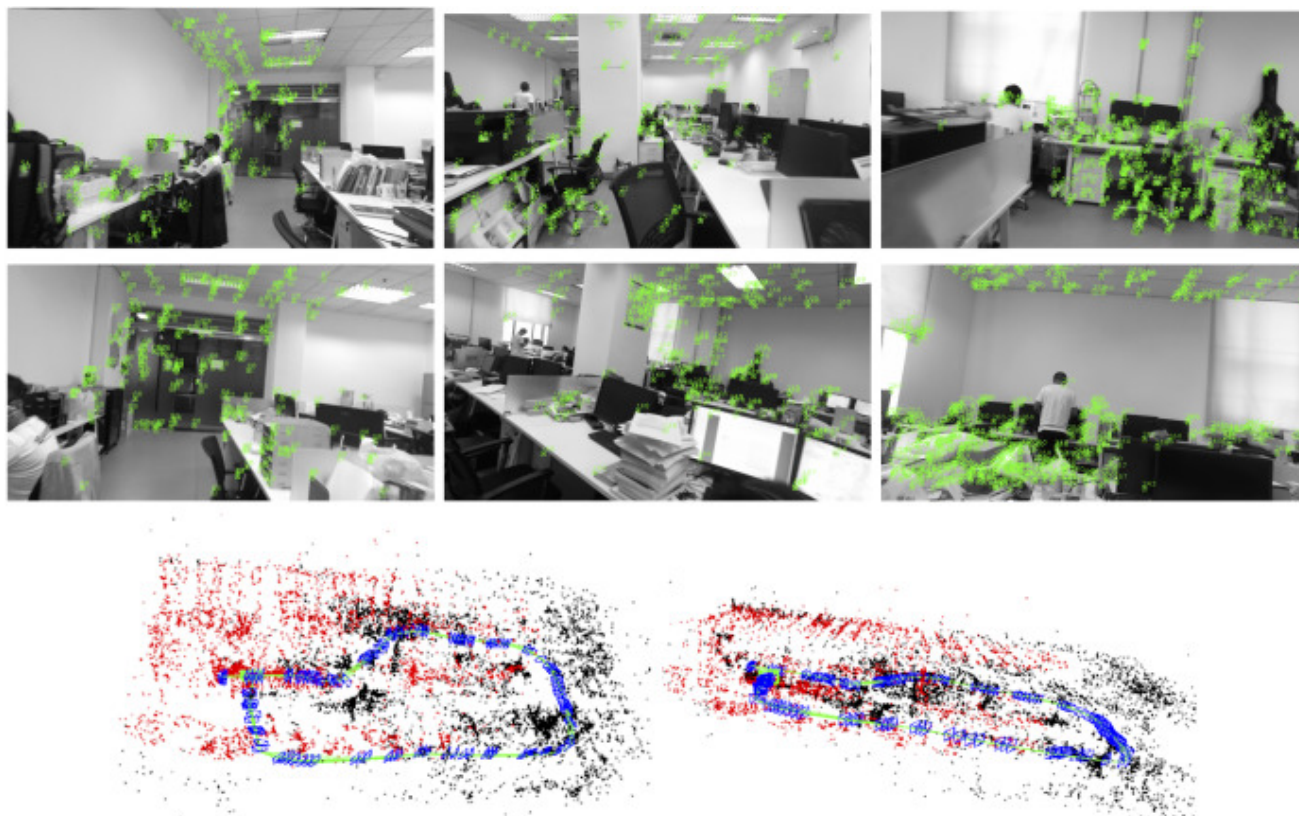
where \mathbf{E}_{prior} is a prior term.

$$\begin{aligned}
\mathbf{E}_{prior}(j) &= \rho \left([\mathbf{e}_R^T \mathbf{e}_v^T \mathbf{e}_p^T \mathbf{e}_b^T] \sum_p ([\mathbf{e}_R^T \mathbf{e}_v^T \mathbf{e}_p^T \mathbf{e}_b^T])^T \right), \\
\mathbf{e}_R &= \text{Log} \left(\bar{\mathbf{R}}_{bw}^j \mathbf{R}_{wb}^j \right), \quad \mathbf{e}_v = {}_w \bar{\mathbf{v}}_b^j - {}_w \mathbf{v}_b^j, \\
\mathbf{e}_p &= {}_w \bar{\mathbf{p}}_b^j - {}_w \mathbf{p}_b^j, \quad \mathbf{e}_b = \bar{\mathbf{b}}^j - \mathbf{b}^j,
\end{aligned} \tag{9}$$

where $(\bar{\cdot})$ is the estimated states that result from the previous optimization in (5). This optimization, which links two consecutive frames and uses a prior, is repeated until the map changes [26].

4.2. Local mapping, loop closing and IMU initialization

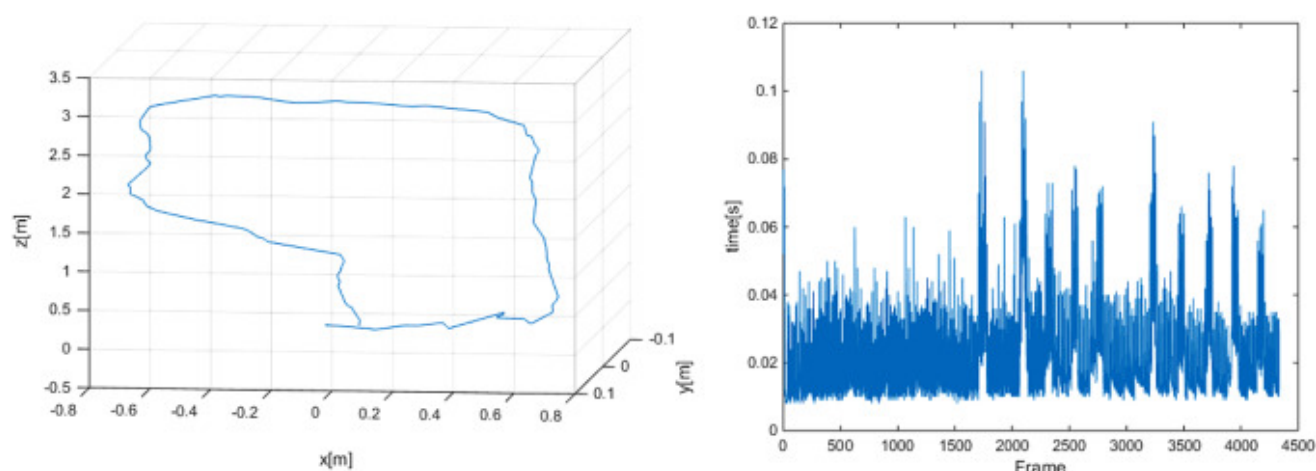
Compared with that of SVL, the local mapping thread of SVL-VI should add an IMU error item into the [optimization objective](#) when performing local BA after a new keyframe is inserted. The loop closing thread is not related to IMU when detecting loops, but the scale is observable in SVL-VI, and thus pose graph optimization is performed with 6 DoF instead of 7. IMU [initialization](#) is performed after a few seconds [26] running of the monocular SLAM. The gyroscope, accelerometer bias, scale, velocity and gravity direction should be estimated during this part. The complete IMU initialization can be found in [26].



[Download : Download high-res image \(2MB\)](#)

[Download : Download full-size image](#)

Fig. 8. Top view shows the sample frames. Bottom view shows two simulation interfaces from different observation directions. The red or black points represents the features in or out of view, respectively. The green lines connect [keyframes](#). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



[Download : Download high-res image \(331KB\)](#)

[Download : Download full-size image](#)

Fig. 9. [Keyframe](#) trajectory and processing time per frame of the physical experiment.

5. Experiments

The proposed approaches are evaluated on the EuRoC MAV dataset [32], which contains several sequences that provide two pictures of a stereo camera, along with gyroscope and accelerometer measurement data. The dataset also contains ground truth measured using Vicon and Leica equipment. The experiment was performed by processing *left image* only. Section 5.1 compares our approach with the state-of-the-art showing that the proposed method can achieve comparable accuracy at a considerably faster rate. Section 5.1.1 presents details of the status of SVL against state-of-the-art approaches, namely, ORB and DSO. Section 5.1.2 provides details of the status of SVL-VI against the monocular and stereo SLAM of OKVIS. Finally, a physical experiment was performed to verify the robustness and feasibility of the proposed system.

5.1. Simulation results

We evaluate the accuracy of SVL on the sequences of the EuRoC dataset. We start processing the sequences from the first frame that contains extremely shaky motion used to initialize the IMU bias, such that the simulation result may vary slightly from [13] and [26], which crop the beginning of each sequence and process it when evaluating the dataset. For SVL, the number of image pyramid levels is set to 4, scale factor is set to 2.0, local BA window size is set to 10 [keyframes](#), the number of features for each frame is set to 1000, and the number of cache features in tracking is set to 200. The same configurations are used for ORB-SLAM. For SVL-VI, the number of image pyramid levels, scale factor, and local BA window size are set to the same values as those in SVL. The number of features for each frame is set to 1200 and cache features number in tracking is set to 300.

Table 1. Trajectory accuracy in euroc dataset (raw ground-truth).

Sequence	SVL			ORB			DSO		
	Meantime (s)	RMSE (m)		Meantime (s)	RMSE (m)		Meantime (s)	RMSE (m)	
MH_01_easy	0.0130	0.0455		0.0303	0.0421		0.02508	0.0588	
MH_02_easy	0.0135	0.0452		0.0285	0.0361		0.02953	0.0449	
V1_01_easy	0.0156	0.0974		0.0305	0.0947		0.02921	0.0987	
V2_01_easy	0.0139	0.1078		0.0276	0.0586		0.02424	0.0961	
Sequence	SVL-VI			OKVIS (mono)			OKVIS (stereo)		
	Meantime (s)	RMSE (m)	Scale	Meantime (s)	RMSE (m)	Scale	Meantime (s)	RMSE (m)	Scale

Sequence	SVL		ORB		DSO				
	Meantime (s)	RMSE (m)	Meantime (s)	RMSE (m)	Meantime (s)	RMSE (m)	Meantime (s)	RMSE (m)	
MH_01_easy	0.0182	0.0569	1.0001	0.0433	0.2849	1.0461	0.0506	0.1656	1.0193
MH_02_easy	0.0186	0.0593	1.0632	0.0219	0.2807	0.9461	0.0475	0.1332	0.9843
V1_01_easy	0.0191	0.0994	0.9887	0.0144	0.0983	0.9985	0.0322	0.0424	1.0046
V2_01_easy	0.0194	0.3236	1.0277	0.0137	0.3718	0.9959	0.0332	0.3238	1.0109

We save the keyframe trajectories only for SVL, SVL-VI, and ORB, and all the frame trajectories for DSO and OKVIS according to an [open-source code](#). For an effective comparison, the number of features for each frame in DSO is set to 800. The speed of this set is closer to that of SVL. [Table 1](#) shows the translation [Root Mean Square Error](#) (RMSE) of the frame trajectory for each sequence, as proposed in [33]. We also measure the ideal scale factor that will optimally align the estimated trajectory and ground-truth. For Visual-Inertial SLAM, this scale factor can be measured and is observable.

As shown in [Table 1](#), for visual-only SLAM, SVL achieves an accuracy comparable with those of ORB and DSO. The extremely shaky motion at the beginning of each sequence has a worse impact on DSO than on SVL or ORB. Thus the accuracy of *Fast* configuration (number of features is 800) DSO is inferior to those of the other two approaches. SVL does not extract or match features in non-keyframes. Its accuracy is slightly worse than that of ORB. However, SVL is significantly faster than the other two methods under similar precision, thereby making it more suitable for a lightweight MAV or robot.

For visual-inertial SLAM, SVL-VI can successfully recover metric scale and achieve higher accuracy with faster speed than OKVIS. Stereo-based OKVIS achieves higher accuracy than mono-based OKVIS at the expense of longer computation time. The meantime of mono-OKVIS in *MH_01_easy* sequence is particularly long because of its robustness. SVL-VI exhibits better performance in terms of computation time and accuracy. However, the speed advantage is less apparent than visual-only SLAM because of the similar IMU data processing modes. RMSE is computed before scale factor correction, such that the RMSE of SVL-VI is higher than that of SVL in some sequences. The estimated motion trajectory is shown in [Fig. 4](#).

5.1.1. Simulation results of SVL

The top chart in [Fig. 5](#) illustrates the processing time per-frame in *MH_01_easy*. SVL spends less time to achieve [localization](#) and mapping. The MAV is stalled on the platform for some time between about the 500th and 1000th frames in this sequence. Given that no keyframe is created among these frames, the processing time of DSO and SVL is shorter than that of ORB, which requires features extraction and matching. As the number of processed frames increases, the processing time of ORB becomes considerably longer while no significant

increase is observed for SVL processing time. This result makes working for a long time possible. Fig. 6 shows position and attitude error over time. No evident loop is found in this sequence, where the advantage of loop correction over to odometry is negligible. The position estimation of SVL is similar to that of ORB, as shown in Fig. 6, because of the analogous feature matching and keyframe optimization. The state of each keyframe is saved on the plot after initialization is completed, such that the pose of SVL is not shown in the first frame.

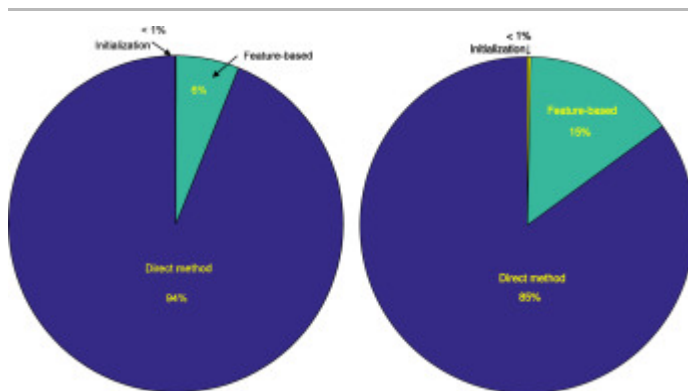
5.1.2. Simulation results of SVL-VI

The bottom chart in Fig. 5 illustrates the processing time per-frame of SVL-VI and OKVIS in *MH_01_easy*. SVL-VI applies *preintegration* [25] theory to shorten IMU data processing time while OKVIS adopts a *slide window* to shorten back end optimization time. The times spent by SVL-VI and monocular OKVIS are basically the same and less than that of stereo OKVIS. However, monocular OKVIS spends a long time in the last frames because of its poor robustness. Fig. 7 illustrates that SVL-VI successfully recovers the metric scale and achieves comparable accuracy. The curves of monocular and stereo OKVIS are similar because they adopt the same strategy, but the fluctuation of mono-OKVIS is larger.

5.2. Experimental results

To verify the robustness and feasibility of the proposed system, we perform a physical experiment in a real-world environment. The Kinect2 camera is connected to an Intel Core i7-4710MQ laptop with 16 Gb RAM by the Robot Operating System (ROS). We only use an RGB image of Kinect, with a resolution of 960×540 . We hold the camera while walking around our laboratory with several students studying at their seats or walking along the corridors, which is a slightly dynamic scene. The light at one side of the laboratory is bright, which increases the difficulty in scene tracking. Finally, we walk back to the starting point, and the last frame observation direction of the camera differs from that of the first to prevent pose correction via loop closure. We only test SVL, which exhibits poorer robustness than SVL-VI.

Fig. 8 illustrates that the SVL has successfully completed scene tracking and localization. The ground-truth of the physical experiment is unavailable, and thus we only present the keyframes trajectory and processing time per-frame in Fig. 9.



Download : [Download high-res image \(130KB\)](#)

Download : [Download full-size image](#)

Fig. 10. Left view shows the ratio of the frames processed by the two methods. Right view shows the time taken by the two methods throughout the process. [Initialization](#) occurs in the first few frames.

5.3. Discussion

SVL can be considered a combination of ORB and SVO. The method for ORB is adopted in keyframes, and SVO is adopted in non-keyframes. In the actual implementation, the method for the original model is improved in order to complete the combination of the two approaches. Frames that use the direct method account for the majority, and they require less time per-frame than feature-based methods. Compared with SVO, SVL extracts feature points at keyframes, improves the precision via the local mapping thread, and completes loop closure detection through the loop closure thread. Compared with ORB, SVL completes acceleration via direct method for non-key frames. As shown in [Fig. 10](#), in the simulation of the *MH_01_easy* dataset, 94% of the frames use the direct method, and they accounts for 85% of the time. The specific proportion of frames using the feature-based method is determined via the keyframe selection strategy and the number of direct method tracking failures. When camera motion is more vigorous, more frames will adopt the feature-based method to ensure accuracy. By contrast, more frames will adopt direct method to ensure speed when the motion is slow. Therefore SVL can achieve good balance between speed and accuracy according to camera motion and environment. The result provides valuable preferences for the design and improvement of other SLAM methods.

6. Conclusion

In this study, we propose a semi-direct monocular visual SLAM system SVL that exhibits the loop closure detection ability, and spends less time to achieve comparable accuracy with state-of-the-art feature-based method. The gain in speed is attributed to the fact that feature extraction and matching are not required for motion estimation in non-keyframes. Feature extraction and matching in [keyframes](#) guarantee the global or [local optimization](#) and loop closure detection capability of SLAM. SVL combines the advantages of direct and feature-based methods, thereby solving the problem of drift in long-term navigation via loop closure detection and achieving comparable accuracy with state-of-the-art through excellent real-time performance. We also realize the fusion of visual and inertial on this system to build SVL-VI, which lays the foundation for the application of this technology to multi-sensor platforms. The simulation result illustrates that our method is the fastest SLAM with loop closure detection to our knowledge. The physical experiment demonstrates the feasibility and robustness of SVL. Our approach will play a major role in lightweight platforms, particularly for long-term navigation and [localization](#). In our future, we will use an RGB-D or stereo camera to improve system robustness.

References


- [1] Adams Martin, Vo Ba-Ngu, Mahler Ronald, Mullane John
Slam gets a phd: new concepts in map estimation
IEEE Robot. Autom. Mag., 21 (2) (2014), pp. 26-37
[CrossRef](#) [View Record in Scopus](#) [Google Scholar](#)
- [2] Scaramuzza Davide, Fraundorfer Friedrich
Visual odometry [tutorial]
IEEE Robot. Autom. Mag., 18 (4) (2011), pp. 80-92
[CrossRef](#) [View Record in Scopus](#) [Google Scholar](#)
- [3] Blösch Michael, Weiss Stephan, Scaramuzza Davide, Siegwart Roland
Vision based mav navigation in unknown and unstructured environments
Robotics and automation (ICRA), 2010 IEEE international conference on, IEEE (2010), pp. 21-28
[CrossRef](#) [View Record in Scopus](#) [Google Scholar](#)
- [4] Weiss Stephan, Achtelik Markus W, Lynen Simon, Achtelik Michael C, Kneip Laurent, Chli Margarita, Siegwart Roland
Monocular vision for long-term micro aerial vehicle state estimation: A compendium
J. Field Robot., 30 (5) (2013), pp. 803-831
[CrossRef](#) [View Record in Scopus](#) [Google Scholar](#)
- [5] Comport Andrew I., Malis Ezio, Rives Patrick
Real-time quadrifocal visual odometry
Int. J. Robot. Res., 29 (2–3) (2010), pp. 245-266
[CrossRef](#) [View Record in Scopus](#) [Google Scholar](#)
- [6] Tykkälä Tommi, Audras Cédric, Comport Andrew I
Direct iterative closest point for real-time visual odometry
Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE (2011), pp. 2050-2056
[View Record in Scopus](#) [Google Scholar](#)
- [7] Kerl Christian, Sturm Jürgen, Cremers Daniel
Robust odometry estimation for rgb-d cameras
Robotics and Automation (ICRA), 2013 IEEE International Conference on, IEEE (2013), pp. 3748-3754
[View Record in Scopus](#) [Google Scholar](#)
- [8] Meilland Maxime, Comport Andrew I
On unifying key-frame and voxel-based dense visual slam at large scales
Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, IEEE (2013), pp. 3677-3683
[View Record in Scopus](#) [Google Scholar](#)

- [9] Klein Georg, Murray David
Parallel tracking and mapping for small ar workspaces
Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on, IEEE (2007), pp. 225-234
[View Record in Scopus](#) [Google Scholar](#)
- [10] Strasdat Hauke, Montiel JMM, Davison Andrew J
Real-time monocular slam: Why filter?
Robotics and Automation (ICRA), 2010 IEEE International Conference on, IEEE (2010), pp. 2657-2664
[CrossRef](#) [View Record in Scopus](#) [Google Scholar](#)
- [11] Raul Mur-Artal, Juan D Tardos, Orb-slam2: an open-source slam system for monocular, stereo and rgb-d cameras. arXiv preprint [arXiv:1610.06475](#), 2016.
[Google Scholar](#)
- [12] Engel Jakob, Schöps Thomas, Cremers Daniel
Lsd-slam: large-scale direct monocular slam
European Conference on Computer Vision, Springer (2014), pp. 834-849
[CrossRef](#) [View Record in Scopus](#) [Google Scholar](#)
- [13] Engel Jakob, Koltun Vladlen, Cremers Daniel
Direct sparse odometry
IEEE Trans. Pattern Anal. Mach. Intell. (2017)
[Google Scholar](#)
- [14] Forster Christian, Pizzoli Matia, Scaramuzza Davide
Svo: fast semi-direct monocular visual odometry
Robotics and Automation (ICRA), 2014 IEEE International Conference on, IEEE (2014), pp. 15-22
[View Record in Scopus](#) [Google Scholar](#)
- [15] Martinelli Agostino
Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination
IEEE Trans. Robot., 28 (1) (2012), pp. 44-60
[CrossRef](#) [View Record in Scopus](#) [Google Scholar](#)
- [16] Mourikis Anastasios I, Roumeliotis Stergios I
A multi-state constraint kalman filter for vision-aided inertial navigation
Robotics and automation, 2007 IEEE international conference on, IEEE (2007), pp. 3565-3572
[View Record in Scopus](#) [Google Scholar](#)
- [17] Wu Kejian, Ahmed Ahmed, Georgiou Georgios A, Roumeliotis Stergios I

A square root inverse filter for efficient vision-aided inertial navigation on mobile devices

Robotics: Science and Systems, Citeseer (2015)

[Google Scholar](#)

- [18] Bloesch Michael, Omari Sammy, Hutter Marco, Siegwart Roland
Robust visual inertial odometry using a direct ekf-based approach
Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, IEEE (2015), pp. 298-304
[CrossRef](#) [View Record in Scopus](#) [Google Scholar](#)
- [19] Indelman Vadim, Williams Stephen, Kaess Michael, Dellaert Frank
Information fusion in navigation systems via factor graph based incremental smoothing
Robot. Auton. Syst., 61 (8) (2013), pp. 721-738
[Article](#)  [Download PDF](#) [View Record in Scopus](#) [Google Scholar](#)
- [20] Usenko Vladyslav, Engel Jakob, Stückler Jörg, Cremers Daniel
Direct visual-inertial odometry with stereo cameras
Robotics and Automation (ICRA), 2016 IEEE International Conference on, IEEE (2016), pp. 1885-1892
[CrossRef](#) [View Record in Scopus](#) [Google Scholar](#)
- [21] Concha Alejo, Loianno Giuseppe, Kumar Vijay, Civera Javier
Visual-inertial direct slam
Robotics and Automation (ICRA), 2016 IEEE International Conference on, IEEE (2016), pp. 1331-1338
[CrossRef](#) [View Record in Scopus](#) [Google Scholar](#)
- [22] Engel Jakob, Sturm Jürgen, Cremers Daniel
Accurate figure flying with a quadrocopter using onboard visual and inertial sensing
Imu, 320 (2012), p. 240
[View Record in Scopus](#) [Google Scholar](#)
- [23] Kneip Laurent, Chli Margarita, Siegwart Roland, *et al.*
Robust real-time visual odometry with a single camera and an imu
BMVC (2011), pp. 1-11
[View Record in Scopus](#) [Google Scholar](#)
- [24] Leutenegger Stefan, Lynen Simon, Bosse Michael, Siegwart Roland, Furgale Paul
Keyframe-based visual-inertial odometry using nonlinear optimization
Int. J. Robot. Res., 34 (3) (2015), pp. 314-334
[CrossRef](#) [View Record in Scopus](#) [Google Scholar](#)
- [25] Forster Christian, Carlone Luca, Dellaert Frank, Scaramuzza Davide
On-manifold preintegration for real-time visual-inertial odometry

IEEE Trans. Robot., 33 (1) (2017), pp. 1-21

[View Record in Scopus](#) [Google Scholar](#)

- [26] Mur-Artal Raúl, Tardós Juan D

Visual-inertial monocular slam with map reuse

IEEE Robot. Autom. Lett., 2 (2) (2017), pp. 796-803

[View Record in Scopus](#) [Google Scholar](#)

- [27] Tianqiang Peng, Fang Li

Bag of visual word model based on binary hashing and space pyramid

Eighth International Conference on Digital Image Processing (ICDIP 2016),

International Society for Optics and Photonics (2016)

pp. 100335T–100335T

[Google Scholar](#)

- [28] Mur-Artal Raul, Montiel Jose Maria Martinez, Tardos Juan D

Orb-slam: a versatile and accurate monocular slam system

IEEE Trans. Robot., 31 (5) (2015), pp. 1147-1163

[CrossRef](#) [View Record in Scopus](#) [Google Scholar](#)

- [29] Kümmerle Rainer, Grisetti Giorgio, Strasdat Hauke, Konolige Kurt, Burgard Wolfram

g 2 o: A general framework for graph optimization

Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE (2011),

pp. 3607-3613

[CrossRef](#) [View Record in Scopus](#) [Google Scholar](#)

- [30] Rublee Ethan, Rabaud Vincent, Konolige Kurt, Bradski Gary

Orb: an efficient alternative to sift or surf

Computer Vision (ICCV), 2011 IEEE international conference on, IEEE (2011), pp. 2564-2571

[CrossRef](#) [View Record in Scopus](#) [Google Scholar](#)

- [31] Strasdat Hauke, Montiel JMM, Davison Andrew J

Scale drift-aware large scale monocular slam

Robot.: Sci. Syst. VI, 2 (2010)

[Google Scholar](#)

- [32] Burri Michael, Nikolic Janosch, Gohl Pascal, Schneider Thomas, Rehder Joern, Omari Sammy, Achtelik MarkusW, Siegwart Roland

The euroc micro aerial vehicle datasets

Int. J. Robot. Res., 35 (10) (2016), pp. 1157-1163

[CrossRef](#) [View Record in Scopus](#) [Google Scholar](#)

- [33] Sturm Jürgen, Engelhard Nikolas, Endres Felix, Burgard Wolfram, Cremers Daniel

A benchmark for the evaluation of rgb-d slam systems

Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, IEEE (2012), pp. 573-580

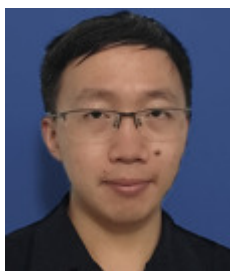
[View Record in Scopus](#) [Google Scholar](#)



Shao-peng Li received Master's degree in automatic control science and engineering from High-Tech Institute of Xi'an, China, in 2016 and is currently perusing the Ph.D. degree in control science and engineering from Tsinghua University. His research interests include computer vision, simultaneous localization and mapping, robotics and artificial intelligent.



Tao Zhang received the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China, in 1999 and the Ph.D. degree in electrical engineering from Saga University, Saga, Japan, in 2002. He became an Associate Professor with Saga University in 2002 and a Research Scientist in the National Institute of Informatics, Tokyo, Japan, in 2003. In 2006, he became an Associate Professor in the Department of Automation, Tsinghua University. His current research interests include pattern recognition, nonlinear system control, robotics, control engineering and artificial intelligent.



Xiang Gao received Ph.D. degree in automatic control science and engineering from Tsinghua University, Beijing, China, in 2017 and is currently doing postdoctoral research in Department of Computer Science, Technical University of Munich, Germany. His research interests include computer vision, simultaneous localization and mapping, robotics and artificial intelligent.



Duo Wang received the B.S. degree in automation from the Harbin Institute of Technology, China, in 2015. Currently he is purchasing the PhD at the Department of Automation, Tsinghua University, Beijing, P.R. China. Currently his research interests are about deep learning/machine learning, and its applications in computer vision and robotics vision.



Yong Xian received the Ph.D. degree in control science and engineering from High-Tech Institute of Xi'an, China, in 2002. He became an professor with High-Tech Institute of Xi'an in 2008. His current research interests include design and optimization of spacecraft orbit, computer vision and artificial intelligent.

¹ SVL is the abbreviation for semi-direct, visual and loop closure detection.

[View Abstract](#)

© 2018 Elsevier B.V. All rights reserved.



About ScienceDirect

Remote access

Shopping cart

Advertise

Contact and support

Terms and conditions

Privacy policy

We use cookies to help provide and enhance our service and tailor content and ads. By continuing you agree to the **use of cookies**.

Copyright © 2021 Elsevier B.V. or its licensors or contributors. ScienceDirect ® is a registered trademark of Elsevier B.V.

ScienceDirect ® is a registered trademark of Elsevier B.V.

