Robust Tightly-Coupled Visual-Inertial Odometry with Pre-built Maps in High Latency Situations

Hujun Bao, Weijian Xie, Quanhao Qian, Danpeng Chen, Shangjin Zhai, Nan Wang, Guofeng Zhang

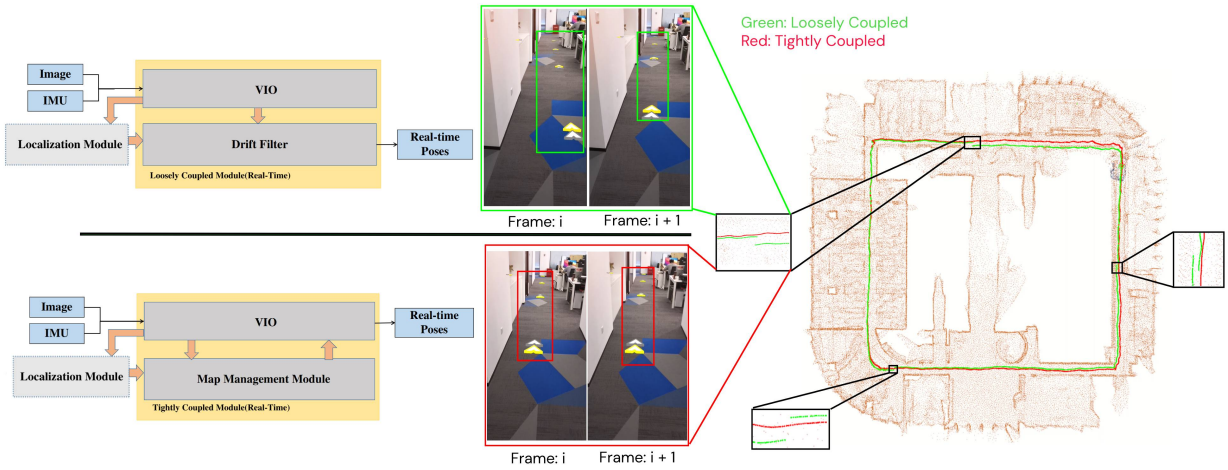


Fig. 1: The two flow diagrams on the left are system overviews of the loosely-coupled method (ARCore-based) and tightly-coupled method (Ours). The figure on the right shows the effect in the actual scene test, including the trajectory and the view of the real-time AR demo at a certain moment. We select two consecutive frames for localization and AR effect comparison. We can see jumpiness of the loosely-coupled method both in trajectory and AR views, while not for our method.

Abstract— In this paper, we present a novel monocular visual-inertial odometry system with pre-built maps deployed on the remote server, which can robustly run in real-time on a mobile device even in high latency situations. By tightly coupling VIO with geometric priors from pre-built maps, our system can tolerate the high latency and low frequency of global localization service, which is especially suitable for practical applications when the localization service is deployed on the remote server. Firstly, sparse point clouds are obtained from the dense mesh by the ray casting method according to the localization results. The dense mesh can be reconstructed from the point clouds generated by Structure-from-Motion. We directly use the sparse point clouds in feature tracking and state update to suppress drift. In the process of feature tracking, the high local accuracy of VIO is fully utilized to effectively remove outliers and make our system robust. The experiments on EurocMav datasets and simulation datasets show that compared with state-of-the-art methods, our method can achieve better results in terms of both precision and robustness. The effectiveness of the proposed method is further demonstrated through a real-time AR demo on a mobile phone with the aid of visual localization on the remote server.

Index Terms—Pre-built Map, VIO, Tightly-Coupled, High Latency

1 Introduction

High-precision localization is fundamental to the areas of robotics, autonomous vehicles, augmented reality and virtual reality. GPS is widely used to provide a global position on the earth. However, the provided localization precision of ordinary GPS is insufficient for AR application, and GPS cannot work in indoor scenes.

With the rise of the digital twin and high-precision maps, the demand for AR and VR of large scenes combined with high-precision maps gradually becomes prosperous. In large-scale environments, it is a challenging problem to effectively suppress the accumulation error of long-distance tracking. Moreover, the memory size of high-precision maps is very large, so map-based services, such as global localization, often run only on remote servers, bringing network latency and bandwidth limitations.

In recent years, visual inertial odometry (VIO) [5, 12, 20, 29, 31, 35] have received a lot of attention. It can estimate the six-degrees-of-freedom (6DoF) pose in real time, even in GPS-denied environments. With the help of IMU, VIO is much more robust than pure visual Simultaneous Localization and Mapping (SLAM) [22, 30, 50]. At present, a series of excellent AR developer platforms in the industry have adopted similar visual-inertial tracking solutions, such as ARKit[1], ARCore[2], Hololens[3]. However, even with the excellent SLAM systems like ARCore, ARKit, and Hololens, the accumulated drift in long-distance tracking has not been thoroughly addressed.

Unlike VIO which focuses on local tracking, localization techniques

- *H. Bao and G. Zhang are with the State Key Lab of CADCG, Zhejiang University. E-mails: {baohujun, zhangguofeng}@zju.edu.cn.*
- *W. Xie and D. Chen are with the State Key Lab of CADCG, Zhejiang University and SenseTime Research. D. Chen is also affiliated with Tetras.AI. E-mails: xieweijian@sensetime.com, chendanpeng@tetras.ai.*
- *Q. Qian, S. Zhai and N. Wang are with SenseTime Research. N. Wang is also affiliated with Tetras.AI. E-mails: qianquanhao1@sensetime.com, zhaishangjin@sensetime.com, wangnan@tetras.ai.*
- *G. Zhang is the corresponding author.*

Manuscript received xx xxx. 202x; accepted xx xxx. 202x. Date of Publication xx xxx. 202x; date of current version xx xxx. 202x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.202x.xxxxxxx

[1] https://developer.apple.com/documentation/arkit
[2] https://developers.google.com/ar
[3] https://www.microsoft.com/en-us/hololens

based on pre-built maps mainly focus on solving global poses. We can categorize the localization methods into visual structure-based methods and geometric structure-based methods based on the map representation. The visual structure-based methods [41, 43, 44, 49] are suitable for solving the global localization. Generally, the map is composed of sparse point clouds and keyframes, which sparse point clouds and keyframes can be built from Structure-from-Motion [45]. The localization based on geometric structure [6,13,15,53,54] considers the visual localization problem as the registration problem between the local visual structure and the dense global map, which usually requires a relatively accurate initial guess.

Compared with VIO, the localization algorithm based on a pre-built map has better global accuracy, requiring more computation, especially in large-scale environments. In addition, as each frame is localized independently, the smoothness of the localization trajectory is worse than the trajectory of VIO. An affordable way to combine the advantages of VIO and pre-built maps is to fuse the pre-built map into the VIO tracking process. In recent years, some studies have begun to focus on this direction [13, 16, 25, 32, 34, 52–54].

We categorized these coupled approaches into loosely-coupled and tightly-coupled ones, depending on whether or not the localization poses or map structures are added to the state update as additional measurements to improve the VIO's real-time trajectory accuracy. As shown in Fig. 1, without additional measurements to VIO, the loosely-coupled approaches do not affect VIO's original state update. Therefore, they can only correct accumulated drift by continuously optimizing the 6-DoF transformation between localization coordinate and VIO coordinate, leading to the jumpiness of the trajectory. In contrast, the tightly-coupled structure-based approach can correct the accumulated drift more smoothly than the loosely-coupled approaches. However, in the previous structure-based tightly-coupled algorithms, only real-time poses are considered to obtain structural information from the map, limiting the tracking accuracy and not suitable for the application scenarios of map services deploying on the remote server.

The main contributions of this paper are listed as follows:

- We propose a novel monocular visual-inertial odometry system that utilizes low-frequency global localization combined with pre-built maps to run robustly in real time on mobile devices even with high-latency localization delays.
- We design a complete scheme to generate the association between the pre-built map and local features in the sliding window of VIO, including point cloud generation, feature tracking, and outlier removal. Our method can work well under the localization service with high delay and low frequency, which is highly suitable for the application scenarios of deploying the localization module on the remote server.
- We use both the real-time pose and global localization pose to extract point clouds from the pre-built map. For these two different kinds of map points, we design two kinds of constraints and add them to VIO' status update to smoothly correct the accumulated drift online.
- Although our tightly-coupled method can work well in most cases, there may still be some extreme cases that lead to the failure of our method, such as long-time localization fault, pedestrian occlusion, etc. If the tightly-coupled algorithm fails, we adopt the loosely-coupled method to help restore the system to the tightly-coupled state.

Compared with previous works, our method can better tolerate the time delay and error of localization. The experiments on EurocMav [3] and simulation datasets demonstrate that our method has better accuracy and robustness than state-of-the-art methods. We also have implemented an AR Demo that can run in real-time on a consumer-grade mobile phone to prove the effectiveness of the proposed method in practical application scenarios.

2 Related Work

2.1 VIO/SLAM

VIO systems can be categorized into filter-based [12, 29, 51] and optimization-based [19, 35] approaches. One of the filter-based representations is MSCKF [29], which combines geometric constraints and IMU measurements in a multi-state constrained Extended Kalman filter (EKF). This method designs a constant-size sliding window of IMU poses in the state vector without maintaining features, and effectively reduces the computational complexity. SR-ISWF [51] uses the square root formula to obtain good numerical properties, which enables the use of a single-precision format to perform the numerical operation, resulting in considerable acceleration compared to the inverse filter (INVF). VINS-Mono [35] is an excellent optimization-based visual-inertial SLAM system with camera-IMU extrinsic calibration and IMU bias estimation. The designing of robust initialization procedures and lightweight backend management make the system have good robustness and accuracy. With the development of deep learning, data-driven methods have been used in inertial navigation. TLIO [23] designs a ResNet network to estimate the 3D displacement, which was used to fuse with the IMU raw data in an EKF framework. RNIN-VIO [7] improves the network by learning the regularity of humans' motion in time series. Moreover, [7] proposes a multi-sensor fusion system, which shows good robustness in extremely challenging environments, coupling with the visual, IMU, and NIN(Neural Inertial Navigation) measurements. However, for various reasons, with the tracking distance increasing, drift accumulation of the VIO system is still inevitable. For VIO, the correction of drift can only rely on the loop closure, which is a very demanding condition.

2.2 Localization

Global localization is a classical problem in computer vision and robotics, which concerns retrieving absolute poses within a known scene. The classic method of solving this problem is to find the 2D-3D correspondences between the 2D features of the query image and the 3D points of the pre-built map. Then, the camera pose can be calculated by applying a Perspective-n-Point (PnP) [11, 17] solver inside a RANSAC loop [10]. In this framework, the pre-built map is usually generated by the Structure-from-Motion [40, 45], consisting of keyframes and sparse point clouds. Traditionally, the candidate frames most similar to the query image are retrieved [2, 46] from the pre-built map, and then 2D-3D matches are generated through local feature descriptor matching. In this framework, the core problem is to select the most suitable interest point detector. Traditional interest points include Brisk [18], Brief [4], ORB [38], SIFT [24], and so on. ORB algorithm is often used in SLAM because of its fast extraction speed. However, the ORB's ability to match varying viewpoints and lighting is not enough for the localization algorithm. In recent years, many learning-based interest points detection [9] and matching methods [42] have appeared and achieved good results. However, the computational costs of these learning-based detection and matching methods are rather high, so they are not suitable for deployment on mobile devices.

2.3 Coupled Approaches

Using the localization results to correct the drift accumulation of VIO is the core problem to be solved in this work. The previous methods can be divided into two categories, i.e. loosely-coupled methods and tightly-coupled methods, as shown in Table 1.

2.3.1 Loosely-Coupled Approaches

In the loosely-coupled approaches, the VIO/VO module is an individual module. VIO/VO's optimization or state update will not add localization poses or pre-built map information as new measurements to improve the real-time accuracy of the VIO/VO. They can only correct accumulated drift at a low (location request) frequency by jump. Platinsky et al. [32] request the pose by image-based localization and optimize the local-to-global transform over the history windows. Mascaro et al. [27] and Qin et al. [33] use pose graph optimization to optimize the transform between GPS coordinate and VIO coordinate.

Table 1: Comparations of different methods

Method	Couple Type	Camera	IMU	Real-time On Mobile	Localization Frequency	Robustness
[32]	Loosely-coupled with Global Pose	Mono	Y	Yes	High	Weak
[27]	Loosely-coupled with GPS	Mono	Y	No	High	Weak
[33]	Loosely-coupled with GPS	Stereo	Y	No	High	Weak
[34]	Loosely-coupled with Pre-built Map	Mono	Y	No	No	Weak
[52]	Loosely-coupled with Pre-built Map	Mono	N	Yes	No	Weak
[21]	Tightly-coupled with GPS	Mono	Y	No	High	Weak
[8]	Tightly-coupled with GPS	Mono	Y	No	High	Weak
[54]	Tightly-coupled with Localization Pose	Stereo	Y	No	No	Weak
[26]	Tightly-coupled with Structure	Mono	Y	Yes	No	Medium
[14]	Tightly-coupled with Structure	Stereo	N	No	No	Medium
[13]	Tightly-coupled with Structure	Stereo	Y	No	No	Medium
[53]	Tightly-coupled with Structure	Mono	N	No	No	Medium
[25]	Tightly-coupled with Structure	Mono	N	No	No	Medium
Our Method	Tightly-coupled with Global Pose and Structure	Mono	Y	Yes	Low	Strong

Qin et al. [34] loads the pre-built map, and then the frame in the sliding window will continuously do loop detection with the loaded map. Once a loop is detected, the error is corrected through 4DoF pose map optimization. Similar to [34], by localizing to the pre-built map, Yamaguchi et al. [52] use 7-DoF optimization to estimate the scale and transformation of the VO map and the pre-built map.

2.3.2 Tightly-Coupled Approaches

The tightly-coupled approaches add global localization poses or pre-built maps as additional measurements to either EKF-based or optimization-based framework to improve the real-time accuracy of VIO. Furthermore, we can categorize the tightly-coupled algorithms into the tightly-coupled with poses [8, 21, 54] and the tightly-coupled with structure [13, 14, 25, 53].

Li et al. [21] uses double-differenced GNSS to fuse with IMU and camera in a EKF-based VIO framework to achieve accuracy navigation performance. Cioffi et al. [8] add the global position measurements, which is estimated by IMU preintegration, to an optimization-based VIO system to improve the accuracy and global consistency. Zuo et al. [54] use distribution transform (NDT)-based method to estimate the rotation and translation between the semi-dense map, which is obtained by stereo matching, and the prior LIDAR map. The results of the NDT-based registration are used in the MSCKF's state update along with the conventional sparse visual feature measurements to correct accumulated drift. It is difficult for the tightly-coupled methods based on localization poses to model the localization error, especially with the low localization frequency.

Another method of correcting drift is to use the structure of the pre-built map. Lu et al. [25] exploit the planar structure obtained from both vision and prior LIDAR data and use it as the anchoring information to fuse the heterogeneous map. By adding co-planarity constraints to global bundle adjustment, the error of visual SLAM is effectively reduced. Lynen et al. [26] propose a system coupled with a loaded 3D point cloud reconstructed by SfM. Despite the use of compression representation with compact features, this method is still difficult to deal with very large scenes and is sensitive to environment changes. Huang et al. [13] proposed a complete visual-inertial localization system based on hybrid map representation. They designed an efficient data association module to associate map components with generating temporal landmarks, which will improve feature tracking. However, [13] uses real-time pose to extract point clouds and uses these points in following visual tracking, which has hidden dangers even with outlier removal strategies. Because the point clouds obtained from the pre-built map and the real-time poses are interdependent and mutually affected. The final accuracy will worsen if the real-time pose is inaccurate or point clouds have noise. Ye et al. [53] proposed to use coplanar constraints to deal with the inaccuracy of the real-time pose. Using the real-time pose to render vertex and normal maps from the prior surfel map, they get the global planar information for the sparse tracked points in the image. The final optimization can accurately estimate the global 6DoF camera pose with the absolute scale. However, this method does not take full advantage of the pre-built map and is easy to fall into degeneration states. Moreover, it is sensitive to map noise.

In this paper, we use both the real-time poses and global localization poses to extract point clouds from the pre-built map. We propose a map management module and efficient data association strategy to deal with the localization latency and low-frequency localization. Then, we use different constraints to integrate structural information into VIO's state update for different types of map points. In addition, we analyze how to determine the degeneration state and how to recover from the degeneration state.

3 Overview

3.1 System Overview

The framework of our system is illustrated in Fig. 2. Our system consists of three modules: map management module, VIO module, and localization module.

The localization module accepts the query image from the VIO module and returns its global pose. In our system, the localization module can run in a low frequency and adopt different localizing algorithms according to the actual situation, as long as it can return the global pose of the query image. We can even deploy it on a remote server and request it over the network. We will describe how we implement the localization module in supplemental materials. However, it is not the focus of our work as it is configurable.

The map management module is responsible for managing the point clouds from the pre-built map. The detail of map management module will be introduced in Sect. 4.1. Once the map management module obtains a successful localization result from the localization module or a query frame from VIO, it will use the ray casting [37] algorithm to obtain sparse 3D point clouds from the dense mesh. The pre-built maps are typically deployed on the server. The first-time localization module localizes success. It will return the global localization pose and the simplified dense mesh of the corresponding region for local ray casting. For the outdoor scene shown in the supplementary video, the whole area is about 20,000 square meters, and the size of the whole simplified dense mesh is 20M, which is acceptable for online transmission. We can further compress the mesh and use block management and transmission, but that is not the focus of this paper.

In the VIO module, we constantly re-attempt to associate the point clouds with frames in sliding window (Sect. 4.2). After generating map observation, we present how to add different constraints based on different map point types into state update, as introduced in Sect. 5. After each state update in VIO progress, we will check if the system is in a degeneration state. We introduce how to detect degeneration state and recover from degeneration state in Sect. 6. Besides sending localization requests in a degeneration state, the VIO module sends a

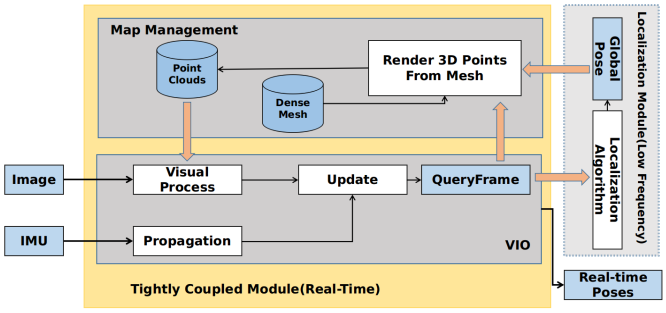


Fig. 2: Detailed system overview.

query image at a predefined low frequency.

Our method uses ray casting to get structural information to align with VIO, while [13] uses ray casting to supplement 3D points for real-time tracking. Therefore, our map management module can manage point clouds asynchronously, regardless of latency. We use ray casting to obtain 3D points corresponding to all feature points on the image to obtain enough point clouds even at a low request frequency. Once the point clouds loaded are done and can still be observed in the latest field of view, the VIO module will build the association of local feature points and map points.

3.2 Notation

We now define notations and frame definitions that we use in the paper. We consider $(\cdot)^G$ as the global frame of VIO. $(\cdot)^C$ is the camera frame, and $(\cdot)^I$ is the body frame or IMU frame. $(\cdot)^M$ is the map frame, which is defined in a pre-built map coordinate. We use both rotation matrix $\boldsymbol{R}$ and Hamilton quaternion $\boldsymbol{q}$ to represent rotation. $\bigotimes$ represents the multiplication operation between two quaternions. $\boldsymbol{q}_B^A, \boldsymbol{p}_B^A$ are rotation and translation from the B coordinate to A coordinate. $\boldsymbol{p}_{f_j}^A$ is the position of the point f_j in A coordinate. I_k is the body frame while taking the k^{th} image. C_k is the camera frame while taking the k^{th} image. M_k is the map frame while taking the k^{th} image. $\pi(\cdot)$ donates the normalized projection operation, which can project $[x,y,z]$ into the normalized coordination:

$$\pi([x,y,z]^\top) = [\frac{x}{z}, \frac{y}{z}]^\top. \tag{1}$$

4 Visual Processing

4.1 Point Cloud Management

The first step of the tightly-coupled algorithm is to obtain high-quality point clouds from the pre-built map. High-quality point clouds need to meet the following conditions:

- The number of points has to be sufficient. If the initial number of points is insufficient, it is difficult to set an appropriate threshold to remove outliers.
- Feature descriptors of points obtained from the pre-built map can match feature points of the current scene so that VIO's visual tracking can use these map points.
- The positions of the points should be accurate enough, otherwise it will cause great difficulty in removing outliers.

Point Cloud Extraction: As we mention in Sect. 2.2, the localization module will generate 2D-3D matches and estimate the camera pose with PnP. A simple way to generate point clouds is to directly create point clouds through 2D-3D matches generated by the localization module. However, the sparse point clouds generated by 2D-3D matches have significant weaknesses. One disadvantage is that the number of 2D-3D matches available may decrease with changing the scenarios, viewpoint, and illumination. Another disadvantage for generating point clouds from 2D-3D matches is that many 3D points may not be sufficiently observed during the reconstruction of the pre-built map, so the positions of these 3D points may be inaccurate.

In this work, we choose to obtain the sparse point clouds from the pre-built dense mesh according to the localization poses. Inspired by techniques widely used in volumetric dense mapping, we use ray casting to obtain sparse point clouds from the dense mesh. The input of this method is the query image, the pose estimated by the localization module or VIO, and the pre-built dense mesh. If global localization poses returned in time from the localization module, we preferentially use global localization poses to obtain point clouds. Otherwise, we use the VIO pose to obtain point clouds. For the convenience of explanation, we call the points obtained by the real-time pose as local map points, while the points obtained by the global localization poses as global map points. The specific steps of extracting sparse point clouds from the dense mesh are as follows:

- Feature points are extracted from the query image using a feature detector algorithm that runs on VIO. In this work, we use an ORB feature detector. For each frame, we extract 500 ORB points.
- We use the ray casting method for each 2D feature point to get its position and normal from the dense mesh. In the end, we can get the point clouds.

Fig. 3 shows the point cloud generated by 2D-3D matches and point cloud generated by ray casting of the same sequence with the same localization frequency, which proves that our method provides much more clean and structured points.

Descriptor of Points: The descriptor of each 3D point of the point clouds comes from its corresponding 2D feature point descriptor in the query image. Since the descriptor of each 3D point in the point cloud comes from the query image, it is independent of the viewpoints and illumination of the data collected for the generation of the pre-built map. So we do not need to collect plenty of data of different viewpoints and illumination conditions like in [32], which dramatically reduces the data collection effort.

Normal of Points: Inspired by [53], we try to use poses from VIO to capture the points with normal from dense mesh in the absence of global pose information. Assuming that the error between the real-time pose and the ground truth pose is small, the points obtained by the real-time pose are supposed to be coplanar with the points obtained by the ground truth pose and have the same normal direction. The ray casting method shoots a ray from the camera through the 2D pixel, intersecting the dense mesh. The normal of the intersected point is the average of the intersected triangle's three vertices, which can be computed during the generation of the dense mesh.

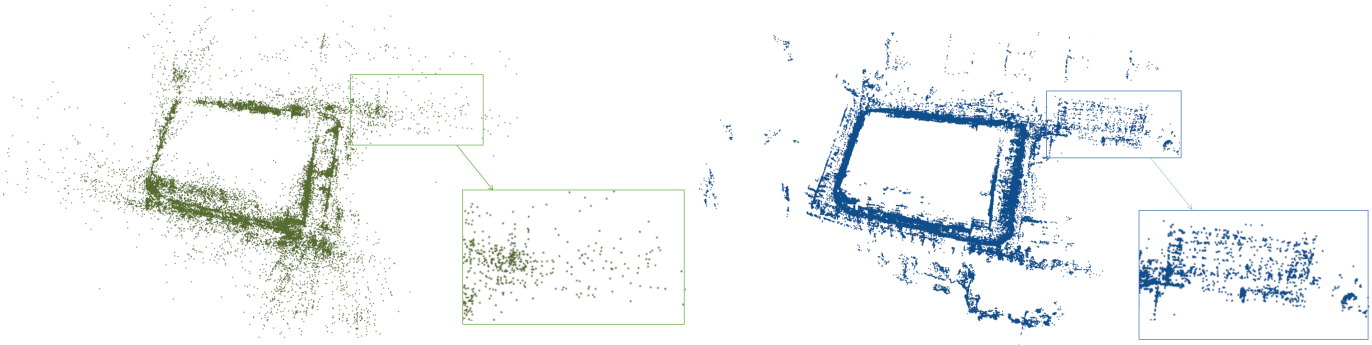
4.2 Generate Map Point Observation

In dynamic scenarios, noise is unavoidable, even if the point cloud comes from an ideal pre-built map. If outliers are not strictly removed, incorrect observations added to MSCKF's status updates can have a negative impact on the VIO system and even cause it to fail. Therefore, the removal of outliers is the most critical step for a tightly-coupled approach.

We design a rigorous outlier removal algorithm, as illustrated in Fig. 4. The key idea for removing outliers is that both the VIO's trajectory and point clouds are accurate in short-range tracking. The problem becomes how to align the point cloud structure of the sliding window with the point cloud structure obtained from the pre-built map. ICP algorithm [39] is a standard method of point clouds alignment, but it is not suitable for a real-time system based on mobile devices due to its high computational complexity. Therefore, we use the feature tracking of the sliding window to achieve point cloud alignment, which is more effective than ICP.

Map Point Association: For the k^{th}image, we project the map points into this frame for projection-based matching. We can project a map point to image coordinates as follows:

$$\boldsymbol{x}_{m_j}^{C_i} = \pi\left(K\left(\boldsymbol{R}_G^{C_i}\left(\boldsymbol{R}_M^G \boldsymbol{p}_{m_j}^M + \boldsymbol{p}_M^G\right) + \boldsymbol{p}_G^{C_i}\right)\right), \tag{2}$$



(a) (b)

Fig. 3: (a) Point clouds generated by 2D-3D matches in real-world outdoor test. The feature type we used is SuperPoint [9]. (b) Point clouds generated by ray casting in real-world outdoor test. The point cloud of (b) has a better structure and a more abundant number of points. The number of points in the point clouds obtained by 2D-3D matching is about 25% of the number of point clouds obtained by ray casting.

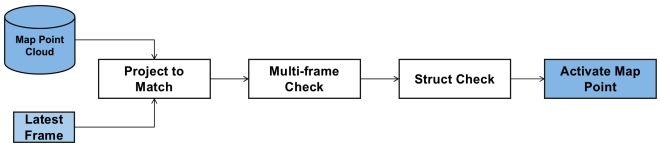


Fig. 4: Outlier removal process.

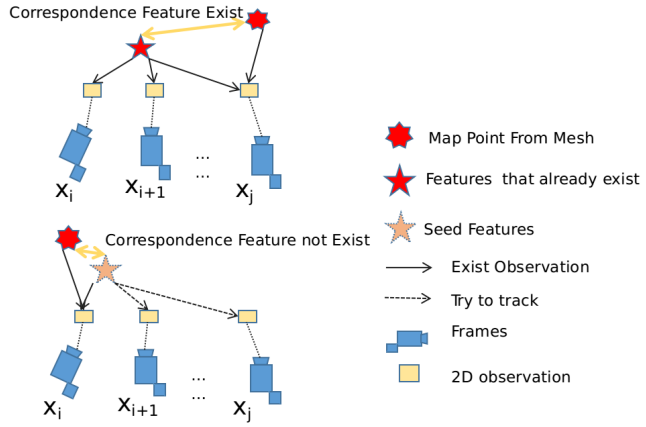


Fig. 5: Generating an association between map points and features.

where $\boldsymbol{p}_{m_j}^{M}$ is the map point position obtain from the pre-built map, K is the camera intrinsic matrix.

The process of projection-based matching is similar to [30]. Through projection matching, we can get several 2D-3D matches. Using the PnP solver inside a RANSAC loop, we can get the camera pose of C_i and the inliers of these 2D-3D matches.

Despite projection-based matching, we also use FAST [36] detector and KLT [48] sparse optical flow tracker to provide feature measurements as [35]. To maintain a minimum number (100-120) of features in each image, new fast corner features are detected for tracking. The next step is to build the correspondences of the map points and the local features. As shown in Fig. 5, if a map point and a local feature tracked by optical flow match to the same 2D point in k^{th} image, then we can bind the map point to this local feature. If the 2D point has not been tracked by any local feature, we will generate a seed local feature, and try to track this seed local feature in the next image. In this way, each map point will correspond to a local feature. For the convenience of explanation, we define the correspondence of local features and map points as $\mathscr{F}_{total}$.

Multi-Frame Check: Each local feature records observations in several frames, which can be used for multi-frame verification of map points. We can use Equation 2 to project map point to each observation of its corresponding local feature and calculate the reprojection error as follows:

$$\boldsymbol{z}^{j,i} = \left\| \boldsymbol{x}_{m_j}^{C_i} - \mu_{f_j}^{C_i} \right\|_2, \tag{3}$$

where $\mu_{f_j}^{C_i}$ is the observation of the j^{th} feature in i^{th} frame.

Then we can find the candidate correspondence of local feature and map point by the following criteria:

- If $\boldsymbol{z}^{j,i} > a_p$, s.t. $\exists \mu_{f_j}^{C_i}$, where a_p is a const reprojection threshold, we will skip this correspondence this time.
- If the track length of local feature is not sufficient, we will skip this correspondence this time.
- Otherwise, this correspondence will be added to $\mathscr{F}_{candidate}$.

Note that we will not delete the correspondence that is not added to $\mathscr{F}_{candidate}$, which will be re-checked next time.

Struct Check: In [13], the map points obtained from the pre-built map will directly be involved in pose estimate after multi-frame verification. However, we believe that it is not reliable to separately verify the accuracy of points. Only when there are enough matches in $\mathscr{F}_{candidate}$, which means that alignment of local feature point cloud and pre-built map point cloud is reliable, we can activate these map points and add them to VIO's status updates as constraints. Otherwise, none of the map points in clusters will be added to the VIO's status updates as additional constraints.

5 Visual-Inertial State Estimate

This section presents the proposed visual-inertial estimate that tightly fuses visual and inertial measurements and prior map constraints to bound tracking errors. Fig. 6 shows the factor graph of our sliding window.

5.1 State Vector

We define the states of our system as follows:

$$\boldsymbol{X}_k = [\boldsymbol{X}_{I_{k-m+1}}^{\top}, ..., \boldsymbol{X}_{I_k}^{\top}, \boldsymbol{X}_{E_k}^{\top}]^{\top}, \tag{4}$$

where m is the length of the sliding window and set to 12 in this paper. $\mathbf{X}_{I_i}$ $(i = k-m+1, ..., k)$ denotes the state vector of the cloned IMU poses at frame i. Each cloned state is defined as:

$$\boldsymbol{X}_{I_i} = [\boldsymbol{q}_{I_i}^{G^{\top}}, \boldsymbol{p}_{I_i}^{G^{\top}}]^{\top}, \tag{5}$$

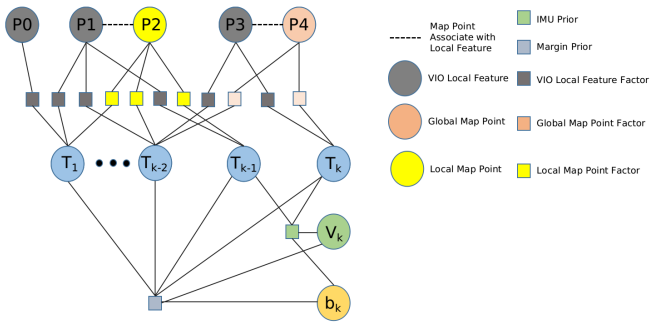


Fig. 6: Factor graph representation for our VIO.

where $\boldsymbol{q}_{I_i}^G$ is the quaternion representation of the orientation of frame i in the global frame $\{G\}$, $\boldsymbol{p}_i^G$ is the position of frame i in global frame $\{G\}$. And the last part $\boldsymbol{X}_{E_k}$ is defined as follows:

$$\boldsymbol{X}_{E_k} = [\boldsymbol{b}_{g_k}^\top, \boldsymbol{b}_{a_k}^\top, \boldsymbol{v}_{I_k}^{G\top}]^\top, \tag{6}$$

where $\boldsymbol{b}_{g_k}$ and $\boldsymbol{b}_{a_k}$ correspond to the bias of gyroscope and acclerater. $\boldsymbol{v}_k^G$ is the velocity of frame k in $\{G\}$.

5.2 Visual Measurement

1) Local Feature Cost: When local feature tracks have reached maximum track length or have lost track, they will be triangulated and further refined by bundle adjustment (BA). Same with standard MSCKF methods, 3D points will be projected into nullspace, and only the camera with IMU motion states remain in the state vector. The constraints of local visual features are:

$$C_z(\tilde{\boldsymbol{X}}_{k+1}) = ||\boldsymbol{H}_x\tilde{\boldsymbol{X}}_{k+1} - \tilde{\boldsymbol{r}}_z||^2_{\sigma\boldsymbol{I}_2}, \tag{7}$$

where σ is the Gaussian noise of local feature measurement and $\boldsymbol{I}_2$ is the 2×2 identity matrix. $\tilde{\boldsymbol{r}}_z$ is the local feature measurement residual.

2) Global Map Point Cost: In Sect. 4.2, we introduced how to establish the association between map points and *tracks* and activate the constraints of map points. In particular, the global map point residual is:

$$\tilde{\boldsymbol{r}}_{gm}^j = \sum_i \omega_j \tilde{\boldsymbol{r}}_{gm}^{j,i} = \sum_i \omega_j \left(\pi(\boldsymbol{R}_G^{C_i}\boldsymbol{p}_{m_j}^G + \boldsymbol{p}_G^{C_i}) - \mu_j^{C_i}\right), \tag{8}$$

$$\boldsymbol{p}_{m_j}^G = \boldsymbol{R}_M^G\boldsymbol{p}_{m_j}^M + \boldsymbol{p}_M^G, \tag{9}$$

where $\boldsymbol{p}_{m_j}^M$ is the map point position obtained from pre-built map, $\mu_j^{C_i}$ is the observation of the j^{th} feature that happpens in the i^{th} frame, and ω_j is the weight of this map point, which can be calculated as follows:

$$\omega_j = \begin{cases} 0 & \text{if } l_j < \alpha \\ b^{(l_j-\alpha)} & \text{if } l_j < \beta \\ b^{(\beta-\alpha)} & \text{other} \end{cases} \tag{10}$$

where b, α, β are constant threshold, and l_j is the track length of local feature.

So, we can write the linearized cost term of global map point measurements as:

$$C_z\left(\tilde{\boldsymbol{X}}_{k+1}\right) = \left\|\omega_j\left(\boldsymbol{H}_x^j\tilde{\boldsymbol{X}}_{k+1} - \tilde{\boldsymbol{r}}_{gm}^j\right)\right\|^2_{\sigma\boldsymbol{I}_2}, \tag{11}$$

where σ is the Gaussian noise of global map point measurement and $\boldsymbol{I}_2$ is the 2×2 identity matrix.

3) Local Map Point Cost: Since the point cloud obtained from real-time pose is interdependent with the state of the sliding window, even if the association between local map point and local feature is inaccurate, it can still easily be activated through the verification in Sect. 4.2. Therefore, we would like to add weaker constraints for the local map points than global map points constraints. Inspired by [53], we assume that the error between the real-time pose and the ground truth pose is small. The activated local map point and its corresponding local feature can still be associated with the same plane and share the same normal value.

Consider the j^{th} local map point's corresponding feature is first observed in I_k frame, the residual of j^{th} local map point is defined as:

$$\tilde{\boldsymbol{r}}_n^j = \sum_i \tilde{\boldsymbol{r}}_n^{j,i} = \sum_i \boldsymbol{n}_{m_j}^{G\top}\left(\boldsymbol{p}_{m_j}^G - \left(\boldsymbol{R}_{C_i}^G\boldsymbol{p}_{f_j}^{C_i} + \boldsymbol{p}_{C_i}^G\right)\right), \tag{12}$$

$$\boldsymbol{n}_{m_j}^G = \boldsymbol{R}_M^G\boldsymbol{n}_{m_j}^M, \tag{13}$$

$$\boldsymbol{p}_{f_j}^{C_i} = \boldsymbol{R}_G^{C_i}\left(\boldsymbol{R}_{C_k}^G\left(d_kK^{-1}\pi^{-1}\left(\mu_j^{C_k}\right)\right) + \boldsymbol{p}_{C_k}^G\right) + \boldsymbol{p}_G^{C_i}, \tag{14}$$

where $\boldsymbol{n}_{m_j}^M$ is the normal of the local map point obtained from the pre-built map. d_k is the depth of the feature in C_k, and $\mu_j^{C_k}$ is the first observation of the local feature in the C_k frame. π^{-1} is the back projection function that turns a pixel location into a unit vector. K is the camera intrinsic matrix.

So, the local map point measurement contributes a linearized cost term as:

$$C_n\left(\tilde{\boldsymbol{X}}_{k+1}\right) = \left\|\boldsymbol{H}_x\tilde{\boldsymbol{X}}_{k+1} - \tilde{\boldsymbol{r}}_n^j\right\|^2_\sigma, \tag{15}$$

where σ is the Gaussian noise of local map point measurement.

5.3 State Update

All the local features constraints, map constraints, IMU integration measurements, and prior measurements construct a linearized error state equation. The estimate of IMU integration measurements cost and prior measurements cost is the same with [51]. Then we use [28,51]'s square root inverse filter to update all our states $\boldsymbol{X}_{k+1}$.

6 Degeneration Analysis

6.1 System Recovery

Although our tightly-coupled method can work well in most cases, there may still be some extreme cases that lead to the failure of our method, such as long-time localization fault, pedestrian occlusion, etc. According to Equation 2, if the error of $\boldsymbol{q}_M^G$ and $\boldsymbol{p}_M^G$ is too large, the projection-based matching will fail, which leads to the failure of activating the map constraints and thus unable to suppress drift accumulation. In this case, we need to have a strategy to recover from this degenerate situation. We constantly check the number of 2D-3D matches generated by projection-based matches and the number of activating map point constraints. If the number of map point tracking and the number of map point constraints remain low, a global localization request must be sent. To avoid the failure of global localization, we will select the latest image with good visual features to make the localization request. When we get a new localization result from the localization module, we can estimate new $\hat{\boldsymbol{q}}_M^G$ and $\hat{\boldsymbol{p}}_m^G$ by optimizing the following problem:

$$\min_{\hat{\boldsymbol{q}}_M^G, \hat{\boldsymbol{p}}_M^G}\left\{\sum\left\|\boldsymbol{q}_i^G \otimes \boldsymbol{q}_i^{M^{-1}} \otimes \hat{\boldsymbol{q}}_G^M\right\|_2 + \sum\left\|\boldsymbol{p}_i^G - \hat{\boldsymbol{q}}_M^G\boldsymbol{p}_i^M - \hat{\boldsymbol{p}}_M^G\right\|_2\right\}, \tag{16}$$

where $\boldsymbol{q}_i^G$ is the quaternion representation of the orientation of the query frame i in {G}, $\boldsymbol{p}_i^G$ is the position of query frame i in {G}, $\boldsymbol{q}_i^M$ is the quaternion representation of the orientation of the query frame i in pre-built map {M}, $\boldsymbol{p}_i^M$ is the position of query frame i in {M}. $\boldsymbol{q}_i^M$ and $\boldsymbol{p}_i^M$ is estimate by localization algorithm.

By optimizing the problem in Equation 16, we can get $\hat{\boldsymbol{q}}_M^G, \hat{\boldsymbol{p}}_M^G$ and its Hessian matrix $\boldsymbol{H}_{rel}$. Then, we can estimate its entropy [1] as follows:

$$entropy = 0.5 \times \log\left((2\pi e)^k det\left(\boldsymbol{H}_{rel}\right)\right). \tag{17}$$

To avoid the mistake of updating these matrices, we update the $\boldsymbol{q}_M^G$ and $\boldsymbol{p}_M^G$ only if the following criteria satisfied:

Table 2: Evaluation on general localization performance on EuRoC MAV dataset with APE (m). The first column is the name of method used, the bold number indicates the best performing one. The results of ORB (offline) are listed for explanation and reference only.

Dataset	V101	V102	V103	V201	V202	V203
BVIO	0.055	0.064	0.086	0.054	0.106	0.129
RTC-VIO	**0.020**	**0.023**	**0.035**	**0.021**	0.027	**0.047**
OpenVINS	0.050	0.084	0.078	0.068	0.064	0.081
VINS-Mono (loop)	0.039	0.037	0.087	0.076	0.105	0.330
ORB (online)	0.427	1.176	0.985	0.417	0.864	2.308
GMM W/ Map	0.023	0.057	0.058	0.047	0.040	0.392
DSL (left cam)	0.035	0.034	0.045	0.026	**0.023**	0.103
MSCKF (w/Map)	0.056	0.055	0.087	0.069	0.089	0.149
ORB (offline)	0.041	0.017	0.029	0.051	0.017	0.030

Table 3: Evaluation on general localization performance on EuRoC MAV dataset with APE (m). Compare the accuracy improvements brought by different tightly coupled algorithms. With GT means use ground-truth pose to obtain point clouds from the pre-built map.

Dataset	V101	V102	V103	V201	V202	V203
GMM w/ map	0.023	0.057	0.058	0.047	0.040	0.392
RTC-VIO	**0.020**	**0.023**	**0.035**	**0.021**	**0.027**	**0.047**
GMM w/ GT	0.022	0.039	0.045	0.027	0.032	0.151
RTC-VIO w/ GT	**0.013**	**0.019**	**0.028**	**0.017**	**0.023**	**0.029**

- $norm\left(\boldsymbol{p}_M^G - \hat{\boldsymbol{p}}_M^G\right) > p_{threshold}$
- $entropy > \lambda_e$

where $p_{threshold}$, λ_e are constant threshold.

6.2 Initialization

As outlier removal relies on the VIO module, the VIO module needs to be initialized independently before using the constraints of the pre-built map. Referring to [7], we use different initialization strategies depending on different motion types. Actually, after the initialization of the VIO, the system will directly fall into a degeneration state, which can be recovered using the method introduced in Sect. 6.1.

7 Experiments

In this section, we compare our method with several state-of-the-art methods on both real and simulated datasets. We conduct these comparative experiments on a desktop PC with an Intel i7-8700 CPU (3.2GHz *12) and 24 GB RAM. In addition, we also test the effectiveness of our approaches in different cases, such as map noise, localization time delay, and localization frequency. Finally, we implement a demo that can run in real-time on a mobile phone and compare it with ARCore.

For the convenience of explanation, in the experiments, we name our basic VIO as **BVIO**, and the VIO coupled with the map as **RTC-VIO**.

7.1 Real-World Data

We first compare our method with several state-of-the-art methods, including OpenVINS [12], VINS-Mono [35], ORB-SLAM3 [5], MSCKF (w/map) [54], GMM-Loc [13], DSL [53] on the EurocMav datasets. The EurocMav datasets provide 20 Hz stereo grayscale images and 200 Hz ADIS16448 IMU and ground-truth room scans (LIDAR Map). There are 11 sequences in the Euroc datasets, but we only use the ViconRoom sequences because only the ViconRoom sequences have the prior LIDAR map. We use MeshLab[4] to recover the dense mesh, which our method depends on, from the sparse point cloud. The result of GMM-Loc comes from its open-source project[5], which is a stereo prior map based system. Unlike previous work, our method requires the input of global localization poses. To make a fair comparison, we add the translation noise with a mean of 2cm and a variance of 0.25cm to ground-truth poses. If there is no particular explanation in the following experiments, our method will take these poses with noise as the localization poses.

To evaluate the accuracy of the compared localization algorithms, we compute the absolute pose error (APE). For BVIO, OpenVINS, ORB-SLAM3, and VINS-Mono, we use evo[6] to estimate the transformation matrix from the estimated trajectory to the ground truth trajectory. Meanwhile, it is unnecessary to estimate the transformation matrix for tightly-coupled methods, which the methods already coupled with the pre-built map. To exclude the interference of random factors, the data in the Table 2 is the average value of 5 runs. ORB (offline) corresponds to the finally outputted poses by ORB-SLAM3, while the ORB (online) corresponds to the real-time poses estimated by ORB-SLAM3, both generated by the monocular inertial system. We found that the online estimated poses by ORB-SLAM3 are jittering, and APE is quite large. The reason might be that ORB-SLAM3 uses a multi-map strategy and will generate a new map if the tracking temporarily fails. So the online estimated poses may jump "map-to-map". In addition, the online estimated poses may not converge well, and the optimization at the back end may result in sudden changes in the online trajectory. The poses and maps are continuously optimized at the back end, and multiple maps may be fused finally. However, for real-time AR applications, the refined poses in a post-processing stage are meaningless. So we mainly compare the online estimated results of ORB-SLAM3. Table 2 shows that BVIO has compatible accuracy with OpenVINS and VINS-Mono. By comparing RTC-VIO and BVIO, we can get that the method proposed in this paper can significantly improve the accuracy of VIO.

We also compare with other visual (-inertial) localization algorithms based on pre-built maps. The experiments show that the accuracy of our method is the best on most datasets. Only V202 has a slightly lower accuracy than DSL(left cam).

Although we also use the ray casting method to obtain 3D points from the pre-built map as in [13], the coupled method is different. In Table 3, we compare the effects of our methods and GMM(w/ map) on accuracy improvement. As shown in Table 3, our coupled method can improve the accuracy much more effectively than the GMM. In most datasets, the results obtained by our method with the noise poses as the localization results are even better than those obtained by the GMM method with the ground truth poses as the localization results.

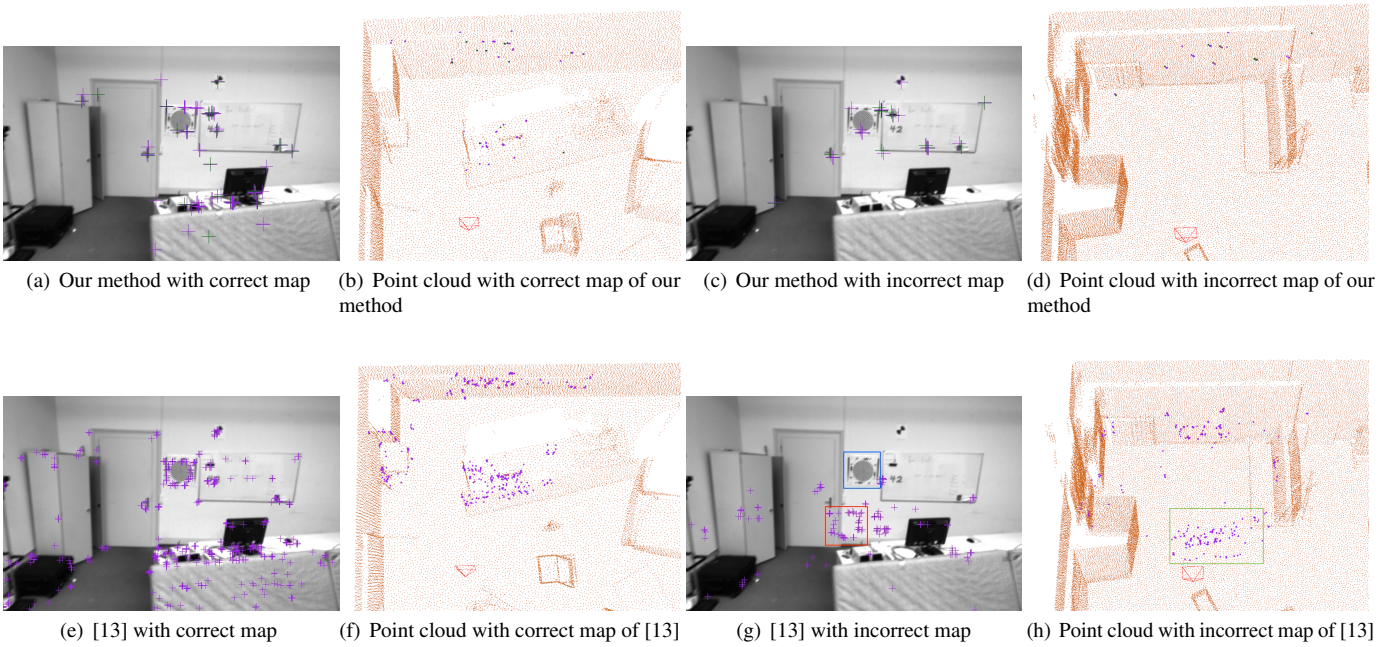
When verifying robustness, previous experiments were limited to adding noise to the prior map. However, compared with noise, changes in scene layout are more common in practical applications. Based on this, we conduct an exciting experiment. The V1 sequences and V2 sequences in the EurocMav datasets are the two different scenarios in the same room with different obstacle configurations. We set the map of V2 as the pre-built map of the V1 sequence and set the map of V1 as the pre-built map of V2, respectively, to re-evaluate the accuracy. The results are shown in Table 4. We are surprised to find that our method helps improve the accuracy of VIO even when the scene changed a lot, while the GMM approach results in a significant drop in trajectory accuracy.

The images in Fig. 7 are screenshots of our method and GMM algorithm. The points in Fig. 7 are 3D points from the pre-built map after outlier removal, which are actually used by the algorithm. It can be seen that our method can well filter out the outliers introduced by environmental changes, while the GMM's algorithm cannot. This experiment shows that our method can detect apparent changes in the scene to avoid the impact of incorrect information on the accuracy of VIO. It also shows that our method can find stable and unchangeable structures from the scenes with significant changes to help improve the

[4] https://www.meshlab.net/

[5] https://github.com/hyhuang1995/gmmloc/

[6] https://github.com/MichaelGrupp/evo.git



(a) Our method with correct map (b) Point cloud with correct map of our method (c) Our method with incorrect map (d) Point cloud with incorrect map of our method

(e) [13] with correct map (f) Point cloud with correct map of [13] (g) [13] with incorrect map (h) Point cloud with incorrect map of [13]

Fig. 7: The purple points are obtained by the real-time pose, while the green points are obtained by the localization pose. The images in the left column show the projection result of these 3D points into the 2D image with the ground truth pose. The pictures in the right column are the screenshots of these points in the 3D space. The correct projection positions of the points in the red box in Fig. 7(g) should be that of the blue box. The points in the green box in Fig. 7(h) are all wrong and fall out of the image when projected back with the ground truth pose so that we cannot see these points in Fig. 7(g).

Table 4: Evaluation on general localization performance on EuRoC MAV dataset with APE (m). V101, V102, V103 use the pre-built map of V2 sequences, while V201, V202, V203 use the pre-built map of V1 sequence. With GT means use ground-truth pose to obtain point clouds from the pre-built map.

Dataset	V101	V102	V103	V201	V202	V203
GMM w/ wrong map	0.469	0.366	0.413	0.851	0.831	1.987
RTC-VIO w/ wrong map	**0.029**	**0.039**	**0.037**	**0.022**	**0.032**	**0.052**
GMM w/ wrong map & GT	0.422	0.399	0.309	0.758	0.803	0.596
RTC-VIO w/ wrong map & GT	**0.023**	**0.023**	**0.033**	**0.019**	**0.028**	**0.041**

accuracy of VIO.

7.2 Evaluation on Synthetic Dataset

Synthetic Dataset: To further validate our system, we compared the proposed method within the AirSim simulator [47]. The simulation datasets provide 30 Hz synchronous RGB images and depth images and 200 Hz MPU-6000 IMU. The datasets contain three sequences, two of which move indoors with different trajectories, and one moves outdoors on a downtown road. The track length of the two indoor sequences is 124.0m and 75.4m, respectively, and the outdoor sequence is 505.3m. We do not compare with [54] on the simulation datasets since it did not provide the code. [13] uses binocular data to triangulate points by stereo matching, while our simulation datasets only have the monocular image. For fairness, we modify the source code of [13] and directly use the ground truth depth instead of stereo matching. The result of the DSL comes from the authors of DSL. In addition, the OpenVINS fails to initialize on all of these sequences, so we do not record its results. As shown in Table 5, our method achieves the best results on all simulation data.

Table 5: Evaluation on general localization performance on synthetic dataset with APE (m). The first line is the name of method used, and the bold number indicates the best performing one. All methods use the pre-built map without noise.

Dataset	BVIO	RTC-VIO	ORB (online)	VINS-Mono (loop)	GMM	DSL
indoor	0.230	**0.023**	0.826	0.159	-	0.050
indoor patial	0.062	**0.020**	3.300	0.078	0.068	0.060
outdoor	2.253	**0.19**	14.702	2.963	25.463	0.383

Robustness: Referring to [53], we add Gaussian noise with different noise levels σ to the origin point cloud and regenerate the mesh. The APE w.r.t different noises are shown in Table 7. The experiments show that even with a standard deviation $\sigma = 1.0m$, the accuracy of our method is still better than the state-of-the-art VIO system. In addition, [53] fails in all three synthetic datasets when we add standard deviation $\sigma = 0.1m$ to the pre-built map.

Ablation Experiments: To intuitively evaluate the role of each module in the system, we performed some ablation experiments. In the ablation experiments, we set the localization latency to 400ms and the interval of sending localization requests to 1000ms. These parameters come from our real-world test. Moreover, we add the noise with $\sigma = 0.1$ to the pre-built map and add a zero-mean, white Gaussian noise with covariance 5 cm and 3 degrees to the localization pose.

Ablation Analysis: Comparing ①② in Table 6, we find that the number of point clouds from the pre-built map will significantly affect the coupled result. The reason is that if the number of points is not sufficient, the outlier removal may be difficult. Therefore, compared with generating point clouds using 2D-3D matches, the ray casting method, which can get more points for each frame, has more significant

Table 6: Evaluation on experiement of ablation with APE (m). *LP* stands for using real-time poses to obtain local map points. *GP* stands for using localization poses to obtain global map points. *M* stands for using the multi-frame check for each point. *S* stands for using struct check. *R* stands for randomly selected 25% point clouds obtained by ray casting. *LR* indicates that the local map points form reprojection constraints. ① is the result of baseline. ② is the ablation experiment on the number of points. ③④⑤ are ablation experiments on outlier removal. ⑥⑦⑧ are ablation experiments on map point constraints.

	stages						indoor	indoor partial	outdoor
	LP	GP	M	S	R	LR			
①	✓	✓	✓	✓	-	-	0.089	0.060	0.226
②	✓	✓	✓	✓	✓	-	0.307	0.135	0.456
③	✓	✓	-	✓	-	-	0.096	0.085	0.227
④	✓	✓	✓	-	-	-	0.098	0.078	0.239
⑤	✓	✓	-	-	-	-	0.102	0.082	0.253
⑥	-	✓	✓	✓	-	-	0.102	0.051	0.270
⑦	✓	-	✓	✓	-	-	0.162	0.094	1.680
⑧	✓	✓	✓	✓	-	✓	0.105	0.244	0.331

Table 7: Evaluation general localization performance on synthetic datasets with APE (m) on different maps. We add gaussian noise σ(m) to mesh vertices and reconstruct the faces of mesh with meshlab.

Dataset	σ:0	σ: 0.1	σ: 0.3	σ: 0.5	σ: 1.0
indoor	0.023	0.049	0.076	0.084	0.148
indoor patial	0.020	0.026	0.036	0.047	0.054
outdoor	0.190	0.212	0.215	0.217	0.258

advantages. Comparing ①⑥⑦ in Table 6, we can find that both local map points and global map points can improve the accuracy of the coupled algorithm, among which global map points are more helpful for improving the accuracy. By comparing ①⑥⑧, we can draw another important conclusion: using reprojection constraints for local map points will take a negative effect. The reason is that the local map points and real-time poses are interdependent and mutually affected. Whether there is noise in local map points or error accumulation in the real-time pose, the accuracy of the coupled methods will decrease under the influence of each other, which also explains the poor accuracy of [13] in Table 4. ①③④⑤ in Table 6 show that our outlier removal strategies are effective.

Latency and Interval Experiements: For our method, high network latency and low localization frequency will make global map constraints difficult to take effect, and local map constraints play a significant role. We verify the effect of local map constraints under different localization frequencies and different latency. Table 8 shows that even when the time delay is 1200ms, our method can still get better results than BVIO. Since *indoor partial* is a local-regional move dataset, the accuracy of *indoor partial* cannot truly reflect the role of each module. Ignoring *indoor partial*, we find that local map point constraints can effectively suppress drift accumulation when the interval and delay of localization requests gradually increase. For more experiments on the synthetic dataset, please refer to supplementary materials.

Table 8: Evaluation on general localization performanceon synthetic dataset with APE (m) under different time delay. The interval of sending localization request is set to 1000ms.

Delay (ms)	indoor		indoor partial		outdoor	
	w	w/o	w	w/o	w	w/o
200	**0.035**	0.037	0.037	**0.036**	**0.267**	0.300
400	**0.041**	0.049	0.041	**0.037**	**0.284**	0.327
800	**0.077**	0.088	0.049	**0.046**	0.378	**0.330**
1200	**0.121**	0.146	**0.047**	0.054	**0.458**	0.567

Table 9: Evaluation on ablation of local map point constraints with APE (m) under different localization frequencies. The latency of localization pose is set to 400ms.

Interval (ms)	indoor		indoor partial		outdoor	
	w	w/o	w	w/o	w	w/o
1000	**0.041**	0.049	0.041	**0.037**	**0.284**	0.327
2000	**0.060**	0.065	0.043	**0.042**	**0.319**	0.368
4000	**0.064**	0.068	**0.043**	0.052	**0.528**	0.541
8000	**0.076**	0.105	**0.052**	0.055	**0.528**	0.541
12000	**0.092**	0.129	0.061	**0.056**	**0.527**	0.588

7.3 AR Demo on a Mobile Phone

To better reflect the advantages of our method in practical applications, we have implemented an AR demo that can run in real-time on a mobile phone with the aid of visual localization on the remote server.

The AR demo runs on a HUAWEI MATE20 PRO. We reconstructed two open scenes, an indoor scene and an outdoor scene, and the data used for scene reconstruction were panoramic images collected with a camera five months before the actual test. Therefore, both the environment and the device have changed significantly, which is a big challenge for the localization algorithm and our coupled method. In addition, because the localization algorithm is deployed on the server, there is a network latency between 300 and 500 milliseconds per localization request. The localization interval is set to 1000ms. Due to the lack of ground truth, we can only verify drift accumulation through loop closure. For comparison, we also developed a loosely-coupled demo based on ARCore, which is called as ARCore-LC. The implemented details can be found in the supplementary document.

In our supplementary video, all demo sequences run in real-time. We saved real-time poses and recorded the screen. Fig. 1 is a post-processing composite of the saved poses. Fig. 1 shows that our method can smoothly suppress drift accumulation in real-time. The trajectory of the loosely-coupled algorithm shows several jumpiness, corresponding to several significant AR jumpiness. Please refer to the supplementary video for better viewing. Both RTC-VIO and ARCore-LC request the same localization algorithm on the same server, so there is no difference in localization accuracy.

In addition to comparing with ARCore-LC, we also compare with Hololens 2. Please refer to our supplementary materials.

8 Conclusion

We have developed a monocular VIO system tightly-coupled with pre-built maps in real-time on a mobile device. With the ray casting method, we reproject the sparse feature points to the dense model to obtain their 3D positions, which will be directly added to the feature tracking and state update of VIO to suppress the drift accumulation of the VIO system in real-time. In the fusion stage, we use VIO's local high precision characteristics to effectively remove outliers, which makes our system have good robustness to map noise. Our method can tolerate high location delay and low location frequency well and is especially suitable for deploying location service on the server. The experimental results show that our method achieves the highest accuracy and robustness compared with the state-of-the-art methods. Finally, a real-time AR demo on a mobile phone is presented to further verify the effectiveness of our method. In the future, we will investigate how to couple with more information such as NIN [7, 23] and GNSS to reduce the frequency of localization requests further.

Acknowledgments

The authors would like to thank Youji Feng, Liyang Zhou, Fei Jiao, Mingxuan Jiang, Chongshan Sheng, Yuequ Cai for their kind help in the development of the global localization service and the real-time AR demo. This work was partially supported by NSF of China (No. 61932003).

REFERENCES

[1] N. A. Ahmed and D. Gokhale. Entropy expressions and their estimators for multivariate distributions. *IEEE Transactions on Information Theory*, 35(3):688–692, 1989.

[2] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5297–5307, 2016.

[3] 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*, 35(10):1157–1163, 2016.

[4] M. Calonder, V. Lepetit, C. Strecha, and F. Brief. Binary robust independent elementary features. In *Proceedings of the European Conference on Computer Vision*, pp. 778–792, 2010.

[5] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós. Orb-slam3: An accurate open-source library for visual, visual-inertial and multi-map slam. *arXiv preprint arXiv:2007.11898*, 2020.

[6] T. Caselitz, B. Steder, M. Ruhnke, and W. Burgard. Monocular camera localization in 3D lidar maps. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1926–1931. IEEE, 2016.

[7] D. Chen, N. Wang, R. Xu, W. Xie, H. Bao, and G. Zhang. Rnin-vio: Robust neural inertial navigation aided visual-inertial odometry in challenging scenes. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 275–283. IEEE Computer Society, 2021.

[8] G. Cioffi and D. Scaramuzza. Tightly-coupled fusion of global positional measurements in optimization-based visual-inertial odometry. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5089–5095. IEEE, 2020.

[9] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 224–236, 2018.

[10] 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*, 24(6):381–395, 1981.

[11] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003.

[12] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang. Openvins: A research platform for visual-inertial estimation. In *Proc. of the IEEE International Conference on Robotics and Automation*. Paris, France, 2020.

[13] H. Huang, H. Ye, J. Jiao, Y. Sun, and M. Liu. Geometric structure aided visual inertial localization. *arXiv preprint arXiv:2011.04173*, 2020.

[14] H. Huang, H. Ye, Y. Sun, and M. Liu. Gmmloc: Structure consistent visual localization with gaussian mixture models. *IEEE Robotics and Automation Letters*, 5(4):5043–5050, 2020.

[15] Y. Kim, J. Jeong, and A. Kim. Stereo camera localization in 3D lidar maps. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–9. IEEE, 2018.

[16] M. Labbé and F. Michaud. Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, 36(2):416–446, 2019.

[17] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009.

[18] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International conference on computer vision*, pp. 2548–2555. Ieee, 2011.

[19] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[20] M. Li and A. I. Mourikis. High-precision, consistent ekf-based visual-inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013.

[21] T. Li, H. Zhang, Z. Gao, X. Niu, and N. El-Sheimy. Tight fusion of a monocular camera, mems-imu, and single-frequency multi-gnss rtk for precise navigation in gnss-challenged environments. *Remote Sensing*, 11(6):610, 2019.

[22] H. Liu, G. Zhang, and H. Bao. Robust keyframe-based monocular slam for augmented reality. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 1–10. IEEE, 2016.

[23] W. Liu, D. Caruso, E. Ilg, J. Dong, A. I. Mourikis, K. Daniilidis, V. Kumar, and J. Engel. Tlio: Tight learned inertial odometry. *IEEE Robotics and Automation Letters*, 5(4):5653–5660, 2020.

[24] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[25] Y. Lu, J. Lee, S.-H. Yeh, H.-M. Cheng, B. Chen, and D. Song. Sharing heterogeneous spatial knowledge: Map fusion between asynchronous monocular vision and lidar or other prior inputs. In *Robotics Research*, pp. 727–741. Springer, 2020.

[26] S. Lynen, T. Sattler, M. Bosse, J. A. Hesch, M. Pollefeys, and R. Siegwart. Get out of my lab: Large-scale, real-time visual-inertial localization. In *Robotics: Science and Systems*, vol. 1, p. 1, 2015.

[27] R. Mascaro, L. Teixeira, T. Hinzmann, R. Siegwart, and M. Chli. Gomsf: Graph-optimization based multi-sensor fusion for robust uav pose estimation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1421–1428. IEEE, 2018.

[28] P. S. Maybeck. *Stochastic models, estimation, and control*. Academic press, 1982.

[29] 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. IEEE, 2007.

[30] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

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

[32] L. Platinsky, M. Szabados, F. Hlasek, R. Hemsley, L. Del Pero, A. Pancik, B. Baum, H. Grimmett, and P. Ondruska. Collaborative augmented reality on smartphones via life-long city-scale maps. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 533–541. IEEE, 2020.

[33] T. Qin, S. Cao, J. Pan, and S. Shen. A general optimization-based framework for global pose estimation with multiple sensors. *arXiv preprint arXiv:1901.03642*, 2019.

[34] T. Qin, P. Li, and S. Shen. Relocalization, global optimization and map merging for monocular visual-inertial slam. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1197–1204. IEEE, 2018.

[35] T. Qin, P. Li, and S. Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[36] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, vol. 1, pp. 430–443, May 2006. doi: 10.1007/11744023_34

[37] S. D. Roth. Ray casting for modeling solids. *Computer graphics and image processing*, 18(2):109–144, 1982.

[38] 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. Ieee, 2011.

[39] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling*, pp. 145–152. IEEE, 2001.

[40] M. R. U. Saputra, A. Markham, and N. Trigoni. Visual slam and structure from motion in dynamic environments: A survey. *ACM Computing Surveys (CSUR)*, 51(2):1–36, 2018.

[41] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12716–12725, 2019.

[42] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4938–4947, 2020.

[43] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1744–1756, 2016.

[44] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, et al. Benchmarking 6DoF outdoor visual localization in changing conditions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8601–8610, 2018.

[45] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited.

In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.

[46] J. L. Schönberger, T. Price, T. Sattler, J.-M. Frahm, and M. Pollefeys. A vote-and-verify strategy for fast spatial verification in image retrieval. In *Asian Conference on Computer Vision*, pp. 321–337. Springer, 2016.

[47] S. Shah, D. Dey, C. Lovett, and A. Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.

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

[49] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson. City-scale localization for cameras with known vertical direction. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1455–1461, 2016.

[50] W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao. Robust monocular SLAM in dynamic environments. In *IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2013, Adelaide, Australia, October 1-4, 2013*, pp. 209–218. IEEE Computer Society, 2013.

[51] 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. In *Robotics: Science and Systems*, vol. 2, 2015.

[52] M. Yamaguchi, S. Mori, H. Saito, S. Yachida, and T. Shibata. Global-map-registered local visual odometry using on-the-fly pose graph updates. In *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, pp. 299–311. Springer, 2020.

[53] H. Ye, H. Huang, and M. Liu. Monocular direct sparse localization in a prior 3D surfel map. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8892–8898. IEEE, 2020.

[54] X. Zuo, P. Geneva, Y. Yang, W. Ye, Y. Liu, and G. Huang. Visual-inertial localization with prior lidar map constraints. *IEEE Robotics and Automation Letters*, 4(4):3394–3401, 2019.