Adaptive VIO: Deep Visual-Inertial Odometry with Online Continual Learning

Youqi Pan   Wugen Zhou   Yingdian Cao   Hongbin Zha

National Key Lab of GAI,   Institute for AI,   School of IST

PKU-SenseTime Joint Lab of MV

Peking University

panyouqi@stu.pku.edu.cn   {zhouwugen, yingdianc}@pku.edu.cn   zha@cis.pku.edu.cn

Abstract

*Visual-inertial odometry (VIO) has demonstrated remarkable success due to its low-cost and complementary sensors. However, existing VIO methods lack the generalization ability to adjust to different environments and sensor attributes. In this paper, we propose **Adaptive VIO**, a new monocular visual-inertial odometry that combines online continual learning with traditional nonlinear optimization. Adaptive VIO comprises two networks to predict visual correspondence and IMU bias. Unlike end-to-end approaches that use networks to fuse the features from two modalities (camera and IMU) and predict poses directly, we combine neural networks with visual-inertial bundle adjustment in our VIO system. The optimized estimates will be fed back to the visual and IMU bias networks, refining the networks in a self-supervised manner. Such a learning-optimization-combined framework and feedback mechanism enable the system to perform online continual learning. Experiments demonstrate that our Adaptive VIO manifests adaptive capability on EuRoC and TUM-VI datasets. The overall performance exceeds the currently known learning-based VIO methods and is comparable to the state-of-the-art optimization-based methods.*

1. Introduction

Obtaining reliable motion estimation in unknown environments is critical to many vision and robotics tasks, such as augmented reality (AR), unmanned aerial vehicle (UAV), and autonomous driving. Simultaneous localization and mapping (SLAM) is one of the critical approaches that employs onboard sensors to estimate agent trajectory and build a map of local environments. Researchers have extensively investigated visual-inertial SLAM (VI-SLAM) and visual-inertial odometry (VIO) due to their low-cost and complementary sensors. It often presents a more accurate and robust trajectory estimation than visual odometry (VO) or inertial odometry (IO).
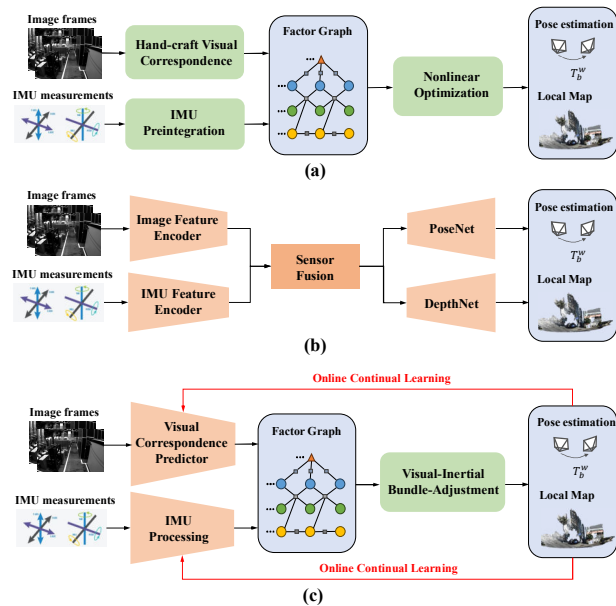

Figure 1. Frameworks of different VIO methods. Learning-based modules are colored in orange. Traditional computational modules are colored in green. **(a)** Classic optimization-based method. **(b)** End-to-end learning-based method. **(c)** Learning-optimization-combined method with online continual learning (ours).

A classic VIO system is mainly composed of visual association, IMU preintegration, and back-end nonlinear optimization [16, 24, 30, 37, 42] (or filtering [16, 17]), as shown in Fig.1(a). Classic methods are featured in the systematic framework and fine-grained pipeline, working well in favorable conditions. However, they are less accurate and may even fail in challenging scenarios (*e.g.* low-light condition, abrupt movement), which can be attributed to the reliance on low-level and hand-crafted visual features. In addition, trajectory drift caused by IMU bias is also one of the critical reasons affecting the system's performance, while traditionally modeling IMU bias as a random walk is insufficient to reflect its evolutionary patterns.

In order to overcome the shortcomings of classic methods that rely on pre-defined features, many learning-based approaches have been proposed. End-to-end learning-based VO can extract features from image streams and directly generate pose and depth estimation without explicit optimization, which has shown promising results in recent years. Similarly, some end-to-end learning-based VIO methods are developed. As shown in Fig.1(b), these methods typically use two separate networks to extract image and IMU features, fuse them through a fusion network, and subsequently get pose and depth estimations from networks. Nonetheless, these methods suffer from poor generalization in complex motion scenarios, with lower accuracy than classic methods.

In this paper, we propose a novel VIO system named Adaptive VIO. As shown in Fig.1(c), unlike classic methods or end-to-end learning-based VIO, our approach integrates learning with classic computations. We leverage the strengths of neural networks in predicting visual correspondence and IMU bias, replacing the traditional optical flow or hand-craft feature matching and random walk modeling for IMU bias. On the other hand, the refined estimates obtained through classic optimization serve as feedback information, generating loss functions for the predictor network, thus enabling self-supervised learning. Finally, thanks to the learning-optimization-combined framework and feedback mechanism, we can conduct online continual learning, enabling the VIO system to adapt across different environments and achieve better tracking performance.

Our main contributions can be summarized as follows:

- We propose a novel learning-optimization-combined VIO, which predicts visual correspondence and IMU bias through learning approaches and obtains accurate state estimation through classic nonlinear optimization.
- We introduce a feedback mechanism for the system, utilizing the estimation from nonlinear optimization to construct loss functions, updating the networks in a self-supervised manner.
- We develop online continual learning to refine the networks in different scenarios. Experiments demonstrate the strong generalization and adaptive capabilities of our method, yielding overall results comparable to state-of-the-art VIO systems.

2. Related Work

2.1. Classic VIO

In the past decade, VIO has been an active topic in the field of SLAM. Due to the complementary sensors, VIO exhibits enhanced robustness compared to VO across various scenarios and makes scale observable in monocular setups.

Classic VIO systems are mainly based on tightly coupled approaches, wherein visual and IMU constraints are fused through filtering or nonlinear optimization. Filter-based methods, such as MSCKF [17, 22], and ROVIO [2], use the extended Kalman filter (EKF) to propagate and update the current state. While nonlinear-optimization-based methods, like VINS [24, 25], ORB-SLAM3 [6] and DM-VIO [30], adopt local visual-inertial bundle adjustment for the state estimation, achieving more accurate pose tracking.

Classic VIO uses optical flow or hand-craft features for visual association, constructing motion constraints based on photometric or reprojection errors. As for IMU processing, modeling IMU bias as the random walk is a common practice [11]. Classic VIO methods have gained widespread application, but the manually defined data association and IMU bias modeling are often inaccurate in challenging scenarios, leading to suboptimal results or even failure.

2.2. Learning-based VO and VIO

In recent years, learning-based methods in VO, IO, and VIO have been extensively researched[1, 4, 45], yielding promising results.

End-to-end learning-based VO utilizes pose and depth estimation networks to replace classic modules of tracking and mapping, trained in either a supervised [40, 43] or self-supervised [18, 45] manner. Some end-to-end learning-based VIO approaches have also been proposed [9, 14, 28]. For instance, SelfVIO [1] leverages networks to encode and adaptively fuse visual and IMU information, estimating depth and pose by self-supervised learning as VO.

Despite promising results on some datasets, end-to-end learning-based VO and VIO exhibit lower accuracy than classic approaches, particularly struggling to get accurate pose estimates under complex motions. Therefore, some propose to combine learning techniques with traditional modules [3, 7, 32]. DROID-SLAM [33] and DPVO [35] combine iterative visual correspondence updates with differentiable bundle adjustment, demonstrating excellent performance across multiple datasets. iSLAM [12] combines learning-based VO with graph optimization, further enhancing performance through self-supervised learning. From the IMU perspective, learning-based data preprocessing and bias estimation techniques are also proposed [4, 7, 21, 44]. Zhang et al. [44] learn to denoise IMU measurements and use preintegration loss derived from ground truth poses to train the network. Buchanan et al. [4] explicitly propose models for the dynamics of bias by networks and incorporate them into factor graph optimization, replacing traditional random walk models.

These methods combining learning with classic computations provide us with significant inspiration. To the best of our knowledge, our approach is the first to simultaneously integrate learning-based visual association and IMU bias modeling into a VIO framework and refine them by self-supervised learning.
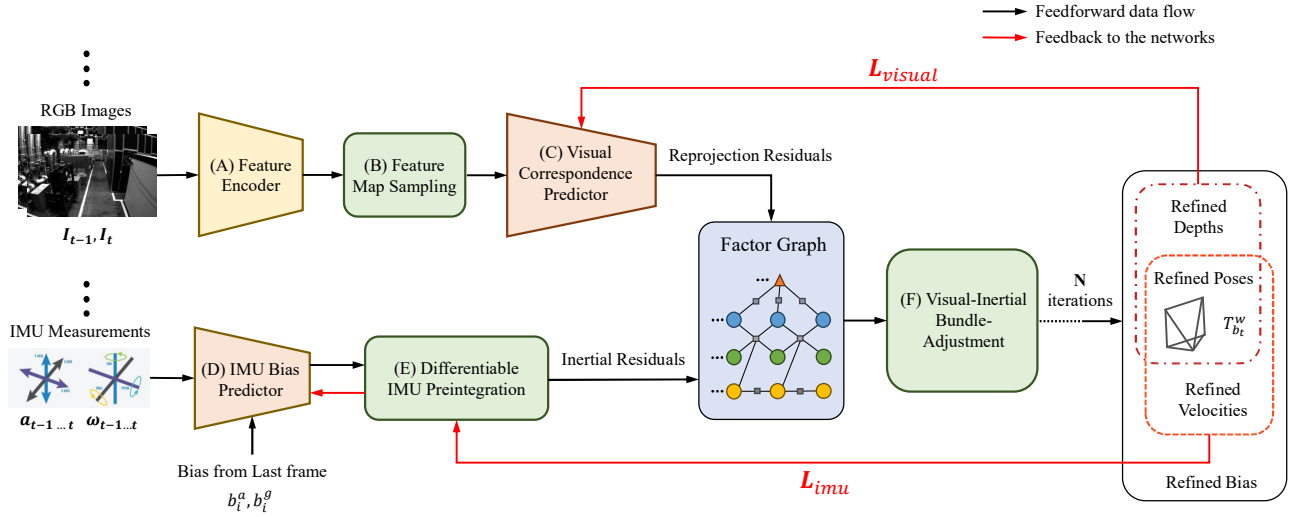
Figure 2. The tracking pipeline of our VIO. The green modules (B, E, F) denote manually designed algorithms. The yellow (A) and orange (C, D) trapezoids represent modules implemented by neural networks, and orange modules can get online continual learning.

2.3. Online continual learning

Generally speaking, classic VO/VIO algorithms are manually pre-defined, and learning-based ones are usually pre-trained on dedicated datasets and then inference on scenes with similar distribution. These methods may suffer from domain shift problems if directly applied to a different scenario, leading to inferior performance.

To address this issue, Li et al. [19, 20] propose an online-learning VO framework to generalize better on unseen environments and use meta-learning techniques to facilitate fast adaptation. Vödisch et al. [36, 38] introduce online continual learning for SLAM, incorporating a replay buffer to prevent catastrophic forgetting and using asynchronous network updates to optimize system efficiency.

While these online learning methods have shown adaptability in autonomous driving scenarios such as Cityscape[10] and KITTI [13], they often fail in complex environments and under complex motions. In our method, the states from optimization provide feedback signals to the networks, enabling self-supervised online continual learning, which has proved effective in EuRoC [5] and VI-TUM datasets [27].

3. Method

In this section, we introduce our monocular VIO method in detail. We start with the unique framework design, which combines classic optimization with deep learning techniques (Sec.3.1). Next, we present the feedback mechanism and self-supervised updates for the networks (Sec.3.2). Finally, we delve into more details of the VIO system, focusing on online continual learning, which is pivotal for the

system to adapt to diverse scenarios and achieve improved performance gradually (Sec.3.3).

3.1. Learning-optimization-combined framework

The tracking pipeline of our VIO system is illustrated in Fig.2, featured by the framework that integrates learning and classic optimization. The orange shapes in the diagram represent computations performed by the neural network, while the blue shapes indicate traditional manual calculations.

(A) Feature encoder receives the latest RGB image frame and encodes it into feature maps by convolutional neural networks, providing stable feature encoding for subsequent visual correspondence.

(B) Feature map sampling module randomly selects several feature points and their neighborhood from the feature map, generating feature patches. These patches are considered as keypoints, facilitating subsequent matching and depth and pose estimation. The module can be differentiable [35], which made the gradient backpropagation possible.

(C) Visual correspondence predictor predicts visual matching relationship among the keyframes in the factor graph. It takes initial matches generated by reprojection as input and outputs the updates relative to the initial matching, which can also be viewed as reprojection residuals. The pose and depth estimation are generated from IMU state propagation or visual-inertial bundle adjustment, which will be detailed in the following content.

(D) IMU bias predictor takes IMU bias estimation from the previous time step, along with the accelerometer and gyroscope measurements between previous and current image

frames, as inputs to predict the IMU bias $\hat{\mathbf{b}}_a$, $\hat{\mathbf{b}}_g$ for the current time step. The IMU timestamps and bias configuration follow the settings in [6, 24]. Specifically, we synchronize the IMU preintegration interval with the timestamps of the image frames, assuming that the bias remains fixed within the interval.

(E) Differentiable IMU preintegration integrate IMU data in body poses of the previous image frame, which is independent of the initial conditions and can be treated as a single observation between two adjacent image frames. Assuming the timestamps of ith and $i+1^{th}$ IMU frames are between the timestamps of kth and $k+1^{th}$ image frames. Let $\hat{\boldsymbol{\alpha}}_i^{b_k}$, $\hat{\boldsymbol{\beta}}_i^{b_k}$, $\hat{\boldsymbol{\gamma}}_i^{b_k}$ represent the preintegration of translation, velocity and rotation until the ith IMU frame, where b_k denotes the preintegration is under the body coordinate at the kth image frame. The body coordinate aligns with the IMU coordinate. Then, the preintegration until $i+1^{th}$ IMU frame can be expressed in the following form[24]:

$$\hat{\boldsymbol{\alpha}}_{i+1}^{b_k} = \hat{\boldsymbol{\alpha}}_i^{b_k} + \hat{\boldsymbol{\beta}}_i^{b_k}\delta t + \frac{1}{2}\mathbf{R}(\hat{\boldsymbol{\gamma}}_i^{b_k})\hat{\mathbf{a}}_i\delta t^2 \qquad (1)$$

$$\hat{\boldsymbol{\beta}}_{i+1}^{b_k} = \hat{\boldsymbol{\beta}}_i^{b_k} + \mathbf{R}(\hat{\boldsymbol{\gamma}}_i^{b_k})\hat{\mathbf{a}}_i\delta t \qquad (2)$$

$$\hat{\boldsymbol{\gamma}}_{i+1}^{b_k} = \hat{\boldsymbol{\gamma}}_i^{b_k} \otimes \begin{bmatrix} \frac{1}{2}\hat{\boldsymbol{\omega}}_i\delta t \\ 1 \end{bmatrix} \qquad (3)$$

where $\hat{\mathbf{a}}_i$, $\hat{\boldsymbol{\omega}}_i$ are acceleration and angular velocity measurements of ith IMU frame, after subtracting the network-predicted bias $\hat{\mathbf{b}}_a$, $\hat{\mathbf{b}}_g$, respectively. $\otimes$ represents the product in quaternion.

The propagation from i^{th} to $i+1^{th}$ IMU frame is differentiable with respect to $\hat{\mathbf{a}}_i$ and $\hat{\boldsymbol{\omega}}_i$. Henceforth, the preintegration from the kth to $k+1^{th}$ image frame is also differentiable, which allows the gradient backpropagate from preintegration to the IMU bias predictor.

The propagation of covariance and approximate bias update techniques are also adopted in our system, which are detailed in [11]. The differentiation and backpropagation of Lie Group can refer to [29, 34, 39].

(F) Visual-inertial bundle adjustment (VIBA) are standard techniques for solving state variables in classic VIO systems, which we also adopt with some distinctions.

Our factor graph comprises states within a sliding window of n image frames, incorporating visual constraints, IMU constraints, and their interrelations. The full state vector in the factor graph is:

$$\mathcal{X} = \{\mathbf{p}_b^w, \mathbf{q}_b^w, \mathbf{v}_b^w, \mathbf{b}_a, \mathbf{b}_g, \mathbf{d}_{0..m}\}_{0..n} \qquad (4)$$

where $\mathbf{p}_b^w$, $\mathbf{q}_b^w$ and $\mathbf{v}_b^w$ denote translation, quaternion and velocity of the body. Bolded $\mathbf{b}_a$ and $\mathbf{b}_g$ are biases of the accelerator and gyroscope. $\mathbf{d}_{0..m}$ represents the depth of m features from the frame.

The visual constraints in the factor graph are reprojection residuals, which reflect the coordinate error between the matches predicted by the network and the matches projected under current states. We construct reprojection residuals for each feature, and the network provides the confidence of each residual.

Reprojection residuals:

$$\mathbf{r}_v = \sum_{i,j \in \mathcal{G}} \|\hat{p}_j^t - KT_s^t \mathbf{d}(p_i^s)K^{-1}p_i^s\| \qquad (5)$$

where p_i^s, $\hat{p}_j^t$ are a pair of matching points predicted by the network. T_s^t denotes the coordinate transformation between the frames of the matching points in the image. K is intrinsic to the camera. All such matching points constitute visual constraints in the factor graph $\mathcal{G}$.

IMU constraints can be categorized into two types: preintegration residuals and bias residuals. For two consecutive image frames I_k and I_{k+1}, the following expressions are given:

$$\mathbf{r}_{\mathcal{I}_{k+1}^k} = \{\mathbf{r}_{\Delta \mathrm{p}_{k+1}^k}, \mathbf{r}_{\Delta \mathrm{v}_{k+1}^k}, \mathbf{r}_{\Delta \mathrm{R}_{k+1}^k}, \mathbf{r}_{\Delta (\mathrm{b}_a)_{k+1}^k}, \mathbf{r}_{\Delta (\mathrm{b}_g)_{k+1}^k}\} \qquad (6)$$

Preintegration residuals:

$$\mathbf{r}_{\Delta \mathrm{p}_{k+1}^k} = \mathrm{R}_w^{b_k}\Big(\mathbf{p}_{b_{k+1}}^w - \mathbf{p}_{b_k}^w - \mathbf{v}_{b_k}^w \Delta t_k \\ + \frac{1}{2}g^w \Delta t_k^2\Big) - \hat{\boldsymbol{\alpha}}_{b_{k+1}}^{b_k}(\mathbf{b}_a, \mathbf{b}_g) \qquad (7)$$

$$\mathbf{r}_{\Delta \mathrm{v}_{k+1}^k} = \mathrm{R}_w^{b_k}\Big(\mathbf{v}_{b_{k+1}}^w - \mathbf{v}_{b_k}^w + g^w \Delta t_k\Big) \\ - \hat{\boldsymbol{\beta}}_{b_{k+1}}^{b_k}(\mathbf{b}_a, \mathbf{b}_g) \qquad (8)$$

$$\mathbf{r}_{\Delta \mathrm{R}_{k+1}^k} = \mathrm{Log}\Big(\mathbf{R}(\hat{\boldsymbol{\gamma}}_{b_{k+1}}^{b_k}(\mathbf{b}_g))^{\mathrm{T}}\mathrm{R}_w^{b_k}\mathrm{R}_{b_{k+1}}^w\Big) \qquad (9)$$

where $\hat{\boldsymbol{\alpha}}_{b_{k+1}}^{b_k}, \hat{\boldsymbol{\beta}}_{b_{k+1}}^{b_k}, \hat{\boldsymbol{\gamma}}_{b_{k+1}}^{b_k}$ denotes preintegration terms and g^w is gravity under the world coordinate.

Bias residuals:

$$\mathbf{r}_{\Delta (\mathrm{b}_a)_{k+1}^k} = (\mathbf{b}_a)^k - (\hat{\mathbf{b}}_a)^k \qquad (10)$$

$$\mathbf{r}_{\Delta (\mathrm{b}_g)_{k+1}^k} = (\mathbf{b}_g)^k - (\hat{\mathbf{b}}_g)^k \qquad (11)$$

where $\hat{\mathbf{b}}_a$, $\hat{\mathbf{b}}_g$ are the bias predicted by the network.

We model accelerator and gyroscope bias as a Gaussian distribution with the network-predicted bias value as the mean, and the variance is manually set. Similar methods are also elaborated in [4]. In addition to the differences in bias residuals compared to traditional random walk methods, there are other distinctions in treating bias. Firstly, we only conduct preintegration by the network-predicted bias $(\hat{\mathbf{b}}_a, \hat{\mathbf{b}}_g)$. We never reintegrate measurements by the updated bias $(\mathbf{b}_a, \mathbf{b}_g)$ after optimization. Second, the updated bias from the current frame serves as inputs to the predictor, along with measurements from the next frame, generating a

new bias prediction instead of being directly passed to the next time step.

The final optimization objective is composed of IMU residuals and visual residuals, which can be written as:

$$\mathcal{X}^* = \arg\min_{\mathcal{X}} \Big(\|\mathbf{r}_{\mathrm{v}}\|^2_{\Sigma_p} + \sum_{k=0}^{n} \|\mathbf{r}_{\mathcal{I}_{k+1}^k}\|^2_{\Sigma_{\mathcal{I}_{k+1}^k}} \Big) \quad (12)$$

Considering the Gauss-Newton method is naturally differentiable, we use it to solve the factor graph, iterating twice for each timestamp. This facilitates gradient backpropagation to the neural network during training and online adaptation.

3.2. Feedback and self-supervised update

In our approach, factor graph optimization can provide feedback signals for visual and IMU networks, enabling self-supervised learning updates.

The feature encoder and visual correspondence predictor adopt similar network structures to DPVO [35]. The difference is that our predictor does not compose a hidden state, making the inference of the correspondences in each iteration an independent process irrelevant to the previous network state. The modifications make our pipeline more akin to the classic SLAM, enhancing system interpretability.

The poses and depths optimized by VIBA can be used to construct photometric loss through reprojection:

$$\mathbf{L}_{visual} = \sum_{i \in \mathcal{G}} \|I^t(p_i^*) - I^s(p_i)\| \quad (13)$$

$$p_i^* = K T_s^t \mathbf{d}(p_i^s) K^{-1} p_i^s \quad (14)$$

where I represents the intensity of the pixel.

The bias prediction network consists of normalization layers, fully connected layers, and a GRU [8] module. The bias updated from the previous timestamp is initially normalized and then encoded as hidden states for the GRU. Simultaneously, the IMU measurements are normalized and concatenated with the normalized bias. After encoding by a fully connected layer, they serve as inputs to the GRU. The GRU's output is then transformed into the current bias estimation via another fully connected layer.

After visual-inertial bundle adjustment, the refined poses and velocities are feedback to the network, generating loss function as:

$$\mathbf{L}_{imu} = \mathbf{L}_{\Delta\mathrm{p}} + \mathbf{L}_{\Delta\mathrm{v}} + \mathbf{L}_{\Delta\mathrm{R}} \quad (15)$$

$$\mathbf{L}_{\Delta\mathrm{p}} = \mathrm{R}_w^{b_{n-1}} \Big(\mathbf{p}_{b_n}^w - \mathbf{p}_{b_{n-1}}^w - \mathbf{v}_{b_{n-1}}^w \Delta t + \frac{1}{2} g^w \Delta t^2 \Big) - \hat{\boldsymbol{\alpha}}_{b_n}^{b_{n-1}}(\hat{\mathbf{b}}_a, \hat{\mathbf{b}}_g) \quad (16)$$

$$\mathbf{L}_{\Delta\mathrm{v}} = \mathrm{R}_w^{b_{n-1}} \Big(\mathbf{v}_{b_n}^w - \mathbf{v}_{b_{n-1}}^w + g^w \Delta t \Big) - \hat{\boldsymbol{\beta}}_{b_n}^{b_{n-1}}(\hat{\mathbf{b}}_a, \hat{\mathbf{b}}_g) \quad (17)$$

$$\mathbf{L}_{\Delta\mathrm{R}} = \mathrm{Log}\Big(\mathbf{R}(\hat{\boldsymbol{\gamma}}_{b_n}^{b_{n-1}}(\hat{\mathbf{b}}_g))^{\mathrm{T}} \mathrm{R}_w^{b_{n-1}} \mathrm{R}_{b_n}^w \Big) \quad (18)$$

The IMU loss is almost the same with preintegration residuals Eq.(7)-(9), while the constrained entities shifted from the system's states to the parameters of the networks. Besides, there are some other slight differences to be noticed. 1) The loss function constrains the bias predicted by the network rather than the bias updated by VIBA. 2) In the feedback loop, we manually set the weights for each loss function (all set to 1 here), while in VIBA, the weights for each residual are determined by the covariance [11]. 3) Bias residuals Eq.(10)-(11) are not included in the loss function.

The feedback-based self-supervised update mechanism enhances the consistency between the learning and classic optimization modules, collectively improving the overall performance.

3.3. Online continual learning and VIO system

Performance degradation is often encountered when a learning algorithm is transferred to an unseen environment. In such cases, adaptive fine-tuning of the network is a common practice. However, in our scenario, network fine-tuning often requires customized data preprocessing and training strategy, posing an additional workload and delaying the deployment of the VIO system.

To reconcile the contradiction between VIO deployment and network adaptation, we propose a "learning within VIO" strategy known as online continual learning. In this mechanism, the VIO system can be considered an automatic dataloader, responsible for organizing and optimizing training data and feeding it to the neural network. In addition to being able to run continuously alongside VIO, our online continual learning mechanism is also highly flexible. 1) We can start or stop it at any time without interrupting the execution of VIO. 2) We can train the visual correspondence or the IMU bias predictor independently or simultaneously.

The VIO system can be summarized as follows:

Initialization: Our VIO system starts with an initialization process, which includes map initialization and IMU initialization. After IMU initialization, we align the body coordinate and recover the metric scale of the pose and the map. The initialization process provides the system with a good initial state and builds the platform for online learning.

Tracking: The pipeline of tracking are mainly shown in Fig.2. We compute the initial pose of the incoming frame by IMU state propagation and add it to the factor graph. The factor graph maintains a sliding window containing states of the latest 10 keyframes. For efficiency considerations, during online continual learning, the visual constraints for

Methods	Online adaptation to EuRoC [5]					Online adaptation to TUM-VI [27]					
	MH1	MH2	MH3	MH4	MH5	room1	room2	room3	room4	room5	room6
DPVO [35]	0.087	0.055	0.158	0.137	0.114	0.251	0.503	0.261	0.085	0.197	0.059
$\mathbf{V}_{PT} + \mathbf{I}_{RW}$	0.093	0.070	0.154	0.141	0.137	0.271	0.487	0.258	0.073	0.245	0.068
$\mathbf{V}_{AD} + \mathbf{I}_{RW}$	0.043	0.042	0.108	0.107	0.095	0.143	0.369	0.237	0.063	0.165	0.053
	54% ↑	40% ↑	30% ↑	24% ↑	31% ↑	47% ↑	24% ↑	8% ↑	14% ↑	33% ↑	22% ↑
$\mathbf{V}_{PT} + \mathbf{I}_{AD}$	0.085	0.065	0.150	0.130	0.136	0.249	0.460	0.251	0.067	0.242	0.059
	9% ↑	7% ↑	3% ↑	8% ↑	1% ↑	8% ↑	6% ↑	3% ↑	8% ↑	1% ↑	13% ↑
$\mathbf{V}_{AD} + \mathbf{I}_{AD}$	0.041	0.039	0.109	0.100	0.094	0.138	0.351	0.231	0.065	0.158	0.052
	56% ↑	44% ↑	29% ↑	29% ↑	31% ↑	49% ↑	28% ↑	10% ↑	11% ↑	36% ↑	24% ↑

Table 1. Evaluation of online adaptation ability of the proposed modules to new environments, reported with metric-scaled RMSE ATE. In this table, PT, RW denotes the pre-trained visual model and IMU bias model of random walk, and AD is the abbreviation of adaptation. We statistic the improvements of each adaptation module than our baseline model $\mathbf{V}_{PT} + \mathbf{I}_{RW}$ in percentage.

each keyframe are only derived from several keyframes preceding and following it. Otherwise, the visual constraints of each keyframe may encompass several keyframes with optical flow magnitudes below a threshold, forming a co-visibility relationship.

Feedback: After tracking, we utilize the feedback-based self-supervised updates described in Sec.3.2 to implement online continual learning for the networks. During online adaptation, we fix the weights of the feature encoder and fine-tune the predictors for both visual and IMU. All inputs and constraints for the networks come from the factor graph, including the images, IMU measurements, and state estimates, generating loss functions as Eq.(13-18)

Keyframing: After each tracking session, keyframe culling is performed. Our keyframing strategy is similar to DPVO [35]. However, considering the temporal constraints of IMU, the gap between two keyframes should not exceed 3 frames.

4. Experiments

4.1. Implementation details

Our method is implemented by Python and PyTorch [23], with specific components like VIBA utilizing C++ and CUDA programming for acceleration. The visual network requires pre-training, while the IMU bias network does not. In the following content, we will present the methods of pre-training and online continual learning separately, primarily focusing on the latter.

Datasets: The TartanAir[41] dataset is adopted to pre-train our visual model, which is a large-scale simulated dataset widely used in visual learning tasks. We choose the EuRoC [5] and TUM-VI [27] datasets for online continual learning and validation. The EuRoC is captured by a UAV visual-inertial platform, while the TUM-VI is collected by a handheld visual-inertial device. Both datasets include environments with complex lighting conditions and intricate motion patterns, making them widely used in VIO. To give the quantified results, we align the estimated trajectories with the provided ground-truth and compute the Root Mean Square Error (RMSE) of the Absolute Trajectory Error (ATE) [31].

Pre-training settings: The visual part of our network (module (A)and(C) in Fig.2) requires pre-training. Following the strategy in DPVO [35], we train our visual model on the TartanAir dataset [41] for 240000 iterations with a batch size of 1. Note that due to minor modifications in our networks compared to DPVO [35](discussed in Sec.3.2), the performance after pre-training may not be identical.

Online continual learning settings: After pre-training the visual network, we perform online continual learning on the EuRoC and TUM-VI datasets. Our PC configuration includes an Intel i9-9900 CPU and an NVIDIA GTX 3090 GPU with 24GB of VRAM.

During online continual learning, our VIO performs tracking and carries out gradient backpropagation for each incoming frame. This process constitutes one iteration of training. For tracking stability, we may not update the networks at every iteration. Completing an entire sequence is considered as one epoch, and online continual learning requires training multiple epochs, involving the continuous replay of one sequence.

For the EuRoC dataset, we perform online continual learning in MH_01. Similarly, we conduct continual learning in room1 of the TUM-VI dataset. In both settings, we replay the sequences for 60 epochs. The visual predictor's network weights update every 100 iterations, while the IMU bias predictor's weights update with each iteration. The learning rate for the visual predictor is set to be 1×10^{-5}. For the IMU bias predictor, since it is not pre-trained, we conduct visual BA for the first 30 epochs to ensure tracking stability, with the learning rate of 1×10^{-4}. After that, we

Sequence	Sensor	MH1	MH2	MH3	MH4	MH5	V11	V12	V13	V21	V22	Avg.
MSCKF[17]	M+I	0.42	0.45	0.23	0.37	0.48	0.34	0.20	0.67	0.10	0.16	0.34
OKVIS[16]	M+I	0.33	0.37	0.25	0.27	0.39	0.094	0.14	0.21	0.09	0.17	0.23
ROVIO[2]	M+I	0.21	0.25	0.25	0.49	0.52	0.10	0.10	0.14	0.12	0.14	0.23
VINS-Mono[24]	M+I	0.15	0.15	0.22	0.32	0.30	0.079	0.11	0.18	0.08	0.10	0.17
Kimera[26]	S+I	0.11	0.10	0.16	0.24	0.35	0.05	0.08	0.07	0.08	0.10	0.13
Online VIO[15]	M+I	0.14	0.13	0.20	0.22	0.20	0.05	0.07	0.16	0.04	0.11	0.13
VI-DSO[37]	M+I	0.062	**0.044**	0.117	0.132	0.121	0.059	0.067	0.096	0.040	0.062	0.08
DM-VIO[30]	M+I	0.065	**0.044**	0.097	0.102	**0.096**	0.048	**0.045**	**0.069**	**0.029**	**0.050**	**0.06**
iSLAM[12]	S+I	0.500	0.391	0.656	1.285	1.088	0.521	0.405	0.397	0.421	0.580	0.58
SelfVIO[1]	M+I	0.19	0.15	0.21	0.16	0.29	0.08	0.09	0.1	0.11	0.08	0.15
Ours	M+I	**0.050**	0.055	**0.069**	**0.092**	0.124	**0.035**	**0.045**	0.073	0.052	0.086	0.07

Table 2. Evaluation of visual-inertial odometry systems on EuRoC dataset, with RMSE ATE (m), SE(3)-aligned. The upper list is all classic VIO methods; the bottom is learning-based systems. The letters "M", "S" and "I" denote monocular, stereo, and IMU.

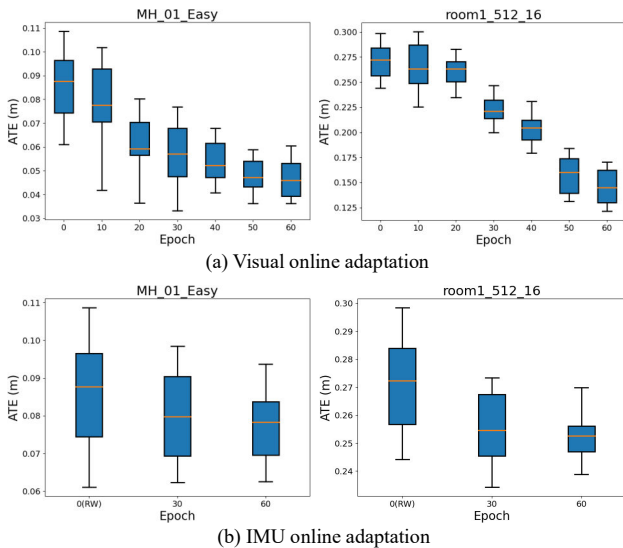(a) Visual online adaptation

(b) IMU online adaptation

Figure 3. RMSE ATE changes in MH_01 (EuRoC) and room1 (TUM-VI) during online continual learning of visual and IMU.

perform VIBA with the learning rate of 1×10^{-6}.

4.2. Evaluation of online continual learning

To evaluate the effectiveness of online continual learning for visual and IMU bias networks, we take the pre-trained visual model and IMU bias random walk as our baseline model and then compare it with visual adaptation, IMU adaptation, and joint adaptation. The results are summarized in Tab.1. To eliminate the influence of the metric scale, all results are Sim(3) aligned and averaged over five trials. Additionally, since the visual networks of our method are mainly borrowed from DPVO [35], it's reasonable to present their results for reference, as shown in the first row of Tab.1, where the results are Sim(3) aligned and are the

median of five trials.

Compared to the baseline, online continual learning for either visual or IMU networks improves trajectory accuracy, and the joint adaptation achieves the best performance, resulting in over 10% improvements across all sequences.

To further validate the continual adaptability of our method, we perform additional statistical tests during online continual learning. We conduct 10 validation experiments for the visual model after every 10 epochs to assess its trajectory accuracy and distribution. The overall statistical results are illustrated in Fig.3(a). We can observe that the ATE distribution of trajectories continually decreases during adaptation. For the IMU model, although the continual learning of IMU bias contributes less significantly to the improvement of trajectory accuracy compared to the visual model, we statistically compare its errors with those of the random walk model in 10 experiments after every 30 epochs. As shown in Fig.3(b), the IMU bias model presents lower trajectory drift and variance than bias random walk, thus leading to more robust performance.

Two examples of each dataset are selected for trajectory comparison of our model, with and without online continual learning. As shown in Fig. 4, the trajectories of our online adaptation model are closer to the ground-truth than that of our pre-trained model on four sequences, as clearly indicated by the red boxes.

4.3. Evaluation of overall performance

To evaluate the overall performance of our VIO system, we compare our method with state-of-the-art VIO approaches on the EuRoC [5] and TUM-VI [27] datasets. Our system constructs a more extensive set of keyframe association constraints containing temporally adjacent keyframes and spatially neighboring keyframes, as described in Sec. 3.3. We perform online continual learning on MH_01 of the EuRoC
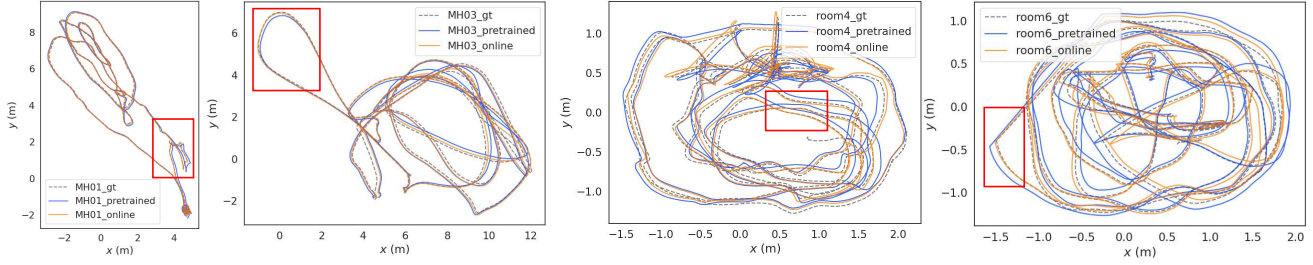
Figure 4. Estimated trajectories comparison on EuRoC (two sub-figures on the left) and TUM-VI dataset (two sub-figures on the right).

Seq.	ROVIO stereo	VINS mono	OKVIS stereo	DM-VIO mono	**Ours** mono
corr.1	0.47	0.63	0.33	0.19	**0.14**
corr.2	0.75	0.95	0.47	0.47	**0.28**
corr.3	0.85	1.56	0.57	**0.24**	0.50
corr.4	**0.13**	0.25	0.26	**0.13**	0.27
corr.5	2.09	0.77	0.39	**0.16**	0.21
mag.1	4.52	2.19	3.49	2.35	**1.18**
mag.2	13.43	3.11	2.73	2.24	**1.17**
mag.3	14.80	**0.40**	1.22	1.69	3.74
mag.4	39.73	5.12	**0.77**	1.02	2.49
mag.5	3.47	0.85	1.62	**0.73**	1.43
mag.6	X	2.29	3.91	**1.19**	3.12
room1	0.16	0.07	0.06	**0.03**	0.05
room2	0.33	0.07	0.11	0.13	**0.04**
room3	0.15	0.11	0.07	0.09	**0.02**
room4	0.09	0.04	**0.03**	0.04	0.04
room5	0.12	0.20	0.07	0.06	**0.04**
room6	0.05	0.08	0.04	**0.02**	0.04
Avg.	5.071	1.099	0.949	**0.634**	0.867

Table 3. Evaluation of VIO methods on TUM-VI dataset, with RMSE ATE (m), SE(3)-aligned. The corr. and mag. represent corridor and magistrale sequences, respectively.

dataset and room1 of the TUM-VI dataset, then generalize to other sequences in the same dataset. The results are computed as the median of three trials. All trajectories are in metric scale and SE(3)-aligned with the ground-truth.

Results on EuRoC: As presented in Tab.2, we choose the state-of-the-art classic VIO methods and learning-based approaches as comparison. Compared with classic methods, our method outperforms most approaches in terms of RMSE ATE and is comparative to the performance of the DM-VIO [30]. Furthermore, our method exhibits significantly superior performance compared to other learning-based VIO methods. We also observe the performance of our method exceeds the VIO system Kimera [26] and iS-LAM [12], both with stereo-inertial settings.

Results on TUM-VI: To evaluate the generalization

ability of our method, we also conduct experiments on another TUM-VI dataset [27], which is a highly challenging handheld dataset with large-scale scenes, compared with the EuRoC dataset [5].

We compare our method to the classic state-of-the-art VIO methods as presented in Lukas et al. [30]. The results are reported in Tab.3. Our method can achieve better accuracy on 7 sequences than alternative methods, even compared to the DM-VIO method, which shows the best result among all methods. However, in other sequences, DM-VIO reported better results than ours, which can be attributed mainly to its more robust initialization and long-term scale refinement.

5. Conclusion

This paper presents a novel VIO system named Adaptive VIO, which combines online continual learning with classic optimization. We employ neural networks to predict visual correspondence and IMU bias and then construct visual-inertial bundle adjustment to tightly couple both sensor measurements to refine the state estimation. The refined estimates can be fed back to the front-end networks that are updated through online continual learning, enabling our system to adapt to new environments. Experimental results illustrate that the online continual learning of our method can improve the overall system performance, whether for visual adaptation, IMU adaptation, or joint of them. Compared with classic and learning-based state-of-the-art VIO systems, our method can achieve competitive results and show potential adaptation to unseen scenarios. In the future, we plan to explore extending the online feedback mechanism to various networks and improving the robustness and efficiency of the system.

Acknowledgement

References

[1] Yasin Almalioglu, Mehmet Turan, Muhamad Risqi U. Saputra, Pedro P.B. de Gusmão, Andrew Markham, and Niki Trigoni. Selfvio: Self-supervised deep monocular visual–inertial odometry and depth estimation. *Neural Networks*, 150:119–136, 2022. 2, 7

[2] Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. Robust visual inertial odometry using a direct ekf-based approach. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 298–304, 2015. 2, 7

[3] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J. Davison. Codeslam - learning a compact, optimisable representation for dense visual slam. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2560–2568, 2018. 2

[4] Russell Buchanan, Varun Agrawal, Marco Camurri, Frank Dellaert, and Maurice Fallon. Deep imu bias inference for robust visual-inertial odometry with factor graphs. *IEEE Robotics and Automation Letters*, 8(1):41–48, 2023. 2, 4

[5] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35 (10):1157–1163, 2016. 3, 6, 7, 8

[6] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6): 1874–1890, 2021. 2, 4

[7] Danpeng Chen, Nan Wang, Runsen Xu, Weijian Xie, Hujun Bao, and Guofeng 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)*, pages 275–283, 2021. 2

[8] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 5

[9] Ronald Clark, Sen Wang, Hongkai Wen, Andrew Markham, and Niki Trigoni. Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, page 3995–4001. AAAI Press, 2017. 2

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset. In *CVPR Workshop on the Future of Datasets in Vision*. sn, 2015. 3

[11] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33 (1):1–21, 2017. 2, 4, 5

[12] Taimeng Fu, Shaoshu Su, and Chen Wang. islam: Imperative slam. *arXiv preprint arXiv:2306.07894*, 2023. 2, 7, 8

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 3

[14] Liming Han, Yimin Lin, Guoguang Du, and Shiguo Lian. Deepvio: Self-supervised deep learning of monocular visual inertial odometry using 3d geometric constraints. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6906–6913, 2019. 2

[15] Euntae Hong and Jongwoo Lim. Visual-inertial odometry with robust initialization and online scale estimation. *Sensors*, 18(12):4287, 2018. 7

[16] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015. 1, 7

[17] Mingyang Li and Anastasios I Mourikis. High-precision, consistent ekf-based visual-inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013. 1, 2, 7

[18] Shunkai Li, Fei Xue, Xin Wang, Zike Yan, and Hongbin Zha. Sequential adversarial learning for self-supervised deep visual odometry. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2851–2860, 2019. 2

[19] Shunkai Li, Xin Wang, Yingdian Cao, Fei Xue, Zike Yan, and Hongbin Zha. Self-supervised deep visual odometry with online adaptation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6338–6347, 2020. 3

[20] Shunkai Li, Xin Wu, Yingdian Cao, and Hongbin Zha. Generalizing to the open world: Deep visual odometry with online adaptation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13179–13188, 2021. 3

[21] Wenxin Liu, David Caruso, Eddy Ilg, Jing Dong, Anastasios I. Mourikis, Kostas Daniilidis, Vijay Kumar, and Jakob Engel. Tlio: Tight learned inertial odometry. *IEEE Robotics and Automation Letters*, 5(4):5653–5660, 2020. 2

[22] Anastasios I. Mourikis and Stergios I. Roumeliotis. A multistate constraint kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3565–3572, 2007. 2

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 6

[24] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018. 1, 2, 4, 7

[25] Tong Qin, Jie Pan, Shaozu Cao, and Shaojie Shen. A general optimization-based framework for local odometry estimation with multiple sensors, 2019. 2

[26] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1689–1696, 2020. 7, 8

[27] David Schubert, Thore Goll, Nikolaus Demmel, Vladyslav Usenko, Jörg Stückler, and Daniel Cremers. The tum vi benchmark for evaluating visual-inertial odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1680–1687. IEEE, 2018. 3, 6, 7, 8

[28] E. Jared Shamwell, Kyle Lindgren, Sarah Leung, and William D. Nothwang. Unsupervised deep visual-inertial odometry with online error correction for rgb-d imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2478–2493, 2020. 2

[29] Joan Sola, Jeremie Deray, and Dinesh Atchuthan. A micro lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537*, 2018. 4

[30] Lukas von Stumberg and Daniel Cremers. Dm-vio: Delayed marginalization visual-inertial odometry. *IEEE Robotics and Automation Letters*, 7(2):1408–1415, 2022. 1, 2, 7, 8

[31] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012. 6

[32] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. *arXiv preprint arXiv:1806.04807*, 2018. 2

[33] Zachary Teed and Jia Deng. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Advances in neural information processing systems*, 2021. 2

[34] Zachary Teed and Jia Deng. Tangent space backpropagation for 3d transformation groups. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10338–10347, 2021. 4

[35] Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. *arXiv preprint arXiv:2208.04726*, 2022. 2, 3, 5, 6, 7

[36] Niclas Vödisch, Daniele Cattaneo, Wolfram Burgard, and Abhinav Valada. Continual slam: Beyond lifelong simultaneous localization and mapping through continual learning. In *Robotics Research*, pages 19–35, Cham, 2023. Springer Nature Switzerland. 3

[37] Lukas Von Stumberg, Vladyslav Usenko, and Daniel Cremers. Direct sparse visual-inertial odometry using dynamic marginalization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2510–2517, 2018. 1, 7

[38] Niclas Vödisch, Daniele Cattaneo, Wolfram Burgard, and Abhinav Valada. Covio: Online continual learning for visual-inertial odometry. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2464–2473, 2023. 3

[39] Chen Wang, Dasong Gao, Kuan Xu, Junyi Geng, Yaoyu Hu, Yuheng Qiu, Bowen Li, Fan Yang, Brady Moon, Ab-hinav Pandey, et al. Pypose: A library for robot learning with physics-based optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22024–22034, 2023. 4

[40] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2043–2050. IEEE, 2017. 2

[41] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020. 6

[42] Xin Wang, Youqi Pan, Zike Yan, and Hongbin Zha. Visual-inertial odometry based on kinematic constraints in imu frames. *IEEE Robotics and Automation Letters*, 7(3):6550–6557, 2022. 1

[43] Fei Xue, Xin Wang, Shunkai Li, Qiuyuan Wang, Junqiu Wang, and Hongbin Zha. Beyond tracking: Selecting memory and refining poses for deep visual odometry. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8575–8583, 2019. 2

[44] Ming Zhang, Mingming Zhang, Yiming Chen, and Mingyang Li. Imu data processing for inertial aided navigation: A recurrent neural network based approach. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3992–3998, 2021. 2

[45] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6612–6619, 2017. 2