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).

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.

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).
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, SelfVO [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.
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 $\hat{b}_a, \hat{b}_g$ 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 $i^{th}$ and $i + 1^{st}$ IMU frames are between the timestamps of $k^{th}$ and $k + 1^{st}$ image frames. Let $\hat{\alpha}^{b_k}_i, \hat{\beta}^{b_k}_i, \hat{\gamma}^{b_k}_i$ represent the preintegration of translation, velocity and rotation until the $i^{th}$ IMU frame, where $b_k$ denotes the preintegration is under the body coordinate at the $k^{th}$ image frame. The body coordinate aligns with the IMU coordinate. Then, the preintegration until $i + 1^{st}$ IMU frame can be expressed in the following form [24]:

$$\hat{\alpha}^{b_k}_{i+1} = \hat{\alpha}^{b_k}_i + \frac{1}{2} R(\hat{\gamma}^{b_k}_i) \hat{\omega}_i t^2$$  

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

$$\hat{\gamma}^{b_k}_{i+1} = \hat{\gamma}^{b_k}_i \odot \left[ \frac{1}{2} \hat{\omega}_i t^2 \right]$$  

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

The propagation from $i^{th}$ to $i + 1^{st}$ IMU frame is differentiable with respect to $\hat{a}_i$ and $\hat{\omega}_i$. Henceforth, the preintegration from the $k^{th}$ to $k + 1^{st}$ image frame is also differentiable, which allows the gradient backpropagation 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:

$$X = \{ p^{w}_b, q^{w}_b, v^{w}_b, b_a, b_g, d_{0..m} \}_{0..n}$$

where $p^{w}_b, q^{w}_b$ and $v^{w}_b$ denote translation, quaternion and velocity of the body. Bolded $b_a$ and $b_g$ are biases of the accelerometer and gyroscope. $d_{0..m}$ 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:**

$$r_v = \sum_{i,j \in \mathcal{G}} \| p^w_i - KT_i^w d(p^w_i) K^{-1} p^w_j \|$$

where $p^w_i, p^w_j$ are a pair of matching points predicted by the network. $T_i^w$ 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:

$$r^{\Delta p}_{k+1} = \{ r^{\Delta p}_{b_k+1}, r^{\Delta v}_{b_k+1}, r^{\Delta R}_{b_k+1}, r^{\Delta (b_a)}_{b_k+1}, r^{\Delta (b_g)}_{b_k+1} \}$$

**Preintegration residuals:**

$$r^{\Delta p}_{b_k+1} = R^{b_k}_w (p^{w}_{b_{k+1}} - p^{w}_{b_k} - v^{w}_{b_k} \Delta t_k)$$

$$+ \frac{1}{2} g^{w} \Delta t_k^2 - \hat{\alpha}^{b_k}_{b_k+1} (b_a, b_g)$$

$$r^{\Delta v}_{b_k+1} = R^{b_k}_w (v^{w}_{b_{k+1}} - v^{w}_{b_k} - g^{w} \Delta t_k)$$

$$- \hat{\beta}^{b_k}_{b_k+1} (b_a, b_g)$$

$$r^{\Delta R}_{b_k+1} = \text{Log} \left( R(\hat{\gamma}^{b_k}_{b_k+1} (b_g)) \right) R^{b_k}_w R^{w}_{b_{k+1}}$$

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

**Bias residuals:**

$$r^{\Delta (b_a)}_{b_k+1} = (b_a) - (\hat{b}_a)^k$$

$$r^{\Delta (b_g)}_{b_k+1} = (b_g) - (\hat{b}_g)^k$$

where $\hat{b}_a, \hat{b}_g$ are the biases 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 $(\hat{b}_a, \hat{b}_g)$. We never reintegrate measurements by the updated bias $(b_a, b_g)$ 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:

\[
X^* = \arg \min_X \left( \| \mathbf{r}_v \|_2^2 + \sum_{k=0}^{n} \| \mathbf{r}_{\mathbf{b}_k} \|_2^2 \right) 
\]  

(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:

\[
L_{\text{visual}} = \sum_{i \in G} \left\| I^i(p^*_i) - I^i(p_i) \right\| 
\]  

(13)

\[
p^*_i = KT^{-1}_i \mathbf{d}(p^*_i) K^{-1} p^*_i 
\]  

(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 initialized 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:

\[
L_{\text{imu}} = L_{\Delta \mathbf{p}} + L_{\Delta \mathbf{v}} + L_{\Delta \mathbf{R}} 
\]  

(15)

\[
L_{\Delta \mathbf{p}} = R_{\mathbf{b}_n}^{-1} \left( \mathbf{p}_{\mathbf{b}_n} - \mathbf{p}_{\mathbf{b}_{n-1}} - \mathbf{v}_{\mathbf{b}_{n-1}} \Delta t \right) + \frac{1}{2} g^w \Delta t^2 - \epsilon_{\mathbf{b}_n}^{-1} (\mathbf{\bar{b}}_a, \mathbf{\bar{b}}_g) 
\]  

(16)

\[
L_{\Delta \mathbf{v}} = R_{\mathbf{b}_n}^{-1} \left( \mathbf{v}_{\mathbf{b}_n} - \mathbf{v}_{\mathbf{b}_{n-1}} + g^w \Delta t \right) - \beta_{\mathbf{b}_n}^{-1} (\mathbf{\bar{b}}_a, \mathbf{\bar{b}}_g) 
\]  

(17)

\[
L_{\Delta \mathbf{R}} = \log \left( R(\gamma_{\mathbf{b}_n}^{-1} (\mathbf{\bar{b}}_g)) ^T R_{\mathbf{b}_n}^{-1} R_{\mathbf{b}_n} \right) 
\]  

(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 data loader, 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 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
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 over our baseline model $V_{PT} + I_{RW}$ 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. 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 240,000 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 MH01. 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 $1 \times 10^{-5}$. 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 $1 \times 10^{-4}$. After that, we
perform VIBA with the learning rate of $1 \times 10^{-6}$.

### 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 MH01 of the EuRoC.

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 ‘T’ denote monocular, stereo, and IMU.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MSCKF[17]</td>
<td>M+I</td>
<td>0.42</td>
<td>0.45</td>
<td>0.23</td>
<td>0.37</td>
<td>0.48</td>
<td>0.34</td>
<td>0.20</td>
<td>0.67</td>
<td>0.10</td>
<td>0.16</td>
<td>0.34</td>
</tr>
<tr>
<td>OKVIS[16]</td>
<td>M+I</td>
<td>0.33</td>
<td>0.37</td>
<td>0.25</td>
<td>0.27</td>
<td>0.39</td>
<td>0.094</td>
<td>0.14</td>
<td>0.21</td>
<td>0.09</td>
<td>0.17</td>
<td>0.23</td>
</tr>
<tr>
<td>ROVIO[2]</td>
<td>M+I</td>
<td>0.21</td>
<td>0.25</td>
<td>0.25</td>
<td>0.49</td>
<td>0.52</td>
<td>0.10</td>
<td>0.10</td>
<td>0.14</td>
<td>0.12</td>
<td>0.14</td>
<td>0.23</td>
</tr>
<tr>
<td>VINS-Mono[24]</td>
<td>M+I</td>
<td>0.15</td>
<td>0.15</td>
<td>0.22</td>
<td>0.32</td>
<td>0.30</td>
<td>0.079</td>
<td>0.11</td>
<td>0.18</td>
<td>0.08</td>
<td>0.10</td>
<td>0.17</td>
</tr>
<tr>
<td>Kimeva[26]</td>
<td>S+I</td>
<td>0.11</td>
<td>0.10</td>
<td>0.16</td>
<td>0.24</td>
<td>0.35</td>
<td>0.05</td>
<td>0.08</td>
<td>0.07</td>
<td>0.08</td>
<td>0.10</td>
<td>0.13</td>
</tr>
<tr>
<td>Online VIO[15]</td>
<td>M+I</td>
<td>0.14</td>
<td>0.13</td>
<td>0.20</td>
<td>0.22</td>
<td>0.20</td>
<td>0.05</td>
<td>0.07</td>
<td>0.16</td>
<td>0.04</td>
<td>0.11</td>
<td>0.13</td>
</tr>
<tr>
<td>VI-DSO[37]</td>
<td>M+I</td>
<td>0.062</td>
<td>0.044</td>
<td>0.117</td>
<td>0.132</td>
<td>0.121</td>
<td>0.059</td>
<td>0.067</td>
<td>0.096</td>
<td>0.040</td>
<td>0.062</td>
<td>0.08</td>
</tr>
<tr>
<td>DM-VIO[30]</td>
<td>M+I</td>
<td>0.065</td>
<td>0.044</td>
<td>0.097</td>
<td>0.102</td>
<td>0.096</td>
<td>0.048</td>
<td>0.045</td>
<td>0.069</td>
<td>0.029</td>
<td>0.050</td>
<td>0.06</td>
</tr>
<tr>
<td>iSLAM[12]</td>
<td>S+I</td>
<td>0.500</td>
<td>0.391</td>
<td>0.656</td>
<td>1.285</td>
<td>1.088</td>
<td>0.521</td>
<td>0.405</td>
<td>0.397</td>
<td>0.421</td>
<td>0.580</td>
<td>0.58</td>
</tr>
<tr>
<td>SciVIo[1]</td>
<td>M+I</td>
<td>0.19</td>
<td>0.15</td>
<td>0.21</td>
<td>0.16</td>
<td>0.29</td>
<td>0.08</td>
<td>0.09</td>
<td>0.1</td>
<td>0.11</td>
<td>0.08</td>
<td>0.15</td>
</tr>
<tr>
<td>Ours</td>
<td>M+I</td>
<td>0.050</td>
<td>0.055</td>
<td>0.069</td>
<td>0.092</td>
<td>0.124</td>
<td>0.035</td>
<td>0.045</td>
<td>0.073</td>
<td>0.052</td>
<td>0.086</td>
<td>0.07</td>
</tr>
</tbody>
</table>

Figure 3. RMSE ATE changes in MH01 (EuRoC) and rooml (TUM-VI) during online continual learning of visual and IMU.
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

We thank anonymous reviewers and AC for their fruitful comments and suggestions. This work is supported by the NSF, China (U22A2061, 62176010), and 230601GP0004.
References


[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


