

DeepVIO: Self-supervised Deep Learning of Monocular Visual Inertial Odometry using 3D Geometric Constraints

Liming Han¹, Yimin Lin¹, Guoguang Du¹, Shiguo Lian¹

Abstract—This paper presents a self-supervised deep learning network for monocular visual inertial odometry (named DeepVIO). DeepVIO provides absolute trajectory estimation by directly merging 2D optical flow feature (OFF) and Inertial Measurement Unit (IMU) data. Specifically, it firstly estimates the depth and dense 3D point cloud of each scene by using stereo sequences, and then obtains 3D geometric constraints including 3D optical flow and 6-DoF pose as supervisory signals. Note that such 3D optical flow shows robustness and accuracy to dynamic objects and textureless environments. In DeepVIO training, 2D optical flow network is constrained by the projection of its corresponding 3D optical flow, and LSTM-style IMU preintegration network and the fusion network are learned by minimizing the loss functions from ego-motion constraints. Furthermore, we employ an IMU status update scheme to improve IMU pose estimation through updating the additional gyroscope and accelerometer bias. The experimental results on KITTI and EuRoC datasets show that DeepVIO outperforms state-of-the-art learning based methods in terms of accuracy and data adaptability. Compared to the traditional methods, DeepVIO reduces the impacts of inaccurate Camera-IMU calibrations, unsynchronized and missing data.

I. INTRODUCTION

6-DoF motion estimation is one of the key challenges in the fields such as robotics and autonomous driving. Because of low-cost, and easy hardware setup, camera-based solutions [1] have drawn a large attention by the community. Therefore, in the last decade, impressive results have been demonstrated in the contexts of Visual Odometry (VO) and Visual Simultaneous Localization and Mapping (VSLAM) [2]. For example, the direct-based representative DSO [3] and the feature-based representative ORB-SLAM [4] both achieve very high localization accuracy in the large-scale environment, and real-time performance with commercial CPUs. However, they still face some challenging issues when they are deployed in non-textured environments, serious image blur or under extreme lighting conditions. Recently, many Visual Inertial Odometry (VIO) systems [5], [6], [7] are proposed to eliminate these limitations, which combine measurements from an inertial measurement unit (IMU) with a camera to improve motion tracking performance. However, the current VIO systems heavily rely on manual interference to analyze failure cases and refine localization results. Furthermore, all these VIO systems require careful parameter tuning procedures for the specific environment they have to work in.

¹Liming Han, Yimin Lin*, Guoguang Du, Shiguo Lian are all with the AI Department, CloudMinds Technologies Inc., Beijing, 100102, China liming.han, george.du, scott.lian@cloudminds.com, *linyimin2012@hotmail.com

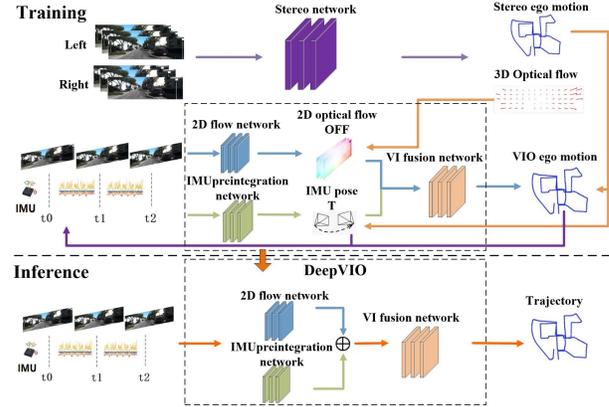


Fig. 1. The pipeline of the DeepVIO. Novel 3D optical flow and stereo ego motion are used as 3D geometric constraints to supervise 2D optical flow learned from 2D flow network, ego-motions estimated from the IMU preintegration network and VI fusion network, the state of the IMU is updated when it receives the feedback from VI fusion network. More details can be found in Sec. III.

In recent years, deep learning based VO has drawn significant attentions due to its potentials in learning capability and the robustness to camera parameters and challenging environments [8], [9]. These data-driven VO methods have successfully learned new feature representations from images that are used to further improve the motion estimation. Inspired by this, a deep learning based VIO [10] is also demonstrated to fuse the estimated pose from deep Convolutional Neural Networks (CNNs) and the IMU data with the Long Short-Term Memory (LSTM) networks. Although supervised deep learning methods have achieved state-of-the-art results on motion estimation, it is impractical to gather large amounts of ground truth data. Moreover, the trained models usually do not generalize well to unseen scenes without fine-tuning on sufficient ground truth data. These limitations suggest some methods [11], [12] to look for various unsupervised learning VO/VIO schemes, and consequently learn a mapping from pixels to flow, depth and camera motion without trajectory ground truth. Nevertheless, most unsupervised methods learn from photometric and temporal consistency between consecutive frames in a monocular video, which are prone to scale-ambiguity. Meanwhile, they try to regress the pose but probably result in high drift since there are not enough data covering various moving speed of single camera. Recently, 2D optical flow is widely used as self-supervisory signals to learn unsupervised ego-motion system, but it has aperture problem due to the missing structure in local parts of the single camera. Moreover, it

cannot explicitly handle dynamic objects and textureless environments from consecutive frames that inevitably results in inaccurate flow predictions.

To overcome these limitations, a stereo camera based approach is proposed in this paper as shown in Fig. 1. This approach not only provides synthetic 2D optical flow as supervision from its precise 3D optical flow, but also solves the scale and speed ambiguity using its absolute 6-DoF pose. Additionally, an IMU status update scheme is designed to improve IMU pose estimation. Some experimental results will be given to show that, the proposed DeepVIO is able to estimate VIO in a real world scale from monocular camera and IMU sequences. In summary, our main contributions are as follows:

- (1) It is the first time to present a self-supervised end-to-end monocular VIO network with the supervisory signals obtained from the stereo sequences.
- (2) It takes 3D geometric constraints including 3D optical flow and 6-DoF pose to penalize inconsistencies in the estimated 2D OFF, IMU pose and VIO trajectory.
- (3) We update additional bias for IMU using the pose feedback from FC-fusion network similar to traditional tightly-coupled VIO methods.
- (4) Our experimental results achieve good performance in terms of accuracy and data adaptability compared to state-of-the-art learning based VO and VIO systems on KITTI and EuRoC datasets.

II. RELATED WORK

A. Traditional Methods

VIO fuses raw camera and IMU data in a single pose estimator and leads to more robust and higher accuracy even in complex and dynamic environments. In the past several decades, most tightly-coupled VIO systems can be divided into filtering-based and optimization-based approaches. Filtering-based representative is MSCKF[13], which combines geometric constraints with IMU measurements in a multi-state constraint Extended Kalman Filter (EKF). The method has low computational complexity and is capable of very accurate pose estimation in large-scale real environments. ROVIO[14] is another popular filtering based VIO. It uses an EKF to fuse the IMU data and intensity errors in the update step. On the other hand, optimization-based representative is OKVIS[15], which is also called a keyframe-based VIO system. VINS[7] is a tightly-coupled, nonlinear optimization-based method, which is used to obtain highly accurate odometry by fusing preintegrated IMU measurements and feature observations.

B. Supervised or Semi-supervised Learning Methods

Supervised learning based approaches have recently achieved great advances in motion estimation [16]. Avoiding the hand-crafted features extraction used in previous methods, the deep neural networks have successfully learned good feature representation directly from a lot of images. [17] presents a LSTM to learn the relation between camera poses and inertial sensor measurements. This estimation is then

appropriately combined with the output of a visual tracking system using a linear Kalman Filter to provide a robust final pose estimation. In addition, [10] fuses the estimated pose from [8] and the IMU data with the LSTM. The fusion network is trained jointly in an end-to-end way, and the proposed fusion system demonstrates comparable performance with traditional sensor fusion methods. However, these supervised learning approaches need large amounts of ground truth data for training, which is expensive to obtain in practice.

C. Unsupervised Learning Methods

In recent years, large progress has been achieved in the development of depth estimation with unsupervised learning methods [18], [19]. More and more unsupervised paradigms for ego-motion estimation have been proposed to exploit brightness constancy and spatial smoothness to train depth or flow models. All the networks are jointly optimized during training, and then they can be applied independently at test time. For instance, [20] learns depth and ego-motion from monocular video in an unsupervised way. The CNNs is trained by a photometric reconstruction loss, which was obtained by warping nearby views to the target using the computed depth and pose. In addition, [21] addresses unsupervised learning of scene depth, robot ego-motion and object motions where the supervision is provided by geometric structure of monocular videos as input. Furthermore, many recent efforts [11], [22] explore the geometric relationships between depth, camera motion, and flow for unsupervised learning of depth and flow estimation models. However, all the methods focus on monocular setup, and an extension to the above frameworks is to use stereo pairs for training. The main advantage is that this avoids issue with the depth-speed ambiguity that exists in monocular 3D reconstruction. [23] and [24] both use stereo pairs to define loss function for training the networks from spatial and temporal dense information. At test time their frameworks are able to estimate depth and two-view odometry from a monocular sequence.

To the best of our knowledge, [12] is the first unsupervised VIO system. The network learns to integrate IMU measurements and generate trajectories which are then corrected online according to the Jacobians of scaled image projection errors with respect to a spatial grid of pixel coordinates. However, it needs depth sensor, like RGBD or Lidar sensor, as input in order to achieve absolute scale recovery. Different from this, this paper proposes a self-supervised VIO system using a low-cost and easy-to-use stereo camera instead.

III. METHOD

The proposed DeepVIO mainly consists of CNN-Flow, LSTM-IMU and FC-Fusion networks, which jointly compute continuous trajectories from monocular images and IMU data as shown in Fig. 2 (a). More specifically, the OFF and 2D optical flow are calculated by the CNN-Flow network through the monocular sequences (e.g., $I_{L,t-1}$ and $I_{L,t}$). Meanwhile, the relative 6-DoF pose (e.g., IMU-se3) between the adjacent

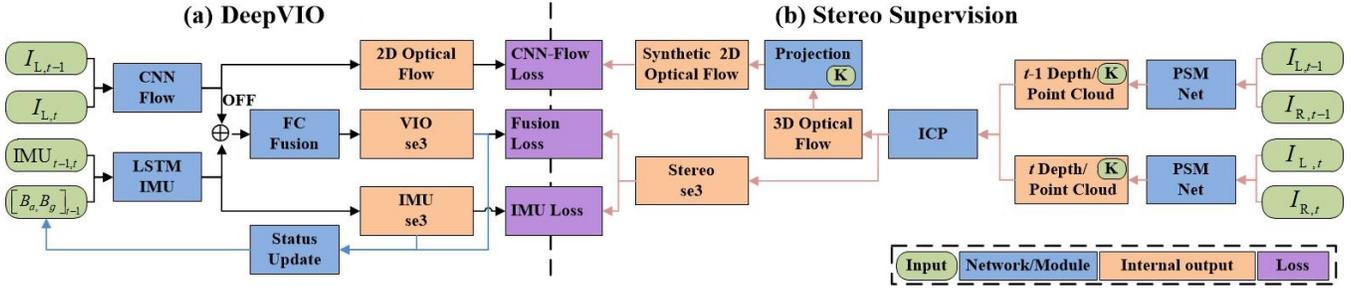


Fig. 2. Illustration of our proposed framework in training and inferring phase. DeepVIO consists of CNN-Flow, LSTM-IMU and FC-Fusion, which is trained by the supervisory signals obtained from 3D geometric constraints of stereo sequences (e.g., CNN-Flow Loss, IMU Loss and Fusion Loss), it updates IMU status using the pose feedback from FC-fusion network.

two frames is calculated by IMU preintegration network (LSTM-IMU) through the IMU data (e.g., $IMU_{t-1,t}$) and status $S[B_a, B_g]$ (bias of accelerometer and gyroscope). Finally, the concatenated features from OFF and IMU-se3 are fed into the FC-Fusion network to calculate the trajectory of the monocular camera (e.g., VIO-se3). Note that, VIO-se3 and IMU-se3 give IMU status corrections as a feedback through the status update module.

Here, we employ stereo sequences as supervision for learning DeepVIO in Fig. 2 (b). In particular, the pretrained depth network (e.g., PSMNet) is firstly applied to estimate dense depth images from left and right images at time $t-1$ (e.g., $t-1$ Depth) and t (e.g., t Depth), respectively. After that, we can recover 3D point clouds from each pair of left image and its depth. Next, the 6-DoF relative pose (e.g., Stereo-se3) and 3D optical flow are calculated from the two point clouds (e.g., $t-1$ -th and t -th point clouds) via the well-known Iterative Closest Point (ICP) method [25]. Moreover, we synthesize a 2D optical flow by projecting the 3D optical flow into its view. During DeepVIO training, the 3D geometric constraints are:

- **Optical Flow Loss.** It provides pixel-level constraints for training CNN-Flow network from minimizing the differences between synthetic 2D optical flow and inferred 2D optical flow.
- **IMU Loss.** LSTM-IMU is optimized by minimizing the differences between its inferred IMU-se3 and the supervision Stereo-se3.
- **Fusion Loss.** Similar to IMU Loss, the output of FC-Fusion network (e.g. VIO-se3) is also constrained by Stereo-se3.

More details are presented in the following sections.

A. Stereo Network as Supervision

Fig. 2 (b) shows how to generate 3D geometric constraints including 3D optical flow and Stereo-se3 from stereo sequences.

1) *Depth and Point Cloud:* In most stereo systems, the disparity map is calculated through traditional stereo matching algorithms [26] or directly estimated by an end-to-end CNNs [27]. In this paper, we chose the state-of-the-art PSMNet [27] since it provides the accurate and dense disparity map. Note that, PSMNets output scene disparity

map $q_L = (x_L, y_L, x_L - x_R)$, where the coordinates (x_L, y_L) give the position of a point P in left image plane, and its matching point (x_R, y_R) in right image where $y_L = y_R$ in such rectified stereo images. Then the depth d_L (see Fig. 3(c)) are calculated as,

$$d_L = \frac{fb}{q_L} = \frac{fb}{x_L - x_R} \quad (1)$$

where b is the baseline and f is the camera focal length. Once we get the depth d_L , the 3D point cloud \mathbf{c} is reconstructed as

$$\mathbf{c} = \mathbf{K}^{-1}d_L[x_L, y_L, 1]^T \quad (2)$$

where \mathbf{K} is the intrinsic parameter of left camera [28]. Fig. 3(d) shows the 3D point cloud converted from each pixel of depth map with absolute scales.

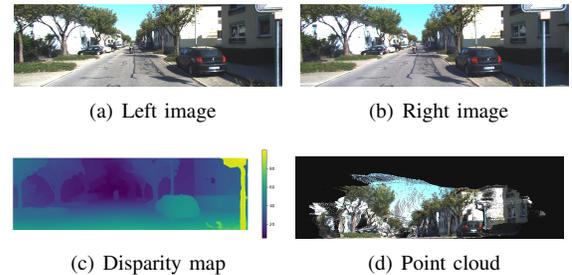


Fig. 3. Illustration of disparity map and point cloud obtained from stereo images. Where (a) is the left image, (b) is the right image, (c) is the disparity map and (d) is the 3D point cloud, respectively.

2) *Stereo-se3 with ICP:* In order to reduce the estimation error from depth prediction of PSMNet, only a part of points whose depth values satisfy $d_1 < d_L < d_2$ are selected to reconstruct the 3D point cloud, where d_1 and d_2 are the nearest and farthest depth threshold. For instance, the depth estimated from the sky region has a large error and is necessary to eliminate in our case. Then we generate two adjacent 3D point clouds \mathbf{c}_{t-1} and \mathbf{c}_t follows the Eq.(2). Then we implement the traditional ICP algorithm as,

$$\{I(\mathbf{c}_{t-1}, \mathbf{c}_t), \mathbf{R}, t\} = \text{ICP}(\mathbf{c}_{t-1}, \mathbf{c}_t) \quad (3)$$

where $I(\mathbf{c}_{t-1}, \mathbf{c}_t)$ is the index of matched 3D points, and \mathbf{R}, t is rotation and translation. Obviously, they have the following properties as.

$$\mathbf{T} = \left\{ \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \mid \mathbf{R} \in SO(3), \mathbf{t} \in R^3 \right\} \quad (4)$$

thus Stereo-se3 $[\omega, v]$ is a 6-dimension vector (R^6) that is calculated as,

$$se3 = \log \{ \mathbf{R}, \mathbf{t} \} = \left\{ \begin{pmatrix} \omega & v \\ 0 & 0 \end{pmatrix} \mid \omega \in so(3), v \in R^3 \right\} \quad (5)$$

3) *3D Optical Flow*: Finally, the 3D optical flow in the scene can be calculated by the corresponding 3D points in two point clouds as,

$$(v_X, v_Y, v_Z) = \Delta I(\mathbf{c}_{t-1}, \mathbf{c}_t) = \mathbf{c}_{t-1} - \mathbf{c}_t \quad (6)$$

where $\mathbf{v}_{3D} = (v_X, v_Y, v_Z)$ is the 3D optical flow in the real world. In order to visualize our 3D optical flow, we project the sparse \mathbf{v}_{3D} to left and right views as,

$$\begin{aligned} (v_x, v_y, 1)_L &= \frac{\mathbf{K}\mathbf{v}_{3D}}{d_L(x, y)} \\ (v_x, v_y, 1)_R &= \frac{\mathbf{K}(\mathbf{R}\mathbf{v}_{3D} + \mathbf{t})}{d_L(x, y)} \end{aligned} \quad (7)$$

Thus we can show them in left and right images (see arrows in Fig. 4(a) and (b)) using the Eq.7.

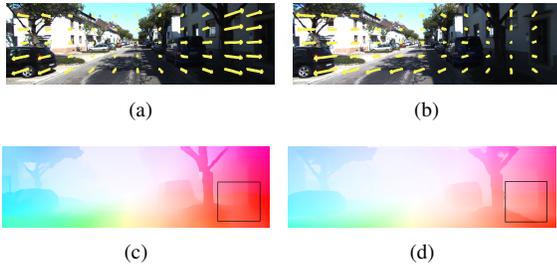


Fig. 4. Illustration of 3D optical flow from (a) left and (b) right views, respectively (shown as yellow arrows), (c) synthetic 2D optical flow calculated from our 3D optical flow and (d) 2D optical flow obtained from FlownetC. It can be seen from (c) and (d) that our 3D optical flow represents the real world more accurately.

4) *Synthetic 2D Optical Flow*: We obtain a synthetic 2D optical flow named as $\mathbf{v}_{2D} = (v_x, v_y)$ by projecting every pixel of 3D optical flow to the left image using Eq.(7), as shown in Fig. 4(c). In the KITTI dataset, true value of the optical flow is provided using Velodyne HDL-64E, but it is a sparse optical flow. In contrast, our synthetic 2D optical flow is a dense and each pixel has a true corresponding optical value.

Intuitively, traditional monocular 2D optical flow algorithms, which calculate the movement of pixels in the XY directions with image correlation, are unable to decide the motion correctly. Learning based optical flow methods also suffer from such limitation especially for the textureless region, as shown using a rectangle in Fig. 4(d). The reason is that depth information is lost when it is captured by a camera

in real world. As a result, there are ambiguities between the camera and object motion in the scene. These ambiguities can be resolved when using a range camera, such as stereo camera [29]. Therefore, 3D optical flow obtained from stereo sequences is a true representation of the camera motion with XYZ directions in the 3D real world. In order to prove our 3D optical flow is competent to serve as supervision, an accurate comparison between 2D optical flow and our synthetic one will be described in section IV-D.

B. CNN-Flow Network

It has been proved that the monocular sequences can be used to calculate the 2D optical flow with the CNNs network. Therefore, we employ the CNN-Flow network structure which is the same as FlownetC [30], [31]. Two consecutive images stacked together are fed into the CNNs. Then, the CNNs outputs an intermediary vector OFF whose dimension is $N \times 6 \times 20$, where N is determined by the image size. Let $c7s1-k$ denote a 7×7 Convolution-InstanceNorm-ReLU layer with k filters and stride 1. The input of CNN-Flow network is merged stereo images. The network consists of:

$c7s2-64, c5s2-128, c5s2-256, c3s1-256, c3s2-512, c3s1-512, c3s2-512, c3s2-1024, c3s1-1024$.

Then 2D optical flow map is refined by a upconvolutional layers, which consists of unpooling and a convolution. Note that, synthetic 2D optical flow is used as the supervision value when the CNN-Flow network is training, thus the CNN-Flow can converge correctly. Finally, the OFF is fed into FC-Fusion network as visual input.

C. LSTM-IMU Network

In recent years, the deep learning network has made initial attempts in IMU data processing. For example in VINET, a LSTM-style network is trained to learn the relative pose from the previous frame and subsequent inertial measurements to the next frame. However, the LSTM network training is usually difficult to converge because of the noises and changes of IMU data. In most traditional VIO systems, the IMU motion is calculated as a time-varying model, and the state of IMU is constantly updated. Therefore, this paper proposes a learning method considering the state of IMU.

1) *Preintegrated Network*: LSTM is a time recurrent neural network which is potential for obtaining poses from IMU sequences. In this paper, IMU data and state S , are directly fed into LSTM. The preintegrated network is a 2-layers LSTM network and each of them has 6 hidden states. And the dropout probability is 0.5 in each layer of LSTM. Their input dimension is $N \times 12$, where N is the number of IMU frames between two frames of images. Its output is the IMU-se3 with 6 dimensions (e.g. 3 translation and 3 rotation). Note that, Stereo-se3 is employed to constraint IMU-se3.

2) *Status Update Module*: As mention above, our network inputs the state of the IMU, which aims to update the IMU status continuously. To accomplish this, the status error of the IMU is a feedback which is the difference of the motion

between the IMU preintegrated and FC-Fusion networks. The status of IMU is updated as follows[6], [32]:

$$\begin{aligned}\tilde{S}_{t-1} &= \arg \min_{\tilde{S}_{t-1}} (\text{VIO-se3}, \text{IMU-se3}) \\ &= \arg \min_{\tilde{S}_{t-1}} (\rho([e_{\mathbf{R}}^T e_{\mathbf{p}}^T] \Sigma_I [e_{\mathbf{R}}^T e_{\mathbf{p}}^T]^T))\end{aligned}\quad (8)$$

$$e_{\mathbf{R}} = \text{Log}((\Delta \mathbf{R}_{LSTM}(\tilde{S}_{t-1}))^T \Delta \mathbf{R}_{VIO}) \quad (9)$$

$$e_{\mathbf{p}} = \Delta \mathbf{p}_{VIO} - \Delta \mathbf{p}_{LSTM}(\tilde{S}_{t-1}) \quad (10)$$

where \tilde{S}_{t-1} is the best estimate of IMU status at time $t - 1$. ρ is the Huber robust cost function, Σ_I is the information matrix of the pre-integration. $e_{\mathbf{R}}$ and $e_{\mathbf{p}}$ are the error of rotation and translation between IMU-se3 and VIO-se3. $\Delta \mathbf{R}_{LSTM}$ and $\Delta \mathbf{R}_{VIO}$ are the rotations calculated by LSTM-IMU network and FC-Fusion network from time $t - 1$ to t . $\Delta \mathbf{p}_{LSTM}$ and $\Delta \mathbf{p}_{VIO}$ are the translations by LSTM-IMU network and FC-Fusion network from time $t - 1$ to t .

Thereby it updates the state of the IMU every time when it receives the feedback from FC-Fusion, which is highly similar to IMU status updating in traditional tightly-coupled VIO methods.

D. FC-Fusion Network

The combination of OFF and IMU-se3 is fed into the FC-Fusion network, which has 5 fully connected layers. Let fA-B-R denote a fully connected ReLU layer from A to B. dk denotes a dropout and the probability of an element to be zeroed is k. The network with 5 blocks consists of:

d0.5,f6+1024x6x20-4096-R,d0.5,4096-1024-R,f1024-128-R,f128-32-R,f32-6.

FC-Fusion is similar to multisensor fusion function in the traditional method. The output of FC-Fusion is named as VIO-se3 whose dimension is 6. Therefore, the VO trajectory of a camera over a period of time is calculated by the integration of VIO-se3. Note that, Stereo-se3 is also employed to constraint VIO-se3.

E. DeepVIO Loss

1) *CNN-Flow Loss*: The loss for CNN-Flow is the differences between its output and synthetic 2D optical flow within all pixels, i.e.

$$L_{\text{Flow}} = \sum_{x,y} \text{EPE}(\mathbf{v}_{2D}(x,y), \mathbf{v}_{\text{CNN-Flow}}(x,y)) \quad (11)$$

where $\vec{V}_{\text{CNN-Flow}}(x,y)$ is 2D optical flow from the CNN-Flow network. L_{Flow} is the summation of the endpoint error (EPE), which is the standard error used for optical flow estimation [31].

2) *IMU Loss*: The loss of the LSTM-IMU network is shown as follows:

$$L_{\text{IMU}} = \Sigma(\|\omega - \hat{\omega}\| + \beta\|v - \hat{v}\|) \quad (12)$$

where ω and v are the camera motion true value given by Stereo-se3, $\hat{\omega}$ and \hat{v} are the camera motion estimation given by LSTM-IMU. β is the additional scale factor for balancing the translation and quaternion elements.

3) *Fusion Loss*: The loss of the FC-Fusion is shown as follows:

$$L_{\text{VIO}} = \Sigma(\|\omega - \hat{\omega}_1\| + \beta'\|v - \hat{v}_1\|) \quad (13)$$

Where ω and v are the quaternion and translation given by stereo-se3, while $\hat{\omega}_1$ and \hat{v}_1 are estimations given by FC-Fusion. β' is the additional scale factor for balancing the translation and quaternion elements.

For the entire DeepVIO network, the total loss consists of three parts as follows:

$$L = L_{\text{Flow}} + L_{\text{IMU}} + L_{\text{VIO}} \quad (14)$$

IV. EXPERIMENTAL RESULTS

In this section, we evaluate our proposed DeepVIO in comparison to the state-of-the-art algorithms on both indoor and outdoor datasets followed by detailed analysis.

A. Dataset

KITTI dataset: It consists of 389 pairs of stereo images and optical flow maps, 39.2 km visual ranging sequences, a Velodyne laser scanner and a GPS/IMU localization unit, sampled and synchronized at 10Hz. The odometry benchmark consists of 22 stereo sequences, saved in loss less png format. It provides 11 sequences (00-10) with ground truth trajectories for training and 11 sequences (11-21) without ground truth for evaluation. We download the pretrained PSMNet model from this link at https://drive.google.com/file/d/1p4eJ2xDzvQxaqB20A_MmSP9-KORBx1pZ/view.

EuRoC dataset: It contains 11 sequences recorded from a micro aerial vehicle (MAV), flying around two different rooms and an industrial environment. The dataset provides synchronized global shutter WVGA stereo images at 20Hz with IMU measurements at 200Hz and trajectory ground truth. The pretrained model of PSMNet is trained by the rectified frames. In detail, it selects 200 training rectified stereo image pairs with the ground truth disparities obtained using an EuRoC tools (https://github.com/ethz-asl/volumetric_mapping). Image size is 752x480. The whole training data is further divided into a training set (80%) and a validation set (20%).

B. Network Training

For the KITTI dataset, the image is rectified and resized to 640x192 and IMU input data is 12x1. In the model parameters, the batch size is 32. The epoch is 200 and the optimization function is Adam. In our experiment, sequences 00-08 are used for training and 09-10 are used for testing. Note that sequence 03 is abandoned since its IMU data is

not available in KITTI Raw Data, In addition, 5% of KITTI sequences 00-08 are selected as a validation set, which is the same to [12].

For the EuRoC dataset, the image is 640×480 and IMU input data is 12×10 . The batch size, epoch and optimization function are the same to KITTI. We use the same training and test sequences as [9]. The computer graphics card used for training is equipped with Nvidia GeForce GTX1080 Ti with 12G memory.

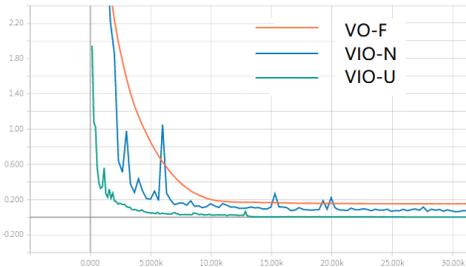


Fig. 5. Comparison of training losses using our models on KITTI.

Fig. 5 shows training losses for KITTI using three kinds of networks: red curve is the loss of VO-F trained only with LSTM-IMU output vector OFF, blue curve is the loss of VIO-N trained without IMU status update module, and green curve is the loss of VIO-U trained with IMU status update module. We can observe that VIO-U converges faster and has lower errors compared to others. The quantification will be shown in the section IV.E.

C. Comparison of the Trajectories

We compare the performance of DeepVIO with other state-of-the-art learning-based and traditional VO or VIO baselines that are DeepVO, VIOlearner, ORB-SLAM-M (without loop closure) and VINS.

Table I shows the results in KITTI test. In order to facilitate comparison, we follow the metric provided by [12], where $t_{rel}(\%)$ is the average translational error percentage on lengths 100 m - 800 m and $r_{rel}(\circ)$ is the rotational error ($\circ/100m$) on lengths 100m - 800m. It is observed that ours outperforms existing methods in terms of trajectory accuracy. In particular, the errors with traditional VINS are extremely worse than us. It is concluded that these traditional VIO methods are not good results for trajectory estimation when the original raw IMU data collected without tight synchronization and the IMU rate is too low in KITTI dataset. A further experiment will be presented to analyze the performance under some extreme cases between a traditional method and ours in section IV.F. On the other hand, it can be seen that our prediction accuracy outperforms the VIOlearner except some cases. The reason is that VIOlearner just directly uses the raw IMU data to infer the trajectory without considering the change of IMU status. In contrast, our IMU status is corrected all the time when the update module receives the feedback from VIO-se3 and IMU-se3. Note that, our method is able to generalize well to unseen scenarios, e.g. seq 09 and seq 10.

We follow the test instructions [8] designed for EuRoC, which reports the results on MH04 and MH05 as shown in Table II. It shows that DeepVIO outperforms DeepVO because of combining IMU data with a camera in our cases. Our performances are slightly worse than VINS since the latter boosts via the high quality of IMU data (e.g. high frequency and good synchronization). In addition, ORB-SLAM-M also outperforms ours especially on translation accuracy. The reason is that our method lacks a module of the local bundle adjustment optimization. The average time of DeepVIO inference are approximate 7.81ms per frame on KITTI and 3.9ms per frame on EUROC, respectively. Fig. 6 shows the comparison of the trajectories on KITTI 09 and EuRoC MH04.

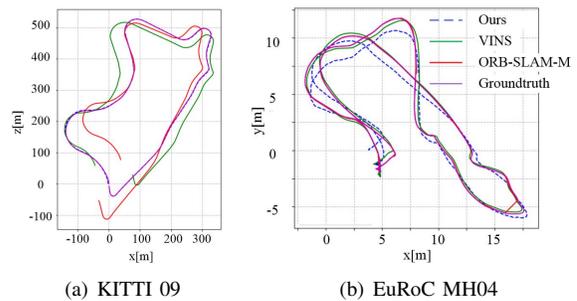


Fig. 6. Trajectories of Ours and existing approaches testing results.

TABLE I
COMPARISON OF TRAJECTORIES ESTIMATION PERFORMANCE WITH EXISTING APPROACHES ON KITTI DATASET.

Seq	Ours		VIOlearner		ORB-SLAM-M		VINS	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
00	11.62	2.45	14.27	5.29	25.29	7.37	18.83	2.49
02	4.52	1.44	4.07	1.48	26.30	3.10	21.03	2.61
05	2.86	2.32	3.00	1.40	26.01	10.62	21.90	2.72
07	2.71	1.66	3.60	2.06	24.53	10.83	15.39	2.42
08	2.13	1.02	2.93	1.32	32.40	12.13	32.66	3.09
09	1.38	1.12	1.51	0.90	45.52	3.10	41.47	2.41
10	0.85	1.03	2.04	1.37	6.39	3.20	20.35	2.73
Average	3.72	1.58	4.49	1.97	25.72	7.19	21.61	2.64

TABLE II
COMPARISON OF TRAJECTORIES ESTIMATION PERFORMANCE WITH EXISTING APPROACHES ON EUROC DATASET.

Seq	Ours		DeepVO		ORB-SLAM-M		VINS	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
MH04	0.69	0.76	≈ 5	≈ 30	0.06	1.76	0.34	0.87
MH05	0.52	0.81	≈ 6	≈ 35	0.05	1.56	0.29	0.69
Average	0.61	0.79	≈ 5.5	≈ 32.5	0.06	1.66	0.32	0.78

D. Optical Flow Error

In order to show our 3D optical flow has the potential for supervising DeepVIO. As described in section III.A, we can calculate the synthetic 2D optical flow from 3D one (see

Fig. 7(c)) according to Eq.(7). Meanwhile, we also calculate 2D optical flow via the original FlownetC network, as shown in Fig. 7(d). Then, both of them are compared to the ground truth (see Fig. 7(b)) provided by KITTI, and the results are visualized in Fig. 7(e) and (f). It is worth to mention that Flownet2 suffer from aperture ambiguity especially in the textureless region, as marked using blue rectangle in Fig. 7(e) and (f). Another advantage is that more dense 2D optical flow is synthesized compared to the sparse one provided by raw lidar data, which is shown as invalid region (e.g. pixel white) in the Fig. 7(b).

To quantify the results, we calculate the average errors of 2D optical flow on KITTI dataset in 50 pairs with non-dynamic targets and 20 pairs with dynamic targets, as shown in Table III. It can be seen that in the scene without dynamic target, the error value of 3D optical flow outperforms Flownet2 which is widely used in learning based VO methods. Compared to Flownet2 with dynamic target (see Fig. 7 (d)), the impact of dynamic targets is eliminated when calculating the motion between two point clouds. Thus, the region of the dynamic target can be removed and the rest of them are used to train our highly accurate DeepVIO. For instance, Fig. 7(a) shows a black car on the left side is moving forward, and the rectangle of Fig. 7 (c) shows our 2D optical flow marks the dynamic region as 0.

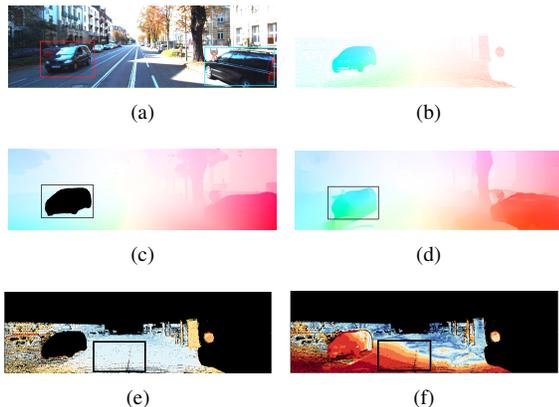


Fig. 7. Comparison of synthetic 2D optical flow and normal 2D optical flow. (a) is the left image, (b) is its ground truth of 2D optical flow from KITTI, (c) is the synthetic 2D optical flow, (d) is the 2D optical flow obtained from Flownet2 and optical flow errors of them, e.g., (e) and (f). Red rectangle is a dynamic object and the black one is textureless road.

TABLE III
ERRORS (IN PIXELS) OF 2D OPTICAL FLOW ON KITTI DATASET

KITTI Dataset	3D optical flow	Flownet2
Non-dynamic targets	4.12	5.63
Dynamic targets	6.31	13.62

E. Comparison of DeepVIO Network

Here, we quantify the performances of the three networks (e.g. VO-F, VIO-N and VIO-U) on KITTI sequence 10.

In Fig. 8, the error of the VO-F is the largest since only visual feature is fed into CNNs. The error of VIO-N is also

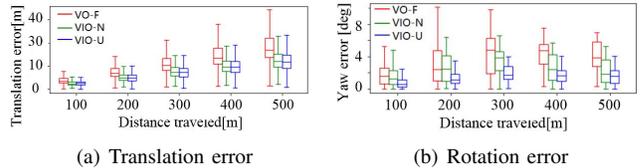


Fig. 8. Translation and orientation errors on the KITTI 10.

worse than VIO-U since the former is significantly affected by the changing status in IMU data. Especially, the faster the camera rotates, the more serious the error is. It can be concluded that our IMU update module improves the location accuracy even when it is rotating rapidly, which is similar to the traditional VIO method.

F. Comparison with Traditional VIO Methods

We conduct four comparative experiments on VIO calibration, sensor synchronization, camera and IMU dropped frames. Table IV indicates that DeepVIO has a better accuracy to these tests compared with the traditional VINS. More specifically, we follow VINET[10] to add camera-IMU calibration errors and trained DeepVIO using the mis-calibrated data (e.g. Mis-c:10° means 10° added into the rotation matrix). It shows that our method is robust to calibration errors, while VINS always has lower accuracy due to the inaccurate calibration parameters. In terms of synchronization, we randomly add 20ms to the IMU timestamp (e.g. Unsyn:20ms). As result, VINS seemed to be confused by the unsynchronized IMU data with a lower accuracy compared to us. The fact is that our VIO system degenerates to VO system when encountering these two situations. As mentioned above, VINS degenerated seriously when the IMU frame rate is low (e.g. IMU-D:90% means dropped rate is 90%). However, in DeepVIO, vision stream pays a greater role in the trajectory inference once the IMU data is missing. Finally, we random drop some frames from image sequences (e.g. Cam-D:50% means dropped rate is 50%) and show that VINS also had a worse performance in rotation. While our IMU stream still work well to compensate the trajectory estimation with less image alignments.

TABLE IV
ROBUSTNESS OF THE DEEPVIO ON SEVERAL CHALLENGING TESTS

	Mis-c: 10°		Unsyn: 20ms		IMU-D: 90%		Cam-D: 50%	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
MH05								
Ours	0.77	1.11	0.59	1.02	0.75	0.93	0.89	1.06
VINS	1.38	2.88	0.65	2.93	0.88	2.94	0.37	3.02

V. CONCLUSIONS

This paper provided an DeepVIO to estimate the absolute trajectory of a visual-inertial sensor from stereo sequences in a self-supervised end-to-end strategy. It has been proved that the 3D geometric constraints containing 3D optical flow

and 6-DoF pose play a very important role in optimizing the CNN-Flow, LSTM-IMU and FC-Fusion network. It is worth to mention that our 3D optical flow was able to eliminate the impacts of dynamic targets and textureless in the scene. Furthermore, the IMU status update scheme reduced the noise from IMU device so that it improved the LSTM-IMU inference accuracy. The experiments in KITTI and EuRoC indicated that our DeepVIO outperforms other state-of-the-art learning-based VO or VIO system in terms of pose accuracy, and generalizes well to scenarios completely unseen. Compared to traditional VIO system, DeepVIO did not need the tightly intrinsic and extrinsic parameters of the camera and IMU, which were inconvenient to obtain in VIO calibration, and had an acceptable VIO result even with poor data, such as unsynchronized and missing ones.

One limitation is that drift problem still exists in our system since the absence of place recognition and relocalization pipelines. In future, we will plan to extend its ability of loop-closure detection and global relocalization similar to traditional visual SLAM.

REFERENCES

- [1] Friedrich Fraundorfer and Davide Scaramuzza. Visual odometry: Part ii: Matching, robustness, optimization, and applications. *IEEE Robotics & Automation Magazine*, 19(2):78–90, 2012.
- [2] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.
- [3] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *arXiv preprint arXiv:1607.02565*, 2016.
- [4] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.
- [5] Ke Sun, Kartik Mohta, Bernd Pfrommer, Michael Watterson, Sikang Liu, Yash Mulgaonkar, Camillo J Taylor, and Vijay Kumar. Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robotics and Automation Letters*, 3(2):965–972, 2018.
- [6] Raul Mur-Artal and Juan Domingo Tardos. Visual-inertial monocular slam with map reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803, 2016.
- [7] Qin Tong, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, PP(99):1–17, 2017.
- [8] 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.
- [9] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. *The International Journal of Robotics Research*, 37(4-5):513–542, 2018.
- [10] Ronald Clark, Sen Wang, Hongkai Wen, Andrew Markham, and Niki Trigoni. Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [11] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018.
- [12] E Jared Shmwell, Sarah Leung, and William D Nothwang. Vision-aided absolute trajectory estimation using an unsupervised deep network with online error correction. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2524–2531. IEEE, 2018.
- [13] Anastasios I Mourikis and Stergios I Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3565–3572. IEEE, 2007.
- [14] 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. IEEE, 2015.
- [15] Stefan Leutenegger, Paul Furgale, Vincent Rabaud, Margarita Chli, Kurt Konolige, and Roland Siegwart. Keyframe-based visual-inertial slam using nonlinear optimization. *Proceedings of Robotics Science and Systems (RSS) 2013*, 2013.
- [16] Ruihao Li, Sen Wang, and Dongbing Gu. Ongoing evolution of visual slam from geometry to deep learning: Challenges and opportunities. *Cognitive Computation*, 10(6):875–889, 2018.
- [17] Jason R Rambach, Aditya Tewari, Alain Pagani, and Didier Stricker. Learning to fuse: A deep learning approach to visual-inertial camera pose estimation. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 71–76. IEEE, 2016.
- [18] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016.
- [19] Clément Godard, Oisín Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017.
- [20] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017.
- [21] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. *arXiv preprint arXiv:1811.06152*, 2018.
- [22] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 36–53, 2018.
- [23] Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7286–7291. IEEE, 2018.
- [24] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 340–349, 2018.
- [25] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Gopcp: A globally optimal solution to 3d icp point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2241–2254, 2016.
- [26] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.
- [27] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018.
- [28] Zhao Cheng, Sun Li, Pulak Purkait, Tom Duckett, and Rustam Stolkin. Learning monocular visual odometry with dense 3d mapping from dense 3d flow. 2018.
- [29] Andreas Wedel, Clemens Rabe, Tobi Vaudrey, Thomas Brox, Uwe Franke, and Daniel Cremers. Efficient dense scene flow from sparse or dense stereo data. In *European conference on computer vision*, pages 739–751. Springer, 2008.
- [30] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.
- [31] Eddy Ilg, Nikolaus Mayer, Tommo Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2462–2470, 2017.

- [32] 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, 2015.