LightVO: Lightweight Inertial-Assisted Monocular Visual Odometry with Dense Neural Networks

Zibin Guo, Mingkun Yang, Ninghao Chen, Zhuoling Xiao*,Bo Yan, Shuisheng Lin and Liang Zhou
School of Communication Information Engineering, University of Electronic Science and Technology of China
Email:zibin_guo@foxmail.com, YmkC1996@163.com, ninghaochen@std.uestc.edu.cn,
{zhuolingxiao, yanboyu, sslin, zlzl}@uestc.edu.cn

***Abstract*—Monocular visual odometry (VO) is one of the most practical ways in vehicle autonomous positioning, through which a vehicle can automatically locate itself in a completely unknown environment. Although some existing VO algorithms have proved the superiority, they usually need another precise adjustment to operate well when using a different camera or in different environments. The existing VO methods based on deep learning require few manual calibration, but most of them occupy a tremendous amount of computing resources and cannot realize real-time VO. We propose a highly real-time VO system based on the optical flow and DenseNet structure accompanied with the inertial measurement unit (IMU). It cascade the optical flow network and DenseNet structure to calculate the translation and rotation, using the calculated information and IMU for construction and self-correction of the map. We have verified its computational complexity and performance on the KITTI dataset. The experiments have shown that the proposed system only requires less than 50% computation power than the main stream deep learning VO. It can also achieve 30% higher translation accuracy as well.**

***Index Terms*—image sequences, visual odometry, neural network, IMU**

I. Introduction

In the development of unmanned vehicles and intelligent robots, it is important for vehicles and robots to autonomously locate and build real-time maps in an unknown environment. As an autonomous positioning solution, visual odometry can provide the required pose information for unmanned vehicles and intelligent robots in an unknown environment. This paper proposes an inertial assisted visual odometry scheme based on deep learning.

In traditional SLAM algorithm, some existing algorithms such as ORB-SLAM2 [1] have achieved relatively high precision. However, these methods rely on optimization and loop closure detection techniques that the vehicle or robot has to reach the passed location on the map to correct the current pose and eliminate accumulated errors. For systems that only consider frame-to-frame estimation, these methods can not work well. There is also a type of VO methods called optical flow method, such as VISO2 [2], which is a high-precision method for estimating the motion of a carrier on the basis of the dense optical flow of two frames. But the optical flow method has a heavy computing burden, it is difficult to apply to scenes that require real-time performance. Peter M. Muller [3] proposed an VO method based on optical flow generated from Flownet. Experiments show that this method has higher real-time performance than the existing optical flow based VO systems. However,it is hard to meet the demand of real-time processing.

LightVO proposed in this research uses the TVNet proposed by Lijie Fan [4], which is an optical flow extraction algorithm based on convolutional neural network. It uses neural network to realize the TV-L1 [5] optical flow extraction algorithm. It has faster running speed than FlownteS. At the same time, we select the Densenet [6] structure and make a modification for pose estimation, which achieves a great reduction of parameter amount and running time. The translation accuracy is even better than Flowdometry [3]. Neural network can learn the characteristics of translation better [7], but a deficiency in the estimation of rotation. In this case, we propose the use of IMU information to correct the VO results. Experiments show that our algorithm has better real-time performance and increases the accuracy of translation estimation.

The proposed method is more than twice as fast as Flowdometry. Our VO solution takes 47ms and Flowdometry takes 103ms in the case of a single Titan XP GPU acceleration. When concerning to accuracy, our VO scheme achieved a translation error of 7.49% on the KITTI dataset [8] and Flowdometry of 10.77%. Our rotation error is slightly more, but the additional IMU corrected program, with few computing resources(0.16ms on 3.20GHz CPU), fixes the flaw and our rotation get good results. Another advantage of our system is that it requires very low data processing rates. Compared with the mainstream visual inertial schemes such as [9], [10], our scheme greatly reduces the data processing rates. In summary, this paper's main contributions are:

- Lightweight visual odometry: The proposed Network enables computational efficiency and real-time frame-to-frame pose estimate.
- A higher precision translation estimate: We achieve the precision of translation estimate about 30% higher than other real-time method.
- Lightweight IMU correction: We implement an inertial correction scheme with very low computational cost in our frame-to-frame pose estimate.

The remainder of this paper is organized as follows: Sec. II overviews existing techniques. Sec. III introduces our frame-to-frame pose estimate systems. Sec. IV extensively evaluates our visual odometry method and compares it with existing

978-1-7281-0962-6/19/$31.00 ©2019 IEEE

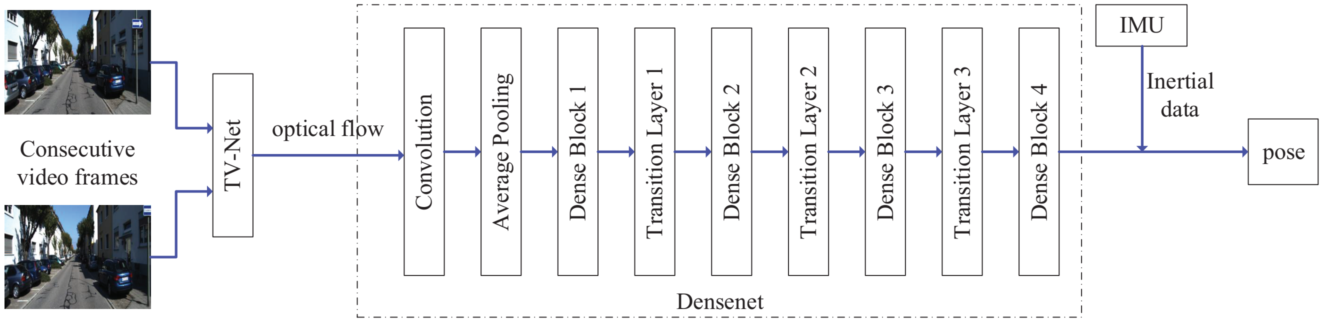


Fig. 1. The architecture of LightVO system

techniques. Sec. V concludes the paper and discusses ideas for future work.

II. Related Work

The present schemes can be divided into three categories: sparse feature based methods, optical flow based methods (direct methods), and deep learning based methods.

A. Sparse Feature based Methods

The VO based on sparse feature method is the current mainstream method [11]. This method extracts the feature points of the image, then performs feature point matching, and uses the matched feature point pairs combined with the camera internal parameters to estimate the pose transformation between the two frames. ORB-SLAM2 [1] is an example of this method. The VO system using the feature method is insensitive to light and image noise, and the operation is relatively stable, but the more it enhances the robustness, the more complex the feature point description and the algorithm is. In addition, its application is limited by the scene and is not suitable for application in scenes with missing features.

B. Direct Methods

The direct method based on the assumption of pixel gray invariance for estimating camera motion has developed rapidly in the past few years [12], [13]. Direct method developed from optical flow [14], which can estimate camera motion by minimizing photometric error without mentioning features from pixel information. The problems faced by the feature point method can be effectively solved. Large-scale direct monocular Simultaneous Location and Mapping (LSD-SLAM) algorithm proposed by Engel et al. [15] is the method employed, it uses a new depth estimation mechanism called sliding window optimization instead of the original Kalman Filter method used by DSO [13] (Direct Sparse Odometry). The direct method can be applied to scenarios that require the construction of semi-dense or dense maps, which is not possible to extract feature points. However, the direct method also has problems such as non-convexity, single pixel no discrimination, and farfetched gray-scale invariance hypothesis. It is only suitable for situations where the motion is small and the overall brightness of the image does not change much.

In order to solve the problem of lacking depth information of monocular VO, the researchers use the idea of combining monocular camera and inertial navigation to achieve better results. The current mainstream fusion schemes have a filtering-based approach and a nonlinear optimization-based approach. OKVIS [9] is an example of the former and VINS-Mono [10] is the latter. These schemes have high requirements on calculation frequency and real-time performance. Once there is a problem in calculation time, the stability of the system will be greatly reduced.

C. Deep Learning based Methods

The neural network method has been applied to many fields recently, VO field is no exception, and has satisfactory achievement. Konda et al. [16] first implement DL-based VO by extracting visual motion and depth information. The change in camera speeds and direction are predicted by the softmax function using a convolutional neural network (CNN). Kendall [17] use CNN to implement an end-to-end positioning system with RGB image as the input and camera pose as the output. Costante et al. [18] replace the RGB image with a dense optical stream as the input to the CNN. The system designs three different CNN architectures for VO feature learning, which realizes the robustness of the algorithm under the conditions of image blur and underexposure.

In recent years, many excellent VO algorithms based on deep learning are proposed. Wang et al. [19] propose a new end-to-end monocular VIO framework based on RCNN using deep recursive convolutional neural networks. Their experiment on KITTI VO dataset shows that the performance of their algorithm is comparable to the most advanced visual-inertial odometry (VIO) methods available today. GeoNet [20] divides the stationary object from the moving object, thus a new cascade structure consisting of two stages is designed to adaptively solve the rigid flow and object motion of the scene. Using unstructured video sequences as input, Xu et al. [21] propose an unsupervised learning framework for monocular depth and camera motion estimation. This method is completely unsupervised and requires only monocular video sequences for training.

Muller [3] puts forward a VO method based on optical flow and depth learning. It extracts optical flow as the input of CNN to calculate the rotation and translation and obtain the estimated result of VO. Experiments show that this method has higher real-time performance than the existing VO system. However, these solutions do not guarantee real-time performance due to the large amount of time required to extract the optical stream or the enormous network, even with a GPU for acceleration.

Benefiting from the research of Lijie Fan [4] and Gao Huang [7], we propose a new solution based on optical flow and deep learning method to solve the real-time problem of visual odometry.

III. System architecture

Our algorithm is an inertial-assisted visual odometry system based on deep learning that involves four steps: 1) preprocess two images using TVNet to calculate optical flow; 2) obtain the frame-to-frame pose estimation using optical flow as the input of the frame-to-frame estimation network; 3) generate the motion map from the cumulative estimation; 4) correct the map to make up the VO deficiency using the IMU information. The system is shown in Fig. 1.

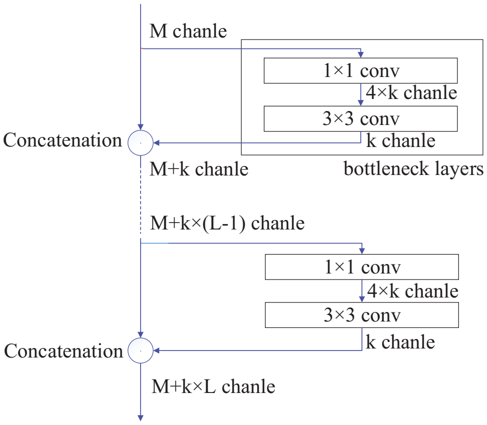


Fig. 2. The architecture of Dense Block

A. Optical Flow Calculation

Traditional optical flow algorithms for high-precision VO are in widespread used, while they can not satisfy VO real-time requirement for relying heavily on computing resources. Recently, the Flownet 2.0 [22] which uses the neural network to calculate optical flow has greatly improved the arithmetic speed, but it is still unfeasible for generic camera sampling frequency. Moreover, it is difficult to cascade the Flownet 2.0 and our network into a system to estimate translation and rotation. According to the paper [4], we find that TVNet performs well in controlling the quantity of parameters and calculation. Above all, we can assemble an end-to-end system by cascading TVNet with our network.

B. Network Structure

We take much superiority over DenseNet into consideration, for instance, enhanced feature propagation, sufficient feature reuse, minor parameter quantity. Therefore, we adopt Dense connection to solve VO problems.

As shown in Fig. 1 which is our network structure, optical flow inputs a 7×7 convolution layer firstly for its large receptive field and an average pooling layer is adjacent to reduce the data size.

Then data will be processed through four dense blocks and three transition layers alternately. We give a detailed description of the dense block in Fig. 2 to indicate that the bottleneck layer is the smallest unit in our block. We design bottleneck layers in each dense block in the way shown in Table I. Every bottleneck layer contains two convolutional layers in sizes of 1×1 and 3×3 respectively. Note that 1×1 kernel can effectively reduce the data dimension to 4K which makes our net with less computation. Then, we give the function of input for i-th bottleneck layer:

$$x_i = H_i([x_0, x_1, \cdots, x_{i-1}])$$

Among them, $[x_0, x_1, \cdots, x_{i-1}]$ represents the feature maps of layer $0, 1, \cdots, i-1$, and H means to concatenate all of these previous tensors. In addition, we add k feature maps amongst each bottleneck layer. Similarly, each transition layer consists of 1×1 convolution kernel and 2×2 pooling layer and compression ratio θ is designed to reduce the feature map dimension. In the end, we connect the last dense block directly to a fully connected layer network contains 1024 hidden units to get frame-to-frame translation and rotation estimation as our final outputs.

TABLE I
Number Of Bottleneck Layers In Dense Block

Layers	Number of bottleneck layers
Dense Block1	L=4
Dense Block2	L=6
Dense Block3	L=8
Dense Block4	L=8

C. Motion Map Generation

The trajectory of VO we plot with ground truth map is a visualization of our results based on KITTI VO/SLAM benchmark [8]. On account of the high sampling frequency of camera which results in smaller movement of the car in a single sampling period, we assume the movement of the car is linear and ignore the influence of the height. Thus, we come up with a simple solution to recover the pose and generate a motion map.

D. IMU Correction

The insensitivity of rotation of our neural network brings about cumulative estimation error in angle which causes great impact on the trajectories reconstruction. In order to reduce the angle error, we combine IMU data with our LightVO using Kalman Filter (KF) under the approximation that VO system in linear variation. Moreover, we pre-integrate IMU data in each sampling period which overlooks the data loss of IMU and sufficiently decreases calculation. The drawback of inertial navigation, as is well-known, is cumulative error caused by quadratic integral error. We set the IMU translation penalty to the observed noise which accumulates with time as shown in Algorithm1.

Algorithm 1 IMU Correction

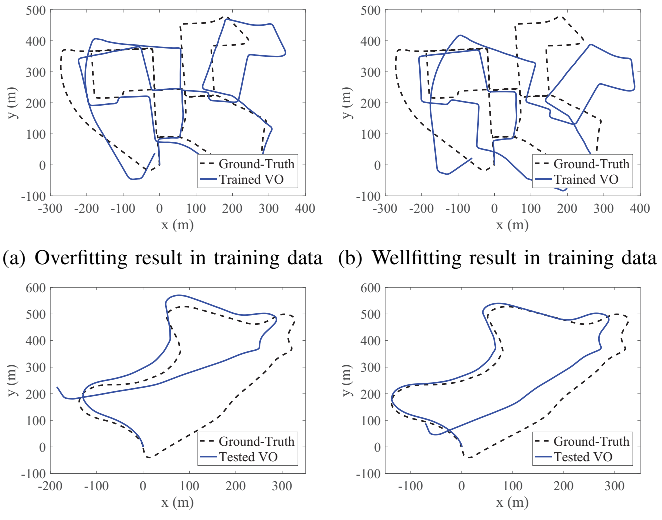
```
Input:
    Visual relative translation vector, $T_{rv}$;
    Visual relative rotation vector, $R_{rv}$;
    IMU relative translation vector, $T_{ri}$;
    IMU relative rotation vector, $R_{ri}$;
    Initial translation, $T_{vi0}$;
    Initial rotation, $R_{vi0}$;
    Kalman filter state, $x$;
    Observation state, $\mu_i$;
    Sequence length, $L$;
    Variance of VO relative vector, $R$;
    Variance of IMU result, $Q$;
Output:
    Corrected translation and rotation matrix, $[T_{vi}, R_{vi}]$;
$x_0 = [T_{vi0}, R_{vi0}]$;
for $k$ in $[1, L]$ do
    $\mu_k = x_{k-1} + [T_{rv}, R_{rv}]$;
    $S = r_{k-1} + R$;
    $Q_{tk} = Q_{t(k-1)} + a \times (b \times k)^c$; (Set the IMU translation penalty)
    $Q_{rk} = Q_{r(k-1)}$;
    $Q_k = \begin{bmatrix} Q_{tk} & 0 \\ 0 & Q_{rk} \end{bmatrix}$;
    $\mu_{ik} = x_{k-1} + [T_{ri}, R_{ri}]$;
    $K = S/(S + Q_t)$;
    $x_k = \mu_k + K \times (\mu_{ik} - \mu_k)$;
    $r_k = I - K \times S$;
end for
return $x$;
```

IV. Experimental Results

In order to demonstrate the performance of our method, our odometry method is evaluated in the well-known KITTI VO/SLAM benchmark [8]. We choose the open source monocular visual odometry scheme VISO2-M [2], the deep learning P-CNN [17] scheme with good effect and the real-time deep learning VO scheme Flowdometry [3] as comparisons, the superiority of our proposed solution has been verified.

A. Dataset

The KITTI VO/SLAM [8] is one of the most widely used benchmarks which contains 22 sequences (00-21) of images. Sequences 00-10 have ground truth measured and calibrated by GPS, while the others only provide raw images. All of these sequences are collected by a 10fps frame rate camera carried by a car which drives in urban area. Therefore, KITTI VO/SLAM benchmark that includes various real-world scenes has enormous practical significance. We regard sequences 00-07 as training set and 08-10 as testing set.



(a) Overfitting result in training data (b) Wellfitting result in training data

(c) Overfitting result in testing data (d) Wellfitting result in testing data

Fig. 3. Overfitting effect of VO results. (a) and (b) show the VO estimate on training data (Seq.00) in overfitting and wellfitting,respectively. (c) and (d) show the VO estimate on test data (Seq.09) in overfitting and wellfitting.

B. Training

Note that we do not put images directly into neural network but regard the optical flow of two adjacent frames which is extracted from TVNet [4] as the input of the network. And labels which are the translation and rotation between every two frames are calculated from the rotation matrix provided by dataset.

Hardware in our training process is Nvidia Geforce Titan XP GPU. We select Adma optimizer with 1×10^{-4} initial learning rate and set batch size to 12. Underfitting and overfitting, as we know, are two major problems causing poor results in deep learning algorithm. In our experiment, overfitting seriously restricts our model performance. Nevertheless, there is no sufficient solution to overfitting problems in VO field. In this paper, we adopt dropout and early stopping to solve this problem. Comparing the results of (a) and (b), (c) and (d) in Fig. 3, we can readily find the impact of overfitting to the trajectory reconstruction. It also refers that overfitting may cause severe loss of pose estimation in an unknown environment from (a) and (c).

TABLE II
Running Time Compared With Other Methods

	Parameter	Optical Flow Calculation [s/fram]	Odometry Calculation [s/fram]	Total Execution [s/fram]
LightVO	14M	0.039	0.008	0.047
Flowdometry[3]	50M	0.08	0.023	0.103

C. VO Results

We select Flowdometry which is proposed in literature [3] as comparative item on account of its better performance than other optical flow schemes for computation speed. Table II shows the running time of our scheme compered with Flowdometry in the same hardware (i.e a single Nvidia Geforce Titan XP GPU). Evidently, LightVO greatly reduces the quantity of parameters and calculation.

Then we compare our results with other models in [2], [3] and [17] (i.e Flowdometry, P-CNN, VISO2-M) in translation and rotation mean error. Fig. 4 indicates that our model exceeds others in translation accuracy but performs the worst in rotation accuracy. We attribute this phenomenon to three facts as follows. First, we only take one axis of rotation into consideration referring to Flowdometry. Besides, rotation samples are limited which makes network could not predict well. Above all, to achieve real-time aim, we have greatly reduced the amount of parameters but sacrificed the accuracy of rotation.

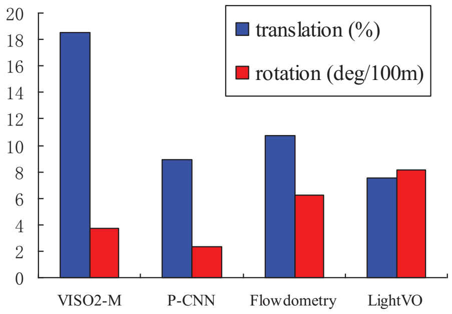


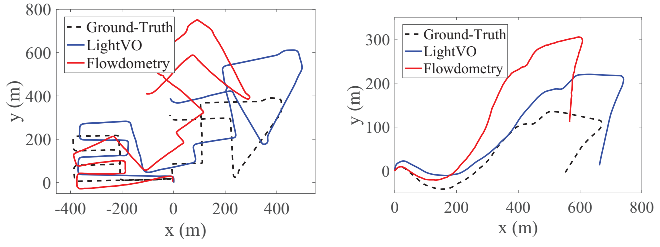
Fig. 4. Translation error and rotation error of the VO result

The robustness of an algorithm is important. Hence we experiment with sequence 08, 09, 10 of KITTI dataset whose collecting environment is different from our training set. Table III shows the results of comparison, we find that traditional dense optical flow method VISO2 and image-based deep learning method P-CNN have a wide fluctuation in sequence 10 while the optical-flow-based deep learning methods (i.e our LightVO, Flowdometry) have a relatively stable performance. It reveals that method such as this can retain high precision and overcome the shortcoming of dense optical flow which is sensitive to illumination by deep learning.

TABLE III
LightVO Result Compered With Other Works

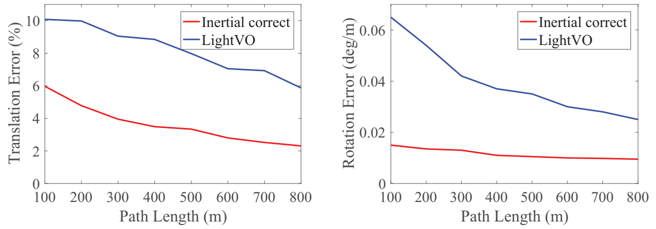
	VISO2-M [2]		P-CNN [17]		Flowdometry [3]		LightVO	
Seq	Trans [%]	Rot [deg/m]	Trans [%]	Rot [deg/m]	Trans [%]	Rot [deg/m]	Trans [%]	Rot [deg/m]
08	19.39	0.0393	7.60	0.0187	9.98	0.0544	6.86	0.0838
09	9.26	0.0279	6.75	0.0252	12.64	0.0804	5.16	0.0675
10	27.55	0.0409	21.23	0.0405	11.56	0.0728	14.01	0.0983
Avg	18.55	0.0376	8.96	0.0235	10.77	0.0623	7.49	0.0813

In Fig. 5 we plot the trajectory reconstruction of sequence 08 and 10 by our LightVO compared with Flowdometry and ground truth. It infers that our LightVO have a relatively stable performance. However, the accumulation error of angle, over time, deviate our trajectory from the ground truth. We think that mainly owing to the lack of rotation information in training set. In urban area, driving straight dataset is certainly more than turning a corner. Thus, the information of tuning data is obviously insufficient.



(a) Trajectories of sequence 08 (b) Trajectories of sequence 10

Fig. 5. Trajectories of VO testing results on Sequence 08,10 compared with Flowdometry



(a) Translation against path length on IMU correct result (b) Rotation against path length on IMU correct result

Fig. 6. IMU-assist VO average errors on translation and rotation against different path lengths

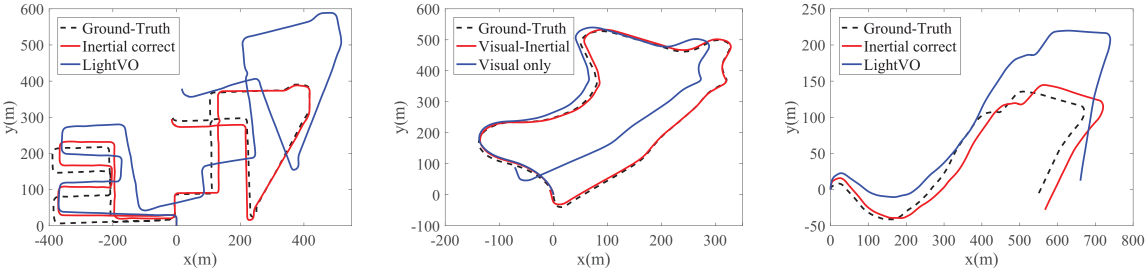
D. Corrected By IMU

Based on the evaluation method in [23], we calculate and plot the average root mean square error of translation and rotation error (RMSE) in Fig. 6. It reveals that error is amplified at a short distance, while error tends to decrease as mileage increases. We attribute this law to deficiency of training data in our training, specifically, our network never learns the condition it meets in test process. It also demonstrates that the estimation error of rotation can be significantly reduced by importing data from IMU. By contrast, there is no obvious change in translation error assembling IMU. It because the accumulative error of accelerometer makes the translation estimation from inertial navigation untrusted.

TABLE IV
IMU Correct Result

	LightVO		Corrected by IMU	
Seq	Trans[%]	Rot[deg/m]	Trans[%]	Rot[deg/m]
08	6.86	0.0838	2.04	0.0247
09	5.16	0.0675	1.98	0.0131
10	14.01	0.0983	3.72	0.0273
Avg	7.49	0.0813	2.28	0.0217

Table IV proves the assistance of IMU in a quantitative way. We can intuitively find the addition of IMU indeed correct the deficiency of LightVO in rotation estimation. It can be seen more intuitively from Fig. 7 that our correction to rotation has a macroscopic optimization effect on the entire VO system, which is exactly what we expect.



(a) Trajectories of sequence 08

(b) Trajectories of sequence 09

(c) Trajectories of sequence 10

Fig. 7. Trajectories of IMU-assist VO testing results on Sequence 08,09,10

V. Conclusion

This paper proposes an IMU-assisted end-to-end monocular VO system based on deep learning. The system inputs two consecutive video frames to calculate the optical flow. Then input the optical flow into the DenseNet to predict translation and rotation information. Due to the structural adjustment of the CNN architecture, a frame-to-frame estimation system with translational accuracy over similar methods is implemented using fewer parameters and lesser network training time. The proposed method has better real-time performance and higher translation precision. Using the IMU data to correct the results with little computing resources, which makes up for the shortcomings of the VO system and obtains satisfying results. We finally implemented an IMU-assisted lightweight deep learning-based visual odometry system.

There are several aspects for future improvements. It is possible to extract key frames and establish multiple loss functions to reduce the cumulative error to some extent. Our IMU correction is a loosely coupled data coupling method. In the future, IMU data can be added to a certain layer of the neural network in order to implement the tight coupling optimization method to implicitly model and obtain better results.

References

[1] R. Mur-Artal, and J. D. Tardos, “ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras,” IEEE Transactions on Robotics, vol. 33, no.5, pp. 1255-1262, 2017.

[2] A. Geiger, J. Ziegler, and C. Stiller, “StereoScan: Dense 3d reconstruction in real-time,” 2011.

[3] P. Muller, and A. Savakis, “Flowdometry: An Optical Flow and Deep Learning Based Approach to Visual Odometry,” 2017.

[4] L. Fan, W. Huang, and C. Chuang, *et al.*, “End-to-End Learning of Motion Representation for Video Understanding,” 2018.

[5] C. Zach, T. Pock, and H. Bischof. ”A duality based approach for realtime tv-l1 optical flow. Pattern Recognition,” pp. 214C223, 2007.

[6] H. Gao, L. Zhuang, L.Maaten, and K.Weinberger, “Densely Connected Convolutional Networks ,” 2016.

[7] R. Clark, S. Wang, H. Wen, A.Markham, and N.Trigoni. “VINet: Visual-Inertial Odometry as a Sequence-to-Sequence Learning Problem,” 2017.

[8] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite,” 2012.

[9] S. Leutenegger, S. Lynen, M. Bosse, R.Siegwart, and P.Furgale, “Keyframe-Based Visual-Inertial Odometry Using Nonlinear Optimization,” International Journal of Robotics Research, vol. 34, no.3, pp. 314-334, 2014.

[10] T. Qin, P. Li, and S. Shen, “VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator,” *IEEE Transactions on Robotics*, vol. PP, no.99, pp. 1-17, 2017.

[11] D. Scaramuzza and F. Fraundorfer, ”Visual Odometry [Tutorial],” *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80-92, Dec. 2011.

[12] C. Forster, M. Pizzoli, and D. Scaramuzza, “SVO: Fast Semi-Direct Monocular Visual Odometry,” 2014.

[13] J. Engel, V. Koltun, and D. Cremers, “Direct Sparse Odometry,” IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 40, no.3, pp. 611-625, 2018.

[14] S. Baker, and I. Matthews, “Lucas-Kanade 20 Years On: A Unifying Framework,” International Journal of Computer Vision, vol. 56, no.3, pp. 221-255, 2004.

[15] J. Engel, T. Schöps, and D. Cremers, “LSD-SLAM: Large-Scale Direct Monocular SLAM,” 2014.

[16] K. Konda , and R. Memisevic, “Learning visual odometry with a convolutional network,“ 2015.

[17] A. Kendall, M. Grimes, and R. Cipolla, “PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization,” 2015.

[18] G. Costante, M. Mancini, P. Valigi, and T. Ciarfuglia, “Exploring Representation Learning With CNNs for Frame-to-Frame Ego-Motion Estimation,” IEEE Robotics & Automation Letters, vol. 1, no.1, pp. 18-25, 2015.

[19] S. Wang, R. Clark, H. Wen, and N. Trigoni, “DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks,” 2017.

[20] Z. Yin, and J. Shi, “GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose,” 2018.

[21] Y. Xu, Y. Wang, and L. Guo, ”Unsupervised Ego-Motion and Dense Depth Estimation with Monocular Video,” 2018.

[22] E. Ilg, N. Mayer, and T. Saikia, *et al.*, “FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks,” 2017.

[23] Z. Zhang, and D. Scaramuzza, “A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry,” 2018.