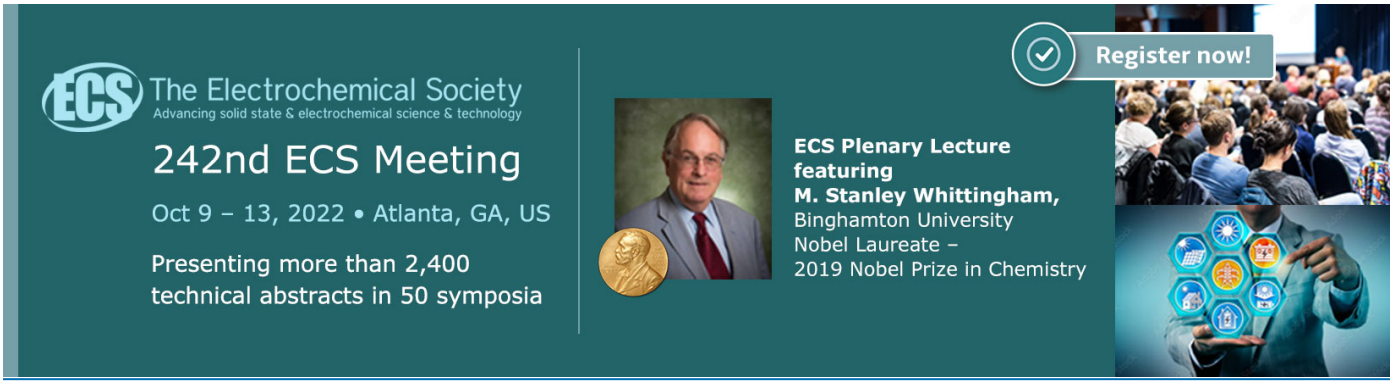
Improved visual inertial odometry based on deep learning



Improved visual inertial odometry based on deep learning

Jiabin Wang[1,a], Faqin Gao[1,b*]

[1] School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou, Zhejiang, 310018, China

[a] E-mail: wjb123wjb123@163.com

[*b] E-mail: gaofaqin@zstu.edu.cn

Abstract. The traditional visual inertial odometry according to the manually designed rules extracts key points. However, the manually designed extraction rules are easy to be affected and have poor robustness in the scene of illumination and perspective change, resulting in the decline of positioning accuracy. Deep learning methods show strong robustness in key point extraction. In order to improve the positioning accuracy of visual inertial odometer in the scene of illumination and perspective change, deep learning is introduced into the visual inertial odometer system for key point detection. The encoder part of MagicPoint network is improved by depthwise separable convolution, and then the network is trained by self-supervised method; A visual inertial odometer system based on deep learning is compose by using the trained network to replace the traditional key points detection algorithm on the basis of VINS. The key point detection network is tested on HPatches dataset, and the odometer positioning effect is evaluated on EUROC dataset. The results show that the improved visual inertial odometer based on deep learning can reduce the positioning error by more than 5% without affecting the real-time performance.

1. Introduction

Odometry is an important part of simultaneous localization and mapping. It can estimate its own motion by carrying sensors without environmental prior information. It is the basis of technologies such as unmanned driving, robot positioning and virtual reality[1] .The visual inertial odometer is an odometer system that integrate a camera and an inertial measurement unit (IMU), which complement each other; the IMU provide scale information for the monocular vision system, and the vision system can alleviate the drift problem of the IMU[2].

The traditional visual inertial odometry scheme use artificially designed rules to extract visual features, such as Harris, FAST, SIFT. [3~4] Although it is easy to extract and describe features, it is not robust enough in scenes with drastic changes in illumination and perspective (such as drone platforms), which limits the usage scenarios and positioning accuracy of the odometry. Deep learning methods could improve the detection robustness by training and adjusting the network, so the application and research of deep learning for visual feature extraction has gradually received attention. TILDE[5] is a regression-based key points detector, which effectively reduces the impact of illumination changes on key points detection, but the effect is not ideal when the viewing angle changes. The GCN network[6] uses a siamese network and a recurrent neural network to achieve key point extraction and descriptor calculation, but the input image needs to be obtained through an RGB-D camera, which is costly. MagicPoint network[7] is a self-supervised key point detection network, which is trained on images generated by programs. It can detect points with significant geometric

features in the image, with good real-time performance and not easily affected by illumination and rotation. However, due to the lack of image training in real scenes, the actual detection effect needs to be improved.

In order to solve the problem that traditional feature extraction methods are too sensitive to changes in illumination and perspective, which affects the positioning of the odometry. We use deep separable convolution to improve the MagicPoint network encoding part; self-labeling of key points is implemented by Homographic Adaptation[8] in the real scenes dataset. Replace the traditional key point detection method with the trained detection network, and finally form a brand new visual inertial odometry based on deep learning. Experimental results show that the detection performance of improved MagicPoint network has been improved, and the visual inertial odometry using the improved MagicPoint network has higher positioning accuracy than other solutions.

2. VISUAL INERTIAL ODOMETRY BASED ON IMPROVED MAGICPOINT NETWORK

We improve the MagicPoint network so that the position accuracy of the visual inertial odometry in scenes with drastic changes in illumination and perspective is improved. Than the improved detection network is used to replace the original detection method in the VINS[9] system to form a visual inertial odometry system based on deep learning.

2.1. VINS profile

VINS is a tightly coupled visual inertial odometry framework proposed by the Hong Kong University of Science and Technology. The system overview is shown in figure 1. We focus on the front-end part of the vision, that is, the improvement of the key point detection part of the image; the improved MagicPoint detection network is applied to the system to improve the positioning accuracy of VINS in scenes with changing viewing angles and lighting.

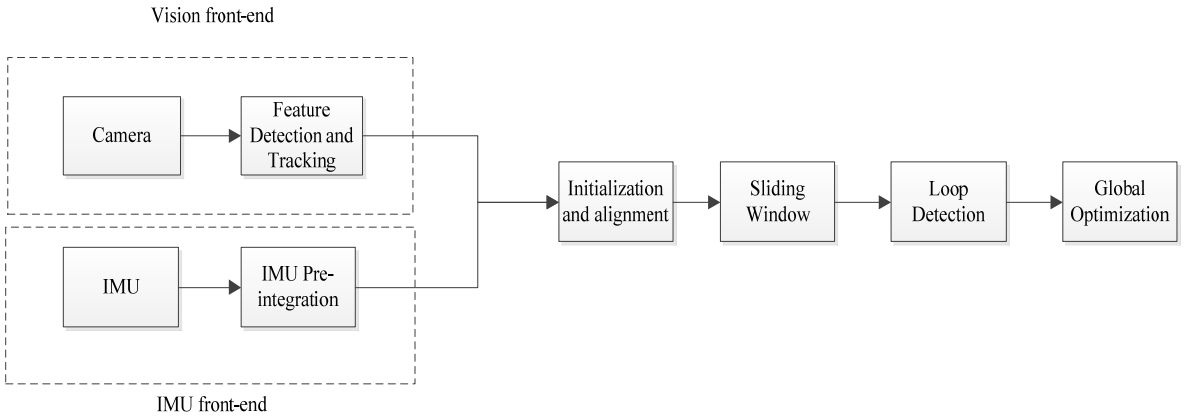


Figure 1. An overview of the VINS-mono system

2.2. Improved MagicPoint network

The MagicPoint network contains two parts: Encoder and Decoder, which are used to extract feature information and detect key points respectively. The Encoder part is originally built based on the VGG architecture. The network structure is simple but the calculation is large, and the receptive field is small. Moreover, the results extracted by the network contain a lot of redundant information, which is not conducive to subsequent use. For the purpose of more effectively extract the information in the image, reduce redundant information; simultaneously increase the detection robustness and the receptive field of the network; we draw on the lightweight network MobileNet V3[10] and redesign the network structure; the bottle neck layer is used to replace ordinary convolution which reduce the amount of network parameters and remove redundant information. Besides, the detection speed is accelerated and the receptive field is increased.

The improved Encoder part has a total of 9 layers. The first, third, and sixth layers have a stride of 2, and the rest have a stride of 1, and the padding is all 0; except for the first layer, which is a normal convolutional layer, the rest are bottle neck layers. The bottle neck layer is composed of 1x1 pointwise convolution, 3x3 depthwise convolution and 1x1 pointwise convolution. The first two layers use the ReLU6 activation function, and the last layer uses the linear activation function. The Decoder part contains 2 layers, the size of the first layer of convolution kernel is 3x3, the number of channels is 256, the stride is 1, and the padding is 1; the second layer of convolution kernel is 1x1, the number of channels is 65, the stride is 1, and the padding is 0. The intermediate tensor changes from 30x40x128 to 30x40x65 after passing through these two layers. Then pixel shuffle[11] is used for upsampling, change the intermediate tensor back to the input image size, and return the probability that each pixel is a key point. The improved network structure diagram is shown in figure 2, where conv3-16 represents a 3x3 convolution kernel, and the number of output channels is 16, bneck3-32 means that the depthwise convolution kernel is 3x3, and the number of output channels is 32.

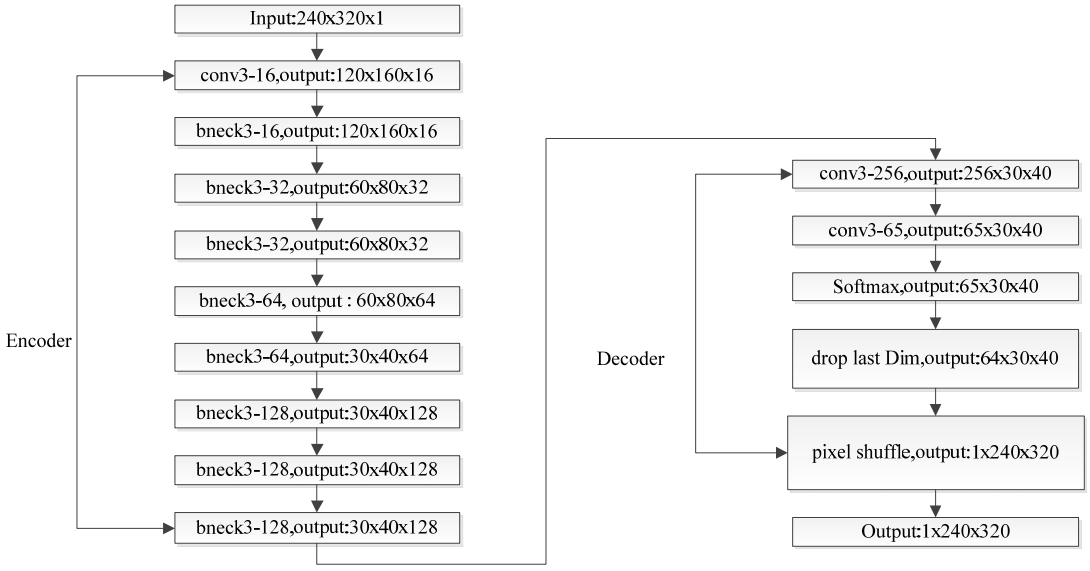


Figure 2. Network structure diagram

2.3. Training process

The improved network model is trained in a self-supervised way. The basic idea is: first use artificially generated images with known key point coordinates for pre-training, then label the real scene images from the model trained in the previous step, finally use the labelled real scene images to obtain the final model. The training process is shown in figure 3. The pytorch framework is selected for training, the optimizer selects the Adam optimizer, and the learning rate is 0.0001.

Figure 3. Training flowchart

Geometry dataset generation: OPENCV is used to generate images including triangles, quadrilaterals, line segments and ellipses, a total of 4000 images. The vertices of triangles, quadrilaterals, end points of line segments and the center of the ellipse are treated as key points.

Image enhancement: Operations such as rotating, flipping, and cropping the image containing geometric figures is performed to obtain more images. OPENCV is used to add random Gaussian noise, motion blur, and brightness change, thereby improving the adaptability of the training model to light and noise.

Pre-training: Since the position of the key points in the generated image is known, the position is still known after image enhancement, so it can be trained without additional labeling. The preliminary detection model is obtained after 100k iterations.

Self-labeling: The model obtained by pre-training has the basic ability to detect key points. In order to enhance the detection ability of the model in the real scene, it is necessary to select the image in the real scene to train the model again. Manual labeling is time-consuming and labor-intensive, so we use the pre-trained model to label the images in the real scene, eliminating the cumbersome labeling process. Homographic adaptation is used to keep 300 key points with the highest confidence as labels. Image enhancement is also performed on real images which improve the model ability to detect changes in perspective and scale.

Final training: The 60k images in the training dataset in MS-COCO2014[12] are selected as the training dataset for this training, and the 20k images in the verification dataset are used as the verification dataset. The annotation is implemented by self-labeling, and the final training model is obtained after 120k iterations.

Loss function: We use the cross-entropy loss function as shown in formulas (1) and (2).

$$L(X,Y)=\frac{1}{H_cW_c}\sum_{\substack{h=1\\w=1}}^{H_c,W_c} l(x_{hw};y_{hw}) \tag{1}$$

$$l(x_{hw};y)=-\lg\left(\frac{\exp(x_{hwy})}{\sum_{k=1}^{65}\exp(x_{hwk})}\right) \tag{2}$$

X represents the set of intermediate tensors corresponding to the predicted key points,Y represents the set of intermediate tensors corresponding to the true values of the key points. H_c=H/8,W_c=W/8 indicates the height and width of the current image, H,W indicates the height and width of the original image, x_{hw}, y_{hw} respectively represent the prediction result and the real result corresponding to the position hw in the intermediate tensor, x_{hwy} represents the y prediction tensor in the set of intermediate tensors, x_{hwk} represents all intermediate tensors.

3. EXPERIMENTAL DETAIL

This part mainly focus on the detection effect of the improved MagicPoint network and compare it with other key point detection methods. Explore the impact of using different key point detection methods on the odometry positioning performance. Some experimental details are also introduced in this part. We measure the run-time of the network in a GTX 1660Ti GPU cases. The system is implemented in C++, and the Pytorch API Libtorch is used to perform key point detection.

3.1. Key point detection dataset and evaluation criteria

HPatches dataset[13] is used to verified the key point detection effect. The evaluation rule adopts FPS (Frame per second) and repetition rate[14]. FPS is used to measure the speed of key point detection. And repetition rate indicates the probability that the key points in the original image will be detected in the image obtained by the original image after illumination and perspective transformation, which is used to measure the robustness of the detection method to illumination and perspective changes.

3.2. Odometry performance test data set and evaluation standard

The EUROC dataset[15] in a drone scene with drastic changes in lighting and viewing angles is used to test the performance of the Odometry.

The positioning effect evaluation of the odometry system use ATE (Absolute Trajectory Error) and RPE (Relative Pose Error). ATE is used to compare the difference between the estimated trajectory and the true trajectory. RPE is used to compare the difference between the estimated attitude change and the real attitude change between two frames with a certain time interval.

3.3. Experimental setup

In order to illustrate the effectiveness of the key point detection algorithm proposed in this article, traditional methods such as Shi-Tomasi harris (ST-harris), SIFT, ORB (without descriptors) and deep learning methods such as TILDE and original MagicPoint (MP) are used. Among them, ST-harris is the key point detection method originally used by VINS. ST-harris, SIFT and ORB are implemented with OPENCV. TILDE method is implemented by borrowing from open source code.

Ten sets of images of illumination changes and viewing angle changes are selected for average repetition rate calculating in the HPatches dataset. Feature point number of per image is limited to 300.

4. Experimental results and analysis

As shown in Table 1, the detection speed of ST-harris key points is relatively the fastest and SIFT is relatively slow under the experimental conditions in this article. The ORB method has the highest repetition rate in the scene of changing perspective. The improved MP method has the highest average repetition rate which is 15.5% higher than ST-harris method and 6% higher than the original MagicPoint network, and the detection speed is also close to the fastest ST-harris, indicating that the improved MP method has good real-time effects and strong robustness to illumination and viewing angle changes. Rep 1 and 2 respectively represent the repetition rate in the scene of lighting change and the scene of viewing angle change. Rep 3 represents the average of Rep 1 and 2. MP1 stands for the original MP method, and MP2 is the improved MP method. Bold data is a better result.

Table 1. Performance comparison of different key point detection methods

Methods/Eval	ST-harris	SIFT	ORB	TILDE	MP1	MP2
FPS	**98.1**	17.3	85.9	19.35	90.3	91.5
Rep 1(%)	0.5932	0.4306	0.6345	0.4317	0.6509	**0.7016**
Rep 2(%)	0.5356	0.5677	**0.6127**	0.2931	0.5716	0.6027
Rep 3(%)	0.5644	0.4991	0.6236	0.3624	0.6112	**0.6521**

Table 2. Comparison of odometry results in different image sequences in the EUROC dataset

Datasets	Evaluations	ST-harris	SIFT	ORB	TILDE	MP1	MP2
MH01	ATE(m)	0.1910	0.1947	0.1931	0.2647	0.1923	**0.1829**
	RPE(deg)	5.8024	4.4885	4.9711	5.8233	5.0158	**4.3863**
MH03	ATE(m)	0.4014	0.4171	0.4107	0.4988	0.413	**0.3996**
	RPE(deg)	8.7454	9.0510	**8.7441**	9.2987	9.3520	9.227
MH05	ATE(m)	0.3885	0.4215	0.4078	0.4783	0.4186	**0.3812**
	RPE(deg)	**5.2959**	5.4461	5.8160	5.7057	5.7410	5.4017
V201	ATE(m)	0.1239	0.1370	0.1337	0.1531	**0.1173**	0.1212
	RPE(deg)	8.3556	6.7658	5.8313	9.9970	6.1253	**5.0834**
V202	ATE(m)	0.2660	0.2902	0.2870	0.2924	0.3157	**0.2551**
	RPE(deg)	13.7554	14.7162	**13.2516**	13.6891	14.8629	13.7586
V203	ATE(m)	0.4147	**0.3214**	0.5920	0.5192	0.4827	0.3223
	RPE(deg)	16.3659	16.4582	**14.0731**	16.1534	16.5280	15.1872
Average	ATE(m)	0.2975	0.2969	0.3373	0.3677	0.3232	**0.2770**
	RPE(deg)	9.7201	9.4876	**8.7812**	10.1112	9.6041	8.8407

It can be seen from Table 2 that the VINS system ATE using MP2 perform well in general; Both ATE and RPE indicators perform well in MH01 and V201 image sequences with sufficient light and slow motion; however, the RPE indicator performs poorly in fast-moving image sequences. The ATE of the improved MP method is also better than other methods in the fast moving and insufficiently illuminated scene MH05, which shows that the improved MP method is less affected by the light. The average ATE and RPE of the VINS system that uses the improved MagicPoint network to detect key

points on all image sequences are reduced by 6% and 9%, respectively, compared with the original VINS system.

5. CONCLUSIONS

This paper proposes a visual inertial odometry system based on the improved MagicPoint network, which reduces the interference of illumination and viewing angle changes to the odometry system, thereby improving the positioning accuracy.

Future work is mainly to optimize the relative pose error when the movement speed is fast and the camera posture changes drastically, and to further reduce the relative pose error under the premise of keeping other performance unchanged.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61402417), the Project of Zhejiang Provincial Natural Science Foundation (LY14F030025), and funded by State Key Laboratory of Geo-information Engineering (SKLGIE2017-M-2-3).

References

[1] Xiong W,Jin J Y,Wang J. (2020) Monocular vision odometer based on deep learning feature point method. Computer Engineering and Science,42(01): 117-124.

[2] Scaramuzza, D, Fraundorfer F, et al. (2011) Visual Odometry .In: IEEE Robotics & Automation Magazine . Hoboken. pp. 80-92.

[3] Smith S M . Fast robust automated brain extraction. (2010) Human Brain Mapping, 17(3):143-155.

[4] Lowe D G . (2004) Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2):91-110.

[5] Verdie Y , Yi K M , Fua P , et al. (2015)TILDE: A Temporally Invariant Learned Detector.In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), New York. pp. 5279-5288.

[6] Tang J , Folkesson J , Jensfelt P.(2018) Geometric Correspondence Network for Camera Motion Estimation. IEEE Robotics & Automation Letters, 3(2):1010-1017.

[7] Detone D, Malisiewicz T , Rabinovich A . (2017) Toward Geometric Deep SLAM[C]//Proc of Computer Vision and Pattern Recognition. New York. pp. 150-154.

[8] Detone D, Malisiewicz T , Rabinovich A. (2018) SuperPoint: Self-Supervised Interest Point Detection and Description. In: Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). New York. pp. 337-33712.

[9] Qin T,Li P ,Shen S. (2017) VINS-Mono: A robust and versatile monocular visual-Inertial state estimator. In: IEEE Transactions on Robotics, 34(4): 1004-1020.

[10] Howard A , Sandler M , Chen B , et al. (2020) Searching for MobileNetV3.In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul. pp. 1314-1324.

[11] Shi W , Caballero J , F Huszár, et al. (2016) Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). New York. pp. 1874-1883.

[12] Lin T Y , Maire M , Belongie S , et al. (2014) Microsoft COCO: Common Objects in Context.In: European Conference on Computer Vision. Zurich. pp. 740-755.

[13] Balntas V , Lenc K , Vedaldi A , et al. (2017) HPatches: A Benchmark and Evaluation of Handcrafted and Learned Local Descriptors.In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). New York. pp. 3852-3861.

[14] Yu H S, Gua F, Guo L F, et al. (2021) Robust monocular vision inertial SLAM Based on improved superpoint network. Chinese Journal of Scientific Instrument, 42(01):116-126

[15] Schneider, Thomas, Nikolic, et al. (2016) The euroc micro aerial vehicle datasets. International Journal of Robotics Research, 35(2):1157-1163.