Parameter Estimation via Deep Learning for Camera Localization

Parameter Estimation via Deep Learning for Camera Localization

Mina Chong[&], Qiming Li[&], Jun Li[*]

Quanzhou Institute of Equipment Manufacturing, Chinese Academy of Science, Quanzhou, China

[&] These authors contributed equally to this work and should be considered co-first authors

[*]Corresponding author's e-mail: junli@fjirsm.ac.cn

Abstract. This paper proposes a method based on deep learning to estimate parameters for camera localization. The parameters are 6-DOF camera pose and regressed from a single RGB image by an end to end convolution neural network. The proposed network makes use of batch normalized method, which relieves the disappearance of gradient, leading to drastic improvements in convergence speed. In addition, inspired by decomposing convolutions, the network breaks large convolutions into more small convolutions to cascade, which reduces the computational complexity, and enhances network feature representations owing to adding a layer of convolution. This CNN-based method is able to learn effective feature for camera localization in state of motion blur and illumination changes, while the traditional SHIF-based methods fail. Experimental results on both indoor and outdoor public datasets show the improved network achieves an increase in accuracy, and outperforms with the compared methods.

1. Introduction

As a fundamental research task, parameter estimation for camera localization plays an important role in the field of mobile robot navigation [1], unmanned driving [2] etc. Traditional geometric methods estimate inliers of camera localization based on SFIT features between images. However, these methods are highly dependent on texture information of images and have complex computation in feature processing. In this paper, we present a method via deep learning to solve the problems above.

In recent years, a great deal of effort has been devoted to studying this issue based on deep learning. There are mainly three aspects for scholars to research: (i) input of the network, including single RGB image [3], optical stream images [4, 5] and IMU data [6]; (ii) structure of the network, such as GoogLeNet, VGG and LSTM, many studies are improved on these basic networks [7-10]. (iii) output of the network, some networks directly output a 6-DOF vector of camera localization [3], while some output the position and angle parameters separately [7]. Since 2016, Kendall et al. first proposed the framework of end-to-end convolution neural network to learn camera localization [3]. So far, many new network models have been put forward to estimate the parameters by training images. The PoseNet network proposed by Kendall et al. laid a foundation for subsequent researchers to use deep learning to solve camera localization problems.

The main motivation of our study is to optimize the PoseNet. Although it has overcome the limitations of traditional methods, there are some shortcomings: it can been found from the experimental results that the error is a little bigger in 7 Scenes of public indoor datasets. For example, in the

Published under licence by IOP Publishing Ltd

experiment of "Stairs", its error of localization is 0.47 m, but the overall space is the size of 2.5 * 2 * 1.5. Therefore, the method proposed in this paper is mainly aimed at to improve the accuracy of indoor scenes.

The contributions of our method can be summarized as follows: (i) decomposing convolutions. The 5*5 convolution are replaced by two 3*3 convolutions, which reduces the number of parameters and overfitting; (ii) adopting Batch Normalized (BN). The use of BN has accelerated the training speed of networks and improved the accuracy of learning. In addition, when BN is applied to a certain layer of convolutional neural network, the internal processing of each mini-batch data is standardized to normalize to the distribution of N (0,1), which reduces the change of the internal neuron distribution. Experimental results show that the improved network is more accurate than the previous network PoseNet, which proves the proposed method is effective.

The rest of this paper is described below. We introduce the details of our method in Section 2. The experimental process and results are showed in Section 3. At last, we conclude the paper briefly in Section 4.

2. Method

In this section, we will describe the proposed method in detail. First, we analyze how the method returns the camera pose by means of deep learning, and use the Euclidean distance loss function to evaluate the training of model (as described in Section 2.1). Second, the network replaces 5*5 convolution by two 3*3 convolutions, which reduces the amount of model parameters and computational complexity at the same time (as described in Section 2.2). Third, we apply the Batch Normalized (BN) method to the network, which is added after convolutional layer to normalize the feature to distribution of $N\ (0, 1)$, which alleviates the problem of gradient disappearance and speeds up the training (as described in Section 2.3).

2.1. Camera pose regression

First, set the image sequence and its corresponding camera localization as a collection:

$$Z = (I, P) = \{(i_1, p_1), (i_2, p_2), \dots (i_n, p_n)\} \tag{1}$$

Where I represents the image and P represents the camera pose. Assuming that there is a functional relationship between elements I and P in set Z, set to f_{IP}, then a mapping relationship we need to learn through convolutional neural networks is:

$$p_i = f_{IP}(i_i) \tag{2}$$

In SHIF-based methods, we know that f_{IP} , a conditional probability distribution for $P(\tilde{p}|\tilde{\imath})$. In convolutional neural network, after the network model is trained, the relationship of the mapping function can be learned. Then, the camera pose of the current image is output during test. Therefore, the conditional probability distribution relationship based on the deep learning network model is $P(\tilde{p}|\tilde{\imath}, Z)$. After training the datasets and its corresponding camera pose, the model fits the parameter ω. When a query image is used, the conditional probability distribution can be converted to:

$$P(\tilde{p}|\tilde{\imath}, Z) \approx P(\tilde{p}|\tilde{\imath}, \omega) \tag{3}$$

Further derivation of the formula above:

$$\mathrm{P}(\tilde{p}|\tilde{\imath}, Z) = \int \mathrm{P}(\tilde{p}, \omega|\tilde{\imath}, Z)d\omega = \int \mathrm{P}(\tilde{p}|\tilde{\imath}, \omega) \cdot \mathrm{P}(\omega|Z)d\omega \tag{4}$$

It can be seen from (4) that the problem of solving the conditional probability is converted into the posterior distribution problem of model parameter ω. Then use the Bayesian formula to convert the parameter posterior distribution problem into:

$$\mathrm{P}(\omega|Z) = \frac{\mathrm{P}(P|I,\omega)\cdot\mathrm{P}(\omega)}{\mathrm{P}(P|I)} \tag{5}$$

Here we should note that in $\mathrm{P}(P|I)$, the P in parentheses indicates the camera pose.

In summary, the conditional probability distribution function of the input image sequence I and the output camera pose P is:

$$\mathrm{P}(\tilde{p}|\tilde{\imath}) = \int \mathrm{P}(\tilde{p}, \tilde{\imath}|\omega) \cdot \frac{\mathrm{P}(P|I,\omega)\cdot\mathrm{P}(\omega)}{\int \mathrm{P}(P|I,\omega)\cdot\mathrm{P}(\omega)d\omega} d\omega \tag{6}$$

Through the above series, we can use MAP (Mean Average Precision) to estimate the posterior distribution of ω and convert the pose estimation problem into the maximum posterior distribution $\mathrm{P}(\omega|Z)$.

The same as PoseNet, our network outputs a 6-DOF vector P, which represents camera localization.

$$P = [x, q] \tag{7}$$

Where x and q indicate camera position and orientation respectively.

In order to make the network to regress the pose better, we use the Euclidean distance loss function:

$$loss(I) = \sum_{i=1}^{n} \lambda(\|x - \bar{x}\|_2 + \beta\|q - \bar{q}\|_2) + (1-\lambda)\sum \omega^2 \tag{8}$$

Among them, x and q respectively represent the camera position and orientation, because the displacement change amount is usually higher than the angle change amount, which causes the error of the displacement in the loss function to be large, so the weight coefficient β is introduced to balance. λ is the proportional coefficient of BN.

The network structure adopted in this paper is the improvement of PoseNet with the help of Inception V2 [11]. RGB images are input to the network, and 6-DOF camera pose are output through end-to-end convolutional neural network after learning. Figure 1 is the overall platform of network. As shown in the figure, the RGB images of 224*224 are input, and the inliers of feature for camera localization are estimated by the network proposed. Finally, multiple local features are fused through the fully connected layer with the scale of 2048, and the camera pose of the image is output. Next we will focus on how our network improves over the PoseNet model.

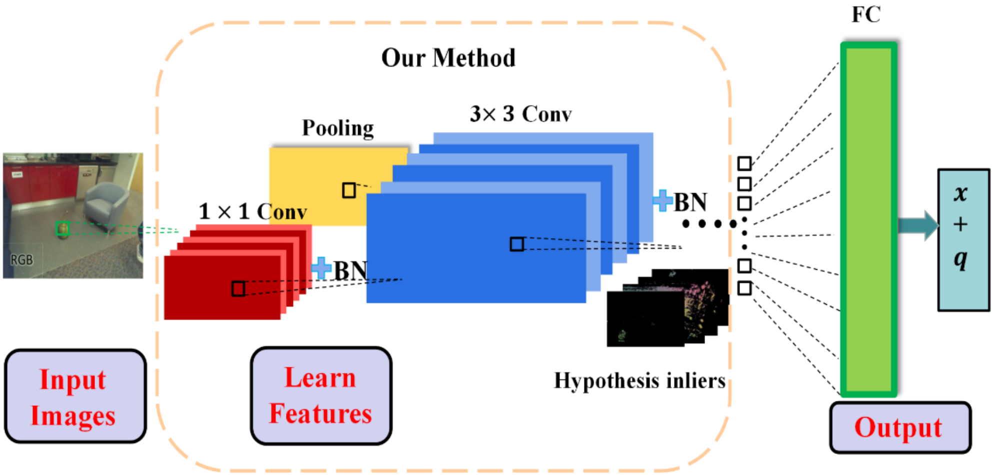

Figure 1. The overall structure of the optimized network

2.2. Decompose convolutions

Generally speaking, theoretical research based on deep learning requires a very large dataset, and the number of the parameters is huge. Therefore, the resources consumed in the calculation are so much. On the basic of VGGNet [12], it is assumed that the feature numbers of the 5*5 and two-level 3*3 convolution outputs are the same. Two 3*3 convolutions replace the 5*5 large convolution, then the calculation of the two-stage 3*3 convolution is 18/25 times ((3*3 + 3*3) / 25) of the former, and the calculation amount is reduced. At the same time, the addition of a layer of convolution also improves the representations of the model and enriches the spatial characteristics. So, in the inception structure of the PoseNet network, a large 5*5 convolution is replaced with two 3*3 convolutions. As shown in Figure 2, it is a schematic diagram of decomposing convolution.

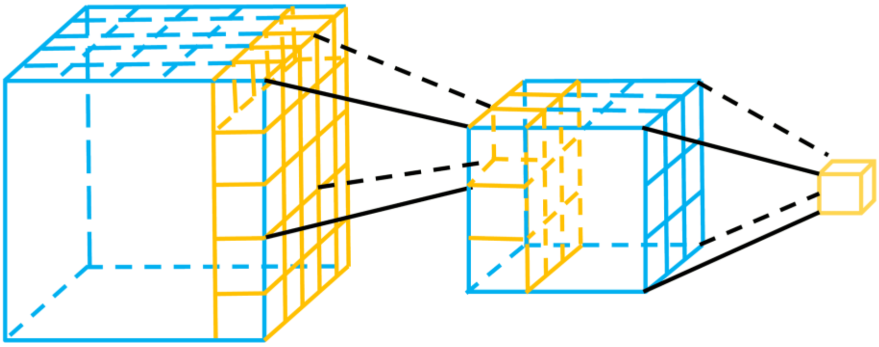

Figure 2. 5*5 convolution is decomposed into two 3*3 convolutions in series

2.3. Batch Normalized (BN)

Batch Normalized method [13] is based on the mini-batch SGD [14]. Although the mini-batch is accurate in gradient when updating and the parallel computation speed is fast. The complexity of parameters is also very high. In deep learning, the parameters of each layer of convolutional neural network are changing and updating during training, especially the Internal Covariate Shift problem usually occurs in the hidden layers. As the layers added, the feature distribution will also gradually deviate to a certain extent during the training process. Generally, the overall distribution (assuming Sigmoid function) will approach the two extremes of upper and lower limits. As a result, gradient disappearance will come up, which is the essential factor of slow convergence.

The proposal of Batch Normalized is to solve the above problems. Through reasonable standardization, the feature distribution of each layer satisfies the standard normal distribution $N\ (0, 1)$, as much as possible. The advantage of this operation is that it is mandatory to keep the input value in a more sensitive area, making the gradient larger and reducing the problem of gradient disappearance. In addition, when the gradient is always large, the efficiency of the neural network for parameter adjustment is also improved, and the speed of tuning for the loss function is also increased, so that the convergence speed is further accelerated during training. Another advantage is that with the rapid learning efficiency, it is possible to continue the iteration, training and learning more features, which improves the accuracy, and reduce the operation of the Dropout after achieving the previous accuracy. The network structure can be simplified simultaneously. In the inception of the PoseNet network, we add Batch Normalized after convolution layer, remove the Dropout, and local response normalization (LRN).

3. Experimental Results

As for experiment, we choose the public datasets: indoor 7 Scenes dataset and outdoor King's College dataset. The indoor dataset is RGB-D images collected by Kinect sensor, mainly selecting scenes with small scale, such as indoor stairs and office corners. There are 2000 to 12,000 images in each scene with a resolution of 640*480 and depth images. In the network adopted in this paper, only RGB images need to be input. As shown in figure 3 [15], it is the indoor scene image. Besides, in order to verify the effectiveness of our network, we choose outdoor King's College dataset for experiments. The King's College dataset is annotated by SFM method. Figure 4 shows the scene of outdoor King's College [16].

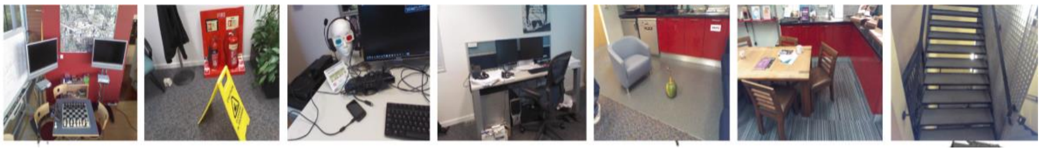

Figure 3. The indoor scenes from left to right：Chess, Fire, Heads, Office, Pumpkin, Red Kitchen and Stairs.

Figure 4. The outdoor King's College scenes

In order to verify the performance of the proposed method, all experiments are conducted using caffe on the graphics workstation GW933. The operating system is 64-bit Ubuntu 16.0, 1TB DDR4-2133MHz R-ECC memory, 4T hard disk, 4 Titan GPU.

In the experiment, we set the initial learning rate to 0.0001, 90% attenuation per 100 iterations, set the momentum factor to 0.9, and the batch_size to 75. We make use of Adagrad algorithm to optimize the network. Snapshot every 2,500 times, and the global weight is attenuated to 0.5.

We compare the proposed network with PoseNet and ScoRe Forest [17]. The 224*224 pixel RGB image inputs to the network, and convolutional neural network regresses parameters of camera localization via training. Table 1 is the experimental result of 7 indoor scenes and outdoor King's College scenes in the public dataset.

Table 1. The comparison results of camera position and orientation error.

Datasets	Images		Size of space	Compared methods		
	Train	Test	Unit:m	SCoRe Forest(Uses RGB-D)	PoseNet	Our Method
Chess	4000	2000	3*2*1	0.03m,0.66°	0.32m,4.06°	0.28m,4.11°
Fire	2000	2000	2.5*1*1	0.05m,1.50°	0.47m,7.33°	0.43m,7.43°
Heads	1000	1000	2*0.5*1	0.06m,5.50°	0.29m,6.00°	0.21m,6.30°
Office	6000	4000	2.5*2*1.5	0.04m,0.78°	0.48m,3.84°	0.43m,3.32°
Pumpkin	4000	2000	2.5*2*1	0.04m,0.68°	0.47m,4.21°	0.43m,4.29°
Red Kitchen	7000	5000	4*3*1.5	0.04m,0.76°	0.59m,4.32°	0.50m,3.83°
Stairs	2000	1000	2.5*2*1.5	0.31m,1.32°	0.47m,6.91°	0.44m,7.13°
King’s College	1200	343	140*40	N/A	1.92m,2.70°	1.81m,2.57°

As can be seen from the table, ScoRe Forest method is currently the highest accuracy algorithm for indoor small scene, but the input data is RGB - D images. Here are something to explain why RGB - D images is more valid for estimating parameters of camera pose: in small indoor scenes, the depth information has a great influence on the hypothesis inliers, because the depth image itself has geometric feature. When the camera is shooting a image indoor, the smaller angle of view rotates, which will make a big difference between the image and image of next frame. Although ScoRe Forest method has the highest accuracy, the disadvantage is that the input is RGB-D image. In practical application, RGB-D image can only be used in small indoor scenes. In large outdoor scenes, compared with RGB, the feature proportion of depth information is little, so ScoRe Forest method is not suitable for large outdoor scenes.

The results of orientation are not accurate as PoseNet in some tests. We explain that the magnitude of position is many times bigger than orientation although we use coefficient β to balance in the loss function, the proportion is still tiny. In the PoseNet, it normalizes the quaternion orientation vector to unit length, which we need to improve in future work.

4. Conclusions

In this paper, we propose a parameter estimation method for camera localization based on deep learning. The RGB image is input to estimate the inliers of camera pose by convolutional neural network, and a 6-DOF camera pose is regressed after network training. The convolutional neural network is an improvement method based on PoseNet. Two small convolutions in series are used to replace the large convolution, which reduces the amount of the model parameters and makes the convergence faster. Moreover, Batch Normalized method is applied to convolutional layer, relieving the problem of gradient disappearance. Experimental results show our method has achieved a good performance, and is more accurate in position and practical.

Acknowledgment

This work was supported by the National Key Research and Development Program of China Grant (No.2016YFC11000502) and Scientific and Technological Project of Quanzhou (No.2018C015).

References

[1] Desouza G N, Kak A C. (2002) Vision For Mobile Robot Navigation: A Survey [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 24(2):237-267.

[2] Zhang X, Gao H, Mu G. (2016) A study on key technologies of unmanned driving [J]. Caai Transactions on Intelligence Technology, 1(1):4-13.

[3] Kendall A, Grimes M, Cipolla R. (2015) Posenet: A convolutional network for real-time 6-dof camera relocalization [C]. IEEE international conference on Computer Vision. Santiago. pp. 2938-2946.

[4] Roberts R, Potthast C, Dellaert F. (2009) Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies [C]. IEEE Conference on Computer Vision and Pattern Recognition. Kyoto. pp. 57-64.

[5] Dosovitskiy A, Fischer P. (2015) Flownet: Learning optical flow with convolutional networks [C]. IEEE international conference on Computer Vision. Santiago. pp. 2758-2766.

[6] Clark R, Wang S, Wen H. (2017) Vinet: Visual-inertial odometry as a sequence-to-sequen ce learning problem [C]. Thirty-First AAAI Conference on Artificial Intelligence,San Francisco. pp. 3995-4001.

[7] Walch F, Hazirbas C, Leal-Taixé L. (2016) Image-based localization with spatial lstms [J]. arXiv preprint arXiv:1611.07890 , 2(6).

[8] Clark R, Wang S, Markham A. (2017) Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization [C]. IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, pp. 6856-6864.

[9] Costante G, Mancini M, Valigi P. (2016) Exploring representation learning with cnns for frame-to-frame ego-motion estimation [J]. IEEE robotics and automation letters, 1(1): 18-25.

[10] Mohanty V, Agrawal S, Datta S. (2016) Deepvo: A deep learning approach for monocular visual odometry [J]. arXiv preprint arXiv:1611.06069.

[11] Szegedy C, Vanhoucke V, Ioffe S. (2016) Rethinking the inception architecture for computer vision [C]. IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, pp. 2818-2826.

[12] Day M J, Horzinek M C, Schultz R D. (2007) Guidelines for the vaccination of dogs and cats. Compiled by the Vaccination Guidelines Group (VGG) of the World Small Animal Veterinary Association (WSAVA).[J]. Journal of Small Animal Practice, 48(9):528-41.

[13] Chang J R, Chen Y S. (2015) Batch-normalized maxout network in network [J]. arXiv preprint arXiv:1511.02583,

[14] Li M, Zhang T, Chen Y. (2014) Efficient mini-batch training for stochastic optimization[C]. International conference on Knowledge discovery and data mining. New York. pp. 661-670.

[15] Ben G, Shahram I,Jamie S, Antonio C. (2013) Real-Time RGB-D Camera Relocalization https://www.microsoft.com/en-us/reserch/project/rgb-d-dataset-7scenes/

[16] Kendall A, Grimes M, Cipolla R. (2015) Posenet: A convolutional network for real-time 6-dof camera relocalization. http://mi.eng.cam.ac.uk/projects/relocalisation/

[17] Shotton J, Glocker B, Zach C. (2013) Scene coordinate regression forests for camera relocalization in RGB-D images [C]. IEEE Conference on Computer Vision and Pattern Recognition. Portland, OR. pp. 2930-2937.