IOP Conference Series: Materials Science and Engineering

Research on Pose Estimation of Mobile Robot Based on Convolutional Neural Network

To cite this article: Min Yue *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **565** 012013

Research on Pose Estimation of Mobile Robot Based on Convolutional Neural Network

Min Yue[1*], Guangyuan Fu[1] and Ming Wu[1]

[1]Xi’an research institute of high-tech, Shanxi, 710025, China

[*]Email: minyue999@126.com

Abstract. In this paper, we propose a new learning scheme for generating camera pose to be used for visual odometry. Pose estimation of mobile robot in unknown environment is formulated as a learning problem. We train a convolutional neural network end to end to compute the 6-dof pose of the robot from a series of images. The architecture is a kind of residual network with multiple attention resblocks, which can give features different weights in order to improve the accuracy of pose prediction. The network estimates ego-motion taking monocular image sequences as input instead of separate images. We call this novel network architecture Posenet. Compared to traditional methods, results are more accurate and in which we can see the potential of pose estimation methods with convolutional neural network.

1. Introduction

Pose estimation is a part of the front-end (visual odometry) in the robot's simultaneous localization and mapping problem [1] which is a necessary prerequisite and preparation for the robot to move autonomously and understand the unknown world. Based on pose estimation, the robot can construct the map of its environment without any sensors other than a camera, thus completing various tasks in many fields. In recent years, SLAM has received more and more attention. There are two main methods to solve SLAM problem of mobile robots, namely filtering method [2-3] and keyframe-based method [4]. In the early SLAM field, the filtering method is mainstream and widely used, but it has some disadvantages such as huge computational cost and tendency to drift. Therefore, recently keyframe-based slam has become more and more popular. We do not need to process every frame of the image sequences so that this kind of method can improve the real-time performance. PTAM [5] is the most famous representative, which divides slam problem into two threads for the first time.

At the same time, some novel methods to solve the pose estimation in SLAM problem have appeared, which are combined with other fields. With the rapid development of deep learning, many fields, such as image recognition and image segmentation [6-8], have made good progress by deep conventional neural network, and the combination of deep learning and SLAM has become one of the research hotspots in the field of robotics. Pose estimation method based on convolutional neural network has attracted more and more attention due to its potential in learning ability and robustness in complex environment.

In this paper, we propose a pose estimation method of monocular robot based on deep learning. The proposed method utilizes residual network to learn the geometric relationship between the image sequences, and the network called Posenet operates in an end-to-end manner, as shown in Figure 1. There are three contributions in this paper: 1) a residual network structure with attention resblock is proposed to improve the accuracy of pose estimation. 2) taking image sequence as input rather than a single image is more conducive for neural network to model features which have temporal characteristics. 3) it proves the application potential of deep learning in robot pose estimation and even slam problem.

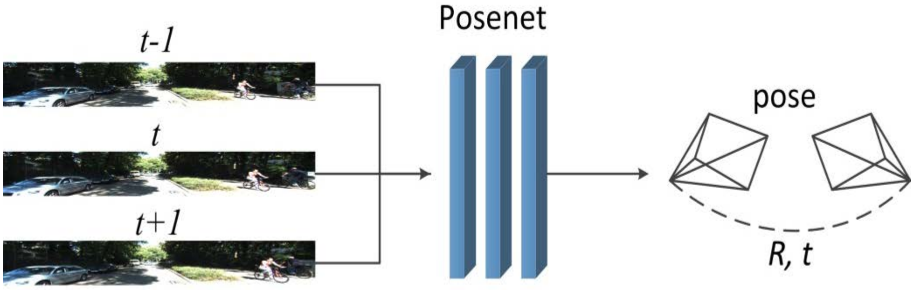

Figure 1. Overview of the proposed Posenet system.

2. Network Architecture

The structure of pose estimation network (Posenet) constructed in this paper is shown in Figure 2. Continuous monocular images are used as input to generate 6-dof pose output. The posenet is a convolutional neural network structure based on ResNet. The original resblock is replaced by the attention resblock to provide weight information for the feature map. It needs three consecutive monocular frames of the image as input in order to predict the pose transformation of intermediate image to the former and the latter frame respectively. Besides, this paper uses rotation and translation errors respectively to construct the loss function and minimize it. The network estimates the 6-dof pose through the end-to-end manner.

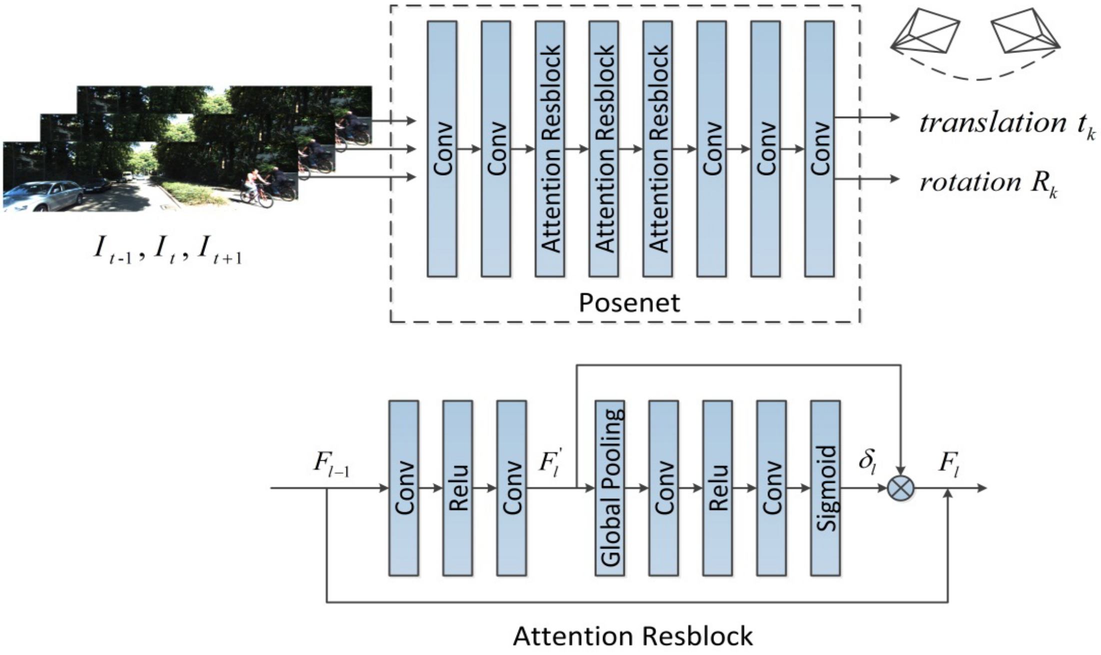

Figure 2. Architecture of the proposed Posenet system.

2.1. Attention Resblock

Different from the traditional resblock in ResNet, the attention resblock can add different weights to the channels in feature map obtained by convolution. Therefore, it is very suitable to deal with problems in which features do not need to be extracted from all pixels of the image or there are some invalid and interfering parts in the image. In the process of pose estimation using convolutional neural network, we take the whole image as the input. However, we can find that there are some pixels or parts of the image that are invalid in pose calculation, such as moving cars and objects that cannot always be photographed in adjacent images as seen in figure 3. These parts interfere with the learning process of the network. Therefore, attention resblock can reduce or eliminate the influence of these parts by adding weights to feature channels.

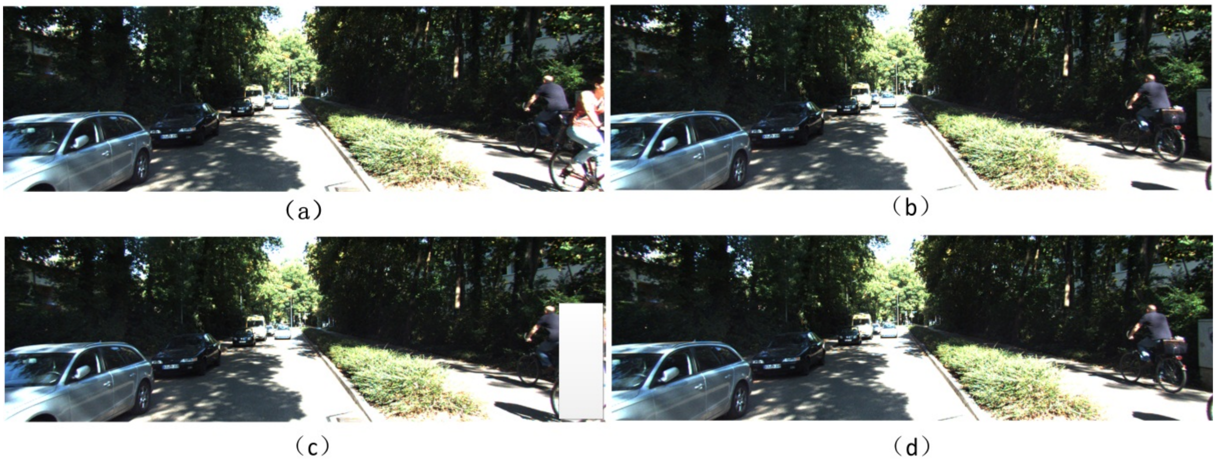

Figure 3. The invalid or interfering parts in the image. Figure (a) and (b) are the same as figure (b) and (d) and (b) and (d) are the next frames of (a) and (c). The girl on the bike covered by a white square in figure (c) is invisible in figure (b) and (d). Besides, the two people on the bike are both moving which would interfere with the pose estimation.

The structure of attention resblock is shown in Figure 2. In the paper [9] and [10], the authors use it to solve the problem of image super-resolution. In fact, the nature of attention resblock is the CA module (channel attention mechanism) [11]. For some specific problems, the importance of each channel is different. We can increase the weight of important channels and decrease the weight of less important channels through the attention mechanism. Let $F_l(l=1,\dots,L)$ denote the intermediate feature in the l^{th} attention resblock. The channel attention coefficient δ_l can be calculated as

$$\delta_l = F_1(F_2(F_3(F_l^{'}))),$$
$$F_3(F_{l,c}^{'}) = \frac{1}{h\times w}\sum_{i=1}^{h}\sum_{j=1}^{w}F_{l,c}^{'}(i,j), \tag{1}$$

where F_1 and F_3 denote the sigmoid function for normalization and the global average pooling function respectively, F_2 denotes the combining operation which includes two convolutional layers with a 1×1 kernel size and a ReLu activation function in between, $F_{l,c}^{'}(i,j)$ denotes the value at position (i,j) in the c^{th} channel of the intermediate feature in the l^{th} attention resblock, *h* and *w* are the shape of the feature map.

After obtaining the channel attention coefficient δ_l, the output of the l^{th} attention resblock F_l can be calculated as

$$F_l = F_{l-1} + \delta_l \cdot F_l^{'}. \tag{2}$$

2.2. Posenet

The Posenet proposed in this paper is mainly based on ResNet, and its structure is shown in Figure 2. The network takes the original image sequence as input rather than the preprocessed images (such as optical flow images) to learn and predict the camera's pose transformation. Specifically, we need to combine each frame image with its previous and subsequent images as input, as shown in figure 4, that is I_{t-1}, I_t, I_{t+1}, to learn and predict the pose transformation of the camera at time t to time $t-1$ and time $t+1$ respectively which are represented by $T_{t\to t-1}$ and $T_{t\to t+1}$. Using $(h, w, 3)$ to represent the shape of a single image, the shape of the input image is $(h, 3\times w, 3)$. The network has 5 convolutional layers and 3 resblocks, and two 6-dof pose can be obtained after the last convolution.

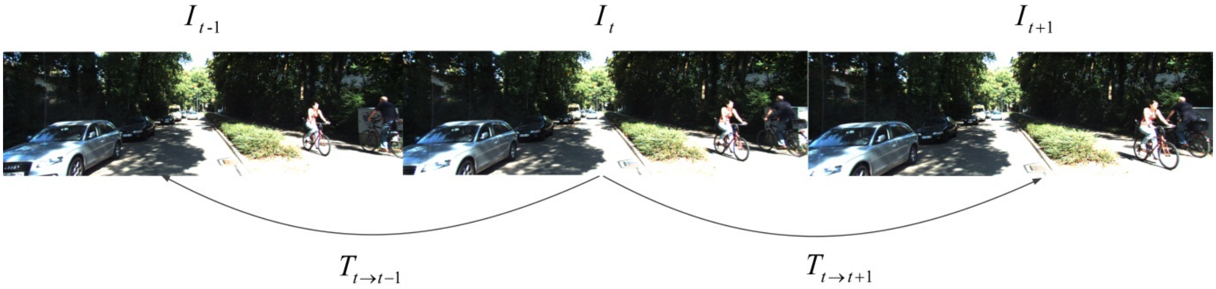

Figure 4. The input image sequence.

3. Training Procedure

3.1. Loss functions

The pose of robot can be divided into rotation and translation. We calculate the errors of these two parts respectively and the final loss function L_{motion} for the motion vectors is

$$\begin{aligned} L_{motion} &= loss(T_{t\to t-1}) + loss(T_{t\to t+1}), \\ loss(T) &= \left\| R_{gt} - R_{pred} \right\|_2 + \left\| t_{gt} - t_{pred} \right\|_2 . \end{aligned} \tag{3}$$

In equation (3), R_{gt} 和 t_{gt} represent the angle of the rotation and the normalized translation ground truth which matches $\|t\|_2 = 1$, R_{pred} 和 t_{pred} denote the 6-dof prediction of motion respectively. These four variables all have three parameters, R_{gt} and R_{pred} both corresponding to $[r_x, r_y, r_z]$ and t_{gt} and t_{pred} both corresponding to $[t_x, t_y, t_z]$.

3.2. Training Schedule

The network training is based on the tensorflow framework with Ubuntu 16.04 system. We train the model from scratch with Adam [12] using a momentum of 0.9 and a weight decay of 0.0002. All of the experiments are based on monocular image sequences. In order to adapt to the requirements of the convolutional neural network for the input image size, this paper carries out a unified operation on all the images in the training process and adjusts the image size. So the shape of the combined three image frames input is $128\times 1248\times 3$, and different kinds of data enhancement methods are used to improve performance and reduce possible overfitting, such as selecting images for scaling randomly. Finally, the posenet is trained for 300k iterations with all other weights fixed.

4. Experiments

4.1. Datasets

We use KITTI dataset as the training and evaluation dataset which is the most common benchmark used in prior work for evaluating the accuracy of pose prediction. The KITTI dataset includes a full suite of data sources such as stereo video, 3D point clouds from LIDAR, and the vehicle trajectory.

Kitti contains real image data from urban, rural and highway scenes, with up to 15 cars and 30 pedestrians in each image, as well as varying degrees of occlusion and truncation. We select the images of camera 02 in KITTI odometry dataset and the first eight sequences as the training set and the last two as the validation set. The number and size of image frames are shown in Table 1.

Table 1. KITTI odometry benchmark sequence 00-10.

Number	**Sequence name**	**Image numbers**	**Image size**
00	2011_10_03_drive_0027	4541	1241,376
01	2011_10_03_drive_0042	1101	1241,376
02	2011_10_03_drive_0034	4661	1241,376
03	2011_09_26_drive_0067	801	1241,376
04	2011_09_30_drive_0016	271	1226,370
05	2011_09_30_drive_0018	2761	1226,370
06	2011_09_30_drive_0020	1101	1226,370
07	2011_09_30_drive_0027	1101	1226,370
08	2011_09_30_drive_0028	4071	1226,370
09	2011_09_30_drive_0033	1591	1226,370
10	2011_09_30_drive_0034	1201	1226,370

4.2. Evaluation of Pose Estimaiton

After training the Posenet, the ego-motion of robot is learned from monocular images. Table 2 reports the ego-motion accuracy of our modes over two sample sequences from the KITTI odometry dataset. The evaluation index is absolute trajectory error(ATE). We can find out that our proposed method outperforms the traditional methods of ORB-SLAM [13] (short).

Table 2. Absolute Trajectory Error (ATE) on the KITTI odometry dataset.

Method	Seq. 09	Seq. 10
ORB-SLAM(full)	0.014+0.008	0.012+0.011
ORB-SLAM(short)	0.064+0.141	0.064+0.130
Ours	0.023+0.027	0.021+0.016

5. Conclusions and Future Work

In this paper, we propose the Posenet, a pose estimation method for mobile robot based on deep learning, and attention resblock is used to distinguish the weights of features extracted from images. The input of the network is monocular original image sequence with three consecutive frames instead of a single or preprocessed image. The proposed method performs better and has more potential than the traditional structure from motion. The next step is focus on improving the robustness and training efficiency of the network.

References

[1] Liu H, Zhang G and Bao H 2016 A survey of monocular simultaneous localization and mapping *J. Comp-Aid. Des. Comp. Graph.* **28** 855-868

[2] Davison A J, Reid I D, Molton N D and Stasse O 2007 Monoslam: real-time single camera slam *IEEE Trans. Pat. Anal. Mach. Intel.* **29(6)** 1052- 1067

[3] Montiel J M M, Civera J and Davison A J 2006 Unified inverse depth parametrization for monocular slam *IEEE. Trans. Robot.* 10.15607/RSS.2006.II.011

[4] Mouragnon E, Dekeyser F, Sayd P, Lhuillier M and Dhome M 2006 Real Time Localization and 3D Reconstruction *Proc. IEEE. Comp. Soc. Conf. Comp. Vis. Pat. Rec.* **1** 363- 370

[5] Klein G 2007 Parallel tracking and mapping for small AR workspaces *Proc. Sixth. IEEE. ACM. Int. Sym. Mix. Aug. Real.* 225-234

[6] Nekrasov Vladimir, Shen Chunhua and Reid Ian 2018 Light-Weight RefineNet for Real-time semantic segmentation

[7] Chen L C, Papandreou G, Kokkinos I, Murphy K and Yuille A L 2016 Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs *IEEE. Trans. Pat. Anal. Mach. Intel.* **40(4)** 834-848.

[8] Ren S, He K, Girshick R and Sun J. 2015 Faster r-cnn: towards real-time object detection with region proposal networks *IEEE. Trans. Pat. Anal. Mach. Intel.* **39(6)** 1137-1149.

[9] Zhang Y, Li K, Li K, Wang L, Zhong B and Fu Y 2018 Image super-resolution using very deep residual channel attention networks

[10] Shi Z, Chen C, Xiong Z, Liu D, Zha Z J and Wu F 2018 Deep Residual Attention Network for Spectral Image Super-Resolution *Eur. Conf. Comp. Vis. Spr. Cham.*

[11] Hu J Shen L, Albanie S, Sun G and Wu E 2017 Squeeze-and-excitation networks Spectral Image Super-Resolution *Eur. Conf. Comp. Vis. Spr. Cham.*

[12] Kingma D P and Ba J 2014 Adam: a method for stochastic optimization *Int. Conf. Learning. Representations.*

[13] Mur-Artal R, Montiel J M M and Tardos J D 2015 ORB-SLAM: a versatile and accurate monocular slam system *IEEE. Trans. Robot.* **31(5)** 1147-1163