Possibility estimation of 3D scene reconstruction from multiple images

E A Dmitriev[1], V V Myasnikov[1,2]

[1]Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086
[2]Image Processing Systems Institute of RAS - Branch of the FSRC "Crystallography and Photonics" RAS, Molodogvardejskaya street 151, Samara, Russia, 443001

e-mail: DmitrievEgor94@yandex.ru, vmyas@geosamara.ru

Abstract. This paper presents a pixel-by-pixel possibility estimation of 3D scene reconstruction from multiple images. This method estimates conjugate pairs number with convolutional neural networks for further 3D reconstruction using classic approach. We considered neural networks that showed good results in semantic segmentation problem. The efficiency criterion of an algorithm is the resulting estimation accuracy. We conducted all experiments on images from Unity 3d program. The results of experiments showed the effectiveness of our approach in 3D scene reconstruction problem.

1. Introduction

3D-scene reconstruction is a classic computer vision problem. Algorithms for 3D-scene reconstruction are prevalent in many spheres like robotics, architecture, design, Earth remote sensing, automated driving systems.

There are several methods for solving considered problem [1, 2]. Binocular stereo vision is one of such methods [1]. This method calculates a disparity between conjugate points on rectified stereo images. The main problem is to find conjugate points. A possible solution is searching key points on stereo images, then getting points descriptors and matching points by metrics values between descriptors [4]. There are more modern approaches that use convolutional neural networks [3, 4, 5].

3D-scene reconstruction using multiple images is a computationally expensive problem [2]. The current level of technology development doesn’t make it possible to reconstruct 3D-scenes in real time with good quality.

This article proposes an algorithm for possibility estimation of 3D-scene reconstruction using several frames of a video sequence in real time. This procedure helps to evaluate images number to get good quality 3D points cloud. The algorithm estimates conjugate pairs number from multiple images using a deep convolutional neural network. The article presents a model of a neural network with a quite small amount of weights. The model is possible to use on mobile devices with a graphical accelerator in real time. We conducted all experiments on images from Unity 3d program.

The article is structured as follows: the second section describes the main terms. Next section describes the model of neural network. The fourth section presents the results of experiments. Finally, we summarize results and tell about future researches.

2. Main terms

Let $I_k^s(n_1,n_2)$ be an RGB image from camera k and scene s, where $(n_1,n_2)\in\mathbf{D}$, $\mathbf{D}=\{(n_1,n_2):n_1=\overline{0,N_1-1},n_2=\overline{0,N_2-1}\}$, $k=\overline{0,K-1}$, $s=\overline{0,S-1}$, N_1,N_2 are height and width of image from camera, K is number of cameras on scene and S is number of scenes. Every scene differs from each other by objects types or relative positions of objects. Let l be the index of the fixed camera and we call the image from this camera as *a relative image*. Let $R_k^s(n_1,n_2)$ be discrete function whose values are points coordinates in space. Every value of $R_k^s(n_1,n_2)$ is projected on the correspondent position of the image plane $I_k^s(n_1,n_2)$. $R_l^s(n_1,n_2)$ is a function whose values are projected on relative image s. To form elements of train and test datasets we consider the following function:

$$P_j^s(n_1,n_2)=\begin{cases}0, R_j^s(n_1,n_2)\neq R_l^s(n_1,n_2)\\ 1, R_j^s(n_1,n_2)=R_l^s(n_1,n_2)\end{cases} \qquad (1)$$

Let $X^G=(x_i,y_i)_{i=0}^{G-1}$ be a dataset, x_i is a tensor, passed through the neural network, y_i is a label tensor and G is the size of the dataset. We form tensor x_i by concatenating $m<K$ different images $I_k^s(n_1,n_2)$ with the relative image from scene s. We choose m less than K in order to form more input and label tensors from one scene. Number of elements from one scene is C_{K-1}^{m-1}.

To get label tensor we consider the following function:

$$A_l^s(n_1,n_2)=\sum_{\substack{j=0\\ j\neq l}}^{m-1}P_j^s(n_1,n_2) , \qquad (2)$$

where j is index of camera. $A_l^s(n_1,n_2)$ values show frames number in set of m images from input tensor (without considering relative image), that contain projection of $R_j^s(n_1,n_2)$ point. We represent all values of $A_l^s(n_1,n_2)$ function in one hot encoding with m bits to get final y_i label tensor. Our task is similar to semantic segmentation task or pixel classification problem.

3. Model description

We considered several fully convolutional neural networks whose output tensor has the same width and height as input tensor [6]. Such networks are used in semantic segmentation task and show good performance. Some of these networks are U-Net [7], SegNet [8]. These networks have comparable number of weights.

Another considered network is LinkNet [9]. This model exploits all features of U-Net and has a smaller number of weights. Specific feature of LinkNet is using several encoders and decoders. Original model consists of four blocks with 11.5 million parameters number.

In this work we use 3 decoder and encoder blocks to make network run faster in real time. We reduced size of max pooling kernel to 2×2 with stride 1 instead of kernel $3\times3$with stride 2. This change in size of max pooling kernel doesn't allow feature maps to decrease fast to save more information. Our model has 3 million number of parameters. Figure 1 shows proposed model while figure 2 and figure 3 demonstrate architecture of encoder and decoder blocks respectively.

We used cross entropy as a loss function. According to [10], cross entropy loss function allows to get local minimum that gives bigger accuracy than mean square distance loss function using random weights initialization.

Let v be an output tensor of neural networks that has the same shape as label tensor. Loss function looks as follows:

$$H\big(y(n_1,n_2),v(n_1,n_2)\big)=-\sum_{i=0}^{m-1}y(n_1,n_2,i)\log v(n_1,n_2,i) . \qquad (3)$$

Loss on whole dataset can be calculated as follows:

$$Q(X^G) = -\frac{1}{G}\sum_{i=0}^{G-1}\sum_{j=0}^{N_1-1}\sum_{k=0}^{N_2-1}\sum_{t=0}^{m-1} y_i(j,k,t)\log\left(v_i(j,k,t)\right) \qquad (4)$$

We used an adaptive stochastic gradient descent Adam as optimization method [11, 12]. We also decreased learning coefficient value if loss on test set remained the same or less than on previous epoch.

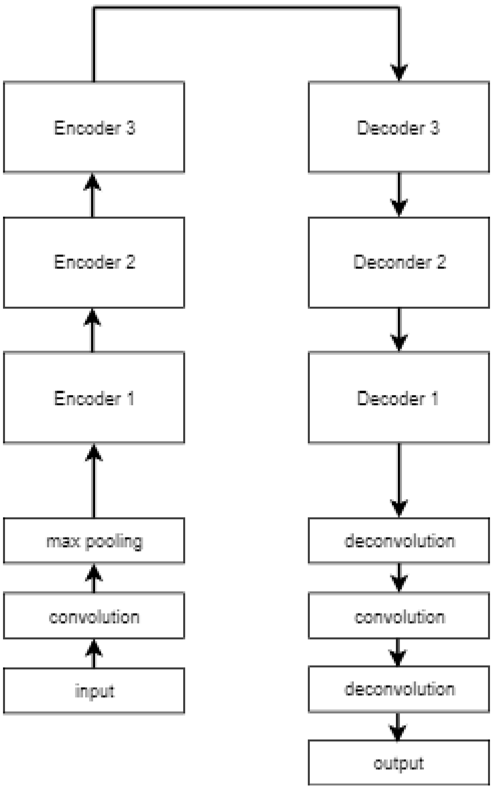

Figure 1. Neural network model.

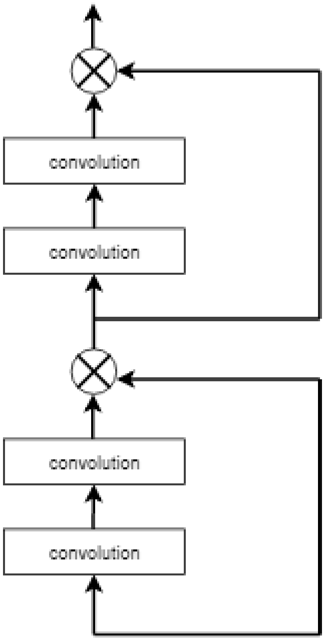

Figure 2. Encoder architecture.

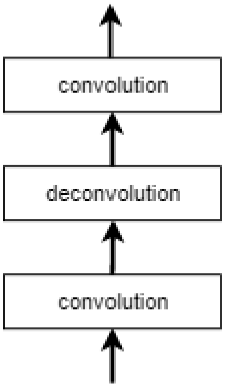

Figure 3. Decoder architecture.

4. Results of experiments

We trained and tested model on dataset of Unity 3d images. Number of cameras K was 8, number of scenes was 23 and number of RGB images m in input tensor was 5. Number of images in dataset was 805. Size of image was 300×300. We split images on 70/30 percent for train and test dataset respectively.

Input tensor contained 15 channels in third dimension. First 3 channels belonged to relative image. Relative image, with non-relative image, label tensor and output tensor are presented on figures 4, 5, 6, 7. Intensity on images 6 and 7 depends on number of conjugate pairs in input tensor.

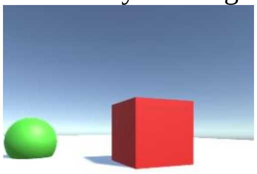

Figure 4. Relative image.

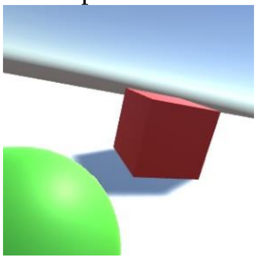

Figure 5. Non-relative image.

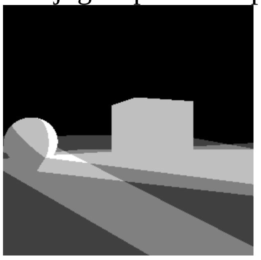

Figure 6. Label tensor.

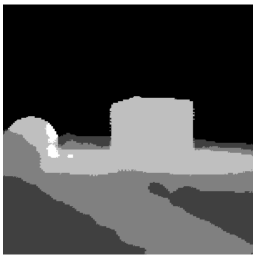

Figure 7. Output tensor.

We used accuracy metric for results estimation. Accuracy metric looks as follows:

$$M = \frac{1}{N_1 N_2 O} \sum_{t=0}^{O-1} \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} \left(\arg\max_u v_t\left(i, j, u\right) == \arg\max_l y_t\left(i, j, l\right) \right), \quad (5)$$

where O is number of images in test dataset, u and l are indexes for output and label tensors respectively.

After training neural network on train dataset accuracy on test dataset was 0.96.

5. Conclusions

In this paper, we presented a new approach for possibility estimation of 3D-scene reconstruction using convolutional neural network. Our model can estimate conjugate pairs number from multiple images.

We conducted experiments and showed effectiveness of our approach. The aim of future researches to propose a method for camera-world rotation matrix and translation vector estimation.

6. References

[1] Horn B 1986 *Robot Vision* (Cambridge: MIT Press)

[2] Choy C B, Xu D, Gwak J Y, Chen K and Savarese S 2016 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **9912 LNCS** 628-644

[3] Savchenko A V 2017 Maximum-likelihood dissimilarities in image recognition with deep neural networks *Computer Optics* **41(3)** 422-430 DOI: 10.18287/2412-6179-2017-41-3-422-430

[4] Lowe D G 2004 Distinctive image features from scale-invariant keypoints *International Journal of Computer Vision* **60** 91-110

[5] Žbontar J and Le Cun Y 2015 Computing the stereo matching cost with a convolutional neural network *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1592-1599

[6] Shelhamer E, Long J and Darrell T 2017 Fully Convolutional Networks for Semantic Segmentation *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39** 640-651

[7] Ronneberger O, Fischer P and Brox T 2015 U-net: Convolutional networks for biomedical image segmentation *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **9351** 234-241

[8] Badrinarayanan V, Kendall A and Cipolla R 2017 SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39** 2481-2495

[9] Chaurasia A and Culurciello E 2018 LinkNet: Exploiting encoder representations for efficient semantic segmentation *IEEE Visual Communications and Image Processing* 1-4

[10] Golik P, Doetsch P and Ney H 2013 Cross-entropy vs. Squared error training: A theoretical and experimental comparison *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* 1756-1760

[11] Kingma D P and Ba J 2014 Adam: A Method for Stochastic Optimization *ArXiv: 1412.6980*

[12] Nikonorov A V, Petrov M V, Bibikov S A, Kutikova V V, Morozov A A and Kazanskiy N L 2017 Image restoration in diffractive optical systems using deep learning and deconvolution *Computer Optics* **41(6)** 875-887 DOI: 10.18287/2412-6179-2017-41-6-875-887

Acknowledgements

The reported study was funded by RFBR according to the research projects 18-01-00748, 17-29-3190.