A Hybrid 2D and 3D Convolution Neural Network for Stereo Matching

Xuan Zeng, Yewen Li, Ziqian Chen, Liping Zhu *
School of Information and Engineering
Minzu University of China
Beijing, China
zlp6681@sina.com

***Abstract*—Stereo matching plays an important role in computer vision and SLAM (simultaneous localization and mapping). In this paper, we propose a novel hybrid 2D and 3D convolution neural network for stereo matching. Unlike existing similarity metric based stereo matching methods which need extra post-processing to finish the matching pipeline, the proposed approach is an end-to-end stereo matching method and it needs much less time for an image pair. Unlike a lot of cost volume and disparity based stereo matching methods which are too complicated to run on performance-constrained devices, the proposed method is much more simple and can run on the real-time sweeping robot that we build. Experimental results on two widely used stereo matching datasets verified the effectiveness of the proposed approach, meanwhile, our real-time SLAM system—the sweeping robot demonstrates that our method can apply to real-time applications.**

***Index Terms*—stereo matching, 3D convolution, SLAM**

I. Introduction

Stereo matching and SLAM have been widely studied in computer vision and pattern recognition communities due to their importance in many real-world applications, such as autonomous vehicles and intelligent robot [1]–[6]. As a critical pre-step of SLAM, stereo matching has drawn a lot of research attention in recent years [7]–[11]. Various stereo matching methods have been proposed, and they can be roughly categorized into two types: Similarity metric based methods and Cost volume & disparity regression based methods.

1) Similarity metric based stereo matching methods [12], [13]. Since the input of stereo matching is two similar images, Siamese Network is born for this problem. Firstly, these methods use Siamese Network to extract features from the two input images, in which 2D convolution is a standard process. Secondly, they apply a similarity metric to extracted depth information. This similarity metric can compute the matching score of the left and right feature maps, based on the matching score the depth image can be obtained. For example, the method presented in [12] exploits the Siamese Network to extract features from the input image, in addition, it adds an inner product layer to compute the matching score of the left and right feature maps. The method proposed by Jure Zbontar and Yann LeCun applies the convolution neural based Siamese Network to extract image features too, as for the similarity metric, it exploits the dot product and the cosine similarity [13].

2) Cost volume & disparity regression based methods [7]–[9]. More recently, Cost volume and & disparity regression based methods were presented, like in Similarity metric based stereo matching methods, these methods use the Siamese Network to extract features too. However, they exploit cost volume which is based on 3D convolution. Besides, they apply more powerful disparity regression to get the depth image. For example, Alex Kendall et al. proposed an end-to-end of learning geometry and context deep neural network for stereo matching, in which they proposed the cost volume and disparity regression approach. What's more, this method was the state-of-the-art method at that time [8]. More recently, Jia-Ren Chang and Yong-Sheng Chen proposed Pyramid Stereo Matching Network [7], which not only takes advantages of the cost volume and disparity but also exploits the pyramid pooling module and encoder-decoder architecture. This method is the state-of-the-art method on the KITTI and Scene Flow stereo matching data set by June 2018. Lidong Yu et al. presented a deep stereo matching method which applies a sub-architecture and end-to-end trainable pipeline. Besides, they also exploit the cost volume and disparity regression and get very good results [9].

A. Motivation

Although improved performance has been reported in the existing stereo matching methods, there still exists much room for improvement. Specifically, there exist the following two shortcomings:

1) Similarity metric based stereo matching methods are not end-to-end methods, some post-processing is still needed, for example, the smoothing process, which will make the pipeline cost more time and difficult to apply in real-time applications. e.g., the method proposed in [13] requiring a minute of GPU computation per image pair, which is far from the need of the real-time application.

2) Cost volume & disparity regression based methods are too complicated and computationally expensive, making them impossible to run on small, performance-constrained devices. For example, the network proposed in [8] and [9] both have 37 layers, the network used in [7] has fewer layers, but the

*Corresponding author.

978-1-5386-7649-3/18/$31.00 ©2018 IEEE
DOI 10.1109/CSE.2018.00028

number of layers still over 30. They are too complicated for performance-constrained devices.

Motivated by the above analysis, we intend to design an approach, which not only is an end-to-end stereo matching method but also should be small enough and can run on performance-constrained devices, such as sweeping robot and miniature drone.

B. Contribution

The major contributions of this paper are summarized as the following two points:

1) In this paper, we proposed a hybrid 2D and 3D convolution stereo matching method, which not only is an end-to-end deep stereo matching but also small enough and can run on performance-constrained devices.

2) We verified the effectiveness of our method on the public KITTI-2012 and KITTI-2015 stereo matching datasets. Experimental results demonstrate that our approach is feasible and can run on the sweeping robot that we developed.

C. Organization

The remainder of this paper is organized as follows. The details of proposed method are provided in Section II. Experimental results are presented in Section III, followed by our conclusions in Section IV.

II. Proposed Method

In this section, we will describe the details of our approach, First, we will detail how we use Siamese Network and 2D convolution to extract features from the input image pairs. Second, we will move to the 3D convolution for depth information extraction. Third, the dilation 3D convolution and implementation detail will be provided.

A. Feature Extraction with Siamese Network and 2D convolution

We use deep learning method to perform stereo matching, like others, we apply Siamese Network and 2D convolution to extract features from two cameras. Since the input of the model is two similar images, it is natural to think of using Siamese Network to extract features. For each subnetwork, like in [7], we use Resblock with 2D convolution to extract features from the input images, and these two subnetworks share parameters with each other. Different from traditional convolution neural network, which use pooling layer to reduce the influence of feature position on the result, we remove the pooling layer to keep the position information. The architecture of our network is shown in Fig.1.

B. 3D Convolution for Depth Information Extraction

Different from recent works [7]–[9], in which, they applied the cost volume and disparity regression in stereo matching, we use more simple 3D convolution. In this paper, we exploit 3D convolution to extract the depth information from the two sets of feature maps, which have been obtained from the feature extraction part. After passing through the feature extraction subnetwork, each input image is represented by a

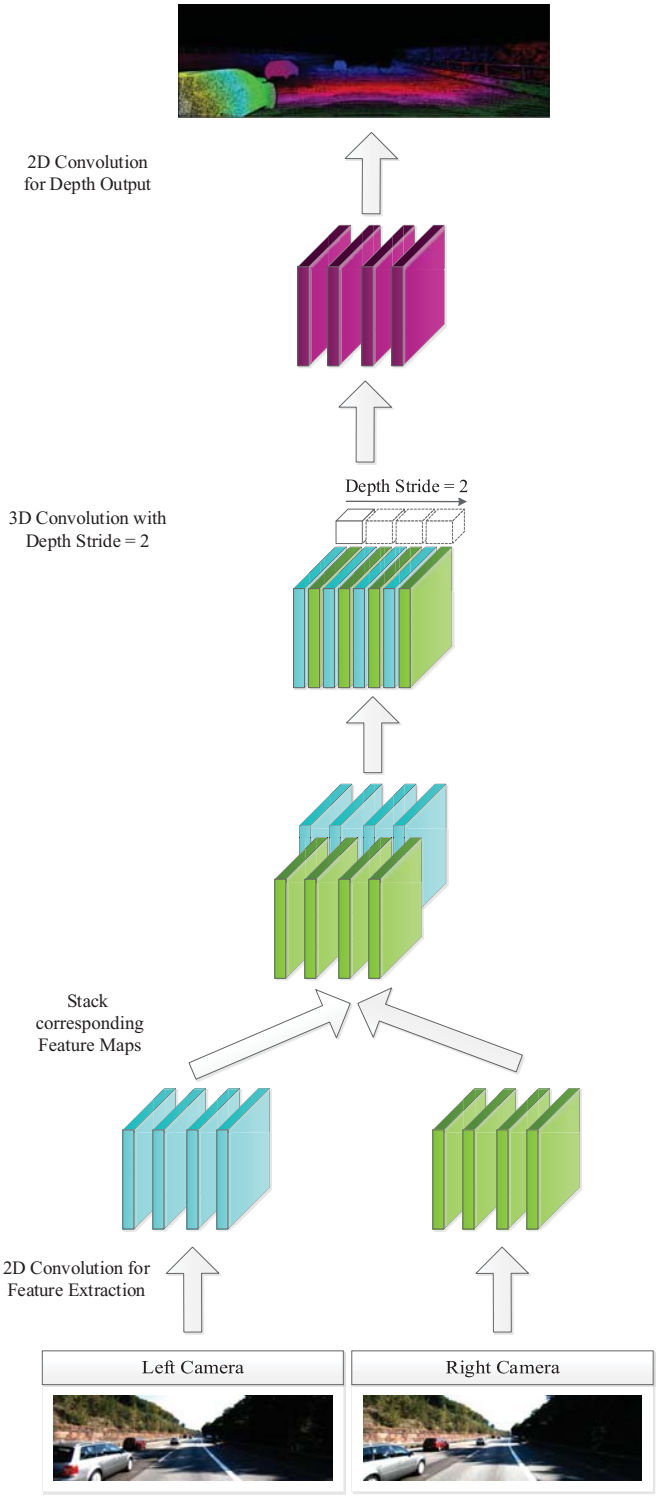


Fig. 1: A hybrid 2D and 3D Convolution Network for Stereo Matching. First of all, 2D convolution was used to extract features from the input images. Then, We stack the corresponding left and right feature maps and apply 3D convolution to extract depth information from them. Finally, we use 2D convolution to get the depth image.

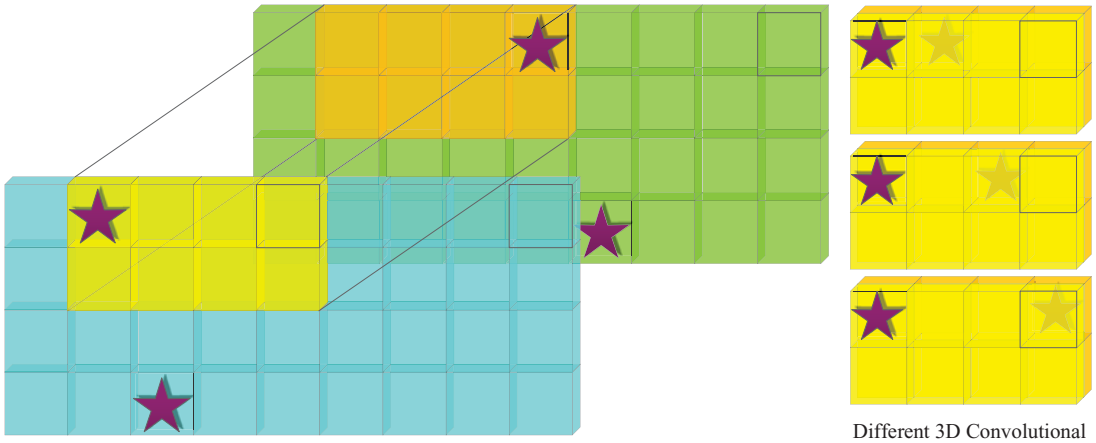


Fig. 2: 3D Convolution for Depth Information Extraction. A pair of patterns that in left and right feature map — in this figure, the pattern is pentagram — can be activated by a certain 3D convolution kernel, which is shown in the right of the figure. Different kernels can predict different image depth. Based on the horizontal distance difference of this pair of patterns, the 3D convolution can extract depth information.

set of feature maps. In order to get the depth information of the input image, the feature maps from two different subnetworks should be take into account at the same time. Different from traditional deep learning method which use similarity metric to compute the similarity of different feature maps or recently proposed cost volume and disparity regression based method, we apply the 3D convolution to extract the depth information from two sets of feature maps.

Before 3D convolution, concatenation of two sets of feature maps is needed. However, we do not concatenate them together simply. As shown in Fig.2, we stack each corresponding feature together, which means stacking the feature maps that computed from the same convolution kernel.

The reason why we use 3D convolution is that different 3D convolution kernels can find different patterns in the stacked feature maps and extract depth features. Because a pair of same pattern located differently in left and right feature maps, there will be a horizontal distance difference between them. However, a certain 3D convolution kernel can find this horizontal distance difference, in other words, a pair of same pattern that in left and right feature maps will be activated by a certain 3D convolution kernel. Based on this horizontal distance difference, the depth features of corresponding position can be obtained and then the depth image can be computed from the depth features.

C. Dilation 3D Convolution and Implementation Details

In order to extend the receptive field of convolution neural network as well as to reduce the parameters of the 3D convolution layers, we exploit the dilation convolution and apply other tricks. The hyperparameters in the 3D convolution layer are specially customized for stereo matching. Firstly, we set the depth stride as 2. The reason behind this is we think it is easier for 3D convolution kernels to learn its parameters when only dealing with corresponding feature maps. Secondly, we use rectangle convolution kernel, like $width \times height \times depth = 16 \times 3 \times 2$. This is because, in the stereo matching dataset, one object appears in different positions in left and right images, and the vertical positions of the same object are almost same, while the horizontal positions vary a lot. Thirdly, for the sake of extending the receptive field of convolution kernel further, we exploit the dilation convolution. Since the 3D convolution and big convolution kernel will increase the computation cost and amount of parameters and it is reasonless to increase the size of convolution kernel blindly. In this paper, we exploit the dilation convolution, which can increase the receptive field and reduce the number of parameters meanwhile. In practice, we apply different size of dilation 3D convolution kernels to stacked features, then concatenate the output feature maps together.

For the final depth regression, we use 2D convolution again. After concatenating the output feature maps that obtained from the 3D dilation convolution, we recognize them as traditional feature maps from 2D convolution, then apply 2D convolution to them. Finally, these 2D convolution layers generate the depth image.

III. Experiments

In this section, we will introduce two experiments. Firstly, we will show the stereo matching result of our approach on KITTI datasets. Secondly, We will describe the SLAM result, which is based on the stereo matching and EKF-SLAM(Extended Kalman Filter-SLAM).

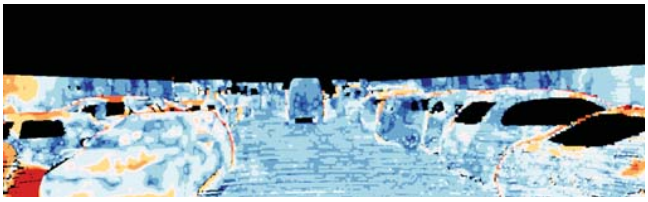
A. Result ON KITTI

Before applying to the real-world application, we verified our stereo matching approach on the KITTI dataset.

(a) Left Image

(b) Right Image

(c) Depth Image



(d) Error Image

Fig. 3: The stereo matching result on KITTI 2015 dataset. The depth image is computed from the left and right images. The depth image and the error image are generated by the KITTI development kit.

KITTI datasets are real-world street views from a driving car. KITTI-2015 dataset [11] contains 200 training stereo image pairs with sparse ground-truth disparities obtained using LiDAR and another 200 testing image pairs without ground-truth disparities. KITTI-2012 dataset [10] contains 194 training image pairs and 195 test pairs. The height of KITTI images is 376 and width is 1240.

For KITTI dataset, we did not use the full-size images, instead, we randomly crop the image to 512×256. Furthermore, for a better result, we only use the bottom part of the image. In other words, we only use the bottom 2/3 of the images. The reason behind this is the top of KITTI depth images have little depth information, which is due to the visual field limitation of the laser scanner used by KITTI. The stereo matching result of our method on the KITTI-2015 dataset is demonstrated in Fig.3. The depth image and the error image were generated by the KITTI development kit.

Compared with the state-of-the-art methods, there is still a gap between our results and their results, but our method is good enough on the sweeping robot.

After actual testing, the accuracy rate of our method was about 87.6% of that of SGBM algorithm, but it was able to achieve the stereo matching speed of 320*240 resolution and about 16-22fps on the Raspberry Pi 3 Model B, it has a 1.2GHz quad-core CPU, 1GB DDR2 RAM and dual-core VideoCore IV GPU. In the same case, SGBM was only 12-18fps, detailed comparison is shown in Fig.4. this means that the limited computing power devices, in combination with other high complexity algorithm (such as SLAM) will have higher availability.

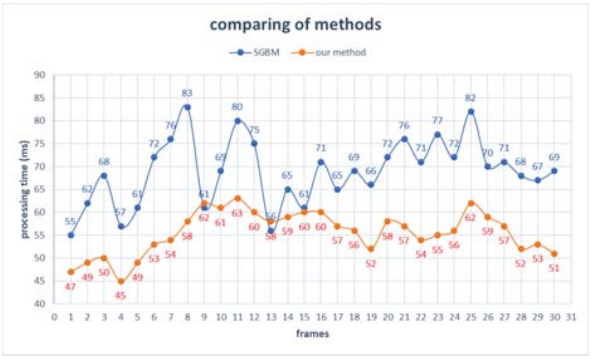


Fig. 4: Comparring of methods.

B. SLAM Result

Base on the deep stereo matching method and EKF-SLAM algorithm, we build our SLAM sweeping robot. The hardware we use is Raspberry Pi 3 Model B, We run PyTorch on it. The sweeping robot we built is shown in Fig.5.

Fig. 5: The sweeping robot we built.

As for the SLAM algorithm, we choose the EKF-SLAM algorithm. EKF-SLAM is a classic SLAM algorithm, and it is

fast enough to run on performance-constrained devices. Two indoor SLAM results are shown in Fig.6 and Fig.7.

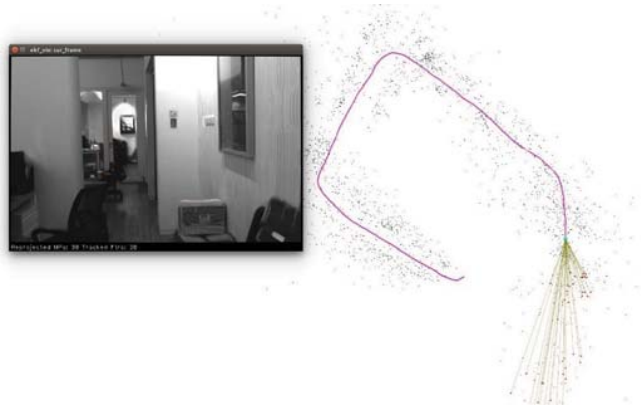


Fig. 6: An indoor SLAM result.

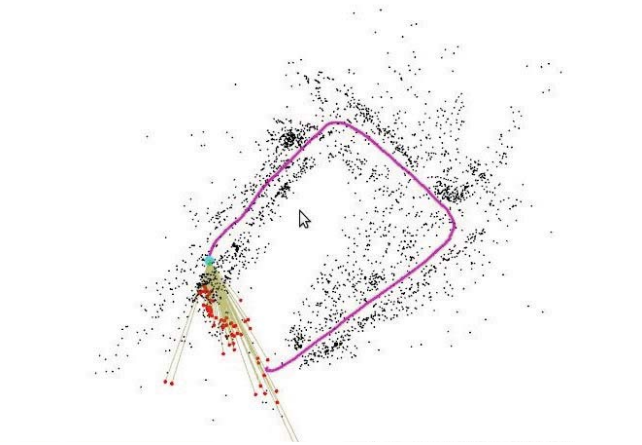


Fig. 7: An indoor SLAM result.

IV. Conclusion

In this paper, we investigate the stereo matching problem and propose a novel end-to-end 2D and 3D convolution neural network. Different from similarity metric based and cost volume & disparity regression based stereo method, the proposed approach employs 3D convolution for depth information extraction. Moreover, our method is much faster than a lot of existing stereo matching methods and can run on the performance-constrained devices. Experiment results on two publicly available datasets demonstrate the effectiveness of our method, and the SLAM based sweeping robot we build shows that our approach can apply to real-time applications. Due to the limitation of time, we did a few comparsions with other state-of-the-art methods, such as SGBM, in the future, we will refine our method and design more experiments.

Acknowledgment

The work described in this paper was supported by the National College Student Innovation Training Program under Project No.GCCX2018110031.

References

[1] K. Lee, S. H. Ryu, C. Nam, and N. L. Doh, "A practical 2d/3d slam using directional patterns of an indoor structure," *Intelligent Service Robotics*, vol. 11, no. 4, pp. 1–24, 2017.

[2] M. J. Tribou, D. W. Wang, and S. L. Waslander, "Degenerate motions in multicamera cluster slam with non-overlapping fields of view," *Image and Vision Computing*, vol. 50, pp. 27–41, 2016.

[3] T. Gulde, S. Kärcher, and C. Curio, "Vision-based slam navigation for vibro-tactile human-centered indoor guidance," in *European Conference on Computer Vision*. Springer, 2016, pp. 343–359.

[4] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *CVPR*, vol. 2, no. 6, 2017, p. 7.

[5] P.-H. Le and J. Kosecka, "Dense piecewise planar rgb-d slam for indoor environments," *arXiv preprint arXiv:1708.00514*, 2017.

[6] K. Tateno, F. Tombari, I. Laina, and N. Navab, "Cnn-slam: Real-time dense monocular slam with learned depth prediction," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6565–6574.

[7] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5410–5418.

[8] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," *CoRR, vol. abs/1703.04309*, 2017.

[9] L. Yu, Y. Wang, Y. Wu, and Y. Jia, "Deep stereo matching with explicit cost aggregation sub-architecture," *arXiv preprint arXiv:1801.04065*, 2018.

[10] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[11] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[12] W. Luo, A. G. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5695–5703.

[13] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *Journal of Machine Learning Research*, vol. 17, no. 1-32, p. 2, 2016.