Оптический журнал

Иконика

Depth map denoising using bilateral filter and progressive CNN$_S$

© 2020 Shuaihao Li*, Weiping Zhu*, Bin Zhang**, ***, Xinfeng Yang*, Min Chen*

*School of Computer Science, Wuhan University, Wuhan, China 430072
**School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China 430072
***Department of Computer Science, City University of Hong Kong, Hong Kong, China 999077

E-mail: chenmin@whu.edu.cn

Submitted 19.03.2019

DOI:10.17586/1023-5086-2020-87-06-51-56

With the advantages of low cost and real-time acquisition of depth and color maps of the object, Time-of-Flight (ToF) camera has been used in 3D reconstruction. However, due to the hardware shortage of ToF sensors, the depth maps obtained by ToF camera has a lot of noise, which limits its subsequent application. Therefore, it is necessary to denoise the depth maps by software method. We propose an algorithm for denoising depth maps by combining the bilateral filter and Progressive Convolution Neural Networks (PCNN). The algorithm takes a single depth map as input. Firstly, the first individual network of the PCNN is used to denoise the depth map, and then the bilateral filter and the second individual network of the PCNN are used to further process, so that the edge information of depth maps can be retained on the basis of fine denoising. Finally, we have carried out experiments on the popular Middlebury dataset. The experimental results show that the proposed algorithm is obviously superior to the traditional methods.

Keywords: *depth map denoising, 3D reconstruction, bilateral filter, Progressive convolution neural networks.*

OCIS codes: *100.0100 100.2980 100.6890.*

Устранение шумов в картах глубин с использованием билатерального фильтра и прогрессивных свёрточных нейронных сетей

© 2020 г. Shuaihao Li, Weiping Zhu, Bin Zhang, Xinfeng Yang, Min Chen

В силу невысокой стоимости и возможности работы в реальном времени, времяпролётные камеры широко используются для получения трёхмерных изображений. Тем не менее, из-за особенностей технического воплощения такие камеры обладают заметным уровнем шумов, что является препятствием их использования. Устранение влияния шумов выполняется путём соответствующей обработки изображений. Предложен алгоритм устранения шумов путём комбинации билатерального фильтра и прогрессивных свёрточных нейронных сетей (PCNN). Алгоритм стартует с единичной карты глубины. На первом шаге для устранения зашумлённости используется первая отдельная свёрточная сеть (PSNN), затем – билатеральный фильтр и вторая свёрточная сеть, в результате чего выделяется информация

о границах. Эксперименты, выполненные с использованием распространенной базы изображений Middlebury, показали очевидные преимущества предложенного алгоритма перед традиционными.

Ключевые слова: *устранение шумов, карты глубины, трёхмерные изображения, билатеральный фильтр, прогрессивная свёрточная нейронная сеть.*

1. INTRODUCTION

As a kind of depth camera, Time of Flight (ToF) camera [1] can obtain accurate depth information from scene to depth camera in real time, which makes it widely used in computer vision fields, such as virtual reality, augmented reality and 3D reconstruction. By combining with the Unmanned Aerial Vehicle (UAV), ToF camera can be used in rapidly 3D reconstruction of large outdoor scenes such as a building [2]. But the depth maps obtained by ToF camera usually contain a lot of noise, which seriously limit the subsequent steps of 3D reconstruction, such as point cloud computing, point cloud registration [3] and point cloud fusion [4], and ultimately limit the surface accuracy of the 3D model [5]. Therefore, the noise of the depth maps must be removed before the 3D reconstruction.

Denoising of depth maps has been one of the major challenges of computer vision in recent years, and many algorithms have emerged, the most representative of which are filtering methods: bilateral filter [6], mean filter [7], median filter [8] and Gaussian filter [9]. Among them, the bilateral filter can maintain the edge information of depth map, but it is poor in the detail retention property. In recent years, with the rise of deep learning research, deep learning-based denoising algorithms have emerged [10], which can maintain good details after denoising depth maps. As a kind of deep learning, the Progressive Convolution Neural Network (PCNN) is superior to its peers for its simple structure and remarkable denoising effect.

In this paper, a depth map denoising algorithm combining the bilateral filter and the PCNN is proposed. Initially, the first individual network of the PCNN is used for denoising the depth map, and then the bilateral filter is adopted to enhance the edge information, and the result is used as the input of, the second individual network of the PCNN to further denoise the details of the depth map. Our experimental results show that the proposed algorithm is superior to the traditional methods in the Peak Signal-to-Noise Ratio (PSNR) evaluation index and time consumption, and the improvement of visual effect is also obvious.

2. RELATED WORK

2.1. The bilateral filter

The bilateral filter [5] is a nonlinear filter composed of spatial filter kernel function and range filter kernel function. The main advantage of bilateral filter is that the two functions can be used for processing the different regions of a single map. In low-frequency regions, the pixel values usually change slowly, hence, spatial filter kernel function is commonly used to remove the noise. While in the high-frequency regions, the pixel values generally change rapidly and therefore requires range filtering to denoise. The edge information of depth map can also be well preserved by selecting different filter kernel functions for denoising in different regions.

The principle of bilateral filter is to suppress pixels that differ from the central pixel pixels. The output pixel value is determined by the weighted sum of the neighborhood pixel value, and its weight coefficient is composed of the spatial neighborhood weight and the value domain similarity weight. The weight coefficient w is expressed as:

$$w=\exp\left(-\frac{(i-k)^2+(j-l)^2}{2\sigma_s^2}-\frac{\|f(i,j)-f(k,l)\|^2}{2\sigma_r^2}\right), \quad (1)$$

where σ_s is the weight coefficient of Gaussian filter, and σ_r is the range similarity weight coefficient, while σ_s and σ_r together determine the performance of bilateral filter. The size of σ_s determines the relative spatial position of pixel points, while σ_r defines the values range of pixel points. $f(i, j)$ and $f(k, l)$ are the pixel values of point (i, j) and point (k, l), and (k, l) is the neighborhood pixel of point (i, j).

2.2. The Progressive Convolution Neural Network (PCNN)

The PCNN is a kind of convolutional neural network, which is composed of two individual convolution neural networks [11]. It uses the output of the previous network as the input to the next network. So, for the PCNN, the input of each individual network is different. The PCNN can gradually learn the high-frequency information of depth map for each individual network. Therefore, the similarity between the depth map output from each independent network and the ground truth depth map tested will be higher and higher, so that the denoising of the depth map will be realized finally.

In theory, each PCNN can be composed of multiple independent networks (i.e. SRCNNs [12]). If we do not freeze the weights of SRCNNs, the PCNN will be equal to a deeper SRCNN with layer number the same as the sum of a single SRCNN. Therefore, in practice, the PCNN consists of only two SRCNNS, and the weights of the first SRCNN will be frozen in the training of the PCNN. For example, in “PCNN(3 + 5)”, 3 + 5 refers to the network structure of the PCNN composed of two SRCNNs, 3 is the layers number of the first SRCNN (the weights of the SRCNN is frozen), and 5 represents the layers number of the second SRCNN (the weights of the SRCNN is learnable). Similarly, “PCNN (5 + 3)” refers to the network structure of the PCNN consisting of two SRCNNs. 5 is the layers number of the first SRCNN (the weights of the SRCNN is frozen), and 3 represents the layers number of the second SRCNN (the weights of the SRCNN is learnable).

In addition, the PCNN must be trained first before denoising depth maps, therefore it needs multiple training datasets to train each individual network independently. The mathematical model of the PCNN can be denoted as

$$\ddot{y}_i = F_{\{S_1,\cdots,S_M\}}(x_i) = S_1\left(S_2\left(\cdots S_M(x_i)\right)\right), \quad (2)$$

where x is a noisy map, y is a noise-free map. x_i and y_i are training samples, S_M denotes M convolutional neural networks, F is the output of S_M. $\ddot{y}_i$ is the predicted value of the i-th map output.

The PCNN framework is shown in Fig. 1 (excluding the bilateral filter).

3. THE DEPTH MAP DENOISING ALGORITHM COMBINING THE BILATERAL FILTER AND THE PCNN

3.1. Algorithm framework

Considering the bilateral filter’s capacity of retaining the edge details while denoising depth map and the PCNN’s global excellent performance in depth map denoising, we utilize these two together aiming to denoise depth map more efficiently. Figure 1 shows the output of the first individual network of the PCNN. The bilateral filter is then used to further denoise in order to enhance the edge details of the depth map. The result is subsequently used as input for the PCNN’s second individual network, to extract the noise that was not removed in the previous step.

3.2. Training individual network

For the proposed algorithm, the key to its implementation is the training of the PCNN, which includes the training of individual network and progressive network.

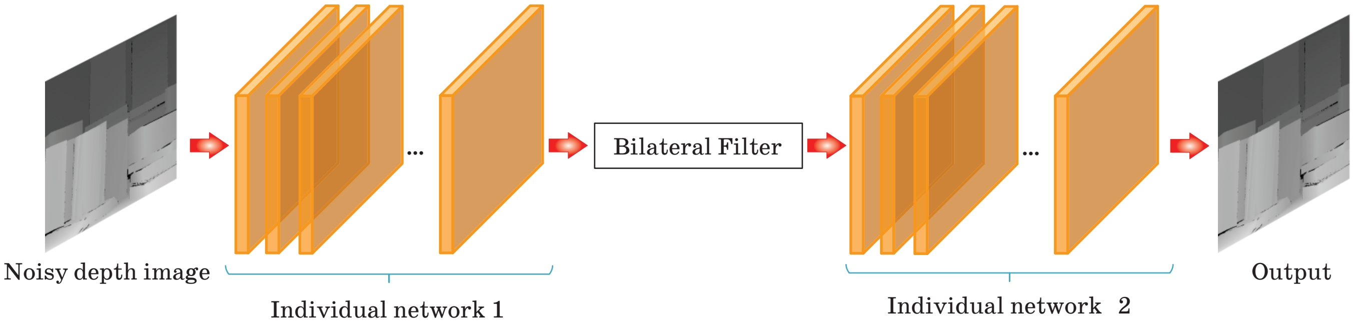


Fig. 1. The algorithm framework combining the bilateral filter and the PCNN.

Since the PCNN is composed of multiple convolutional neural networks, it is necessary to train individual convolution neural networks first. To train the network, we use the back-propagation (BP) algorithm [13]. The BP algorithm is one of the most effective learning methods for convolution neural networks. Its main characteristics are signal forward transmission and error backward propagation. By constantly adjusting the network weight value, the final output of the network is as close as possible to the expected output, so as to achieve the purpose of training.

We selected Middlebury dataset [14] for training and testing. For the training samples, we selected 75 depth maps randomly from Middlebury dataset. In order to improve the validity of the training data, we transformed and expanded the original training samples: first, we flipped the original 75 training samples vertically to obtain 150 maps, and then the 150 depth maps were rotated 90°, 180°, 270° and magnified 3–6 times. We finally get 1800 (i.e. 150×3×4) depth maps as the initial input of the network, that is, the training set.

We first train a three-layer SRCNN as the benchmark, with a network structure of 9–5–5. Where, the size of the convolution kernel in the first layer is 9×9, and the dimension of the convolution kernel is 64. The size of the feature map is $(33 - 9)/1 + 1 = 25$. Second and third layers have the same convolution kernel size and dimension, which are 5×5 and 32, respectively. The size of the feature map obtained after the convolution of the second layer is $(25 - 5)/1 + 1 = 21$, and the size of the feature map obtained after the convolution of the third layer is $(21 - 5)/1 + 1 = 17$. Zero padding is not used during training. In order to obtain faster convergence speed, we use this method to iteratively generate deeper SRCNN, for example, from layer 4 SRCNN 9–5–3–5, to layer 5 SRCNN 9–5–3–3–5, to layer 6 SRCNN 9–5–3–3–3–5, and so on.

The training objective is to minimize the Euclidean distance loss.

Euclidean distance Loss is denominated as the mean of the squared Euclidean distance between the estimated and predicted values of the calculated samples. The formula [15] is

$$Loss = \frac{1}{2N}\sum_{i=1}^{N}\left\| x_{1i} - x_{2i} \right\|_2^2. \quad (3)$$

3.3. Training the PCNN

For the PCNN, the training algorithm and dataset are consistent with the individual network. In the PCNN training, we combined and arranged two SRCNNs in the chain order. In the experiments carried out, the numbers of convolution layers of the first SRCNN and of the second SRCNN were set as 3 + 3, 3 + 5, 5 + 3 and 5 + 5, respectively. Each number denotes the number of layers in the SRCCN. These different fusion schemes are set up to compare the performance of different networks. For example, the accuracy of the PCNN 5 + 3 is lower compared to the PCNN 3 + 5. The reasons are that the weights of the first SRCNN need to be frozen in the training PCNN, and more parameters could be fine-tuned in the PCNN 3 + 5 compared to the PCNN 5 + 3, the accuracy is expected to be better. The feature map output, such as learning rate and weight initialization from the first SRCNN serves as the input of the second SRCNN. If the weights of the first individual network are not frozen, the PCNN is equal to a deeper individual network, and its number of layers is the same as the sum of the layers of the individual network. Therefore, the PCNN consists of only two separate SRCNN, and the weights of the first SRCNN needs to be frozen in the training PCNN.

4. EXPERIMENTAL RESULTS AND ANALYSIS

The experimental environment of proposed algorithm is as follows: Windows 10 64-bit operating system, MATLAB R2018a, Caffe [16], CPU: Intel (R) Core (TM) i7-4770HQ 2.20GHz, RAM: 16G, Intel (R) Iris (TM) Pro Graphics 5200.

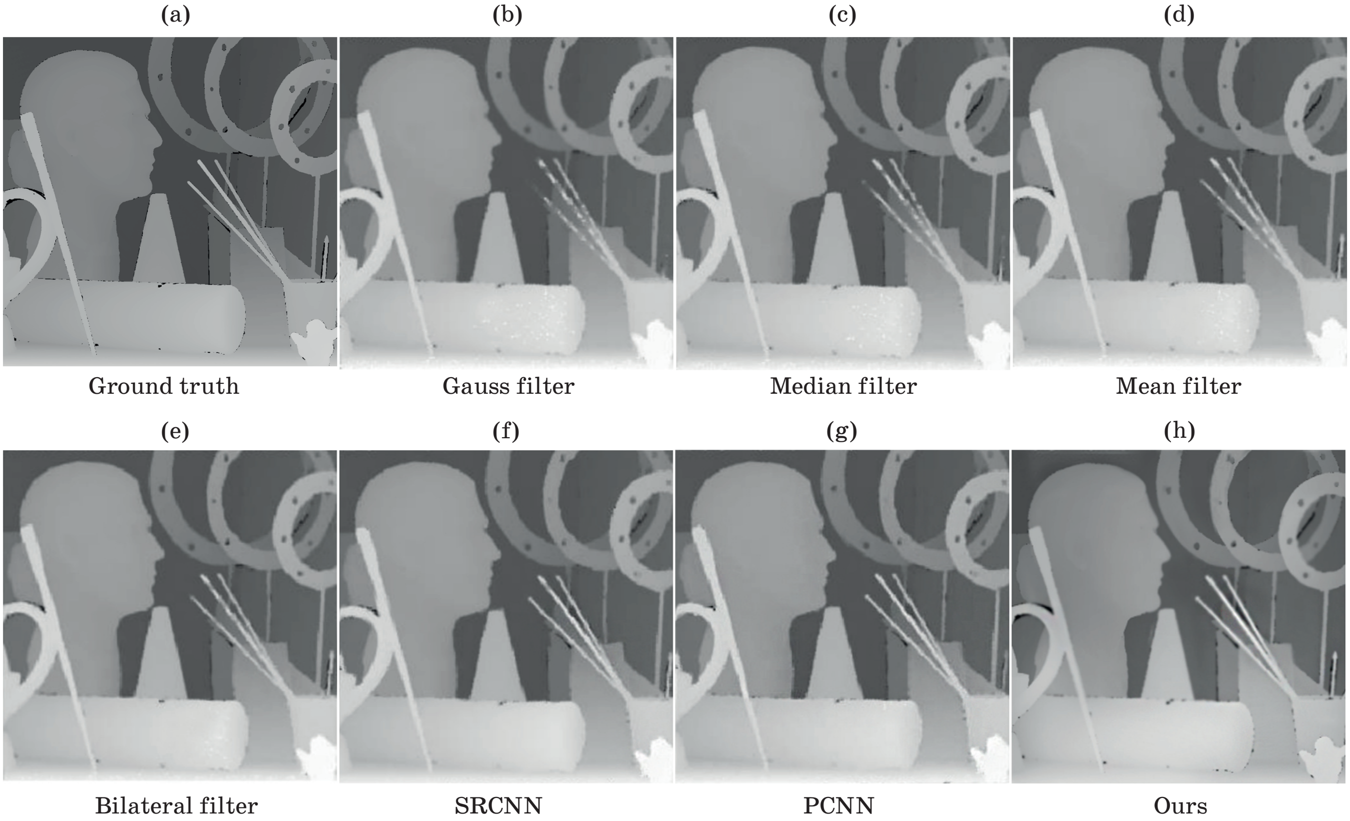
In this paper, the peak signal-to-noise ratio (PSNR) is used to evaluate the denoising quality of depth map. The higher the PSNR, the better the denoising effect.

To illustrate the effectiveness of our algorithm, we compared it with other typical denoising algorithms, the SRCNN [14] and the PCNN on 6 depth maps of the Middlebury dataset, and the results were shown in table 1 and Fig. 2.

As can be seen from the results of Table and Fig. 2, the bilateral filter has more advantages than other traditional algorithms in the index of PSNR, while the algorithm proposed has the best PNSR value and the best subjective visual effect. This shows that the proposed denoising al-

PSNR evaluation of different denoising methods (dB)

Method	Art	Moebius	Books	Laundry	Dolls	Reindeer
Gauss filter	33.29	31.17	31.44	32.13	31.56	31.06
Median filter	33.77	32.25	33.06	33.43	31.25	33.13
Mean filter	34.12	32.40	32.80	33.51	32.09	33.05
Bilateral filter	35.27	36.91	36.91	35.27	36.28	37.21
SRCNN	37.55	38.39	38.51	37.97	38.17	38.95
PCNN	38.21	39.52	39.67	38.81	39.04	40.19
Ours	39.79	40.68	41.15	39.52	40.71	41.34



(a) Ground truth (b) Gauss filter (c) Median filter (d) Mean filter

(e) Bilateral filter (f) SRCNN (g) PCNN (h) Ours

Fig. 2. Qualitative denoising performance comparisons on the Middlebury dataset.

gorithm combining bilateral filter and the PCNN has a good effect on the denoising of depth maps.

We also compared the proposed algorithm with the usual convolutional neural network (i.e. the SRCNN and the PCNN). Our results revealed that despite the SRCNN and the PCNN can also achieve a reasonable level of noise suppression with slightly insignificant lower time complexity. But in the PSNR, they are still inferior to our proposed algorithm.

5. CONCLUSION

In this paper, an algorithm is proposed to utilize both bilateral filter and Progressive Convolution Neural Network (PCNN) to denoise the depth maps. The bilateral filter's capacity of retaining the image's edge details while denoising depth map can be effectively combined with the PCNN's excellent denoising performance by our algorithm. The effective denoising of the depth maps is conducive to automatically classifying the depth maps and selecting the appropriate workflow for rapid 3D reconstruction of buildings. Moreover, the surface of the 3D model reconstructed can be finer and the visual effect is better.

Of course, due to the hardware shortage of ToF camera, the depth maps collected not only have a lot of noise, but also have a lot of holes without depth information. How to effectively recover these missing depth data to improve the quality of depth maps is the focus of the follow-up work.

This study was funded by the National Key Research and Development Program of China (Nos. 2018YFC1604000 and 2016YFB1101702), the Nature Science Foundation of China (grant 61772573), China National Mapping and Geographic Information Bureau Engineering Technology Research Center (ID: SIDT20170101) and Research Center for Culture-Technology Integration Innovation, Key Research Base of Humanities and Social Science of Hubei Province, China (ID: WK201704).

REFERENCES

1. *Schuon S., Theobalt C., Davis J., Thrun S.* Lidarboost: depth superresolution for ToF 3d shape scanning // IEEE Conference on Computer Vision and Pattern Recognition(CVPR). Florida. 23 June 2009. P. 343–350.
2. *Shuaihao Li, Qianqian Li, Min Chen et al.* 3D reconstruction of oil refinery buildings using a depth camera // Chemistry & Technology of Fuels & Oils. 2018. V. 54(5). P. 613–624.
3. *Halber M., Funkhouser T.* Fine-to-Coarse Global Registration of RGB-D Scans // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society. Hawaii. 22 July 2017. P. 1755–1764.
4. *Wang N., Zhang Y., Li Z. et al.* Pixel2Mesh: Generating 3D mesh models from single RGB Maps // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Utah. 20 June 2018. P. 52–67.
5. *Yan S., Wu C., Wang L. et al.* DDRNet: Depth Map denoising and refinement for consumer depth cameras using cascaded // Proceedings of the European Conference on Computer Vision (ECCV). Munich. 12 September 2018. P. 151–167.
6. *Tomasi C., Manduchi R.* Bilateral filtering for gray and color maps // IEEE International Conference on Computer Vision. Bombay. 7 January 1998. P. 839–846.
7. *Greene N., Heckbert P.* Creating raster OmnimaxMaps from multiple perspective views using the elliptical weighted average filter // IEEE Proc Computer Graphics & Applications. 1986. V. 6(6). P. 21–27.
8. *Zhe L.V., Wang F.L., Chang Y.Q. et al.* Region-adaptive Median Filter // Journal of System Simulation. 2007. V. 19(23). P. 5411–5414.
9. *Ito K.* Gaussian filter for nonlinear filtering problems // IEEE transactions on automatic control. 2000. V. 45(5). P. 910–927.
10. *Vincent P., Larochelle H., Lajoie I. et al.* Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion // Journal of Machine Learning Research. 2010. V. 11(12). P. 3371–3408.
11. *Ren H., El-Khamy M., Lee J.* Image super resolution based on fusing multiple convolution neural networks // IEEE Conference on Computer Vision and Pattern Recognition Workshops. Hawaii. 21 July 2017. P. 54–61.
12. *Dong C., Loy C.C., He K. et al.* Image super-resolution using deep convolutional networks // IEEE Transaction on Pattern Analysis and Machine Intelligence. 2016. V. 38(2). P. 295–307.
13. *Rumelhart D.E., Hinton G.E., Williams R.J.* Learning representations by back-propagating errors // Nature. 1986. V. 323(6088). P. 533.
14. *Scharstein D., Pal C.* Learning conditional random fields for stereo // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Minneapolis. MN. 19 June 2007. P. 1–8.
15. *Coleman S.* Black holes as red herrings: topological fluctuations and the loss of quantum coherence // Nuclear Physics B. 1988. V. 307(4). P. 867–882.
16. *Jia Y., Shelhamer E., Donahue J. et al.* Caffe: convolutional architecture for fast feature embedding // Proceedings of the 22nd ACM International Conference on Multimedia. Orlando. Florida. 4 November 2014. P. 675–678.