FAST DEPTH IMAGE DENOISING AND ENHANCEMENT USING A DEEP CONVOLUTIONAL NETWORK

Xin Zhang and Ruiyuan Wu

School of Electronic aningd Information Engineering, South China University of Technology, China

ABSTRACT

We propose a depth image denoising and enhancement framework using a light convolutional network. The network contains three layers for high dimension projection, missing data completion and image reconstruction. We jointly use both depth and visual images as inputs. For the gray image, we design a pre-processing procedure to enhance the edges and remove unnecessary detail. For the depth image, we propose a data augmentation strategy to regenerate and increase essential training data. Further, we propose a weighted loss function for network training to adaptively improve the learning efficiency. We tested our algorithm on benchmark data and obtained very promising visual and quantitative results at real-time speed.

Index Terms— depth image denoise; depth image enhancement; deep convolutional network; data augmentation

1. INTRODUCTION

With the development of affordable and portable depth cameras [1][2], the depth image plays an increasingly important role in fundamental research and daily applications. By utilizing a depth image, people have greatly improved the performance in several key vision-related topics, like segmentation, tracking, recognition, and reconstruction. Many real-world applications have been developed, especially in the human computer interaction field. However, due to the limitation of commercial depth cameras, the quality of depth images is far from satisfactory. First, there is always different shapes of black holes around edges and on dark surfaces. Second, the noise is much stronger when compared with color images. To deal with these issues, depth image denoising and enhancement are usually employed. The denoising step is used to fix corrupted isolated pixels and small regions. The enhancement step aims to improve image details, especially the edges of the depth image.

Several pixel-wise image processing methods have been developed, such as joint bilateral filtering [3], image inpaiting [4], spatial temporal relationship [5], cost-volume [6], wavelet tight frame [7] and low rank matrix [8]. These methods all take advantage of the color-depth relationship and are not fast enough for real-time. Recently, deep learning has become a popular and effective tool for feature representation [9][10] and several pixel-level methods have been successfully proposed [11][12]. We believe the CNN-based framework can provide a possible solution to the depth image denoise and enhancement.

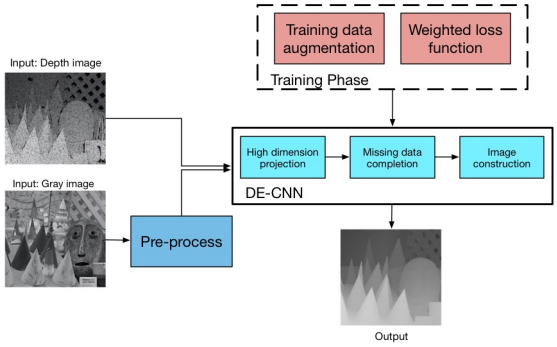


Fig. 1. The flowchart of DE-CNN based depth image denoising and enhancement.

We propose the denoise and enhance convolutional neural network (DE-CNN) to improve the depth image quality, shown in Fig. 1. The DE-CNN is a pixel-wise generative network. We designed the network with three layers and each layer has different structures for different purposes, i.e., 1st layer for high dimensional projection; 2nd layer for missing data completion and 3rd layer for image reconstruction. Moreover, in the training part, we propose the color image preprocessing procedure and depth data augmentation method for data preparation. A weighted map based loss function is also introduced to emphasize edges. By comparing with the most recent state-of-the-art methods, the proposed model is highly computational efficient for real-time applications with very promising results.

2. DEPTH DENOISE AND ENHANCE CONVOLUTION NEURAL NETWORK (DE-CNN)

In order to solve the denoising and enhancement problem depth images, we need a pixel-wise generative model. In-

spired by SRCNN [11] and FCNN [12], we employ the convolutional neural network and design the DE-CNN framework considering unique features of our problem. Firstly, since our goal is to regenerate an image rather than giving a label, the full connection layer should not be employed. Secondly, most CNN-based pixel-based image processing algorithms do not include the max pooling layer to avoid information loss. Here, we add the max pooling to screen out certain corrupted values for denoising. Thirdly, we define a new weighted loss function to take advantage of the depth-color image relationship and emphasize the edges. Therefore, the DE-CNN has three layers with different purposes respectively, i.e., the high dimensional projection, missing data completion and image reconstruction, shown in Fig. 1.

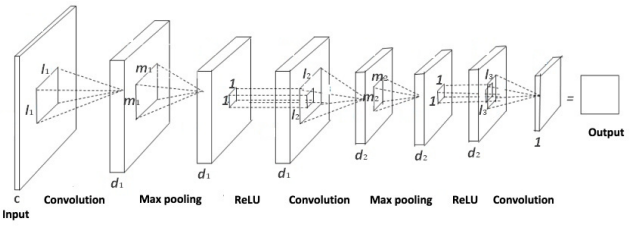


Fig. 2. The structure of DE-CNN

2.1. DE-CNN framework

The DE-CNN has a light structure and every layer of DE-CNN is designed with special goals.

High dimensional projection Since the visual and depth images are highly correlated, we want to project the data into a higher dimensional space to discover the hidden relationship between them. This layer set consists of a convolution layer, a max pooling layer and a rectified linear unit (ReLU) layer. The convolutional layer can extract invisible existing information, while the max pooling process helps to screen out those black holes and noisy parts. The output is the high quality image information in the higher dimensional space.

Missing data completion In this layer, the relation between the visual and depth images is much stronger. Hence, missing depth information can be restored with the help of the corresponding visual image in this space. This layer also consists of a convolutional layer, a max pooling layer and a ReLU layer. The convolutional layer fills up the missing depth image, and the max pooling layer discards the visual information and keeps the depth data.

Image reconstruction After completing the depth image in the high dimensional space, the last step is to reconstruct the image. Here, we only use the convolution operation to summarize and generate the final output depth image. Due to the image spatial relationship, nonlinear operations, like the max pooling and ReLU layer, could ruin this property.

2.2. Loss function definition

For the learning process of DE-CNN, we specifically define the loss function to emphasize the edge influence. Usually, a Euclidean-based distance function is used as the loss function, indicating the difference between the network output and corresponding ground-truth. The general loss function treats every part equally, but we want to emphasize the edges because these parts are always corrupted by large *black holes* and noise. Hence, we define a weighted map based loss function in (1),

$$f_{loss} = \|M \bullet (I_O - I_G)\|^2 \tag{1}$$

where M is the weighted map and I_O and I_G represent the network output and ground-truth images individually. After obtaining the edge information from the ground-truth depth image, in the weighted map we set values around edges close to 1 and those in smooth areas to be much smaller. In this way, we can guide the network to learn stronger explanation capacity around edge regions.

3. DATA PREPROCESSING AND AUGMENTATION

Directly using noisy depth images to train a network can be helpful, but very limited especially for the black holes. Humans can evaluate the missing pixels much easier with the help of the color image. The strong relationship between the depth and color inputs have been used in [8]. In this paper we use the depth and gray images together to complete the denoising and hole-filling tasks in the depth image.

3.0.1. Gray image pre-process

By analyzing the depth image, we observe that the noise is equally distributed and the *black holes* mostly exist around the edges. Hence, we design a gray image pre-processing procedure to emphasize important detail and eliminate useless information. As shown in Fig. 3, the pre-processing procedure has six steps, including the intensity equalization, bilateral filtering, edge extraction, watershed segmentation, segment average padding and intensity quantization. Among these, the goal of "watershed segmentation" and "segmentation average padding" is to combine similar intensity pixels into one region with the same averaged value. After pre-processing, the unnecessary detail is weakened and edges are enhanced.

3.0.2. Depth image pre-processing and training data augmentation

In the dataset, the groundtruth depth image still has some black holes. This fact severely confuses the network since it does not know whether to pad the black area or not. Hence, the first step is to drop training patches whose corresponding groundtruth data contains black areas.

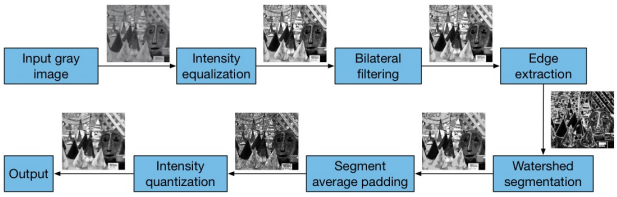


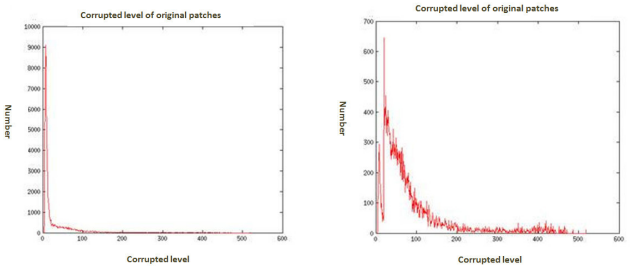
Fig. 3. The pro-process procedure of gray images.

The next issue is the limited number of training samples, especially samples with black holes. The key challenge of depth image enhancement is the black area filling and padding. We statistically analyze the number of connected black pixels in every patch and show the plot in Fig. 4 (a). The number of patches with less than 15 connected black pixels occupy 98% of the total patches while the number of patches with large black areas is very small. To increase the number of these patches, we use a hyperbolic curve as the probability distribution to reorganize the training set, defined as

$$p_i = (e^{crr_i/crr_m} + e^{-crr_i/crr_m})/3,$$

where crr_i is the corrupted level of ith patch and crr_m the max corrupted level among all patches. For pre-defined θ_1 and θ_2 ($0 < \theta_1 < \theta_2 < 1$), patches with p_i less than θ_1 are eliminated by this probability; patches of larger than θ_1 but less than θ_2 are kept by this corresponding probability; patches with greater than θ_2 are duplicated according to their p_i.

We propose a strategy to effectively duplicate specific training patches by randomly rotating a chosen patach 90, 180 or 270 degrees. Fig. 4 (b) shows the processed and augmented result. Now the distribution of corrupted levels is relatively more uniform than shown in Fig. 4 (a). The patches with large black holes now have more influence on the network.



(a) (b)

Fig. 4. The histogram of the number of connected pixels in the training data (a) and processed data set (b).

4. EXPERIMENTS

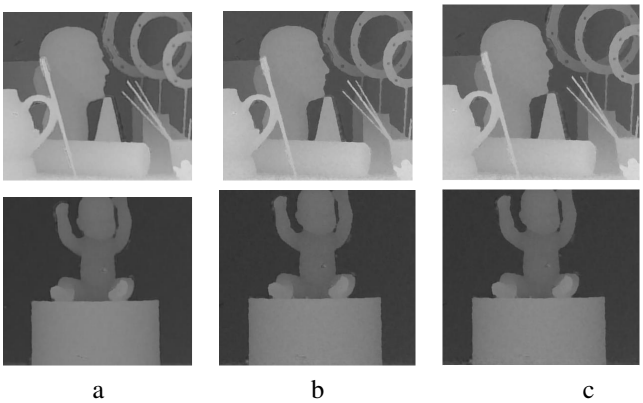
We firstly evaluate and discuss the framework structure of DE-CNN. Then, we compare our proposed DE-CNN with two state-of-the-art depth denoising and enhancement methods in terms of speed, PSNR, and visual effects. The Middlebury dataset [13][14] consists of 30 pairs of ground-truth depth and color images. In [8], the authors manually added black holes in depth images to simulate the noisy pictures captured by real depth cameras. This modified version has been used widely as the benchmark for depth image processing [8].

4.1. DE-CNN framework evaluation

In the following experiment, we use 28 images of the Middlebury set as the training data and the remaining two as the test images. We set the first unit as a convolutional layer of size $1 \times 9 \times 9 \times 128$, a 5×5 max pooling layer and a ReLU layer. The second unit consists of a $128 \times 1 \times 1 \times 64$ convolutional layer, a 3×3 max pooling layer and a ReLU. A single $64 \times 5 \times 5 \times 1$ convolutional layer acts as the last unit. Each experiment is trained using 1.5 million iterations. We evaluate and compare the framework structure from two aspects: (1) single depth input vs. joint depth-RGB input; (2) Euclidean loss function vs. edge based weighted loss function.

4.1.1. Input data comparison

We compare the single depth input and joint depth and color input. The figure and PSNR comparisons in Fig. 5 and Table 1, show the joint input result provides much better results. The large black hole areas have been better padded with clear edges, such as the long brush in test figure one.



a b c

Fig. 5. The DECNN setting comparison: (a) single depth input; (b) joint depth and RGB input; (c)joint input with preprocessing and weighted loss function.

PSNR (dB)	Depth Input	Joint Inputs	Weighted Loss Function
Test One	32.56	33.46	33.68
Test Two	38.96	39.02	39.18

Table 1. The PSNR comparison of different settings on two testing figures.

4.1.2. Loss function

We use the edge based weight maps as the weighted loss function on each output layer to focus on black holes and edges. Results are further improved, as shown in Fig. 5(c) and Table 1. In summary, these experiments have demonstrated the effectiveness of our network design and data preparation.

4.2. Comparison with other algorithms

We also compare DE-CNN with another two recent algorithms that deliver the best results among others. We denote the low rank matrix completion method [8] as LRMC and the data-driven tight framework [7] as DDTF in the following. For a fair comparison, we use the same training set including all 30 images in the Middlebury dataset, and compare the three methods according to their computing efficiency, PSNR and visual quality.

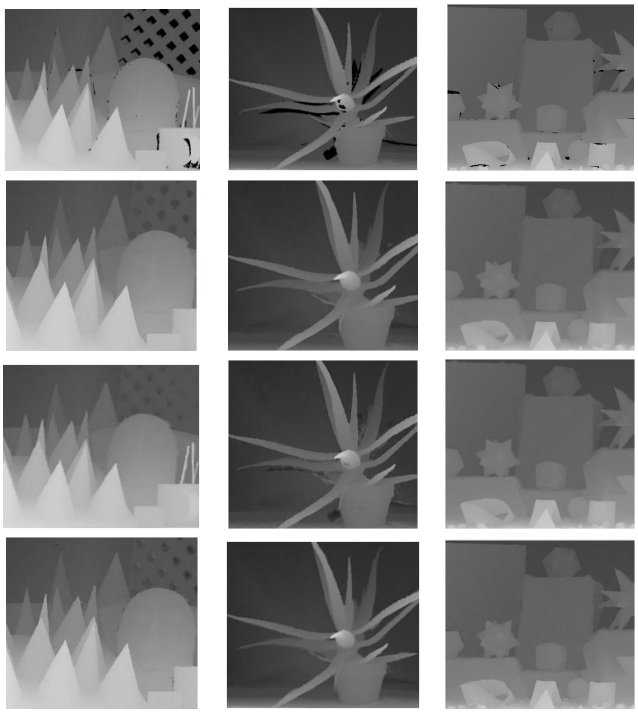


Fig. 6. The visual quality comparison of our DE-CNN with two other methods. First row: input depth image; second row: LRMC's results; third row: DDTF's results and fourth row: proposed DE-CNN results.

Table 2. PSNR(dB) comparison

Middlebury dataset	Flower			Sculpture			Infant 1		
Method	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF
PSNR	**36.24**	35.96	35.63	**33.84**	33.20	32.29	**40.28**	38.85	38.92
Middlebury dataset	Infant 2			Infant 3			Book		
Method	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF
PSNR	41.92	42.59	**42.99**	39.76	42.32	**43.33**	41.37	40.75	**42.12**
Middlebury dataset	Bowling 1			Bowling 2			Cloth 1		
Method	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF
PSNR	37.52	**38.22**	38.00	38.71	**39.37**	38.18	45.01	46.24	**46.86**
Middlebury dataset	Cloth 2			Cloth 3			Cloth 4		
Method	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF
PSNR	41.45	**42.35**	42.33	**42.75**	42.37	42.16	**39.16**	37.43	37.57
Middlebury dataset	Cone			Toy 1			Clay pot		
Method	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF
PSNR	39.16	**39.85**	38.56	40.67	41.91	**42.13**	37.62	42.73	**43.77**
Middlebury dataset	Toy brick 1			Toy brick 2			Window		
Method	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF
PSNR	**39.96**	38.79	39.37	**39.91**	38.56	39.32	38.11	**38.49**	37.91
Middlebury dataset	Bag 1			Bag 2			Origami		
Method	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF
PSNR	**40.39**	39.82	39.06	**39.41**	38.57	38.52	41.07	**41.95**	41.67
Middlebury dataset	Board game			Folder			Elk		
Method	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF
PSNR	39.32	39.41	**40.02**	42.72	42.04	**43.34**	35.36	35.46	**35.90**
Middlebury dataset	Stone 1			Stone 2			Toy 2		
Method	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF
PSNR	43.39	45.26	**45.49**	43.27	46.36	**46.58**	39.77	40.74	**40.80**
Middlebury dataset	Wood			Board			Newspaper		
Method	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF	DE-CNN	LRMC	DDTF
PSNR	**40.84**	39.28	40.80	40.46	39.85	**41.04**	**41.36**	41.32	41.18

- **Speed** After the pre-processing step (it takes 0.05s), DE-CNN takes 0.033 second to process a 352×395 depth image using NVIDIA TITAN X GPU. In comparison, LRMC requires 1.5 minutes for one image. DDTF also needs quite a while.
- **PSNR** PSNR result are summarized in Table 2 for all 30 images. These results show that DE-CNN has comparable denoising and enhancement capacity to state-of-the-art algorithms.
- **Visual Quality** We show the sample image results in Fig. 6. The general visual quality is similar but edges in DE-CNN processed images are sharper than for the other two methods.

5. CONCLUSION

We propose a novel convolutional neural network DE-CNN for pixel-wise depth image denoising and enhancement. It is a light CNN-based network with two units consisting of a convolution layer, max pooling layer and ReLU layer, and one convolution layer in the last unit. The training data preprocessing and augmentation have effectively improved the performance. Based on our experiments, the proposed model has a very high computational efficiency and promising performance for pixel-wise denoising and enhancement. We believe this model can be applied for real-time processing in real-world depth image pre-processing applications. It's worth mentioning that at current stage we still we don't have enough training data. In the future, we will collect more related data and improve the performance of our deep learning framework.

6. ACKNOWLEDGMENT

Authors want to thank Prof. Pickering of UNSW for his suggestions. This work is supported by NSFC (No. 61202292), Fundamental Research Funds for the Central Universities (No. 2015ZZ027) and SCUT-UNSW Research Collaboration Seed Program 2014.

7. REFERENCES

[1] Microsoft, "Kinect," https://dev.windows.com/en-us/kinect.

[2] Intel, "Realsense," http://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html.

[3] T. Matsuo, N. Fukushima, and Y. Ishibashi, "Weighted joint bilateral filter with slope depth compensation filter for depth map refinement," in *International Conference on Computer Vision Theory and Applications*, 2013, pp. 300–309.

[4] L. Wang, H. Jin, R. Yang, and M. Gong, "Stereoscopic inpainting: Joint color and depth completion from stereo images," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[5] J. Fu, S. Wang, Y. Lu, S. Li, and W. Zeng, "Kinect-like depth denoising," in *IEEE International Symposium on Circuits and Systems*, 2012, pp. 512–515.

[6] J. Shen and S. C. S. Cheung, "Layer depth denoising and completion for structured-light RGB-D cameras," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1187–1194.

[7] J. Wang and J. F. Cai, "Data-driven tight frame for multi-channel images and its application to joint color-depth image reconstruction," *Journal of the Operations Research Society of China*, vol. 3, no. 2, pp. 99–115, 2015.

[8] Si Lu, Xiaofeng Ren, and Feng Liu, "Depth enhancement via low-rank matrix completion," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3390–3397.

[9] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal, "The wake-sleep algorithm for unsupervised neural networks," *Science*, pp. 1158–1161, 1995.

[10] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3476–3483.

[11] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super resolution," in *Proceedings of Eurpoean Conference on Computer Vision*, 2014, pp. 184–199.

[12] J. Long, E. Shelhammer, and T. Darrell, "Fully convlutional networks for sematic segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[13] D. Scharstein and R. Szeiliski, "High-accuracy stereo depth maps using structured light," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2003, pp. 195–202.

[14] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," in *ACM Transactions on Graphics*, 2009.