DEPTH SUPER-RESOLUTION WITH DEEP EDGE-INFERENCE NETWORK AND EDGE-GUIDED DEPTH FILLING

*Xinchen Ye**† *Xiangyue Duan**† *Haojie Li**†

*DUT-RU International School of Information Science & Engineering, Dalian University of Technology
†Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province

ABSTRACT

In this paper, we propose a novel depth super-resolution framework with deep edge-inference network and edge-guided depth filling. We first construct a convolutional neural network (CNN) architecture to learn a binary map of depth edge location from low resolution depth map and corresponding color image. Then, a fast edge-guided depth filling strategy is proposed to interpolate the missing depth constrained by the acquired edges to prevent predicting across the depth boundaries. Experimental results show that our method outperforms the state-of-art methods in both the edges inference and the final results of depth super-resolution, and generalizes well for handling depth data captured in different scenes.

Index Terms— Super-resolution, depth image, edge-inference, edge-guided

1. INTRODUCTION

Scene depth perception is one of the most important sources to understand natural scenes, which becomes the basis of 3D modeling, 3DTV, autonomous driving, etc [1–3]. However, the accuracy of depth acquisition are affected due to the complexity of real scenes and the imaging limitation of depth sensors. One of the main degradations of the acquired depth maps is low-resolution (LR), which impedes the development of other depth-based applications. Therefore, effective depth upsampling techniques are needed to yield high-resolution (HR) depth maps recovered from a given LR depth map.

Usually, the basic idea to recover a HR depth map is to use the corresponding color image captured from the same scene because of the strong structural correlations between depth and texture (Fig. 1(a) and (c)). One way is to recast the depth super-resolution task as a global optimization problem [4–8], in which, the data term penalizes the difference between the observation and the recovered depth, while the smooth term regularizes neighboring pixels based on the designed priors. However, these methods often use hand-designed objective functions which cannot approach real image priors well and are typically time-consuming. Another category of depth upsampling methods [9–12] uses designed filters to apply joint filtering on the depth map under guidance of the HR color image. However, these filtering-based methods cannot provide enough information to determine the global structure, and may introduce artifacts in regions where the associated color image has rich textures.

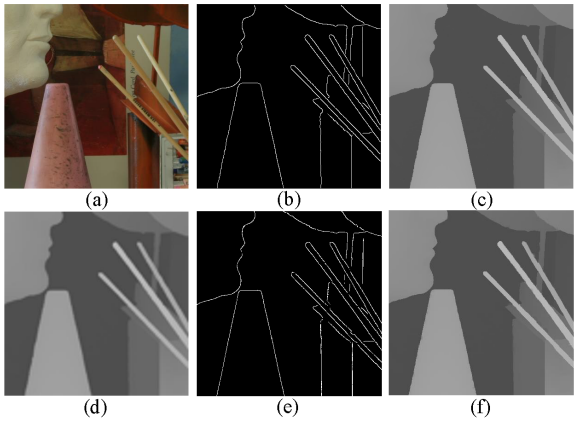


Fig. 1. Depth super-resolution example. (a) Color image; (b) ground truth (GT) edge map; (c) GT depth map; (d) LR depth map upsampling by bicubic interpolation; (e) Our inferred edge map; (f) Our upsampled HR depth map.

A promising category is the learning-based methods [13–15], which learns a relation between LR and HR depth map. Xie *et al.* [14] have learned a HR edge map from extracted LR edges using a MRF framework based on the trained external database of LR-HR edge pairs, and interpolated the depth values via a modified joint bilateral filter to obtain a HR depth map. Recently, Li *et al.* [15] have employed a two-path convolutional neural network (CNN), to learn a end-to-end network to obtain the final HR depth map from LR depth map and the corresponding HR color image. It enjoys a fast testing speed, and delivers more promising performance than the above methods. While the CNN-based method [15] yield powerful representations and are efficient in the evaluation of the network for given input LR depth data, their training is often difficult. Since there exist different scene structures in different training datasets captured by depth sensors, a suffi-

This work was supported by National Natural Science Foundation of China (NSFC) under Grant 61702078, Grant 61772108, Grant 61472059, and by the Fundamental Research Funds for the Central Universities. *Corresponding author: Xinchen Ye (E-mail: yexch@dlut.edu.cn).

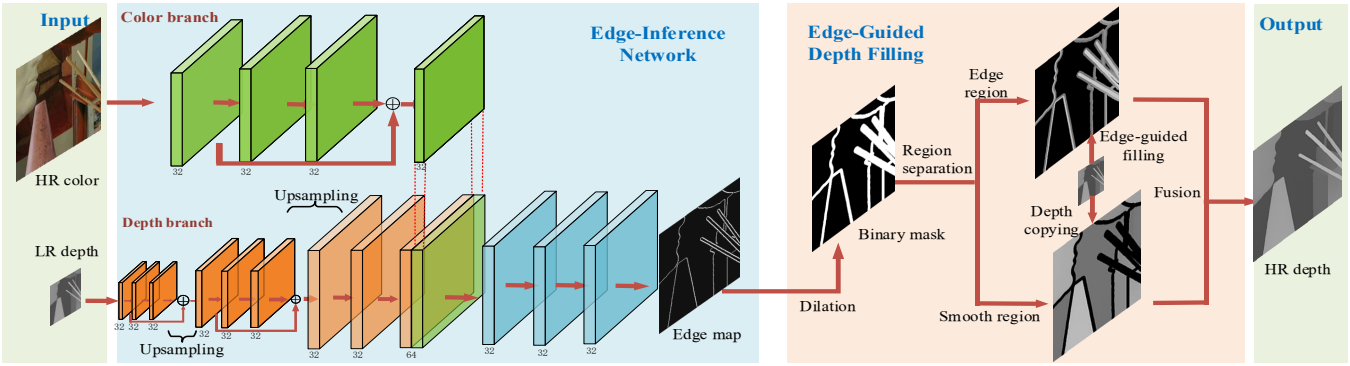


Fig. 2. Depth super-resolution framework. Only 4× upsamping CNN architecture is presented for saving space.

cient amount of training data needs to be acquired to generalize well enough to the test data.

As we observe, depth map mainly contains smooth regions separated by small amounts of edges. Unlike color image, depth map does not have too much texture information, and what really affects the depth quality is the sharpness of depth edges (Fig. 1(b)). Therefore, locating the depth edges precisely is more important to directly infer the depth value on each pixel. Motivated by [14], we focus on learning a binary edge map that indicates the edge location (1 for valid pixels on edges or 0 otherwise) from the LR depth map (Fig. 1(e)). Different from [14] that uses hand-crafted correspondence between LR edges and HR edges, we instead employ the more competing CNN framework to infer an edge location map from the LR depth map.

Another advantage of constructing an edge-inference CNN relates to network training and generalization. Edge map is presented by binary values, and most pixels are zero due to the smooth property of the depth signal. Learning a mapping from LR depth to binary edge location is much easier than learning directly from LR depth to HR depth, since the network does not need to infer the depth value on each pixel and the relative depth between neighboring pixels. Training a depth-to-location network is easier converged and faster than the depth-to-depth one [15]. Therefore, a light-weight network with a small quantity of training data is enough to learning the mapping, in other word, promoting the performance of network generalization. Later experiments will validate our ideas about the generalization of our method.

Following the above analysis, we propose a novel depth super-resolution framework (Fig. 2) based on a deep edge-inference network followed by an edge-guided depth filling process. Firstly, we construct a CNN architecture to learn a binary map of depth edge location from LR depth map. Then, an edge-guided depth filling method is carefully designed to interpolate depth values on the HR image grids constrained by the acquired edges to prevent predicting across the depth boundaries. The depth map to be filled is separated into smooth regions and edge regions, and each region uses different depth filling strategy to achieve a better depth upsampling performance and complexity tradeoff. Experimental results show that our method achieves the state-of-art performance in both the edges inference and the final results of depth super-resolution, and generalizes well for handling depth data captured in various scenes.

2. PROPOSED METHOD

2.1. Deep Edge-Inference Network

As we observe, edges information is of especially importance in textureless depth map. Therefore, we design a CNN to learn the HR edge map $\boldsymbol{E}$ from the given LR depth image $\boldsymbol{D}^l$ and the accompanied HR color image $\boldsymbol{I}$.

As shown in Fig. 2, our network architecture consists of two branches, i.e., color branch and depth branch. The color branch acts as a feature extractor to determine informative edge features from color image. For the depth branch, the LR depth map $\boldsymbol{D}^l$ is upsampled progressively in S levels to a desired HR edge map for the upsampling factors 2^S. For example, the depth branch consists of 2 sub-networks for upsampling an LR image at a scale factor of 4. At each level, the depth branch consists of multiple convolutional layers and one transposed convolutional layer to upsample the extracted features by a scale of 2. The output of each transposed convolutional layer is connected to a convolutional layer for extracting features at the finer level. Then, the upsampled feature maps from depth branch are concatenated with the feature maps extracted from color branch in the same resolution. Finally, some convolutional layers are added to extract the final HR edge map. A thresholding operation is done on the network output to get a binarized edge map.

In the implementation, each convolutional layer consists of 32 filters with the size of 3×3. All the convolutional and transposed convolutional layers (except the top layer) are followed by ReLU activation function. The skip structure [16] is used in both depth branch and color branch to accelerate the training process. We learn the network parameters θ by minimizing the L2 loss $||f(\boldsymbol{D}^l, \boldsymbol{I}, \theta) - \boldsymbol{E}^{gt}||_2^2$, where f is the mapping function, and $\boldsymbol{E}^{gt}$ is the ground truth (GT) edge map extracted from GT depth map by canny operator.

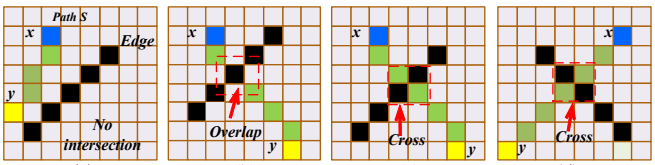


Fig. 3. Illustration of different situations of intersection between the path S (Green) and the depth edge (Black). (a) No intersection; (b-d) All the three different intersection modes.

2.2. Edge-Guided Depth Filling

To our best knowledge, depth map contains large quantities of smooth regions. Therefore, we do not need to design a sophisticated filtering algorithm in dealing with these textureless regions. More attention should be paid to the regions around edges. As shown in Fig. 2, we first dilate the inferred edge map $\boldsymbol{E}$ to get an binary mask, which can effectively separate the depth map into smooth region and edge region. Then, for the smooth region, depth values are directly copied from the LR depth map interpolated by bicubic. For the edge regions, we design a depth filling method with a new edge constraint strategy in the following:

We upsample the LR Depth map $\boldsymbol{D}^l$ to the resolution of HR color image $\boldsymbol{I}$ by filling zeros, and interpolate missing depth values on the HR image grids. For each pixel x in the target HR depth map $\boldsymbol{D}$, its depth $\boldsymbol{D}_x$ is estimated via an joint bilateral filter:

$$\boldsymbol{D}_x = \frac{1}{K} \sum_{y \in \mathcal{N}(x)} G_\sigma \left(\boldsymbol{I}_x - \boldsymbol{I}_y \right) \mathbf{1}(x, y; \boldsymbol{E}) \boldsymbol{D}_y, \qquad (1)$$

where $G(\cdot)$ is the Gaussian kernel with subscript σ as its standard deviation to adjust color difference. K is the normalization factor. $\mathcal{N}(x)$ is the neighborhood of pixel x. $\mathbf{1}(\cdot)$ is a binary indicator that replacing the range kernel, which indicates whether the pixel x and y are the same side of an depth edge. Only pixels at the same side of the edge with x are considered as candidates during averaging to prevent predicting across depth edges. Note that, with the guidance of edge map $\boldsymbol{E}$, a large neighborhood is unnecessary, and contributes little to the estimation of x, which also increases the computational complexity. So we adopt a neighborhood containing four nearest valid pixels around the centering pixel x acquired from LR depth map $\boldsymbol{D}^l$.

To determine the relationship between the pixel x and y, we link x and y by computing the line path S between the two pixels. If no intersection between the path S and the edge (Fig. 3(a)), y can be regarded as a candidate and the binary indicator $\mathbf{1}(\cdot)$ are set at one. Otherwise, we judge x and y at different side of the edge when encountering the overlap case in Fig. 3(b) and other two cross cases in Fig. 3(c)(d), and therefore exclude y by setting the indicator to zero.

Statistically, about 3% pixels of a depth map are selected into the edge region to be filled by our edge-guided filtering strategy, while most pixels are directly copying from LR depth map. Besides, different from [14] that constructs a complicated graph on a large neighborhood to determine the pixels relationship, our algorithm achieves a comparable depth filling results with [14], but a much faster running speed.

3. EXPERIMENTAL RESULTS

Our proposed method is evaluated on the performance of inferred edges, depth super-resolution, and network generalization, separately. To train our network, we use 38 RGB-D images for training and 6 for validation (*Art*, *Book*, *Moebius*, *Reindeer, Laundry,* and *Dolls*) from Middlebury dataset [17]. We randomly extract 13860 depth patches of a fixed size 15×15 from downsampled depth map, and corresponding color patches and edge patches of the squared size 30, 60, 120, and 240 according to 2, 4, 8, and 16 upsamping factors respectively. We train our model with the MatConvNet toolbox. The learning rate is initialized to 1e-3 for all layers and decreased by a factor of 2, 5, and 10 for every 50 epochs progressively. Besides, the depth filling parameter σ is set to be 3. Mean absolute difference (MAD) is used for objective evaluation on the result of depth super-resolution.

3.1. Evaluation on the Performance of Inferred Edges

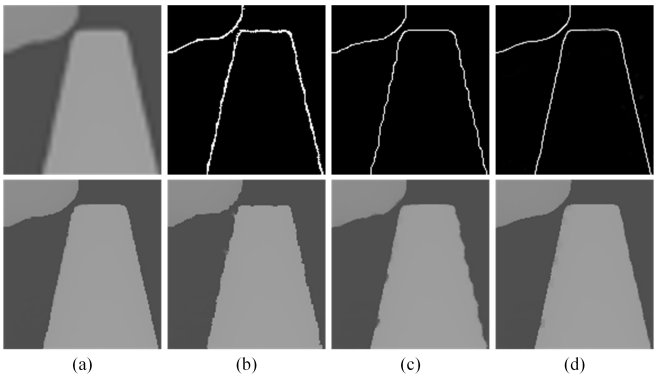


Fig. 4. Edge extraction and upsampling results from 4× LR depth. (a) LR depth upsampling by bicubic and GT; (b) EG [14] (MAD: 1.01); (c) Canny operator + our depth filling strategy (MAD: 0.79); (d) Ours (MAD: 0.61).

Fig. 4 shows the inferred edges and the final depth filling results from 4× downsampling depth map. We compare with the recent edge-guided method (EG) [14], and the method using canny operator to extract edges plus our depth filling strategy to obtain the final HR depth map. Result shows that we extract the most accurate and thin depth edges, while EG presents wider predicted edges than ours, and the edge extracted by canny operator is inclined to distort influenced by the blurry edges from LR depth map. Besides, the upsampling result also demonstrates our superior performance (the lowest MAD, 0.61) on edge prediction.

Table 1. Quantitative depth upsampling results on Middlebury datasets at four subsamping rate.

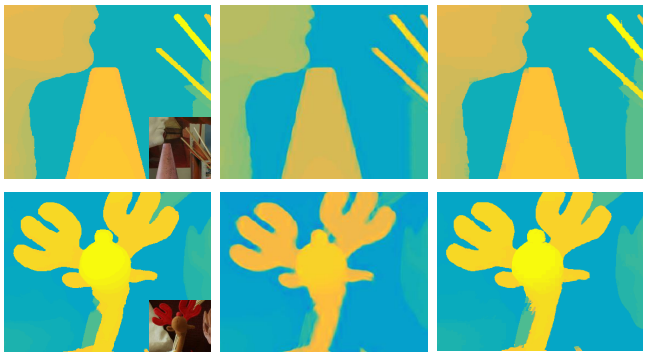
	Art				*Book*				*Dolls*				*Moebius*				*Laundry*				*Reindeer*			
	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×
TGV[6]	0.45	0.65	1.17	2.30	0.18	0.27	0.42	0.82	0.21	0.33	0.70	2.20	0.18	0.29	0.49	0.90	0.31	0.55	1.22	3.37	0.32	0.49	1.03	3.05
AR[7]	0.18	0.49	**0.64**	2.01	0.12	0.22	**0.37**	**0.78**	0.21	0.34	0.50	0.82	0.10	**0.20**	0.40	**0.79**	0.20	0.34	0.53	1.12	0.22	0.40	0.58	1.00
FGI[8]	0.70	1.29	2.41	4.51	0.43	0.74	1.16	1.91	0.54	0.93	1.44	2.12	0.51	0.91	1.59	2.68	0.42	0.72	1.13	1.81	0.50	0.87	1.58	2.72
JGF[11]	0.29	0.47	0.78	1.54	0.15	0.24	0.43	0.81	0.19	0.33	0.59	1.06	0.15	0.25	0.46	0.80	0.21	0.36	0.64	1.20	0.23	0.38	0.64	1.09
EG[14]	-	0.64	-	-	-	0.28	-	-	-	0.33	-	-	-	0.37	-	-	-	0.29	-	-	-	0.40	-	-
DJF[15]	**0.12**	**0.40**	1.07	2.78	**0.05**	**0.16**	0.45	1.00	**0.06**	**0.20**	0.49	0.99	**0.07**	0.28	0.71	1.67	**0.06**	**0.18**	0.46	1.02	**0.07**	**0.23**	0.60	1.36
Ours	0.23	**0.40**	**0.64**	**1.34**	0.12	0.22	**0.37**	**0.78**	0.12	0.22	**0.38**	**0.73**	0.13	0.23	**0.36**	0.81	0.11	0.20	**0.35**	**0.73**	0.15	0.26	**0.40**	**0.80**

Table 2. Quantitative depth upsampling results on chosen depth frames from MPI datasets at four subsamping rate.

	Alley_1-48				*Ambush_2-15*				*Ambush_4-12*				*Ambush_5-41*				*Temple_3-23*			
	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×
EG[14]	-	0.23	-	-	-	0.28	-	-	-	0.76	-	-	-	0.88	-	-	-	0.54	-	-
DJF[15]	**0.07**	0.17	0.46	0.90	**0.06**	0.20	0.48	0.96	**0.21**	0.54	1.14	2.49	0.28	0.72	1.42	2.67	**0.15**	0.40	0.79	1.76
Ours	0.08	**0.13**	**0.23**	**0.43**	0.09	**0.15**	**0.28**	**0.54**	0.25	**0.50**	**0.82**	**1.76**	**0.24**	**0.41**	**0.69**	**1.32**	0.17	**0.30**	**0.51**	**1.12**

3.2. Results on Depth Super-Resolution

Depth Upsampling results on the six testing datasets are shown in Table 1. Our method is compared with six state-of-the-art methods, i.e., total generalized variation (TGV) [6], auto-regressive (AR) [7], fast global interpolation (FGI) [8], joint geodesic filtering (JGF) [11], and edge-guided method (EG) [14], and deep joint filtering (DJF) [15]. We use the same training data with ours to train the network from D-JF. Our method nearly obtains the lowest MAD for 8× and 16× rates, while DJF provides better results for 2× and 4× upsampling rates. This explains that a end-to-end trained network [15] specializes in inferring HR depth for lower upsampling rates, but fails to reverse the process of down-sampling degradation for high upsampling rates from the seriously blurry depth map. Visual results in Fig. 5 also demonstrate this, i.e., large areas of ringing artifacts appear around the depth edges for 8× upsampling rate.

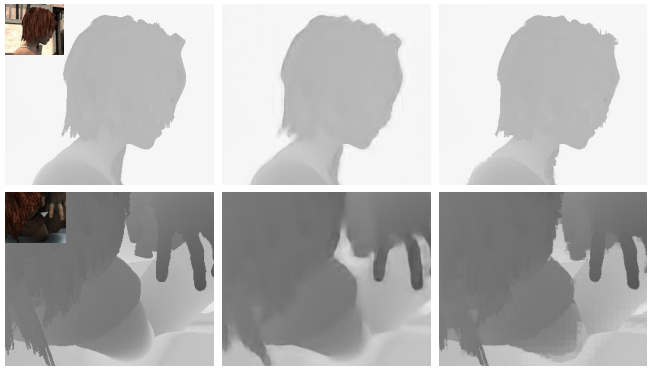


(a) (b) (c)

Fig. 5. Upsampling results from 8× LR depth map. (a) GT; (b) DJF [15]; (c) Ours.

3.3. Evaluation on the Generalization of Our Framework

To test the generalization of our framework, we choose five depth maps from another depth dataset, i.e., MPI Sintel dataset [18], to validate stronger generalization of our method than DJF [15] that uses depth-to-depth training mode. Note that we use Middleburry dataset to train the both methods, but test on MPI dataset which contains totally different scene structures from Middlebury dataset. Table 2 shows that the quantitative results of our method are far better than DJF for 4×, 8× and 16× subsampling rates, which demonstrates that learning a mapping from LR edges to binary edges location is the critical factor that promotes the network generalization. Visual results in Fig. 6 also validates this, i.e., DJF generates blurry results than ours.



(a) (b) (c)

Fig. 6. Evaluation on network generalization. (a) GT; 8× upsampling results: (b) DJF [15] and (c) Ours.

4. CONCLUSION

This paper proposes a novel depth super-resolution framework with deep edge-inference network and edge-guided depth filling. We first construct a CNN architecture to learn the depth edge location from LR depth map. Then, a fast edge-guided depth filling strategy is proposed to interpolate the missing depth. Experimental results show that our method outperforms the state-of-art methods in both the edges inference and depth super-resolution, and generalizes well for handling diverse depth datasets.

5. REFERENCES

[1] Kun Li, Jingyu Yang, Leijie Liu, Ronan Boulic, Yu Kun Lai, Yebin Liu, Yubin Li, and Eray Molla, "Spa: Sparse photorealistic animation using a single rgb-d camera," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 771–783, 2017.

[2] S. A Scherer and A Zell, "Efficient onbard rgbd-slam for autonomous mavs," in *IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 1062–1068.

[3] Xinchen Ye, Jingyu Yang, Hao Huang, Chunping Hou, and Yao Wang, "Computational multi-view imaging with kinect," *IEEE Transactions on Broadcasting*, vol. 60, no. 3, pp. 540–554, 2014.

[4] Jaesik Park, Hyeongwoo Kim, Yu-Wing Tai, Michael S Brown, and Inso Kweon, "High quality depth map upsampling for 3D-TOF cameras," in *Proc. ICCV*, 2011, pp. 1623–1630.

[5] Jingyu Yang, Xinchen Ye, Kun Li, and Chunping Hou, "Depth recovery using an adaptive color-guided auto-regressive model," in *Proc. ECCV*, 2012, pp. 158–171.

[6] David Ferstl, Christian Reinbacher, Rene Ranftl, Matthias Rüther, and Horst Bischof, "Image guided depth upsampling using anisotropic total generalized variation," in *Proc. ICCV*, 2013.

[7] Jingyu Yang, Xinchen Ye, Kun Li, Chunping Hou, and Yao Wang, "Color-guided depth recovery from rgb-d data using an adaptive autoregressive model.," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3443–3458, 2014.

[8] Yu Li, Dongbo Min, Minh N. Do, and Jiangbo Lu, "Fast guided global interpolation for depth and motion," in *Proc. ECCV*, 2016.

[9] D. Min, J. Lu, and M. Do, "Depth video enhancement based on joint global mode filtering," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1176–1190, 2011.

[10] Jiangbo Lu, Keyang Shi, Dongbo Min, Liang Lin, and Minh N Do, "Cross-based local multipoint filtering," in *Proc. CVPR*, 2012, pp. 430–437.

[11] Ming-Yu Liu, Oncel Tuzel, and Yuichi Taguchi, "Joint geodesic upsampling of depth images," in *Proc. CVPR*, 2013, pp. 169–176.

[12] Jonathan T. Barron and Ben Poole, "The fast bilateral solver," in *Proc. ECCV*, 2016.

[13] Oisin Mac Aodha, Neill DF Campbell, Arun Nair, and Gabriel J Brostow, "Patch based synthesis for single depth image super-resolution," in *Proc. ECCV*, 2012, pp. 71–84.

[14] Jun Xie, R. S. Feris, and Ming Ting Sun, "Edge-guided single depth image super resolution," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 428, 2016.

[15] Yijun Li, Jia Bin Huang, Narendra Ahuja, and Ming H-suan Yang, "Deep joint image filtering," in *Proc. ECCV*, 2016.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[17] Middlebury datasets, "http://vision.middlebury.edu/," .

[18] MPI Sintel datasets, "http://sintel.is.tue.mpg.de/," .