DEPTH COMPLETION USING PLANE-RESIDUAL REPRESENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The basic framework of depth completion is to predict a pixel-wise dense depth map using very sparse input data. In this paper, we try to solve this problem in a more effective way, by reformulating the regression-based depth estimation problem into a combination of depth plane classification and residual regression. Our proposed approach is to initially densify sparse depth information by figuring out which plane a pixel should lie among a number of discretized depth planes, and then calculate the final depth value by predicting the distance from the specified plane. This will help the network to lessen the burden of directly regressing the absolute depth information from none, and to effectively obtain more accurate depth prediction result with less computation power and inference time. To do so, we firstly introduce a novel way of interpreting depth information with the closest depth plane label \( p \) and a residual value \( r \), as we call it, Plane-Residual (PR) representation. We also propose a depth completion network utilizing PR representation consisting of a shared encoder and two decoders, where one classifies the pixel’s depth plane label and the other one regresses the normalized distance from the classified depth plane. By interpreting depth information in PR representation and using our corresponding depth completion network, we were able to acquire improved depth completion performance with faster computation, comparing to other recent approaches.

1 INTRODUCTION

Many different computer vision algorithms are becoming more reachable in our everyday life, starting from a smartphone camera and augmented reality (AR) / virtual reality (VR) applications to autonomous driving and even more complicated robotics tasks. In order to solve these problems efficiently with accurate 3D information, 3D reconstruction using an additional depth sensor is preferred, since it can utilize more accurate depth measurements as prior information. One of the major downsides of commercialized depth sensors, such as 3D LiDAR, Kinect, and RealSense, is the sparsity of the measurement. Addressing this problem, various approaches emerged trying to densify, i.e., ‘complete’ the sparse depth measurement into a dense depth map, namely, ‘depth completion’.

In recent years, there have been many different methods trying to solve depth completion using deep learning, starting from [Ma & Karaman, 2018]. The challenging part of these algorithms is that regression-based approaches have difficulties in maintaining the information of the object boundary and may show mixed depth results [Imran et al., 2019]. A few early works addressed these difficulties by maximizing the information from the RGB input and refining the initial depth regression output to get better final results [Cheng et al., 2018; Ma et al., 2019; Lee et al., 2019; Van Gansbeke et al., 2019; Cheng et al., 2020; Cheng et al., 2020; Park et al., 2020]. Other algorithms tried to utilize additional information that can be inferred from the depth map, such as surface normal, to give more geometrical guidance to the training process [Lee et al., 2019; Xu et al., 2019; Qiu et al., 2019]. While these algorithms showed some promising outcomes, they still lack in preserving edge information and often require a large amount of computation power, as in heavy network memory and longer inference time, which are not suitable for real-time applications.

To tackle these problems, we introduce a Plane-Residual (PR) representation, a novel way of interpreting depth information, and an end-to-end depth completion deep learning network. PR representation expresses an absolute depth value of a pixel with two parameters \((p, r)\), where \(p\) refers
to the closest depth plane among a number of pre-defined discretized depth planes, and \( r \) refers to the normalized distance from the selected plane. With Plane-Residual representation, we can factorize the direct depth regression problem into a combination of discrete depth plane classification and plane-by-plane residual regression. We also propose an end-to-end depth completion network using PR representation to execute this idea and present our results with good performance and fast computation. We did extensive ablation studies and compared our algorithm with some state-of-the-art methods, both quantitatively and qualitatively, to prove the effectiveness and the validity of our design choices.

2 RELATED WORK

2.1 DEPTH COMPLETION

Ma & Karaman (2018) proposed depth completion using convolutional neural network (CNN). They utilized the overall network structure from Laina et al. (2016) to solve the depth completion problem using an RGB image and a single-channeled sparse depth input. Their algorithm showed that using only a fraction of additional input depth information boosts a big margin of performance.

Cheng et al. (2018) and Park et al. (2020) addressed the blurriness of the result in depth regression using deep learning, and tried to solve this problem by utilizing post-processing refinement module called convolutional spatial propagation network (CSPN). CSPN learns affinity weights of each pixel to its neighbor pixels, where those weights are used to refine initial depth result iteratively. Park et al. (2020) compensated CSPN by proposing a non-local spatial propagation network (NLSPN), to learn non-local affinity weights that are not restricted to a square-shaped propagation kernel. There are more approaches using different versions of CSPN (Cheng et al., 2020; Cheng et al., 2020), but these works all suffer from slow inference time.

Imran et al. (2019) approached in a slightly different way by transforming the depth information into a series of depth coefficient. They represented the depth value of each pixel by a weighted summation of pre-defined discretized depth planes, and learned those coefficients by multi-label classification. They tried to address the problem that deep learning based depth regression using upsampling might lose the information of the object boundaries. By doing so, they were able to propose relatively light-weighted depth completion network, but lacked on the performance.

2.2 LAYERED DEPTH IMAGE & MULTIPLANE IMAGE

The layered depth image (LDI) is a way of representing the depth information that was introduced by Shade et al. (1998). They tried to view a scene from a single camera, but with multiple pixels along each line of sight. Each ray is highlighted on the depth layers where the ray would cross objects at the depth of each layer. Therefore, a pixel in the image can have multiple depth values on a single ray, preserving the information of the hidden object that can only be seen from the other viewpoints.

The multiplane image (MPI) (Zhou et al., 2018; Tucker & Snavely, 2020) is an image representation technique inspired by a layered depth image. They take a similar concept as an LDI, assuming multiple discrete depth planes spaced with equal disparity. However, they focused on each layer’s occupancy from the reference camera position, rather than depth value itself. This helped their additional contributions on solving view synthesis problem.

While both of an LDI and an MPI contains depth-related information, these types of representations are not suitable for 3D reconstruction, since they were not targeting to solve depth prediction.

3 METHODOLOGY

3.1 PLANE-RESIDUAL REPRESENTATION

As mentioned in section 1 and section 2, directly estimating depth information with deep learning using regression is challenging. Therefore, we try to solve depth estimation by factorizing a challenging regression problem into a combination of relatively simple classification and regression.
Position
Camera
Depth Plane 1
the absolute depth value of a pixel represented as $(p, r)$, where $p$ refers to the closest depth plane that the pixel lies. $r$ is the normalized distance, i.e., residual from the plane depth to its actual value according to $d_{\text{step}}$. Therefore, every $r$ values are in range of $[-0.5, 0.5)$, except $[0, 0.5)$ for depth plane 1 and $[-0.5, 0]$ for depth plane $D$. Given the values of the plane depths $d_p$s, the absolute depth value of a pixel represented as $(p, r)$ is,

$$D(p, r) = d_p + r \times d_{\text{step}}(p, r), \quad d_{\text{step}}(p, r) = \begin{cases} d_{p+1} - d_p & \text{if } r \geq 0 \\ d_p - d_{p-1} & \text{if } r < 0 \end{cases} \quad (1)$$

More intuitive illustration of our PR representation is shown in figure 1.

By using PR representation, we can seamlessly reformulate the depth completion task into a combination of depth plane classification and relative residual regression. Depth plane classification simplifies the densification of the given sparse depth information. Relative residual regression makes the final depth result to be more accurate and reduces discontinuity issue from the discretized depth planes.

3.2 Network Design

Our depth completion network using PR representation consists of a single ResNet-based [He et al., 2016] encoder and two decoders, decoder $P$ and decoder $R$. The overall pipeline of our approach is shown in figure 2.

The encoder takes an RGB image and spare depth information as an input. Input sparse depth is pre-processed into a PR representation. The $P$ part of the PR representation is expressed in a $D$-channeled valid mask, and the $R$ part is expressed in a single-channeled residual map. Each type of input element is processed with a single convolution layer and then concatenated all together before the ResNet-based encoder [Ma et al., 2019; Cheng et al., 2020].

Two decoders are similar structure-wise while having distinguished roles. The basic architecture of each decoder block consists of an upsampling layer and a convolution layer. Skip connection [Ronneberger et al., 2015] from the encoder is given to every matching output of the decoding blocks, which is known to be effective for retaining object boundary information. We chose feature summation rather than feature concatenation for our skip connection, since encoded features are in the similar domain as decoded features [Qiu et al., 2019].

Decoder $P$ outputs a $D$-channeled probability volume, using a softmax function. The output of the decoder $R$ is a single-channeled normalized residual map, $r_{\text{pred}}$. Using equation (1) the final
Figure 2: **Overall pipeline of our depth completion algorithm.** With a single RGB image and a sparse depth input, we solve depth completion with a combination of depth plane classification and residual regression. Please refer to section 3 for detailed information.

The prediction of the network $D_{\text{pred}}$ is acquired by,

$$D_{\text{pred}}(x, y) = \sum_{p=1}^{D} d_p \times \sigma(l_p(x, y)) + r_{\text{pred}}(x, y) \times d_{\text{step}}(p, r).$$  \hspace{1cm} (2)

Here, $\sigma(\cdot)$ is a softmax function, and $l_p(x, y)$ is the raw classification logit at the coordinate $(x, y)$ for depth plane $p$. Unlike equation 1, we used weighted summation for the final depth estimation with the probability volume $\sigma(l_p)$, to make the training process differentiable, and also to smoothen the depth result in the boundary area of two discrete depth planes.

### 3.3 Probability Volume Filtering

In order to achieve more accurate depth plane classification results, we apply channel-wise guided image filtering. Guided image filtering [He et al., 2010; He et al., 2013] uses content information from the guidance image to perform edge-preserving smoothing. Applying this technique to each channel of the cost volume for correspondence matching was introduced in Hosni et al. (2013).

To adopt this idea, we first apply two consecutive convolution layers to the input RGB image to create $D$-channeled guidance images $I_{\text{guide}}$, as in Wu et al. (2018). Then we perform channel-wise guided image filtering to the initial depth plane classification logit $l$, by

$$l_{\text{refined},p}^i = A_{\text{guide},p}^i l^i + b_{\text{guide},p}^i, \forall i \in w_k,$$  \hspace{1cm} (3)

where $i$ is a pixel location, $w_k$ is a window centered at the pixel $k$, and $l_{\text{refined},p}$ is a refined classification logit at depth plane $p$. $A_{\text{guide},p}^i$ and $b_{\text{guide},p}^i$ are determined using mean, variance and covariance values of $I_{\text{guide},p}$ and $l$ in a window $w_k$, where we use an average pooling layer to utilize this equation in the training process. Our final depth result will be modified from equation 2 by substituting $l$ to $l_{\text{refined}}$.

### 3.4 Confidence-Based Regression

As described in equation 2, our final depth prediction is obtained by a weighted summation using depth plane values and depth plane classification probability values. However, it makes it difficult for the decoder $R$ to output good residual prediction, since the values $d_p$ and $\sum_{p=1}^{D} d_p \times \sigma(l_p)$ are different.

To address this problem, we measure depth plane classification confidence to use it as a training loss weight for decoder $R$. The core idea is to give the network optimizer some guidance on where to put heavier weight between ground truth depth map $D_{\text{gt}}$, or ground truth residual map $r_{\text{gt}}$. For confidence measure, we use the channel-wise maximum value of the probability volume $\sigma(l)$ on each pixel, as in,

$$c(x, y) = \max(\sigma(l_1(x, y)), \sigma(l_2(x, y)), \cdots, \sigma(l_D(x, y))),$$  \hspace{1cm} (4)
where \( c(x, y) \) is the confidence value at the pixel coordinate \((x, y)\).

For a pixel with high depth plane classification confidence, more supervision of the ground truth residual value is given, since \( d_p \) will be more similar to \( \sum_{p=1}^{D} d_p \times \sigma(l_p) \). On the other side, for a pixel with low depth plane classification confidence, the decoder \( R \) will be trained with more guidance from \( D_{gt} \).

### 3.5 Loss Function

For accurate depth prediction and effective training, we trained our depth completion network with three loss terms, which are final depth loss, depth plane classification loss, and residual loss.

**Final depth loss.** The first loss term is a simple \( L^1 \) loss between the final depth prediction \( D_{pred} \) from equation (2) and the ground truth depth \( D_{gt} \), as in,

\[
L_D = |D_{gt} - D_{pred}|.
\]

**Depth plane classification loss.** To maximize the benefits of our PR representation, we apply cross-entropy loss for the depth plane classification from decoder \( P \). Therefore, the cross-entropy loss \( H(P_{gt}, l) \) for our depth plane classification logit \( l \) can be described as,

\[
H(P_{gt}, l) = -\sum_{p=1}^{D} P_{gt}(p) \log(\sigma(l_p)).
\]

\( P_{gt}(p) \) is a ground truth depth plane classification binary indicator, showing if the pixel is in the depth plane \( p \).

As described in section 3.3, we also have a refined logit output \( l_{refined} \). We calculate the same loss for both initial and refined classification prediction. Our final depth plane classification loss \( L_P \) will be defined as following:

\[
L_P = \lambda H(P_{gt}, l) + H(P_{gt}, l_{refined}),
\]

with \( \lambda \) being a loss weight parameter which was set to 0.7, empirically.

**Residual loss.** The last loss term is also a simple \( L^1 \) loss, calculated between the residual prediction \( r_{pred} \) and the ground truth residual map \( r_{gt} \). With the confidence explained in section 3.4 our residual loss can be interpreted as,

\[
L_R = c \odot |r_{gt} - r_{pred}|,
\]

where \( \odot \) indicates the Hadamard product.

Our final loss function \( L \) for end-to-end training of our depth completion network is,

\[
L = L_D + L_P + \frac{1}{D} L_R.
\]

Since the residual part of our PR representation is normalized to a value of \( d_{step} \), we penalize the residual loss with the parameter of \( \frac{1}{D} \).

### 4 Experiment

#### 4.1 Dataset & Training

**NYU Depth V2 dataset.** The NYU Depth V2 dataset [Nathan Silberman & Fergus, 2012] contains RGB and depth sequences from a Kinect sensor, in 464 indoor scenes. For both training and testing, we simulate uniform depth point sampling to randomly select 500 depth samples for input depth images. To neglect the image boundary area with missing depth measurements, we downsize each image into a size of \( 320 \times 240 \), and then center-cropped it to be a size of \( 304 \times 228 \). We utilized \( \sim 48K \) image pairs from the official training set and used 654 pairs of fully-labeled dataset for evaluating.

**KITTI Depth Completion dataset.** The KITTI Depth Completion dataset [Geiger et al., 2012; Uhrig et al., 2017] is a driving scene dataset that was captured by stereo RGB cameras and a LiDAR sensor. We use 2D-projected LiDAR measurement as input, and accumulated depth map with
Table 1: **Depth completion evaluation results on the NYU Depth V2 dataset.** Algorithms on the lower block utilizes any kind of iterative post-refinement process. The metrics RMSE and REL are presented in meters (m). Numbers of parameters were driven from officially released codes.

| Method                  | # params (M) | RMSE | REL | \(\delta_1\) | \(\delta_2\) | \(\delta_3\) |
|-------------------------|--------------|------|-----|---------------|---------------|---------------|
| Ma & Karaman (2018)     | 28.39        | 0.230| 0.044| 97.1          | 99.4          | 99.8          |
| Imran et al. (2019)     | 27.02        | 0.118| 0.013| -             | -             | -             |
| Qiu et al. (2019)       | 53.44        | 0.115| 0.022| 99.3          | 99.9          | 100.0         |
| **Ours**                | **21.43**    | **0.104**| 0.015| **99.4**      | **99.9**      | **100.0**     |
| Cheng et al. (2018)     | 256.08       | 0.117| 0.016| 99.2          | 99.9          | 100.0         |
| Xu et al. (2019)        | 28.99        | 0.112| 0.018| 99.5          | 99.9          | 100.0         |
| Park et al. (2020)      | 25.84        | 0.092| 0.012| **99.6**      | 99.9          | 100.0         |

Figure 3: **Qualitative depth completion results on the NYU Depth V2 dataset.** Depth plane classification results were shown in reversed color scheme for better visibility.

11 consecutive frames as ground truth. We ignore the regions with no LiDAR measurements, by bottom-cropping the images with the size of 1216 \(\times\) 352. We used \(\sim\)93K image pairs for training, and selected validation set of 1000 pairs for comparisons with other approaches.

For both datasets, we adopted ResNet-18 [He et al., 2016] model for our encoder, and used ADAM optimizer with learning rate of \(1e^{-3}\), weight decay rate of \(1e^{-6}\), and \(\beta\) of \((0.9, 0.999)\). The learning rate was reduced to its 20\% every 5 epochs, where the training was done for 20 epochs in total. We set the batch size to 32 for the NYU Depth V2 dataset, and 4 for the KITTI Depth Completion dataset, using 4 NVIDIA RTX 2080 Ti GPUs, which take half a day and 3 days in training, respectively.

We set \(d_1\) and \(d_D\) as the minimum and the maximum depth values of the given sparse input samples. We place the intermediate planes equally spaced depth-wise, thereby the value of \(d_{\text{step}}\) is identical in every planes. We chose the number of depth planes \(D\) as 8 for the NYU Depth V2 dataset, and 64 for the KITTI Depth Completion dataset, regarding the maximum range difference between two datasets. Comparison between other ways of choosing \(d_p\) and \(d_{\text{step}}\) will be shown in section 4.3.

4.2 **Comparisons with state-of-the-art**

We use commonly used metrics for evaluation as following: root mean squared error (RMSE), mean absolute error (MAE), relative mean absolute error (REL), root mean squared error of the inverse depth (iRMSE), mean absolute error of the inverse depth (iMAE), and a percentage of predicted pixels with the relative error being smaller than 1.25\(\uparrow\), \(\delta_i\).

In table 1, we present our quantitative evaluation result on the NYU Depth V2 dataset with others. Among the methods with no additional iterative post-processing refinement, our result performs the best, in terms of RMSE and \(\delta_i\). While [Imran et al., 2019] has lower REL error, we believe that it is due to the fact that they used 80 channels of depth bases. Our algorithm uses only 8 depth planes, yet combines the strength of classification and regression, thereby shows better performance on other
Table 2: **Depth completion evaluation results on the KITTI Depth Completion validation dataset.** The metrics are presented in millimeters (mm). Running times were taken from the official KITTI benchmark website.

| Method               | Runtime (s) | RMSE   | MAE   | iRMSE  | iMAE  |
|----------------------|-------------|--------|-------|--------|-------|
| Ma et al. (2019)     | 0.08        | 814.73 | 249.95| 2.80   | 1.21  |
| Xu et al. (2019)     | 0.10        | 811.07 | 236.67| 2.45   | 1.11  |
| Qiu et al. (2019)    | 0.07        | 687.00 | 215.38| 2.51   | 1.10  |
| Chen et al. (2019)   | 0.09        | 785.00 | 217.00| 2.36   | 1.08  |
| Cheng et al. (2020)  | 0.20        | 725.43 | 207.88| -      | -     |
| **Ours**             | **0.06**    | **867.12** | **204.68** | **2.17** | **0.85** |

RGB
Predicted Depth Map
Ground Truth Depth Map
Depth Plane Classification

Figure 4: Qualitative depth completion results on the KITTI Depth Completion dataset. Depth plane classification results were shown in reversed color scheme for better visibility.

metrics. Comparing to the algorithms with iterative post-refinement, we still achieve the second best result, while having the least number of parameters.

Qualitative results on the NYU Depth V2 dataset are also shown in figure 3. Compared to Park et al. (2020), although we did not use any iterative post-refinement processing, our results show even better details in depth map estimations, as the wheels of the chair in the first example. This is due to our depth plane classification, which avoids depth mixing on object boundaries, and our context-guided probability volume filtering. Also, since our decoder can learn normalized residual map plane-wise, it is easier to obtain more details by magnifying information between depth planes.

Table 2 shows the evaluation results on the official KITTI Depth Completion validation dataset. Our algorithm excels others in MAE, iRMSE, and iMAE metrics, while having the shortest inference time. We believe that since the KITTI dataset ground truth depth map is too sparse to make a reliable depth plane ground truth labels, our depth plane classification with decoder is more challenged, thereby falls short on the RMSE metric. Figure 4 shows qualitative results on depth completion and plane classification of our algorithm.

4.3 Ablation Study

**Setting the depth planes.** As mentioned in section 3.1 the values of plane depths and the number of planes can be set arbitrarily. Our proposed way of determining the parameters is to set and according to the sparse input values, and to place intermediate planes equally spaced in between. We name this setup UR, as in uniformly and relatively set depth planes.

In table 3 we present three other setups, which are, UA (uniformly & absolutely), DR (disparity-wise & relatively), and DA (disparity-wise & absolutely). For example, and were set to 0m and 10m in UA setup, regarding the overall depth range of the NYU Depth V2 dataset. The results show that DR and DA suffer from varying values, therefore lack in residual regression. UA performs better than disparity-wise plane settings but worse than UR, since the distance that each plane should represent is more broad. However, DR preserves better object boundary information in close region, as shown in figure 5 (a), because it placed more planes on the front. We believe that the best way to setup depth planes in our PR representation would vary, depending on the task.
Table 3: Ablation study on our design choices. Other options beside selected category were set to default settings of “Ours”. Default settings are; UR depth plane setup, 8 planes, with $l(p)$ filtering, and with confidence-guided learning. The results were taken from the NYU Depth V2 dataset.

| Category                | Method | RMSE  | REL   | $\delta_1$ | $\delta_2$ | $\delta_3$ |
|-------------------------|--------|-------|-------|-------------|-------------|-------------|
| Depth Plane Setup       | UA     | 0.113 | 0.016 | 99.3        | 99.9        | 100.0       |
|                         | DR     | 0.114 | 0.016 | 99.3        | 99.9        | 100.0       |
|                         | DA     | 0.117 | 0.017 | 99.1        | 99.8        | 99.9        |
| Number of Planes        | 4      | 0.111 | 0.016 | 99.4        | 99.9        | 100.0       |
|                         | 16     | 0.105 | 0.015 | 99.4        | 99.9        | 100.0       |
|                         | 32     | 0.106 | 0.015 | 99.4        | 99.9        | 100.0       |
| $l(p)$ Filtering        | No     | 0.105 | 0.015 | 99.4        | 99.9        | 100.0       |
| Confidence-guided       | No     | 0.122 | 0.018 | 99.2        | 99.8        | 99.9        |
|                         | Ours   | 0.104 | 0.015 | 99.4        | 99.9        | 100.0       |

Figure 5: Ablation study on our design choices. (a) Depth plane classification results on different depth planes setups. (b) Improvement on using channel-wise guided image filtering. (c) Failure case of not using confidence-guided residual learning.

We also examine the influence of the number of planes. The number of planes affects the ratio of classification and regression in our depth completion network. As shown in table 3, our result with plane number of 8 performs the best, and the experiments with 16 and 32 number of depth planes show better results than the one with 4 depth planes. Our intuition is that if the plane number is large, it indeed lessens the burden for the decoder $R$, and therefore can acquire better result. However, it would make initial input on each plane to be very sparse, which will make the plane classification part more challenging.

**Probability volume filtering.** We show the effect on our probability volume filtering by using channel-wise guided image filter. In table 3, it shows that using our probability volume filtering increases the overall performance. It is more clearly shown in figure 5 (b), where the depth plane classification result contains more information from the object boundaries.

**Confidence-guided residual regression.** We examine on how confidence-guided adaptive residual learning improves the result. As presented in table 3 using confidence-guided learning improves very much of a performance. Also, in figure 5 (c), the result without confidence-guided learning finds it difficult to predict the right residual values at the boundary regions between two discrete planes, and therefore makes a discontinuity in the final predicted depth map.

5 Conclusion

In this paper, we address the main difficulties in previous depth completion algorithms, which are depth mixing on object boundaries, heavy network computation, and slow inference. We propose a novel way to solve depth completion, by reformulating depth regression problem into a combination of depth plane classification and residual regression. We also introduce Plane-Residual representation, which enables our approach, and also can be used in many other 3D related tasks. We show competitive results with faster computation, thereby validify our idea and design choices.
REFERENCES

Y. Chen, B. Yang, M. Liang, and R. Urtasun. Learning joint 2d-3d representations for depth completion. In IEEE International Conference on Computer Vision (ICCV), 2019.

X. Cheng, P. Wang, and R. Yang. Learning depth with convolutional spatial propagation network. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2020.

Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In European Conference on Computer Vision (ECCV), 2018.

Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In AAAI Conference on Artificial Intelligence (AAAI), 2020.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

K. He, J. Sun, and X. Tang. Guided image filtering. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2013.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

Kaiming He, Jian Sun, and Xiaou Tang. Guided image filtering. In European Conference on Computer Vision (ECCV), 2010.

A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2013.

S. Imran, Y. Long, X. Liu, and D. Morris. Depth coefficients for depth completion. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In International Conference on 3D Vision (3DV), 2016.

B. Lee, H. Jeon, S. Im, and I. S. Kweon. Depth completion with deep geometry and context guidance. In IEEE International Conference on Robotics and Automation (ICRA), 2019.

F. Ma and S. Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In IEEE International Conference on Robotics and Automation (ICRA), 2018.

Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In IEEE International Conference on Robotics and Automation (ICRA), 2019.

Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In European Conference on Computer Vision (ECCV), 2012.

Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In European Conference on Computer Vision (ECCV), 2020.

Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015.
Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In Annual conference on Computer graphics and interactive techniques (SIGGRAPH), 1998.

Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In International Conference on 3D Vision (3DV), 2017.

Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. In International Conference on Machine Vision Applications (MVA), 2019.

Huikai Wu, Shuai Zheng, Junge Zhang, and Kaiqi Huang. Fast end-to-end trainable guided filter. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints. In IEEE International Conference on Computer Vision (ICCV), 2019.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In Annual conference on Computer graphics and interactive techniques (SIGGRAPH), 2018.