Depth Completion using Plane-Residual Representation

Byeong-Uk Lee Kyunghyun Lee In So Kweon
Korea Advanced Institute of Science and Technology
Daejeon, Republic of Korea
{byeonguk.lee, kyunghyun.lee, iskweon77}@kaist.ac.kr

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, while 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, compared to previous 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, obtaining precise and reliable 3D scene information is crucial. 3D reconstruction has been studied for many years and certainly can be categorized as one of the traditional computer vision tasks, but it is still a core technology. There are various ways of obtaining 3D information, such as monocular depth prediction, structure from motion, and multi-view stereo. However, 3D reconstruction using additional sensors alongside an RGB camera like secondary camera, depth sensors, or radar will be more effective, since it can utilize more accurate depth measurements as prior information. Given that most of the recent mobile devices have more than one camera and even a LiDAR sensor, it is widely acceptable that 3D reconstruction by integrating multiple sensor inputs is a more efficient and practical approach.

While stereo matching being one of the most conventional and reliable ways in 3D reconstruction, it shows its weakness in depth prediction on the farther area due to the physical restriction for large baseline between stereo cameras. Therefore, using a depth sensor to obtain more accurate and absolute initial depth information is also preferred. However, 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 and Karaman [18]. 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 [14]. 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 [3, 17, 16, 26, 4, 2, 21]. 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 [16, 28, 22]. 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 and Karaman [18] proposed depth completion using convolutional neural network (CNN). They utilized the overall network structure from Laina *et al.* [15] 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.* [3] and Park *et al.* [21] 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.* [21] 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 [4, 2], but these works all suffer from slow inference time.

Imran *et al.* [14] approached in a slightly different way, by transforming the depth information into a series of depth coefficient for multiple discrete depth basis. 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.

Chen *et al.* [1] introduced a way to maximize the 3D information from the sparse depth input. They applied 2D convolution on image pixels and continuous 3D convolution on point cloud, so that each branch extracts useful features from each side. They fused the features from both sides in a multi-scaled manner, to get more accurate depth completion result, with well-preserved object boundary information.

2.2. Depth Prediction using Multiple Planes

Imran *et al.* [14] used discretized multiple planes and corresponding coefficients. 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.

Fu *et al.* [6] proposed the ordinal regression to estimate the depth from a monocular image. They compared multi-label classification and ordinal regression with predefined multiple depth planes. The proposed deep ordinal regression network (DORN) divided the depth estimation problem into sub-problems of plane-wise binary classification, each determining whether the true depth value is larger than the discrete depth of each plane. However, they used the discrete depth value itself in the final depth generation, therefore, the accuracy is limited.

Meanwhile, plane-sweeping algorithms are famous stereo matching methods using multiple planes. [5, 12, 13] They utilize cost-volume by stacking the photometric loss of the warped image from every discrete plane. Im *et al.* [13] integrated the process into the deep learning framework and proposed an end-to-end plane-sweeping algorithm. However, they need a new image from a different viewpoint with known extrinsic parameters to calculate cost volume. Therefore, they are usually used in a stereo setup or with consecutive image sequences.

Other methods to represent an image with multiple planes are Multi-Plane Image (MPI). Tucker *et al.* [24] and Zhou *et al.* [29] represented a single RGB image as multiple RGBA images. In this kind of representation method, the final image is obtained by the cumulative multiplication of the layers with additional transparency value. However, these methods are not suitable to be directly used in depth prediction, since they were designed for differentiable rendering and view synthesis.

3. Methodology

In this section, we elaborate our new way of representing depth information, which we named Plane-Residual (PR) Representation, with detailed network structure and design choices for our end-to-end depth completion algorithm. Overall pipeline of our approach is shown in Fig. 2.

3.1. Plane-Residual Representation

As mentioned in Sec. 1 and Sec. 2, directly estimating depth information with deep learning using regression is

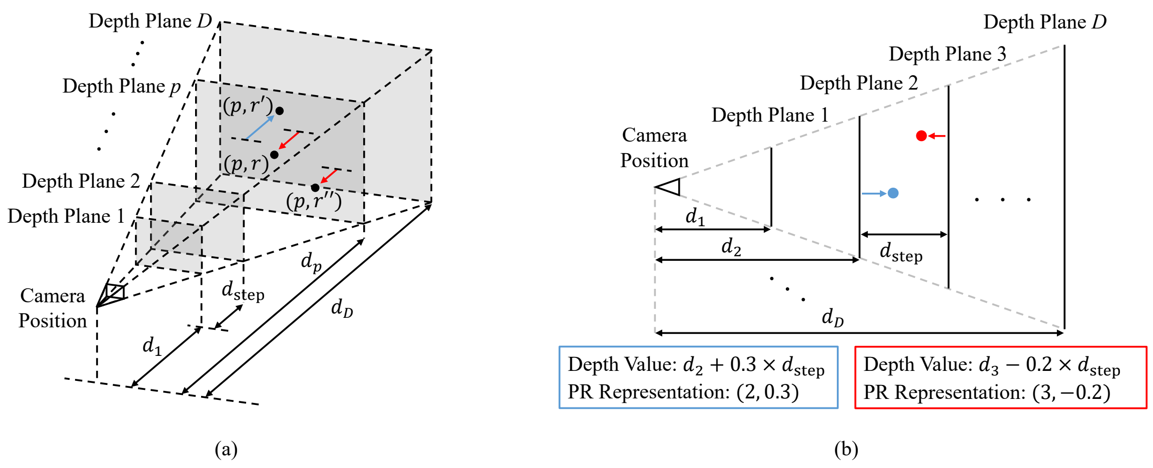

Figure 1. **PR representation.** (a) 3D illustration of our PR representation. (b) 2D illustration and examples of how absolute depth values are interpreted in PR representation. Equations inside the blue-lined and red-lined boxes notate the PR representations from the actual depth values of the blue dot and the red dot, respectively.

challenging. Therefore, we try to solve depth estimation by factorizing a challenging regression problem into a combination of relatively simple classification and regression.

To enable this idea, first we introduce a Plane-Residual (PR) representation, a novel way of interpreting depth information. Our PR representation assumes a set of D frontoparallel planes to the camera, where each plane represents a discrete depth value $d_{p\in\{1,2,\cdots,D\}}$. Depth plane 1 is the closest plane from the camera, and depth plane D is the farthest. The values of $d_1, d_2, \cdots, d_D$ and d_{step}, which is the distance between two adjacent depth planes, can be determined arbitrarily.

Within a depth image, the depth value of each pixel is then expressed 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_{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_ps, the absolute depth value of a pixel represented as (p, r) is,

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

More intuitive illustration of our PR representation is shown in Fig. 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 [10] encoder and two decoders, decoder P and decoder R. The overall pipeline of our approach is shown in Fig. 2.

The encoder takes an RGB image and sparse 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 [17, 2].

Two decoders are similar structure-wise while having distinguished roles. Each decoder block in both decoders consists of a deconvolution [20] layer. Skip connection [23] from the encoder is given to every matching output of the decoding blocks in decoder P, which is known to be effective for retaining object boundary information. Moreover, in our proposed PR representation, residual value r is determined in regard of the depth plane p. Therefore, some kind of guidance from plane classification process should be given when regressing residual map. To give this guidance while maintaining the parameter-wise effectiveness of the whole depth completion network, we chose to give additional skip connection from the decoded features of decoder P into the decoded features of decoder R. More detailed and intuitive architecture of our depth completion network using PR representation is shown in Fig. 3. We chose feature summation rather than feature concatenation for our skip connection, since encoded features are in the similar domain as decoded features [22].

Decoder P outputs a D-channeled probability volume,

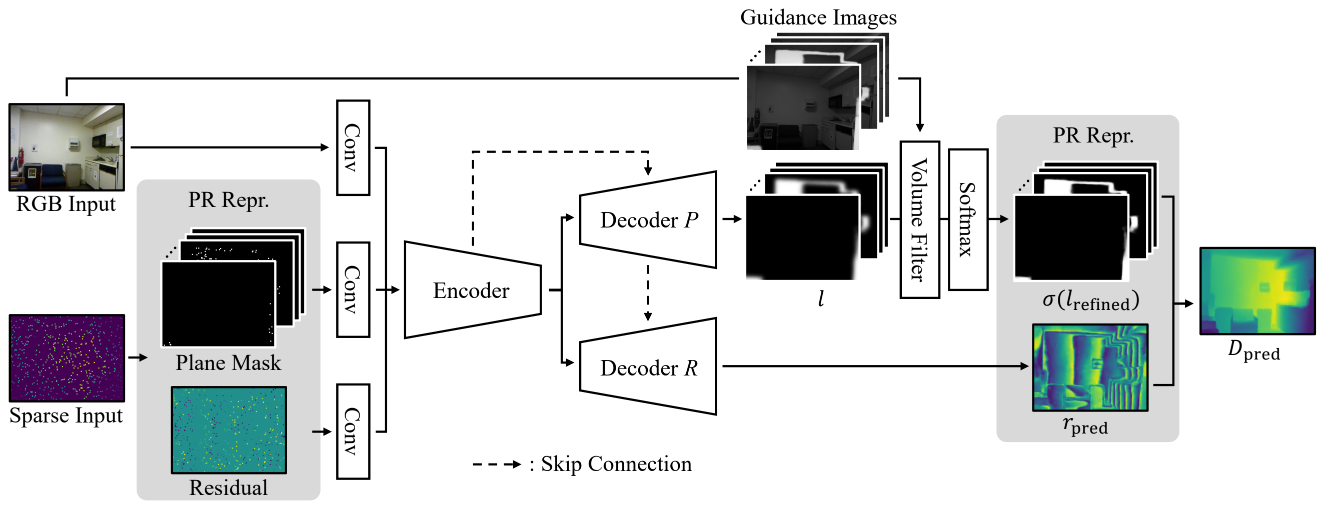

Figure 2. **Overall pipeline of our depth completion algorithm.** With a single RGB image and a sparse depth input, we solve depth completion with a combination of depth plane classification and residual regression. Please refer to Sec. 3 for detailed information.

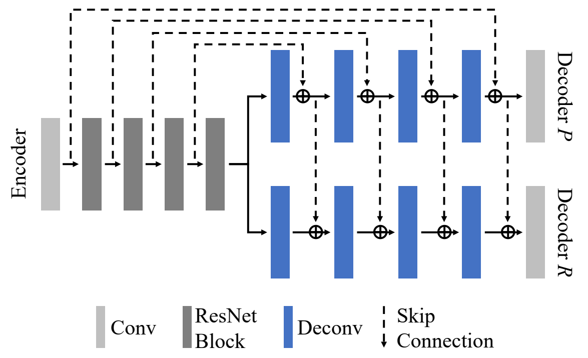

Figure 3. **Illustration of the detailed network architecture and our late feature fusion technique.** Decoded feature maps from decoder P are fused to the decoded feature maps in decoder R, to guide residual regression with plane classification information.

using a softmax function. The output of the decoder R is a single-channeled normalized residual map, r_{pred}. Using Eq. (1), the final prediction of the network D_{pred} is acquired by,

$$\begin{aligned} D_{\text{pred}}(x,y) = &\sum_{p=1}^{D} d_p \times \sigma(l_p(x,y)) \\ &+ r_{\text{pred}}(x,y) \times d_{\text{step}}(\hat{p}, r_{\text{pred}}(x,y)). \end{aligned} \tag{2}$$

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. $\hat{p}$ is the classified depth plane at the pixel (x,y) with the highest probability of $\sigma(l_p(x,y))$. Unlike Eq. (1), we used weighted summation for the final depth estimation with the probability volume $\sigma(l_p)$, 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 [8, 9] 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 and was shown to be effective in [11]. Since our probability volume for depth plane classification has similar structure to the cost volume in plane-sweeping and stereo matching, we believe that this kind of technique can be easily applied to our approach as well.

To adopt this idea, we first apply two consecutive convolution layers to the input RGB image to create D-channeled guidance images I_{guide}, as in [27]. Then we perform channel-wise guided image filtering to the initial depth plane classification logit l, by

$$l^i_{\text{refined},p} = A^k_p I^i_{\text{guide},p} + b^k_p, \ \forall i \in w_k, \tag{3}$$

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^k_p and b^k_p are determined using mean, variance and covariance values of $I_{\text{guide},p}$ and l_p 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 Eq. (2) by substituting l to l_{refined}, as in,

$$\begin{aligned} D_{\text{pred}}(x,y) = &\sum_{p=1}^{D} d_p \times \sigma(l_{\text{refined},p}(x,y)) \\ &+ r_{\text{pred}}(x,y) \times d_{\text{step}}(\hat{p}, r_{\text{pred}}(x,y)). \end{aligned} \tag{4}$$

3.4. Confidence-Based Regression

As described in Sec. 3.2, the residual value r depends on the classified depth plane p of a pixel. Therefore, reliable depth plane classification performance is required for robust residual learning. If the decoder R was trained with supervision of the ground truth residual map while the decoder P gives false depth plane prediction, the final depth map will result in poor depth map quality with discretizing artifacts on the depth plane boundary region.

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_{gt}, or ground truth residual map r_{gt}. For confidence measure, we use the channel-wise maximum value of the probability volume $\sigma(l)$ on each pixel, as in,

$$c(x, y) = \max(\sigma(l_1(x, y)), \cdots, \sigma(l_D(x, y))), \qquad (5)$$

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, and for a pixel with low depth plane classification confidence, the decoder R will be trained with more guidance from D_{gt}.

In the earlier stage of the training, the network can focus more on the depth plane classification. In the latter stage of the training, when the depth plane classification becomes more reliable, the network can learn residual map more efficiently. Also, for the boundary area of two adjacent depth planes, the decoder R will be trained adaptively between the ground truth residual map and the ground truth depth map.

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 $L1$ loss between the final depth prediction D_{pred} from Eq. (4) and the ground truth depth D_{gt}, as in,

$$L_D = |D_{\text{gt}} - D_{\text{pred}}|. \qquad (6)$$

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 $H(P_{\text{gt}}, l)$ for our depth plane classification logit l can be described as,

$$H(P_{\text{gt}}, l) = -\sum_{p=1}^{D} P_{\text{gt}}(p) \log(\sigma(l_p)). \qquad (7)$$

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

As described in Sec. 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:

$$L_P = \lambda H(P_{\text{gt}}, l) + H(P_{\text{gt}}, l_{\text{refined}}), \qquad (8)$$

with λ being a loss weight parameter which was set to 0.7, empirically.

Residual loss. The last loss term is also a simple $L1$ loss, calculated between the residual prediction r_{pred} and the ground truth residual map r_{gt}. With the confidence explained in Sec. 3.4, our residual loss can be interpreted as,

$$L_R = c \odot |r_{\text{gt}} - r_{\text{pred}}|, \qquad (9)$$

where $\odot$ indicates the Hadamard product.

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

$$L = L_D + L_P + \frac{1}{D} L_R. \qquad (10)$$

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 [19] contains RGB and depth sequences in 464 indoor scenes. The dataset was captured by a Kinect sensor, which provides short-ranged semi-dense depth measurements. For both training and testing, we simulate uniform depth point sampling to randomly select 500 depth samples for each image & depth pair. To neglect the image boundary area with missing depth measurements, we downsized each image into a size of 320×240, and then center-cropped it to be a size of 304×228. We utilized $\sim$48K image pairs from the official training set and used handpicked 654 pairs among 1449 pairs of fully-labeled dataset for evaluating, following previous works.

KITTI Depth Completion dataset. The KITTI dataset [7] is a driving scene dataset that was captured by stereo RGB cameras, stereo monochrome cameras, and a LiDAR sensor. The KITTI Depth Completion dataset [25] provides RGB image and 2D-projected LiDAR sensor measurement as a pair. It utilizes 11 consecutive frames to generate denser depth map as a ground truth. For training, we ignored the regions with no LiDAR measurements, by bottom-cropping the images with the size of 1216×256. We used $\sim$93K image pairs for training, and official selected validation set of 1000 pairs for comparisons with other approaches.

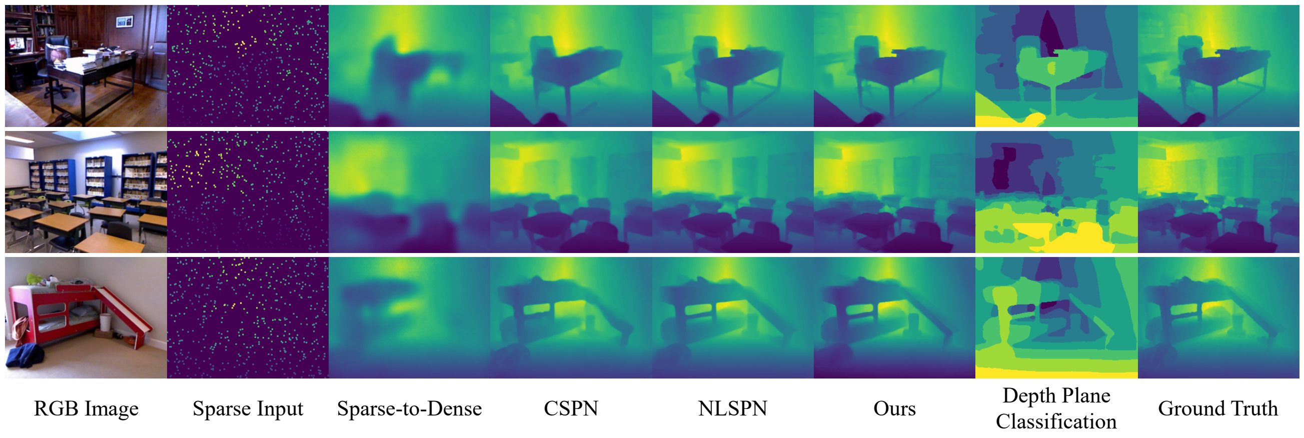

Figure 4. **Qualitative depth completion results on the NYU Depth V2 dataset.** Depth plane classification results were shown in reversed color scheme for better visibility.

Method	# params (M)	RMSE	REL	δ_1	δ_2	δ_3
Sparse-to-Dense [18]	28.39	0.230	0.044	97.1	99.4	99.8
DC-3coeff [14]	27.02	0.131	**0.013**	97.9	99.3	99.8
DC-all [14]	27.02	0.118	**0.013**	97.8	99.4	99.9
DeepLiDAR [22]	53.44	0.115	0.022	99.3	99.9	100.0
Ours	**14.34**	**0.104**	0.014	**99.4**	99.9	100.0
CSPN [3]	256.08	0.117	0.016	99.2	99.9	100.0
Depth-Normal [28]	28.99	0.112	0.018	99.5	99.9	100.0
NLSPN [21]	25.84	**0.092**	**0.012**	**99.6**	99.9	100.0

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.

For both datasets, we adopted ResNet-18 [10] model for our encoder, and used ADAM optimizer with learning rate of $1e^{-3}$, weight decay rate of $1e^{-6}$, and β 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 8 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_{step} is identical in every plane. 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_ps and d_{step} will be shown in Sec. 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^i, δ_i.

In Tab. 1, we present our quantitative evaluation result on the NYU Depth V2 dataset with others. We categorized various depth completion algorithms into two categories, where the first one refers to the methods with no iterative post-processing refinement, and the other one means the opposite. In Tab. 1, the upper block shows the previous approaches and ours, which are in the first category, and the lower block contains algorithms like CPSN [3] and NLSPN [21].

Among the methods with no additional iterative post-processing refinement, our result performs the best in most of the metrics. More specifically, our depth completion network using PR representation outperforms Depth Coefficient (DC) algorithm by Imran *et al.* [14] in terms of RMSE and δ_is. As was introduced in Sec. 2, Depth Coefficient took similar strategy as ours, by defining multiple depth planes with discrete depth values. They tried to predict depth coefficients using multi-class classification, which were then used for weighted summation in the final depth reconstruction. Depth Coefficient used 80 channels of depth bases, and DC-all applied weighted summation using all of these 80 depth values, while DC-3coeff used only three adjacent depth values with the highest probability. However, DC only performs classification, which is why we believe that they show reliable REL score, but it also requires sufficient number of depth channels to maintain certain level of performance. On the other hand, our algorithm uses only 8 depth planes, yet combines the strength of classification and regression, thereby shows much better performance on other metrics.

We also report the number of network parameters of each

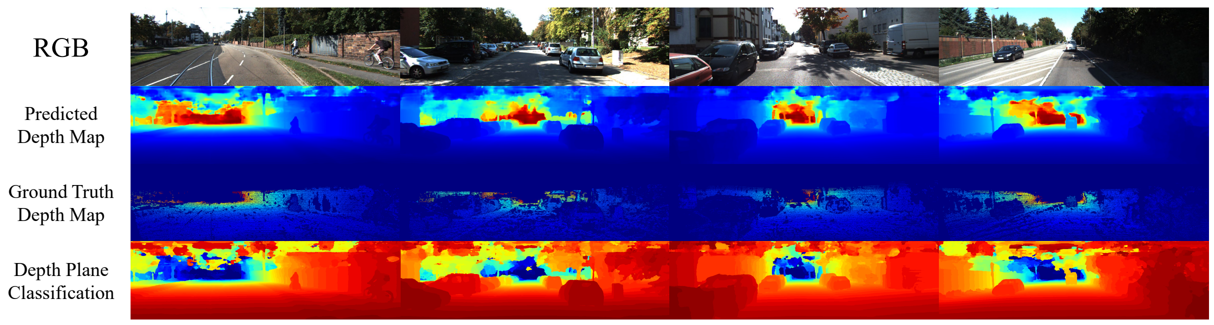

Figure 5. **Qualitative depth completion results on the KITTI Depth Completion dataset.** Depth plane classification results were shown in reversed color scheme for better visibility.

Method	Runtime (s)	RMSE	MAE	iRMSE	iMAE
Self-Supervised S2D [17]	0.08	814.73	249.95	2.80	1.21
DeepLiDAR [22]	0.07	**687.00**	215.38	2.51	1.10
Depth-Normal [28]	0.10	811.07	236.67	2.45	1.11
DC-3coeff [14]	0.15	1212.00	241.00	-	-
DC-all [14]	0.10	1061.00	252.00	-	-
FuseNet [1]	0.09	785.00	217.00	2.36	1.08
CSPN++ [2]	0.20	725.43	207.88	-	-
Ours	**0.01**	867.12	**204.68**	**2.17**	**0.85**

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. Note that a few of the reported methods have different experimental environments.

Category	Method	RMSE	REL	δ_1	δ_2	δ_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**	**0.014**	**99.4**	99.9	100.0

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.

algorithm, to analyze the efficiency of the approach. Compared to CSPN [3], which uses ResNet-50 as encoder, and utilizes heavy and time-consuming iterative post-refinement module, our depth completion network using PR representation performs much better in every metrics, while having only 5$\sim$6% of network parameters. Also, our approach shows comparable result in comparison to NLSPN [21], which is the state-of-the-art method, with $\sim$55% of network parameters. Overall, our result shows the second-best performance in most of the metrics with the smallest number of network parameters, that is 5$\sim$55% of others. This shows that our way of reformulating depth completion problem into a combination of depth plane classification and residual map regression is efficient and effective approach.

Qualitative results on the NYU Depth V2 dataset are also shown in Fig. 4. Compared to NLSPN [21], 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, or the details of the bookshelves in the second 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 R can learn normalized residual map plane-wise, it is easier to obtain more details by magnifying information between depth planes.

Evaluation results on the official KITTI Depth Completion validation dataset is shown in Tab. 2. For the KITTI Depth Completion dataset, we also report running time of each algorithm, where the numbers were retrieved from official KITTI benchmark website. Our algorithm has the shortest inference time, where it is 7 to 20 times faster, comparing to other algorithms. Also, our results excel others in MAE, iRMSE, and iMAE metrics. This again shows that our approach is efficient, in both network mobility and computational speed. Fig. 5 shows qualitative results on depth completion and plane classification of our algorithm.

4.3. Ablation Study

Setting the depth planes. As mentioned in Sec. 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 d_1 and d_D according to the sparse input values, and to place intermediate planes equally spaced in between. We name this setup UR, as in **U**niformly and **R**elatively set depth planes.

In Tab. 3, we present three other setups, which are, UA (**U**niformly & **A**bsolutely), DR (**D**isparity-wise & **R**elatively), and DA (**D**isparity-wise & **A**bsolutely). For example, d_1 and d_D were set to 0m and 10m in UA setup, regarding the overall depth range of the NYU Depth V2

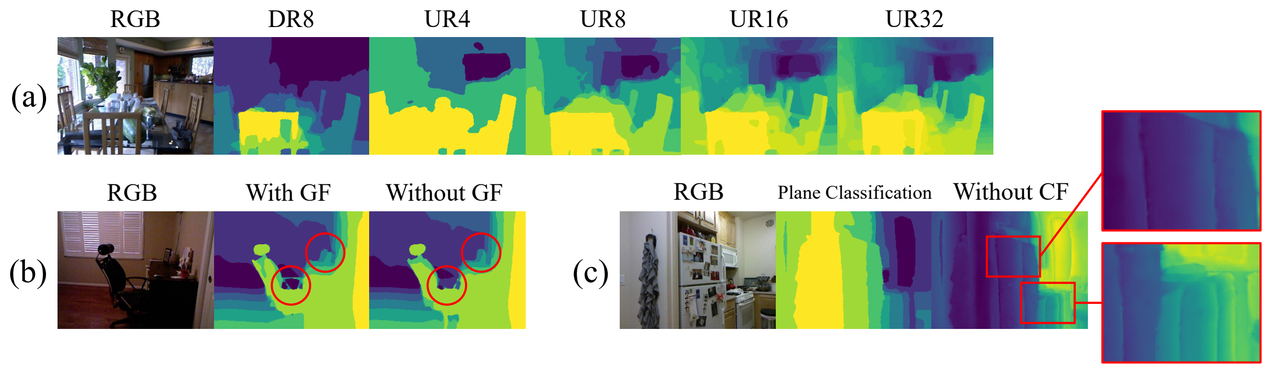

Figure 6. **Ablation study on our design choices.** (a) Depth plane classification results on different depth planes setups. (b) Improvement on using channel-wise guided image filtering. (c) Failure case of not using confidence-guided residual learning.

dataset. Some of the previous multiplane-related depth prediction approaches like DPSNet [13] or DORN [6] claimed that placing depth planes disparity-wise or in increasing log scale would boost the performance compared to uniformly-placed depth plane setup. However, in the case of using PR representation for depth completion, the results show that UR and UA perform better than DR and DA. Unlike DPSNet [13] and DORN [6], which only focus on finding the best depth plane to place a pixel, our depth completion method using PR representation estimates residual map on top of the depth plane classification result. Since we designed our predicted residual map to have a single channel, we believe that disparity-wise depth plane settings like DR and DA suffer from varying d_{step} values, which hardens the residual regression. Also, UA performs better than disparity-wise plane settings but worse than UR, since the distance that each plane should represent is broader. On the other hand, DR preserves better object boundary information in close region, as shown in Fig. 6-(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.

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 Tab. 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 lessens the burden for the decoder R. However, setting more depth planes results in the increased difficulty of depth plane classification. Moreover, it would make initial input on each plane to be very sparse, which will also make the plane classification part more challenging.

Probability volume filtering. We show the effect of our probability volume filtering using channel-wise guided image filter. Although our channel-wise guided filter do increase the quantitative evaluation result of the final predicted depth, it does not boost the performance dramatically, as can be seen in Tab. 3. However in Fig. 6-(b), we can see that the depth plane classification result with our channel-wise guided image filter contains more detailed information on the object boundaries area, compared to the result with no filter applied.

Confidence-guided residual regression. We examine on how confidence-guided adaptive residual learning improves the result. As was explained in Sec. 3.4, in order to make the reconstructed depth result more seamless on the depth plane boundary region, residual learning using depth plane classification-based confidence information is needed. As presented in Tab. 3, using confidence-guided learning improves very much of a performance. Also, in Fig. 6-(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 addressed the main difficulties in previous depth completion algorithms, which are depth mixing on object boundaries, heavy network computation, and slow inference. We proposed a novel way to solve depth completion, by reformulating depth regression problem into a combination of depth plane classification and residual regression. We introduced Plane-Residual (PR) representation as well, where we represent a depth pixel as two separate values as in (p, r), where p is the closest discrete depth plane, and r is the distance from the plane. We showed competitive results with lighter and faster computation using our approach, thereby verified our idea and design choices. We also believe that PR representation can be utilized in various depth-related problems like monocular depth estimation and multi-view stereo matching.

References

[1] 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. 2, 7

[2] 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. 1, 2, 3, 7

[3] 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. 1, 2, 6, 7

[4] X. Cheng, P. Wang, and R. Yang. Learning depth with convolutional spatial propagation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 1, 2

[5] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5515–5524, 2016. 2

[6] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 8

[7] 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. 5

[8] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. In *European Conference on Computer Vision (ECCV)*, 2010. 4

[9] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2013. 4

[10] 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. 3, 6

[11] 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. 4

[12] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2821–2830. IEEE, 2018. 2

[13] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. Dpsnet: End-to-end deep plane sweep stereo. In *International Conference on Learning Representations*, 2018. 2, 8

[14] 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. 1, 2, 6, 7

[15] 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. 2

[16] 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. 1

[17] 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. 1, 3, 7

[18] 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. 1, 2, 6

[19] 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. 5

[20] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015. 3

[21] 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. 1, 2, 6, 7

[22] 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. 1, 3, 6, 7

[23] 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. 3

[24] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[25] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017. 5

[26] 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. 1

[27] 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. 4

[28] 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. 1, 6, 7

[29] 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. 2