Masked Spatial Propagation Network for Sparsity-Adaptive Depth Refinement

Jinyoung Jun
Korea University
jyjun@mcl.korea.ac.kr

Jae-Han Lee
Gauss Labs Inc.
jaehanlee@mcl.korea.ac.kr

Chang-Su Kim*
Korea University
changsukim@korea.ac.kr

Abstract

The main function of depth completion is to compensate for an insufficient and unpredictable number of sparse depth measurements of hardware sensors. However, existing research on depth completion assumes that the sparsity — the number of points or LiDAR lines — is fixed for training and testing. Hence, the completion performance drops severely when the number of sparse depths changes significantly. To address this issue, we propose the sparsity-adaptive depth refinement (SDR) framework, which refines monocular depth estimates using sparse depth points. For SDR, we propose the masked spatial propagation network (MSPN) to perform SDR with a varying number of sparse depths effectively by gradually propagating sparse depth information throughout the entire depth map. Experimental results demonstrate that MPSN achieves state-of-the-art performance on both SDR and conventional depth completion scenarios. Codes are available at https://github.com/jyjunmcl/MSPN_SDR

1. Introduction

Image-guided depth completion is a task to estimate a dense depth map using an RGB image with sparse depth measurements; it fills in unmeasured regions with estimated depths. It is useful because many depth sensors, *e.g.* LiDAR and ToF cameras, provide sparse depth maps only. With the recent usage of depth information in autonomous driving [22] and various 3D applications [34, 45, 61, 75], depth completion has become an important research topic.

Recently, with the success of deep neural networks, learning-based methods have shown significant performance improvement by exploiting massive amounts of training data [51]. They attempt to fuse multi-modal features like surface normal [56, 72] or provide repetitive image guidance [74]. Especially, affinity-based spatial propagation methods have been widely studied [9, 11, 44, 54, 79].

The main function of depth completion is to compensate for the limitations of existing depth sensors, but conventional research on depth completion assumes that the sparsity is fixed for training and testing. However, in practice, sparsity changes significantly, for it is difficult to measure the depths of transparent regions, mirrors, and black objects. Sensor defects also affect the number of measurements. Besides, conventional spatial propagation methods [9, 11, 44, 54, 79] refine the depths of all pixels simultaneously, regardless of the locations of sparse depth measurements. Therefore, erroneous depths can propagate during the refinement when only a few sparse depths are available.

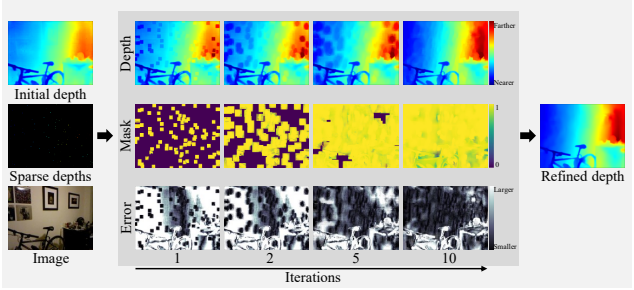

Figure 1. Illustration of the masked spatial propagation process in the proposed MSPN. The initial mask is obtained from sparse depths, assigned 1 if depth values are present and 0 otherwise. For easier comparison, error maps are also provided for each depth map, in which brighter pixels correspond to larger errors. MSPN updates depth maps and masks gradually to generate the final refined depth map.

In this paper, we develop a sparsity-adaptive depth refinement (SDR) framework, which refines monocular dense depth estimates adaptively according to the sparsity of depth measurements. Also, we propose the masked spatial propagation network (MSPN) to propagate the information from sparse depth points to unmeasured regions. First, an off-the-shelf monocular depth estimator is used to estimate an initial depth map from an input RGB image. Next, a guidance network generates guidance features using the input image, the sparse depths, and the initial depth map. Finally, using the guidance features, the proposed MSPN performs iterative refinement to obtain a refined depth map, as shown in Figure 1. The proposed SDR framework can

*Corresponding author.

be trained with a variable number of sparse depths, making it more suitable for real-world applications. In addition, the proposed MSPN performs significantly better than conventional methods in the SDR scenario by generating an adaptive propagation mask according to sparse measurements. Moreover, MSPN provides state-of-the-art performance on the conventional depth completion on NYUv2 [62] and KITTI [22] datasets.

This paper has the following contributions:

- We develop the SDR framework, which refines monocular depth estimates using a variable number of sparse depth measurements.
- For SDR, we propose MSPN to handle a various number of sparse depths by gradually propagating sparse depth information.
- MSPN provides state-of-the-art performance on both SDR and conventional depth completion scenarios.

2. Related Work

2.1. Monocular depth estimation

The goal of monocular depth estimation is to infer the absolute distance of each pixel in a single image from the camera. Traditional approaches are based on assumptions about the 3D scene structure — superpixels [60], block world [25], or line segments and vanishing points [26]. However, such assumptions are invalid for small objects or due to color ambiguity, making monocular depth estimation an ill-posed problem. With the advance of CNNs, training-based methods have successfully overcome the ill-posedness of monocular depth estimation. Earlier CNN methods focused on exploring better architectures [7, 17, 28, 38, 70] or loss functions [6, 16, 29, 38, 39] for more effective training. Recently, as transformer and self-attention [65] have been applied to diverse vision tasks [15], transformers for monocular depth estimation have also been developed [78].

On the other hand, training strategies to utilize 3D information have been attempted. For example, merging depth maps in frequency domain [40], enforcing high-order geometric constraints by defining virtual normal [76], and computing depth attention volume which considers pixel-to-image attention [31], have been attempted. Also, ordinal regression [21], or planar coefficient estimation [55] have been attempted.

2.2. Image-guided depth completion

Whereas monocular depth estimation uses only a single image as input, image-guided depth completion focuses on filling depths of unmeasured regions in an image when sparse depth measurements are provided. Significant progress has been made after the emergence of learning-based methods, overcoming the modality gap between image and depth. Early methods on depth completion concatenate images and sparse depth maps and process them using an encoder-decoder network [51, 52]. Also, attempts have been made to extract image and depth features separately and then leverage them to compensate for the modality gap effectively. For example, multi-branch architecture [30, 53, 58, 74] and coarse-to-fine framework [47, 73] have been studied. More specifically, in the coarse-to-fine framework, variants of affinity-based spatial propagation [48] are dominant [9–11, 44, 54, 73], since it can be easily applied using various encoder-decoder networks [30, 53, 79].

2.3. Spatial propagation network

Motivated by anisotropic diffusion [68], spatial propagation networks (SPNs) refine initial depth estimates using sparse depth measurements [9–11, 44, 54, 73]. They can reduce blurring caused by the encoder-decoder structure with varying spatial resolutions. To reconstruct the depth of a pixel based on spatial propagation, reference pixels, and their referring methods should be determined. The original SPN [48] uses three adjacent pixels in a neighboring row or column and performs refinement in four directions: up, down, left, and right. Convolution is also used to use neighboring pixels [10, 11]. Moreover, to refer to distant pixels, deformable convolution [13] is employed in [44, 54, 73].

While these SPN methods show impressive performance, they do not take the sparsity of available depths into account, which is essential in practical applications. For example, a network trained with 500 sparse depth points is less effective when the number of sparse depth points changes significantly since the initial depth map is estimated based on input sparse depths. In contrast, in this paper, we define a propagation mask, which is updated together with a depth map during the refinement. An initial depth map is first updated using only sparse depth points, and then a wider area is refined based on the propagation mask. As a result, the proposed MSPN can refine monocular depths effectively using a variable number of sparse depths.

2.4. Transformer and self-attention

After the success of transformers [14, 65] in natural language processing, extensive attempts have been made to apply transformers to vision tasks as well. Self-attention is a key component in transformers, which highlights and extracts important features from input. Recently, many transformers have demonstrated even better performances than CNNs in vision tasks [4, 15, 20, 27, 49, 57, 67]. Moreover, there have been researches for cross-attention between different types of features [3, 8, 35], which can combine different modalities. In this paper, we develop an attention-based refinement algorithm for depth completion. Whereas the conventional spatial propagation performs refinement via convolutions, the proposed MSPN refines the depth map using reliable pixels based on the masked attention strategy.

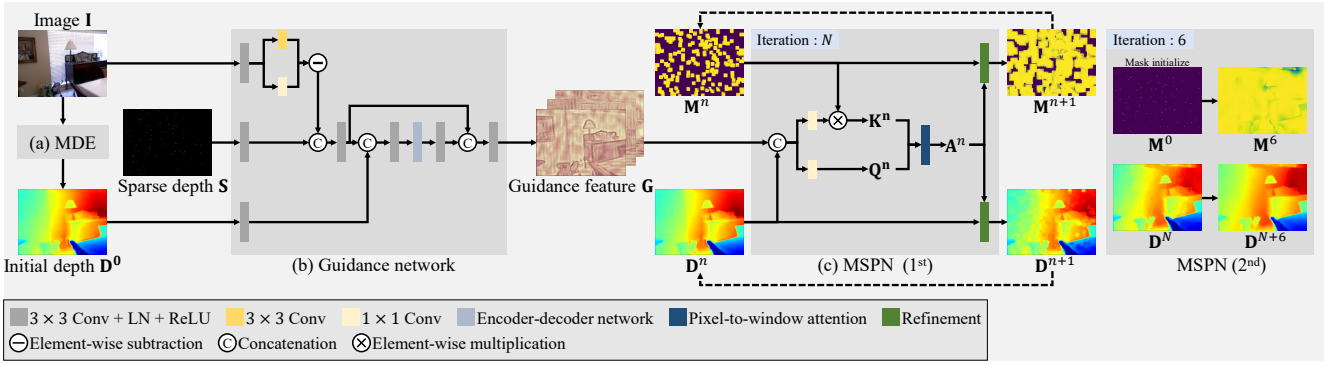

Figure 2. An overview of the proposed SDR framework.

3. Proposed Algorithm

3.1. Conventional depth completion

Let $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$ be an image and $\mathbf{S} \in \mathbb{R}^{H\times W}$ record its sparse depth measurements. The goal of depth completion is to estimate a dense depth map $\mathbf{D} \in \mathbb{R}^{H\times W}$ using $\mathbf{I}$ and $\mathbf{S}$. However, it is challenging due to the different modalities between color and depth. Moreover, while $\mathbf{S}$ provides the accurate depths for measured pixels, it does not provide any information on unmeasured regions. Therefore, previous methods [9, 11, 44, 54, 79] generate a guidance feature $\mathbf{G} \in \mathbb{R}^{C\times H\times W}$ that compensates for the modality gap, as well as an initial depth prediction $\mathbf{D}^0$;

$$(\mathbf{G}, \mathbf{D}^0) = \theta(\mathbf{I}, \mathbf{S}). \tag{1}$$

where θ denotes a multi-head network that estimates both $\mathbf{D}^0$ and $\mathbf{G}$. Then, $\mathbf{D}^0$ is refined N times to obtain the final depth map $\mathbf{D}^N$

$$\mathbf{D}^{n+1} = \phi(\mathbf{D}^n, \mathbf{G}), \quad n = 0, \dots, N-1, \tag{2}$$

where ϕ is a spatial propagation network.

3.2. Sparsity-adaptive depth refinement

The conventional depth completion methods [9, 11, 44, 54, 79] are trained with a fixed number of sparse depths. In practice, however, external factors often change the number of sparse depths, making the conventional methods less effective. In other words, the number of available depths in $\mathbf{S}$ in (1) is variant.

To address this problem, we propose the SDR framework, which consists of three networks: (a) a monocular depth estimator (MDE) that estimates an initial depth map, (b) a guidance network that combines different modality features, and (c) MSPN that recursively updates depth maps and masks. Figure 2 illustrates an overview of the proposed SDR. Note that we may use any off-the-shelf MDE. In this paper, for MDE, we use the pre-trained parameters of the Jun *et al.*'s algorithm [37] to generate $\mathbf{D}^0$ and exclude $\mathbf{S}$ from the generation process. Instead, we use $\mathbf{D}^0$ to generate $\mathbf{G}$ as follows:

$$\mathbf{G} = \Theta(\mathbf{I}, \mathbf{S}, \mathbf{D}^0) \tag{3}$$

where Θ denotes the proposed guidance network.

Next, we refine $\mathbf{D}^0$ using $\mathbf{G}$, based on the sparse depth locations in $\mathbf{S}$. The spatial propagation methods in [11, 54] refine all pixels simultaneously. Thus, erroneous depth values can propagate to neighbors, especially when only a few sparse depths are available. Instead of refining all pixels in a depth map at once, we define a propagation mask and gradually update the depth map from the sparse depth points. We perform the recursive refinement on both $\mathbf{D}^n$ and $\mathbf{M}^n$,

$$(\mathbf{D}^{n+1}, \mathbf{M}^{n+1}) = \Phi(\mathbf{D}^n, \mathbf{M}^n, \mathbf{G}), \; n = 0, \dots, N-1, \tag{4}$$

where Φ denotes the proposed MSPN. The initial propagation mask $\mathbf{M}^0 \in \mathbb{R}^{H\times W}$ is defined as

$$\mathbf{M}^0 = \Xi(\mathbf{S}) \tag{5}$$

where Ξ denotes an indicator function that outputs 1 for each sparse depth point and 0 otherwise.

3.3. Guidance network

As shown in Figure 2(b), the guidance network takes $\mathbf{I}$, $\mathbf{S}$ and $\mathbf{D}^0$ and generates the guidance feature $\mathbf{G}$. To implement the guidance network, we adopt the encoder-decoder architecture in U-Net [59]. The encoder extracts lower-resolution features from input, and the decoder processes the features to yield $\mathbf{G}$.

In a depth map, pixels on a large object or background have similar depths and vary gradually. Such depths are low-frequency components. In contrast, depths on small objects or edge regions are high-frequency components. Since sparse depths do not cover an entire image, extracting high-frequency information from the image is crucial for efficient

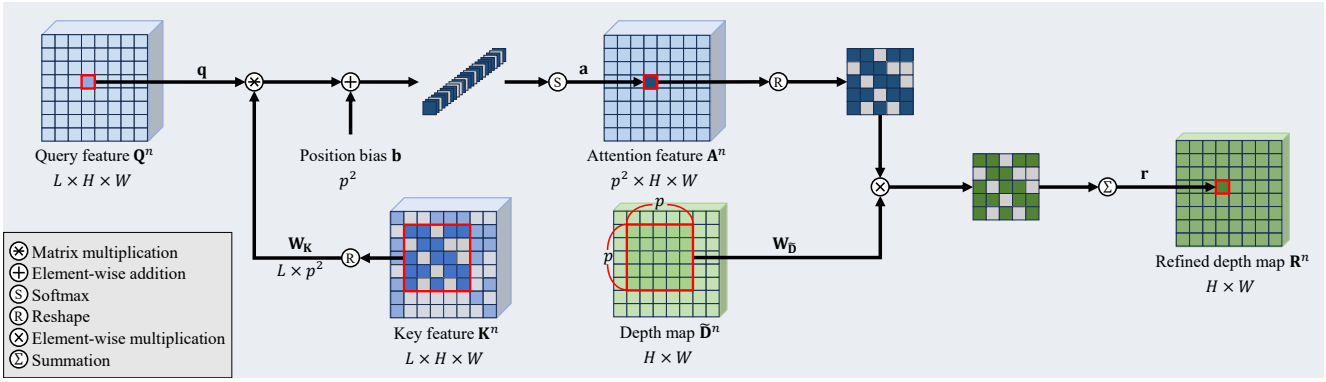

Figure 3. Illustration of the pixel-to-window attention process and generation process of $\mathbf{R}^n$. $\mathbf{M}^{n+1}$ is generated in the same manner.

refinement. Hence, we extract high-frequency features by subtracting the results of 1×1 convolution from those of 3×3 convolution, as in [71]. We extract the high-frequency features from $\mathbf{I}$, merge them with the features from $\mathbf{S}$ and $\mathbf{D}^0$, and feed the result to the encoder. The decoder has five blocks, each consisting of 3×3 transpose convolution, layer normalization [1], ReLU, and a NAF block [5]. More detailed network architecture is provided in the supplemental document.

3.4. Masked spatial propagation network

Using the output $\mathbf{G}$ of the guidance network, MSPN updates a depth map $\mathbf{D}^n$ and a mask $\mathbf{M}^n$ to $\mathbf{D}^{n+1}$ and $\mathbf{M}^{n+1}$, as shown in Figure 2(c). Each pixel in $\mathbf{D}^n$ is refined using its neighboring pixels, while the sparse depth values in $\mathbf{S}$ remain unchanged. Specifically, we replace the depth values in $\mathbf{D}^n$ using $\mathbf{S}$ and generate $\tilde{\mathbf{D}}^n$ by

$$\tilde{\mathbf{D}}^n = (1 - \mathbf{M}^0) \otimes \mathbf{D}^n + \mathbf{M}^0 \otimes \mathbf{S} \tag{6}$$

where $\otimes$ denotes the element-wise multiplication.

Next, we determine reference pixels for the refinement and the strength of the refinement. The conventional spatial propagation methods [9, 11, 44, 54] focus on the selection of reference pixels. However, there are much fewer reliable pixels than unreliable ones, so these methods are less effective when only a small number of sparse depths are provided. Instead, we design the masked-attention-based dynamic filter, which computes attention scores between each pixel and its surrounding pixels. We first generate the query feature $\mathbf{Q} \in \mathbb{R}^{L \times H \times W}$ and the key feature $\mathbf{K} \in \mathbb{R}^{L \times H \times W}$,

$$\mathbf{Q}^n = f_\mathbf{Q}([\tilde{\mathbf{D}}^n, \mathbf{G}]), \quad \mathbf{K}^n = f_\mathbf{K}([\tilde{\mathbf{D}}^n, \mathbf{G}]) \otimes \mathbf{M}^n \tag{7}$$

where $f_\mathbf{Q}$ and $f_\mathbf{K}$ are 1×1 convolutions followed by layer normalization [1]. Also, $[\cdot]$ denotes the channel-wise concatenation. Since $\tilde{\mathbf{D}}^n$ is unrefined, $\mathbf{K}^n$ is a mixture of reliable and unreliable pixel features. Therefore, we mask unreliable pixel features when computing $\mathbf{K}^n$ in (7).

Next, we compute the attention scores between $\mathbf{Q}^n$ and $\mathbf{K}^n$. Let $\mathbf{q} \in \mathbb{R}^L$ be a query pixel feature in $\mathbf{Q}^n$ at location (i, j). Also, let $\mathbf{W}_\mathbf{K} \in \mathbb{R}^{L \times p^2}$ denote the $p \times p$ window of the key features in $\mathbf{K}^n$ centered at (i, j). Note that we compute pixel-to-window attention in order to refine pixel (i, j) using its neighboring pixels. More specifically, the pixel-to-window attention $\mathbf{a} \in \mathbb{R}^{p^2}$ is computed as

$$\mathbf{a} = \text{softmax}(\mathbf{q}^t \mathbf{W}_\mathbf{K} + \mathbf{b}) \tag{8}$$

where $\mathbf{b} \in \mathbb{R}^{p^2}$ denotes the relative position bias [27] in the $w \times w$ window. By performing the attention on all pixels in $\mathbf{Q}^n$, the attention feature $\mathbf{A}^n \in \mathbb{R}^{p^2 \times H \times W}$ is obtained.

Then, using $\mathbf{A}^n$ and $\tilde{\mathbf{D}}^n$, we generate a refined depth map $\mathbf{R}^n \in \mathbb{R}^{H \times W}$. Let $\mathbf{W}_{\tilde{\mathbf{D}}}$ and $\mathbf{W}_\mathbf{M}$ be the $p \times p$ windows in $\tilde{\mathbf{D}}^n$ and $\mathbf{M}^n$, centered at (i, j), respectively. A refined depth pixel $\mathbf{r}$ in $\mathbf{R}^n$ is obtained by

$$\mathbf{r} = \sum_{t=1}^{p^2} \mathbf{a}_t \cdot \mathbf{W}_{\tilde{\mathbf{D}},t} \tag{9}$$

where the subscript t denotes the tth element in the window. Figure 3 illustrates the pixel-to-window attention process and the generation process of $\mathbf{R}^n$.

Finally, depth map $\mathbf{D}^{n+1}$ at the next iteration step is given by

$$\mathbf{D}^{n+1} = (1 - \mathbf{M}^n) \otimes \tilde{\mathbf{D}}^n + \mathbf{M}^n \otimes \mathbf{R}^n. \tag{10}$$

Also, similarly to (9), mask pixel $\mathbf{m}^{n+1}$ at the next iteration step is computed by

$$\mathbf{m}^{n+1} = \sum_{t=1}^{p^2} \mathbf{a}_t \cdot \mathbf{W}_{\mathbf{M},t}, \tag{11}$$

yielding the updated mask map $\mathbf{M}^{n+1}$.

3.5. Refinement strategy

Note that $\mathbf{M}^0$ is defined using $\mathbf{S}$ and gets wider as iteration goes on. We determine the number of refinement iterations adaptively according to the sparsity of $\mathbf{S}$, *i.e.* the number of sparse depth measurements. Suppose that s depth points are valid and uniformly distributed in $\mathbf{S}$. Then, the distance ν_s between adjacent sparse depth points is calculated as

$$\nu_s = \sqrt{\frac{HW}{s}} - 1. \tag{12}$$

In addition, within a $p \times p$ window, each pixel can refine pixels within the distance of $p/2 - 1$. Therefore, to refine all pixels in an image, we set the number N of iterations as

$$N = \frac{\nu_s}{p/2 - 1}. \tag{13}$$

Since the computation in (12) is for an ideal case, N in (13) is a lower limit for the mask to spread throughout the entire image. We, therefore, multiply N by a pre-defined hyperparameter κ for margin. We also set the minimum number of iterations to six to perform sufficient refinement even when a large number of sparse depths are given.

In MSPN, sparse depth points and refined pixels are treated similarly after iterations. This is less effective when a large number of sparse depths are provided. Hence, we use two MSPN layers and initialize the mask to $\mathbf{M}^0$ as shown in Figure 2. The second MSPN refines the depth map by reusing the accurate $\mathbf{S}$ with the mask initialization. This helps to reduce the adverse impacts of monocular depth errors around the sparse depths. Empirically, we fix its iteration number to six. To summarize, two MSPN layers are configured in series and perform refinement for N and six iterations, respectively.

3.6. Loss functions

For NYUv2 [62], we use both L1 and L2 losses for training. Let d_i and $\hat{d}_i$ denote the ith depths in the ground-truth depth map $\mathbf{D}$ and a predicted depth map $\hat{\mathbf{D}}$, respectively. Then, the loss function is defined as

$$\mathcal{L}(\hat{\mathbf{D}}, \mathbf{D}) = \frac{1}{|\mathbf{D}|} \sum_{\sigma=1}^{2} \sum_{i} |\hat{d}_i - d_i|^{\sigma} \tag{14}$$

where $|\mathbf{D}|$ denotes the number of valid pixels in $\mathbf{D}$.

For KITTI [22], we use the scale-invariant logarithmic loss function, which is widely used in monocular depth estimation [2, 78]. It is defined as

$$\mathcal{L}_{SI}(\hat{\mathbf{D}}, \mathbf{D}) = \alpha \sqrt{\frac{1}{|\mathbf{D}|} \sum_{i} {e_i}^2 - \frac{\lambda}{|\mathbf{D}|^2} (\sum_{i} e_i)^2} \tag{15}$$

where $e_i = \log \hat{d}_i - \log d_i$. As in [2, 78], we set $\alpha = 10$ and $\lambda = 0.85$.

4. Experiments

4.1. Datasets

NYUv2 [62]: It contains 464 indoor scenes captured by Kinect v1 and provides 120K and 654 images for training and testing, respectively. As done in [51, 54, 79], we use the uniformly sampled 50K images for training. Also, as in [62], we fill in missing depths using the colorization scheme [42]. The original images of size 640×480 are bilinearly downsampled by a factor of $\frac{1}{2}$ and then center-cropped to 304×228. Sparse depths are randomly sampled from the ground truth depth map.

KITTI Depth Completion (DC) [22]: It contains outdoor scenes captured by HDL-64E. Since the depth maps obtained by HDL-64E are sparse, the measurements are used as input, and the ground truth is generated by combining the information in 11 consecutive frames. In this work, we use the 10K subset in [79] for training and do the test on a 1K validation set. As in [54, 79], the original images are bottom-center-cropped to 240×1216.

4.2. Evaluation metrics

We adopt the top three metrics in Table 1 for NYUv2 and the bottom four for KITTI. For RMSE, NYUv2 uses meters, and KITTI uses millimeters.

Table 1. Evaluation metrics for estimated depth maps. Here, $|\mathbf{D}|$ denotes the number of valid pixels in a depth map $\mathbf{D}$, d_i is the ith valid depth in $\mathbf{D}$, and $\hat{d}_i$ is an estimate of d_i.

REL	$\frac{1}{\lvert\mathbf{D}\rvert} \sum_i \lvert\hat{d}_i - d_i\rvert / d_i$
δ_k	% of d_i that satisfies $\max\left\{\frac{\hat{d}_i}{d_i}, \frac{d_i}{\hat{d}_i}\right\} < k$
RMSE (m, mm)	$\frac{1}{\lvert\mathbf{D}\rvert} \left(\sum_i (\hat{d}_i - d_i)^2\right)^{0.5}$
iRMSE (1/km)	$\frac{1}{\lvert\mathbf{D}\rvert} \left(\sum_i (\frac{1}{\hat{d}_i} - \frac{1}{d_i})^2\right)^{0.5}$
MAE (mm)	$\frac{1}{\lvert\mathbf{D}\rvert} \sum_i \lvert\hat{d}_i - d_i\rvert$
iMAE (1/km)	$\frac{1}{\lvert\mathbf{D}\rvert} \sum_i \lvert\frac{1}{\hat{d}_i} - \frac{1}{d_i}\rvert$

4.3. Implementation details

Network architecture: In Figure 2 (b), we adopt PVT-Base [67] as the encoder of the guidance network. The encoder takes the input of spatial resolution $H \times W$ and yields the output of spatial resolution $H/16 \times W/16$ with 512 channels. The decoder consists of five blocks; each upsamples its input feature using 3×3 transposed convolution, followed by layer normalization, ReLU activation, and a NAF block [5]. To the last four decoder blocks, the encoder features are skip-connected via concatenation. The guidance network outputs $\mathbf{G}$ of resolution $H \times W$ with 64 channels.

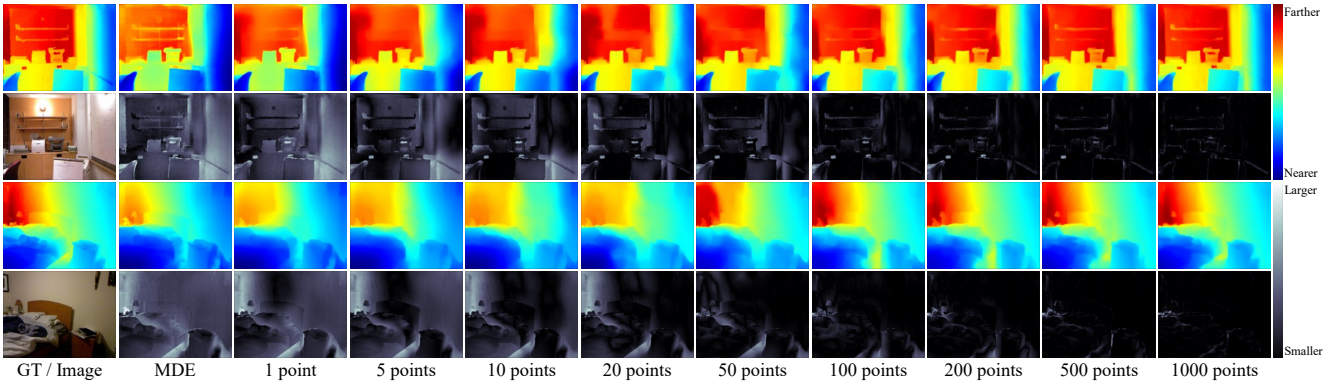

Figure 4. SDR results on NYUv2. For each depth map, the corresponding error map is provided below, in which brighter pixels represent larger errors.

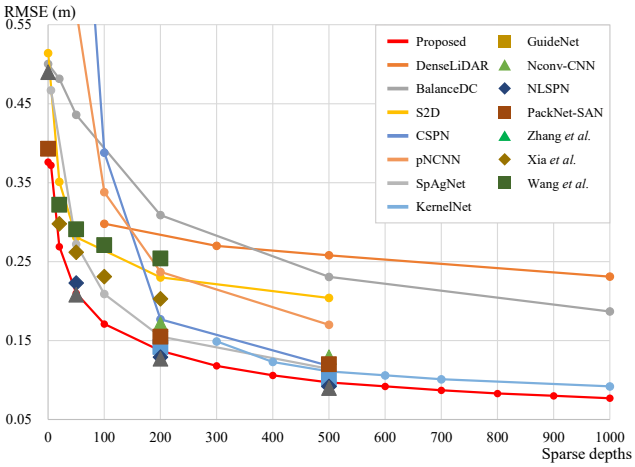

Figure 5. Comparison of the SDR performances on NYUv2.

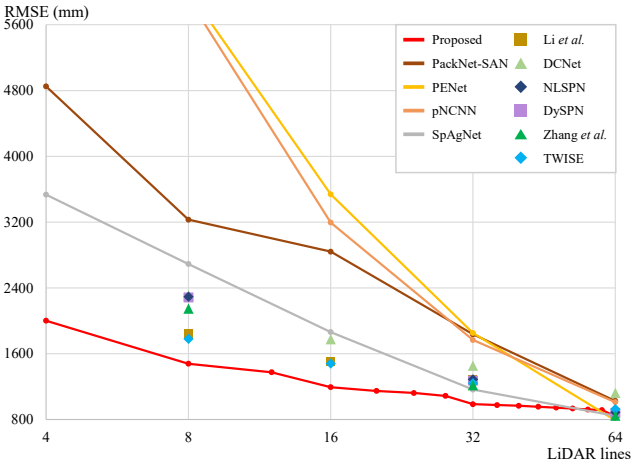

Figure 6. Comparison of the SDR performances on the KITTI validation set.

The detailed network architecture is provided in the supplemental document.

Training: We train models with the AdamW optimizer [50] with an initial learning rate of 10^{-3}, weight decay of 10^{-2}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The batch size per GPU is set to 24 and 16 on NYUv2 and KITTI, respectively, using gradient accumulation. We use a 13×13 window for MSPN. For NYUv2, the model is trained for 36 epochs, and the learning rate is halved after 18, 24, and 30 epochs. For KITTI, the model is trained for 60 epochs, and the learning rate is halved after 30, 36, 42, 48, and 54 epochs. The number of iterations N in (13) is weighted by $\kappa = 2$.

For NYUv2, the number of sparse depth points is uniformly sampled between 10 and 1000. In the case of KITTI [22], each input lidar scan has 64 lines. To simulate a real-world case and to compare with other methods, we sample lines instead of sampling each sparse depth point randomly. The number of sampled lines is set evenly between 4 and 64. To determine the number of iterations, we compute the average number of sparse depth pixels per line and use it to calculate N in (13).

4.4. Sparsity-adaptive depth refinement

We evaluate the SDR performances of the proposed MSPN with other depth completion algorithms [12, 18, 19, 23, 24, 30, 32, 33, 43, 44, 46, 51, 54, 64, 66, 69, 77, 79]. Figures 5 and 6 compare the RMSE performances according to the number of sparse depths on NYUv2 and KITTI, respectively. In Figures 5 and 6, a solid line indicates that a single model is evaluated for various numbers of sparse depths. On the contrary, each symbol means that a separate model is trained and evaluated for a fixed number of sparse depths. The following observations can be made from Figures 5 and 6:

- By comparing the solid lines in Figure 5, we see that the proposed MSPN outperforms all the other methods for all numbers of sparse depths on NYUv2.
- Specifically, some methods are specialized for many sparse depths, and their performances degrade significantly with fewer sparse depths. Conversely, some are specialized for few sparse depths, and their performances

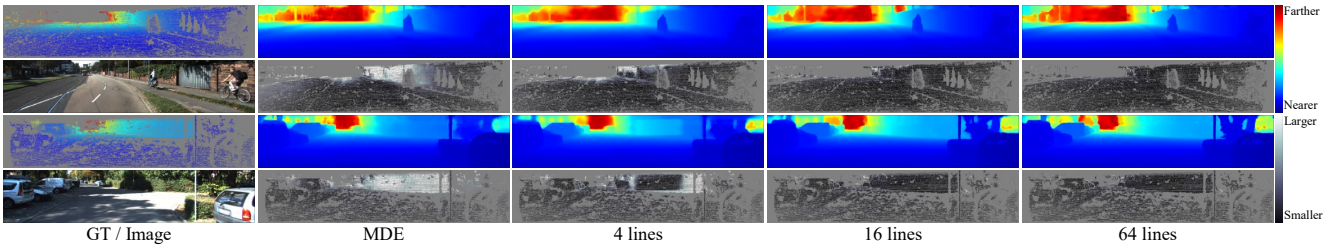

Figure 7. SDR results on KITTI. For each depth map, the corresponding error map is provided below, in which brighter pixels represent larger errors.

improve marginally with more sparse depths.

- On the other hand, the proposed MSPN exhibits similar performances to those symboled methods [54, 79], trained for specific numbers of sparse depths. This indicates that MSPN yields robust results regardless of the number of sparse depths.
- In Figure 6, MSPN significantly outperforms the other methods on KITTI when less than 64 lines are available.
- For KITTI, methods specialized for a certain number of LiDAR lines do not perform well with a small number of lines. On the contrary, MSPN utilizes monocular depth estimation results to perform depth completion effectively, regardless of the number of lines.
- Overall, MSPN yields more reliable depth maps for varying numbers of sparse depths than the conventional algorithms on both indoor and outdoor images. This indicates that MSPN is more suitable for real-world applications.

Figures 4 and 7 show SDR results with varying numbers of sparse depths. We see that depth maps can be improved with just a single sparse depth, and errors are reduced when more sparse depths become available.

More results, including exact metric scores and additional qualitative results, are in the supplemental document.

4.5. Ordinary depth completion

Although the primary focus of MSPN is SDR, we also evaluate the performance of MSPN under an ordinary depth completion scenario. For this ordinary depth completion, we add another decoder head to the guidance network in order to predict an initial depth map and do not use the monocular depth estimator, as in previous works [44, 54, 79]. The detailed network architecture for ordinary depth completion is in the supplemental document.

We train and test our model using a fixed amount of sparse depths. For NYUv2, we use randomly sampled 500 sparse depths from the ground truth and train the network for 72 epochs. For KITTI, we train models specialized for 16 and 64 LiDAR lines, respectively, for 72 epochs. For a fair comparison on KITTI, we use the 10k subset for training provided by [79].

Tables 2 and 3 compare the performances on NYUv2 and KITTI, respectively. We see that the proposed MSPN provides state-of-the-art performances on ordinary depth completion as well. Figure 8 qualitatively compares the results with [51, 54, 79] on NYUv2. We see that MSPN fills in challenging regions with fine details more effectively.

Table 2. Comparison of ordinary depth completion results on NYUv2. In each test, the best result is **boldfaced**.

Method	RMSE(↓)	REL(↓)	$\delta_{1.25}$(↑)
S2D [51]	0.204	0.043	97.8
GuideNet [64]	0.101	0.015	99.5
PackNet-SAN [24]	0.120	0.019	99.4
TWISE [33]	0.097	0.013	**99.6**
NLSPN [54]	0.092	**0.012**	**99.6**
RigNet [74]	0.090	0.013	**99.6**
DySPN [44]	0.090	**0.012**	**99.6**
Zhang *et al.* [79]	0.090	**0.012**	-
Proposed	**0.089**	**0.012**	**99.6**

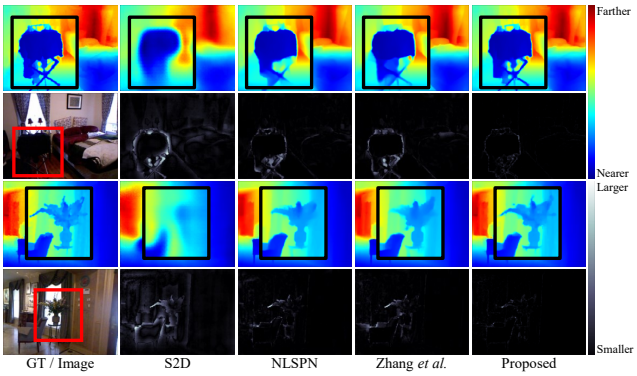

Figure 8. Qualitative comparison of ordinary depth completion results on NYUv2.

4.6. Analysis

MSPN: Figure 9 shows the mask update process of MSPN using 16 LiDAR lines on KITTI. We see that the mask values at the second MSPN layer are less widened than those of the first MSPN layer. This indicates that the two MSPN layers play different roles — The first layer roughly refines an entire image, and the second layer intensively refines near sparse depths. Figure 10 (a) shows the RMSE performances of each MSPN layer on NYUv2. Since the role of the second layer is to refine nearby regions of sparse depths, it has

Table 3. Comparison of ordinary depth completion results on the KITTI validation set. In each test, the best result is **boldfaced**.

Lines	Method	RMSE(↓)	MAE(↓)	iRMSE(↓)	iMAE(↓)
16	NLSPN [54]	1288.9	377.2	3.4	1.4
	DySPN [44]	1274.8	366.4	3.2	1.3
	Zhang *et al*. [79]	1218.6	**337.4**	3.0	**1.2**
	Proposed	**1212.7**	341.8	**2.6**	**1.2**
64	NLSPN [54]	889.4	238.8	2.6	1.0
	DySPN [44]	878.5	228.6	2.5	1.0
	Zhang *et al*. [79]	848.7	**215.9**	2.5	**0.9**
	Proposed	**835.7**	218.5	**2.1**	**0.9**

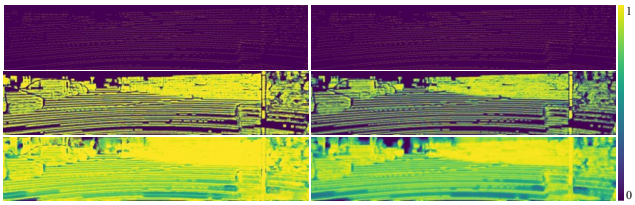

Figure 9. Illustration of the mask update process of MSPN on KITTI. The three rows represent the input, the first masks, and the last masks, respectively. The two columns indicate the masks of the first and second MSPN layers, respectively.

a higher gain as more sparse depths are provided.

Generalization: We assess the generalization capability of the proposed SDR. In this test, the guidance network and MSPN, trained for Jun *et al*. [37], are not fine-tuned. In Figure 10 (b), we use other off-the-shelf networks, [7, 21, 36, 41, 78] for monocular depth estimator and evaluate the SDR performances. We observe similar trends for various MDEs, which indicates that the proposed SDR framework provides robust results without being sensitive to the adopted monocular depth estimators. Figure 10 (c) compares the cross-dataset evaluation performance of SDR on RealSense split of SUN RGB-D [63] with [54] and [79]. We observe that the proposed SDR provides robust performance on unseen cameras. Qualitative results are provided in the supplemental document.

Ablation study: To assess the impact of the mask update process in MSPN, we remove the mask and its update process in MSPN. Figure 10 (d) compares the SDR results of this ablated setting on NYUv2. We see that the ablated setting severely degrades the SDR results, especially with a small number of sparse depths. This indicates that the mask update plays a crucial role in MSPN.

Depth hole filling: We evaluate the depth filling performances of MSPN for large areas with no sparse depths. To this end, we mask the center 114×152 region of a 228×304 ground-truth depth map and train the network to restore the masked region. We compare the results of the proposed MSPN with Zhang *et al*. [79] in Table 4 and Figure 11. Note that MSPN recovers missing regions more faithfully.

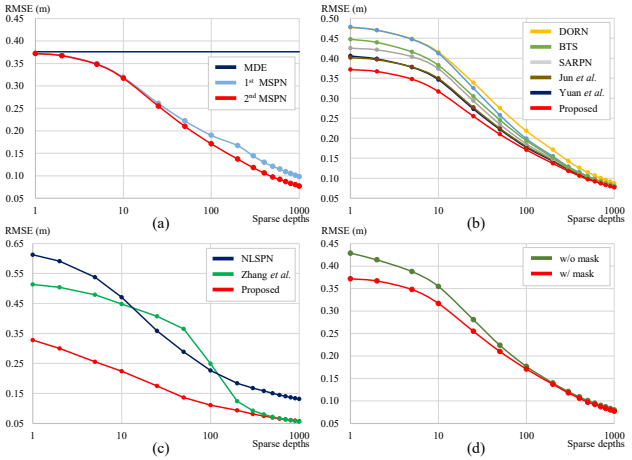

Figure 10. (a) SDR results of each MSPN layer. (b) SDR results using different monocular depth estimators. (c) Cross-dataset evaluation performance on SUN RGB-D. (d) Ablation study of the mask update process in MSPN.

Table 4. Comparison of depth hole filling results on NYUv2. The performance is evaluated only on masked areas.

Method	RMSE(↓)	REL(↓)	$\delta_{1.25}$(↑)	$\delta_{1.25^2}$(↑)	$\delta_{1.25^3}$(↑)
Zhang *et al*. [79]	0.348	0.073	0.925	0.982	0.995
MSPN	**0.325**	**0.064**	**0.936**	**0.986**	**0.996**

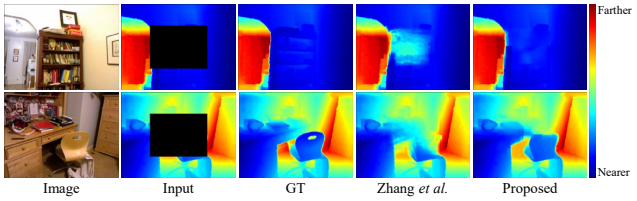

Figure 11. Qualitative comparison of depth hole filling results.

5. Conclusions

In this paper, we proposed the sparsity-adaptive depth refinement (SDR) framework for real-world depth completion. First, a monocular depth estimator generates an initial depth map. Then, the guidance network generates guidance features using the initial depth map, the input image, and sparse depth measurements. Finally, the proposed MSPN gradually refines the initial depth map using the guidance features by propagating sparse depth points to the entire depth map. Extensive experiments demonstrated that the proposed MSPN provides excellent results on both SDR and conventional depth completion scenario.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (No. NRF-2021R1A4A1031864 and NRF-2022R1A2B5B03002310).

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. In *NeurIPS*, 2016. 4

[2] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. AdaBins: Depth estimation using adaptive bins. In *CVPR*, 2021. 5

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2

[4] Chun-Fu Chen, Rameswar Panda, and Quanfu Fan. RegionViT: Regional-to-local attention for vision transformers. In *ICLR*, 2022. 2

[5] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *ECCV*, 2022. 4, 5

[6] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. In *NeurIPS*, 2016. 2

[7] Xiaotian Chen, Xuejin Chen, and Zheng-Jun Zha. Structure-aware residual pyramid network for monocular depth estimation. In *IJCAI*, 2019. 2, 8

[8] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021. 2

[9] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *ECCV*, 2018. 1, 2, 3, 4

[10] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2361–2379, 2019. 2

[11] Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. CSPN++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In *AAAI*, 2020. 1, 2, 3, 4

[12] Andrea Conti, Matteo Poggi, and Stefano Mattoccia. Sparsity agnostic depth completion. In *WACV*, 2023. 6

[13] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 2

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2018. 2

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[16] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 2

[17] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014. 2

[18] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through cnns for guided sparse depth regression. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2423–2436, 2019. 6

[19] Abdelrahman Eldesokey, Michael Felsberg, Karl Holmquist, and Michael Persson. Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end. In *CVPR*, 2020. 6

[20] Jiemin Fang, Lingxi Xie, Xinggang Wang, Xiaopeng Zhang, Wenyu Liu, and Qi Tian. MSG-Transformer: Exchanging local spatial information by manipulating messenger tokens. In *CVPR*, 2022. 2

[21] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018. 2, 8

[22] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012. 1, 2, 5, 6

[23] Jiaqi Gu, Zhiyu Xiang, Yuwen Ye, and Lingxuan Wang. Denselidar: A real-time pseudo dense depth guided depth completion network. *IEEE Robotics and Automation Letters*, 6(2):1808–1815, 2021. 6

[24] Vitor Guizilini, Rares Ambrus, Wolfram Burgard, and Adrien Gaidon. Sparse auxiliary networks for unified monocular depth prediction and completion. In *CVPR*, 2021. 6, 7

[25] Abhinav Gupta, Alexei A. Efros, and Martial Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, 2010. 2

[26] Abhinav Gupta, Martial Hebert, Takeo Kanade, and David Blei. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *NeurIPS*, 2010. 2

[27] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *CVPR*, 2023. 2, 4

[28] Minhyeok Heo, Jaehan Lee, Kyung-Rae Kim, Han-Ul Kim, and Chang-Su Kim. Monocular depth estimation using whole strip masking and reliability-based refinement. In *ECCV*, 2018. 2

[29] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *WACV*, 2019. 2

[30] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Penet: Towards precise and efficient image guided depth completion. In *ICRA*, 2021. 2, 6

[31] Lam Huynh, Phong Nguyen-Ha, Jiri Matas, Esa Rahtu, and Janne Heikkilä. Guiding monocular depth estimation using depth-attention volume. In *ECCV*, 2020. 2

[32] Saif Imran, Yunfei Long, Xiaoming Liu, and Daniel Morris. Depth coefficients for depth completion. In *CVPR*, 2019. 6

[33] Saif Imran, Xiaoming Liu, and Daniel Morris. Depth completion with twin surface extrapolation at occlusion boundaries. In *CVPR*, 2021. 6, 7

[34] Hamid Izadinia, Qi Shan, and Steven M. Seitz. IM2CAD. In *CVPR*, 2017. 1

[35] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver IO: A general architecture for structured inputs & outputs. In *ICLR*, 2022. 2

[36] Jinyoung Jun, Jae-Han Lee, Chul Lee, and Chang-Su Kim. Depth map decomposition for monocular depth estimation. In *ECCV*, 2022. 8

[37] Jinyoung Jun, Jae-Han Lee, and Chang-Su Kim. Versatile depth estimator based on common relative depth estimation and camera-specific relative-to-metric depth conversion. *arXiv preprint arXiv:2303.10991*, 2023. 3, 8

[38] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016. 2

[39] Jae-Han Lee and Chang-Su Kim. Multi-loss rebalancing algorithm for monocular depth estimation. In *ECCV*, 2020. 2

[40] Jae-Han Lee, Minhyeok Heo, Kyung-Rae Kim, and Chang-Su Kim. Single-image depth estimation based on Fourier domain analysis. In *CVPR*, 2018. 2

[41] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 8

[42] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. *ACM Trans. Graph.*, 23(3):689–694, 2004. 5

[43] Ang Li, Zejian Yuan, Yonggen Ling, Wanchao Chi, Chong Zhang, et al. A multi-scale guided cascade hourglass network for depth completion. In *WACV*, 2020. 6

[44] Yuankai Lin, Tao Cheng, Qi Zhong, Wending Zhou, and Hua Yang. Dynamic spatial propagation network for depth completion. In *AAAI*, 2022. 1, 2, 3, 4, 6, 7, 8

[45] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. PlaneNet: Piece-wise planar reconstruction from a single RGB image. In *CVPR*, 2018. 1

[46] Lina Liu, Yiyi Liao, Yue Wang, Andreas Geiger, and Yong Liu. Learning steering kernels for guided depth completion. *IEEE Transactions on Image Processing*, 30:2850–2861, 2021. 6

[47] Lina Liu, Xibin Song, Xiaoyang Lyu, Junwei Diao, Mengmeng Wang, Yong Liu, and Liangjun Zhang. Fcfr-net: Feature fusion based coarse-to-fine residual learning for depth completion. In *AAAI*, 2021. 2

[48] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. In *NeurIPS*, 2017. 2

[49] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2

[50] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[51] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *ICRA*, 2018. 1, 2, 5, 6, 7

[52] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *ICRA*, 2019. 2

[53] Danish Nazir, Alain Pagani, Marcus Liwicki, Didier Stricker, and Muhammad Zeshan Afzal. Semattnet: Toward attention-based semantic aware guided depth completion. *IEEE Access*, 10:120781–120791, 2022. 2

[54] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *ECCV*, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[55] Vaishakh Patil, Christos Sakaridis, Alexander Liniger, and Luc Van Gool. P3Depth: Monocular depth estimation with a piecewise planarity prior. In *CVPR*, 2022. 2

[56] 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 *CVPR*, 2019. 1

[57] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 2

[58] Kyeongha Rho, Jinsung Ha, and Youngjung Kim. Guideformer: Transformers for image guided depth completion. In *CVPR*, 2022. 2

[59] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 3

[60] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. Make3D: Learning 3D scene structure from a single still image. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5):824–840, 2008. 2

[61] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3D photography using context-aware layered depth inpainting. In *CVPR*, 2020. 1

[62] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, 2012. 2, 5

[63] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *CVPR*, 2015. 8

[64] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion. *IEEE Transactions on Image Processing*, 30:1116–1129, 2020. 6, 7

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[66] Tsun-Hsuan Wang, Fu-En Wang, Juan-Ting Lin, Yi-Hsuan Tsai, Wei-Chen Chiu, and Min Sun. Plug-and-play: Improve depth prediction via sparse data propagation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5880–5886. IEEE, 2019. 6

[67] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *CVPR*, 2021. 2, 5

[68] Joachim Weickert et al. *Anisotropic diffusion in image processing*. Teubner Stuttgart, 1998. 2

[69] Zhihao Xia, Patrick Sullivan, and Ayan Chakrabarti. Generating and exploiting probabilistic monocular depth estimates. In *CVPR*, 2020. 6

[70] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous CRFs as sequential deep networks for monocular depth estimation. In *CVPR*, 2017. 2

[71] Ke Xu, Xin Yang, Baocai Yin, and Rynson WH Lau. Learning to restore low-light images via decomposition-and-enhancement. In *CVPR*, 2020. 4

[72] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. Depth completion from sparse LiDAR data with depth-normal constraints. In *ICCV*, 2019. 1

[73] Zheyuan Xu, Hongche Yin, and Jian Yao. Deformable spatial propagation networks for depth completion. In *ICIP*, 2020. 2

[74] Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Jun Li, and Jian Yang. RigNet: Repetitive image guided network for depth completion. In *ECCV*, 2022. 1, 2, 7

[75] Mao Ye, Xianwang Wang, Ruigang Yang, Liu Ren, and Marc Pollefeys. Accurate 3D pose estimation from a single depth image. In *ICCV*, 2011. 1

[76] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *ICCV*, 2019. 2

[77] Sungho Yoon and Ayoung Kim. Balanced depth completion between dense depth inference and sparse range measurements via KISS-GP. In *IROS*, 2020. 6

[78] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. NewCRFs: Neural window fully-connected CRFs for monocular depth estimation. In *CVPR*, 2022. 2, 5, 8

[79] Youmin Zhang, Xianda Guo, Matteo Poggi, Zheng Zhu, Guan Huang, and Stefano Mattoccia. Completionformer: Depth completion with convolutions and vision transformers. In *CVPR*, 2023. 1, 2, 3, 5, 6, 7, 8