LRRU: Long-short Range Recurrent Updating Networks for Depth Completion

Yufei Wang[1], Bo Li[1*], Ge Zhang[1], Qi Liu[1], Tao Gao[2], Yuchao Dai[1*]

[1]Northwestern Polytechnical University and Shaanxi Key Laboratory of
Information Acquisition and Processing [2]Chang'an University

Abstract

Existing deep learning-based depth completion methods generally employ massive stacked layers to predict the dense depth map from sparse input data. Although such approaches greatly advance this task, their accompanied huge computational complexity hinders their practical applications. To accomplish depth completion more efficiently, we propose a novel lightweight deep network framework, the Long-short Range Recurrent Updating (LRRU) network. Without learning complex feature representations, LRRU first roughly fills the sparse input to obtain an initial dense depth map, and then iteratively updates it through learned spatially-variant kernels. Our iterative update process is content-adaptive and highly flexible, where the kernel weights are learned by jointly considering the guidance RGB images and the depth map to be updated, and large-to-small kernel scopes are dynamically adjusted to capture long-to-short range dependencies. Our initial depth map has coarse but complete scene depth information, which helps relieve the burden of directly regressing the dense depth from sparse ones, while our proposed method can effectively refine it to an accurate depth map with less learnable parameters and inference time. Experimental results demonstrate that our proposed LRRU variants achieve state-of-the-art performance across different parameter regimes. In particular, the LRRU-Base model outperforms competing approaches on the NYUv2 dataset, and ranks 1st on the KITTI depth completion benchmark at the time of submission. Project page: https://npucvr.github.io/LRRU/.

1. Introduction

Acquiring accurate and dense scene depth plays a fundamental role in various applications, such as autonomous driving [33] and augmented reality [25]. However, existing depth sensors have inevitable limitations for both indoor and outdoor scenes [39], for example, the depth acquired by

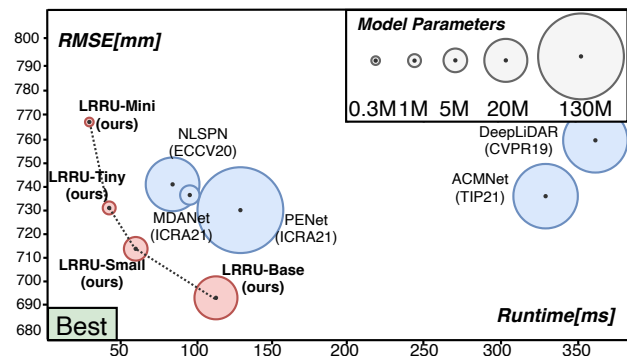*Corresponding authors {daiyuchao, libo}@nwpu.edu.cn.

Figure 1. The performance in terms of RMSE versus runtime on the KITTI test dataset. The bubble size represents the number of parameters. LRRU performs better while maintaining efficiency.

LiDAR is too sparse to be used directly. Thus, depth completion, *i.e.*, estimating the dense depth maps from sparse distance measurements, has attracted extensive research interests in industry and research communities.

Recently, deep learning-based methods [32, 35] have shown dominant performance for this task, which directly maps sparse depth maps to dense depth maps through massive stacked filters and layers. Since the RGB images contain rich semantic cues that are critical for filling unknown depth, some works [22, 23, 30, 42] utilize the RGB information to guide depth completion. Although many advanced networks such as ResNet [10] and Transformer [8, 29, 43] have been exploited, it is still difficult to directly predict an accurate and dense depth map from a sparse input depth map and the corresponding RGB image. Specifically, existing methods employ tens of millions of learnable parameters in exchange for a desirable model capacity to learn robust features, for example, 132M parameters are contained in PENet [11]. Such large-scale networks usually require heavy computing resources, which fail to be applied in the real world, while the method performance drops significantly if the network size is simply reduced [13]. In addition, the predicted depth maps obtained by direct regression suffer from blur effect and distortion of object boundaries, which need to be further refined through extra refinement modules [4]. For example, the popular spatial propagation

networks (SPNs) [3, 4, 11, 17, 21, 26, 41] update the output of the direct-regression methods by a recurrent operation. Thus, designing an efficient depth completion architecture that performs better while maintaining efficiency is essential for further research.

Inspired by the success of the recurrent design [18, 31, 34], we propose the Long-short Range Recurrent Updating (LRRU) network, a novel lightweight deep network framework for depth completion. Unlike existing direct-regression methods, LRRU iteratively updates an initial depth map obtained by a non-learning approach [15]. The initial depth map has coarse but complete scene depth information, which can help relieve the heavy burden of directly regressing the precise dense depth from sparse input depth map. Although existing SPNs [3, 4, 11, 17, 21, 26, 41] have shown that the depth map can be refined by learned spatially-variant kernels that model relevant neighbors and their affinities of each pixel, these methods cannot be directly used in our framework due to the following limitations: (a) content-agnostic update unit: the kernel parameters required for updating are predicted by the features from RGB and sparse depth, which are not adaptively adjusted to the target map (the depth map to be updated); (b) inflexible recurrent strategy: the kernel scope is fixed during the update process, and multiple iterations are required to obtain long-range dependencies and satisfactory results.

To address the above issues, we propose a Target-Dependent Update (TDU) unit and a long-short range recurrent strategy, which make our iterative update process *content-adaptive and highly flexible*. Our TDU predicts the sampling position of the neighbors and the weights (affinities) between them and the reference point by jointly considering the cross-guided and self-guided features. The cross-guided features extracted from RGB images and sparse depths can guide the TDU to avoid irrelevant neighbors, while the self-guided features extracted from the depth map to be updated allow the TDU to be adaptive to the content of the target map. Moreover, our TDU further improves the performance by learning the residual. In addition, we observe that when multiple TDUs of the update process employ the cross-guided features of different scales respectively, the TDU guided by smaller scale cross-guided features will adaptively learn to obtain the neighbors in a relatively large scope, and vice versa. Since our initial depth is obtained by dilating sparse measurement points [15], surrounding points of most pixels are inaccurate. Therefore, at the beginning of the update process, we employ small scale cross-guided features to lead the TDU to predict a large scope, which obtains some long-range but accurate points as neighbors. As the depth map becomes more refined, larger scale cross-guided features are sequentially used to pay more attention to shorter-range neighbors. Due to the elegant recurrent strategy, our LRRU only requires four it-

Table 1. **The configurations of four LRRU variants**, which are obtained by adjusting the number of channels in different stages of the cross-guided feature extraction network.

Models	Number of channels					Params.
	stage1	stage2	stage3	stage4	stage5	
LRRU-Mini	8	16	32	32	32	0.3 M
LRRU-Tiny	16	32	64	64	64	1.3 M
LRRU-Small	32	64	128	128	128	5 M
LRRU-Base	64	128	256	256	256	21 M

erations to achieve satisfactory results.

Extensive experiments on both indoor and outdoor datasets verify the performance of our method. Furthermore, we conduct comprehensive ablation studies to demonstrate the effectiveness of each component. Lastly, we extend our network framework to the depth-only case.

Our main contributions are summarized as:

- We propose a novel lightweight deep network architecture for depth completion, which pre-fills the sparse depth map and iteratively updates it by the proposed Target-Dependent Update (TDU) unit.
- We propose a long-short range recurrent strategy, which dynamically adjusts kernel scopes during the update process to obtain long-to-short range dependencies.
- As shown in Fig. 1 and Table 1, our four LRRU variants achieve state-of-the-art performance across different parameter regimes. Especially, the LRRU-Base model outperforms SOTA methods on NYUv2 [24] and ranks 1st on the KITTI benchmark [2] at the time of submission.

2. Related Work

Depth Completion. In the deep learning era, the straightforward depth completion methods [32, 35, 46] employ various network structures to directly predict dense depth maps. Since the RGB images contain rich texture and semantic information, which are critical to recover the structure details of depth maps, many popular methods [11, 19, 22, 23, 28] fuse the information of RGB images and sparse depth maps to boost the depth completion. Ma *et al*. [23] propose to concatenate the depth maps and the RGB images to form a 4D tensor, which is known as "early-fusion". To reduce the gap between different modalities, some works[11, 19, 22, 28] propose a "later-fusion" method, which extracts the feature of the RGB images and the sparse depth maps separately, and feeds the fusion features into the network. Further, Tang *et al*. [30] propose a feature fusion module based on the guided dynamic convolutional network [9, 12] to better utilize the guidance feature of RGB images. In addition, some works [16, 36, 38] propose to first densify the sparse depth by classical approaches [15, 37], and then learn a residual of the initial depth approximation. However, existing methods usually employ massive parameters and computation to obtain good results. A lightweight and efficient network architecture is lack.
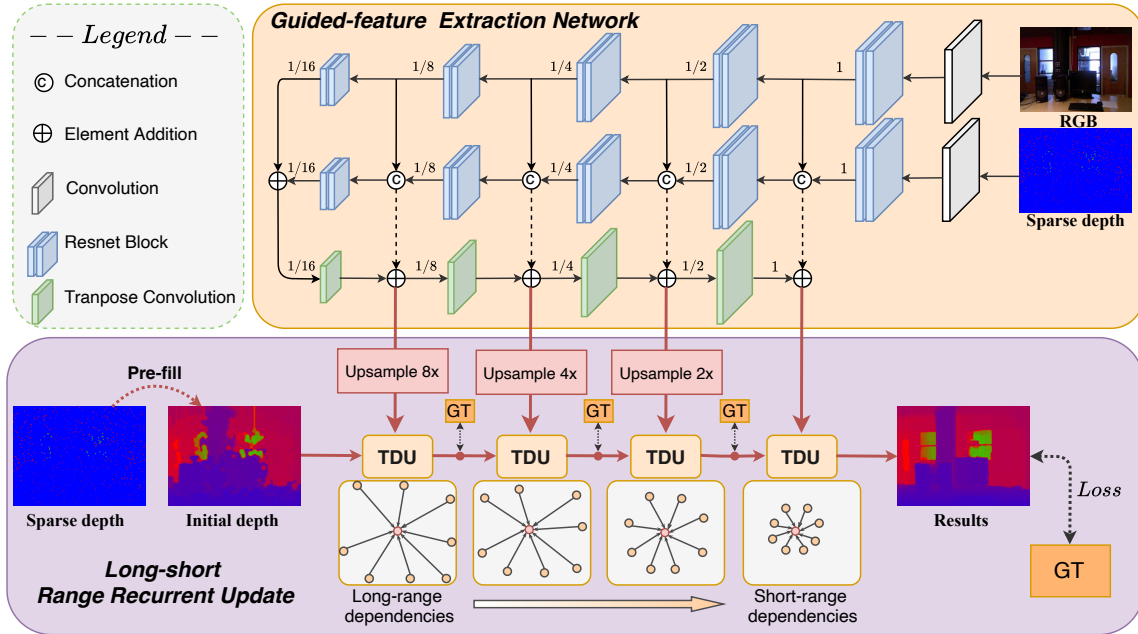
Figure 2. **The Long-short Range Recurrent Updating (LRRU) network**, which extracts the cross-guided features from RGB images and sparse depth map, and then iteratively updates the pre-filled depth map by **the proposed Target-Dependent Update (TDU) unit** according to **our long-short range recurrent strategy**.

Spatial Propagation Networks. The depth maps predicted by the direct-regression methods suffer from blur effect and distortion of object boundaries [4]. To address this issue, some works propose a series of spatial propagation networks (SPNs) [3, 4, 11, 17, 21, 26, 41], which iteratively update the output of direct-regression methods by aggregating the reference and neighbor pixels. The original SPN [20] updates each pixel by three adjacent pixels from the previous row or column. The serial update process is performed in four directions individually, and the results are combined by max-pooling. To make the update process more efficient, Cheng *et al.* [4] propose the convolutional spatial propagation network (CSPN), which updates all pixels simultaneously within a fixed-local neighbors. However, the fixed-local neighborhood configuration will introduce irrelevant points. Furthermore, CSPN++ [3] assembles the results obtained by using different kernel sizes. DSPN [41] and NLSPN [26] predict a non-local neighborhood by learning the offsets to the regular grid, while the difference between them is that DSPN obtains the kernel weights by calculating the similarity between features, while NLSPN learns it by the network. Despite the improved SPN-based methods providing more flexibility in selecting neighbors, they employ fixed kernel weights during the update process, which limits the representation capability of SPN. To alleviate the problem, DySPN [17] gives variable weights to neighbors with different distances through learned attention maps. GraphCSPN [21] leverages graph neural networks (GNN) to integrate 3D information into the update process. However, existing SPNs still use fixed neighbors and are not

able to dynamically adjust them during the update process.

3. Method

Given a sparse depth map, we first densify it by a simple non-learning method [15]. Then, according to the long-short range recurrent strategy (described in Sec. 3.2), our method iteratively updates the initial depth map through the target-dependent update unit (described in Sec. 3.1) to obtain accurate and dense depth map. In Sec. 3.3, we provide implementation details of our method. For the convenience of description, we use **the target depth** (denoted as $\hat{D}^{\mathbf{t}}$) to refer to the depth map to be updated in the t-th update. Thus, $\hat{D}^{\mathbf{1}}$ represents the initial map obtained by [15]. Moreover, $\hat{D}^{\mathbf{t+1}}$ denotes the updated result.

3.1. Target-dependent Update Unit

The proposed Target-Dependent Update (TDU) unit updates the target depth map by learned spatially-variant kernels, which model neighbors and their affinities of each pixel. To avoid irrelevant neighbors brought by the fixed-local neighborhood configuration, our TDU employs fully convolutional networks to predict the kernel weights and sampling position of neighbors as [26], where the sampling position is obtained by learning offsets to the regular grid. However, direct supervisory information for the weights and offsets is typically not available, which often results in the training instability. To overcome the instability, we use the features from RGB images and sparse depth maps to guide our TDU to obtain relevant neighbors, motivated
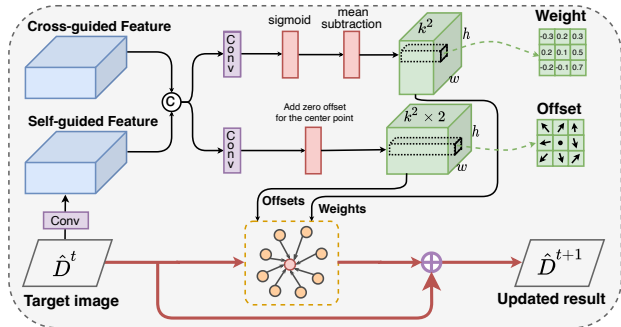
Figure 3. **The Target-Dependent Update (TDU) unit** updates the target map by learning its residual map, where the cross-guided features from guidance images guide TDU to obtain relevant neighbors and the self-guided features from target maps ensure TDU adjust adaptively based on the content of the target map.

by rich structure details in RGB images and the accurate scene depth information in sparse depth maps. Since the dense RGB image and the sparse depth map belong to different modalities, we employ a dual-encoder network similar to [30], which uses two separate sub-networks to extract the features of RGB images and sparse depth maps respectively, and fuses them at multiple scales. However, if only the features from RGB and sparse depth maps are used to guide the TDU, the update operation is independent of the content of the depth map to be updated, which may lead to sub-optimal solutions, especially when the initial depth map is not directly regressed from RGB images and sparse depth maps. Therefore, in addition to RGB images and sparse depth maps, we propose to extract the features from the target depth map itself to guide our TDU. We refer to the features form RGB images and sparse depth maps as the cross-guided features, and the features from the target depth map as the self-guided features. As shown in Eq. (1), the cross-guided features $F_{Cross-guided}$ are extracted from the input RGB image I and sparse depth map S by the feature extraction network f_θ, and the self-guided features $F_{Self-guided}$ are obtained from the target depth map $\hat{D}^{\mathbf{t}}$ through a convolutional layer f_ψ.

$$F_{Cross-guided} = f_\theta(I, S), \; F_{Self-guided} = f_\psi(\hat{D}^{\mathbf{t}}). \quad (1)$$

Weights and Offsets Regression. As shown in Fig. 3, our TDU first concatenates the cross-guided features and the self-guided features, and then learns the weight and offset feature map by two individual 1×1 convolutional layers. To make the weights and offsets converge quickly, we add some restrictions to regulate their behaviors and guide the learning process. Concretely, the weight feature map has k^2 channels, where k is the kernel size and set to 3 in this paper. We apply a sigmoid layer to make the weight larger than zero and smaller than one. In addition, we subtract the mean value from the output of the sigmoid layer to make the sum of the weight to be zero, which performs like a a high-
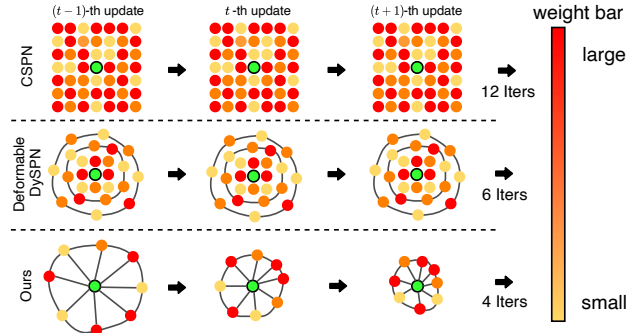

Figure 4. **The Long-short Range Recurrent Strategy**, which dynamically adjust the kernel scope from large to small during the update process, while existing SPNs keep it unchanged.

pass filter [14]. The offset feature map has $2k^2$ channels, which represents the deviation of the sampling point from locations on a regular grid in the x and y directions. However, to ensure that each reference pixel participates in its own update process, we first predict an offset feature map with $2(k^2 - 1)$ channels, and then insert zero into the center of the offset feature map channel [26].

Residual Connection. We observe that the input and output map of the update unit are highly correlated, and share the low-frequency information. Therefore, unlike existing SPNs [3, 4, 11, 17, 21, 26, 41] that directly predict the updated depth map, we propose to learn a residual image of the target depth map to enhance structural details and suppress noise. Given the learned weights and sampling offsets, as shown in Eq. (2), the residual image $\Delta \hat{D}_{\mathbf{p}}^{\mathbf{t}}$ at position $\mathbf{p} = (x, y)$ is obtained by a weighted average.

$$\Delta \hat{D}_{\mathbf{p}}^{\mathbf{t}} = \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p})} \mathbf{W}_{\mathbf{pq}}(F_{Cross-guided}, F_{Self-guided}) \hat{D}_{\mathbf{q}}^{\mathbf{t}}. \quad (2)$$

In Eq. (2), $\mathcal{N}(\mathbf{p})$ denotes the set of neighbors of the position $\mathbf{p}$. Since the offsets are normally fractional, we use the bilinear interpolation to sample local four points as [5] and [47]. The filter weights $\mathbf{W}$ are predicted from the cross-guided and self-guided features. We aggregate depth values from the sparsely chosen locations with the learned weights. Then, we add the residual image to the target depth map as Eq. (3) to obtain the updated depth map $\hat{D}^{\mathbf{t+1}}$.

$$\hat{D}^{\mathbf{t+1}} = \hat{D}^{\mathbf{t}} + \Delta \hat{D}^{\mathbf{t}}. \quad (3)$$

3.2. Long-short Range Recurrent Update Strategy

During the update process, we need an effective recurrent update strategy to enable the TDUs to obtain appropriate kernel weights and scopes for their respective target. Specifically, for the initial depth map obtained by a non-learning method [15], only a few available sparse measurement points and their surrounding points have high accuracy, the surrounding points of most pixels are inaccu-

rate [35]. Therefore, at the beginning of the update process, we should employ a large kernel scope to obtain some long-range but accurate points as neighbors. As the depth map becomes more refined, the kernel scope should be gradually narrowed to pay more attention to short-range points, as they are generally more relevant to the reference point. However, existing recurrent update strategies used by SPNs are not flexible enough to meet the above needs. For example, CSPN [4] and NLSPN [26] use fixed kernel weights and scopes during the update process, which not only limits the representation capability of SPNs, but also requires massive iterations to obtain long-range dependencies. Although CSPN++ [3] and DySPN [17] alleviate this issue by the model ensemble and attention mechanism, their kernel scopes remain unchanged during the update process, even in the deformable DySPN.

In this paper, we propose a long-short range recurrent update strategy that is shown in Fig. 4. The parameters of each TDU, including the kernel weights and sampling positions of neighbors, are learned by considering the cross-guided and self-guided features. We observe that when TDUs of the iterative update process are guided by the cross-guided features of different scales respectively, the TDU guided by smaller scale cross-guided features will adaptively learn to obtain the neighbors with a relatively large scope, and vice versa. We consider this is due to the different receptive fields of the cross-guided features at different scales. Building upon the above observations, we employ the $1/8$-scale cross-guided feature map to guide the TDU of the first iteration to obtain the neighbors with a large scope. In subsequent iterations, the TDU gradually uses the larger scale cross-guided features, such as $1/4$-scale, $1/2$-scale, and full-scale, to obtain the neighbors with smaller scopes. Fig. 5 and Fig. 6 demonstrate the kernel scope changes from large to small during the iterative update process on the KITTI and NYUv2 dataset. Since the proposed recurrent update strategy is highly flexible, we can achieve satisfactory results by using fewer iterations and neighbors.

3.3. Implementation Details

Network Architecture. Our network architecture, shown in Fig. 2 consists of the cross-guided feature extraction network and the long-short range recurrent update module. The cross-guided feature extraction network employs two sub-networks, the depth encoder and the RGB encoder, to extract features from the sparse depth map and corresponding RGB image respectively. The extracted multi-scale RGB features are injected into the depth encoder to fully integrate the information from different modalities. Then, a decoder network is used to learn the residual of the fused multi-scale features. The cross-guided features are first up-sampled to the same resolution as the initial depth map, and utilized in the TDUs of the iterative update process.
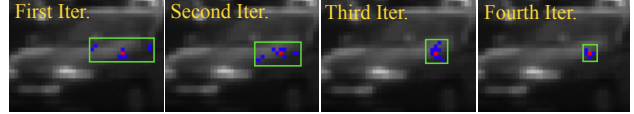

Figure 5. Typical examples to illustrate that large-to-small kernel scopes are dynamically adjusted to capture long-to-short range dependencies during the iterative update process (the red points denote the reference pixel and blue points denote the neighbors).
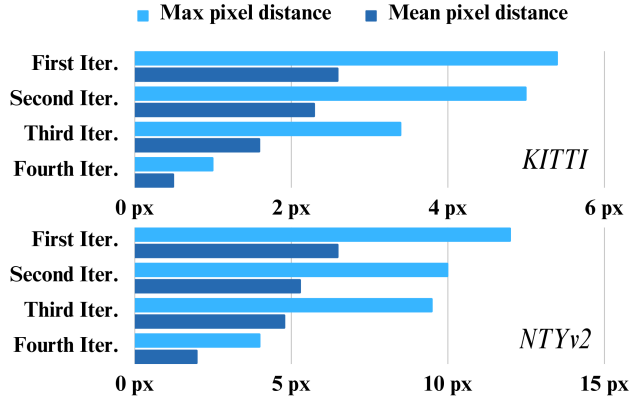

Figure 6. Analysis of the max and mean pixel distances from neighbors to the reference pixel on KITTI and NYUv2 dataset.

Loss Function. We supervise our network through $\mathcal{L}_1$ and $\mathcal{L}_2$ distance between the output and the ground truth D_{gt} over results of each iteration, $\{\hat{D}^2, \cdots, \hat{D}^N\}$, with exponentially increasing weights. The loss is defined in Eq. (4).

$$\mathcal{L} = \sum_{\sigma=1}^{2} \sum_{i=2}^{N} \gamma^{N-i} \|(\hat{D}^i - D_{gt}) \odot \mathbf{1}_{\{D_{gt}>0\}}\|^{\sigma}, \quad (4)$$

where $\mathbf{1}$ indicates whether there is a value in the ground truth, and $\odot$ denotes the element-wise multiplication. We set $\gamma = 0.8$ in our experiments.

Training Details. We employ PyTorch [27] to implement our model, which is trained and tested with GeForce RTX 3090 GPUs. All models are initialized from scratch with random weights. During training, we employ the Adam optimizer with a batch size of 8. We set $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay is 10^{-6}, and the total number of epochs is 45. The initial learning rate is 10^{-3}, and the learning rate remains unchanged for the first 15 epochs and decreases by 50% every 5 epochs.

4. Experiments

4.1. Datasets and Metrics

KITTI Dataset [7]. The KITTI dataset, a popular real-world autonomous driving dataset, consists of sparse depth maps projected from raw LiDAR scans and corresponding RGB images. It contains 86k frames for training, 1k selected frames for validation, and 1k frames without ground truth that need to be tested on the KITTI online benchmark.
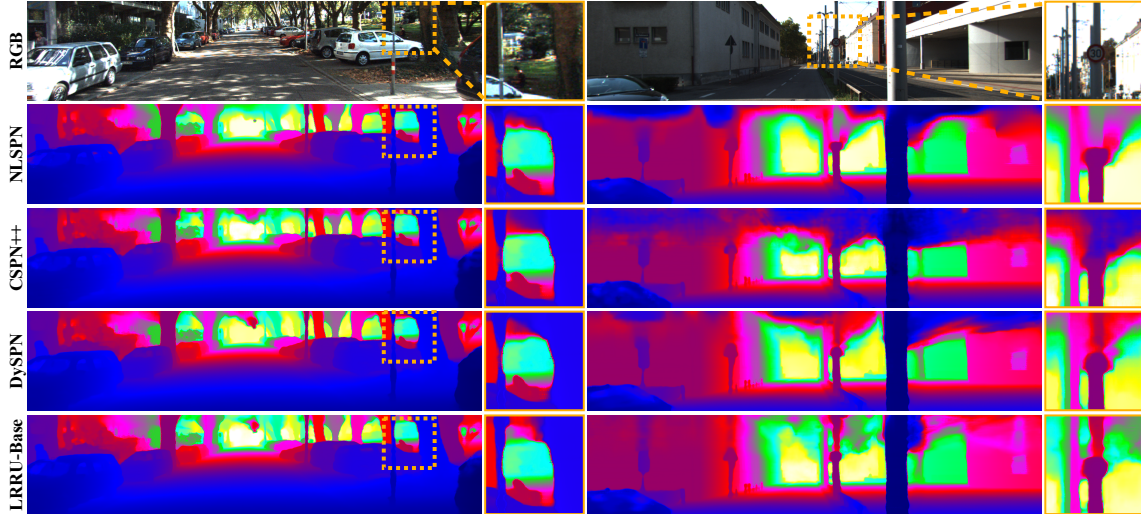
Figure 7. Qualitative comparison with SOTA methods on KITTI test dataset. From top to bottom are RGB images, dense depth maps predicted by NLSPN [26], CSPN++ [3], DySPN [17] and Ours, respectively. We zoom in some representative areas for detailed comparison.

NYUv2 Dataset [24]. The NYUv2 dataset consists of RGB and depth images acquired from 464 different indoor scenes. Following the standard setting [22], we train the model with 50K images sampled from the training set, and test it on 654 official labeled images. For both train and test datasets, the original images of size 640×480 are down-sampled to half and then center-cropped to 304×228.

Evaluation Metrics. Following exiting depth completion methods [22, 26, 30], we employ Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), inverse RMSE (iRMSE), inverse MAE (iMAE), mean absolute relative error (REL), and percentage of pixels satisfying δ_τ for quantitative evaluation.

4.2. Comparison with SOTA Methods on KITTI

Quantitative Comparison. Table 2 reports the quantitative evaluation of our methods and state-of-the-art (SOTA) depth completion methods on the KITTI benchmark, which ranks all methods according to the RMSE value. Since directly predicting dense depth maps from sparse depth maps and RGB images is a difficult "ill-posed" task, most direct-regression methods, such as S2D [22], DeepLiDAR [28], GuideNet [30], ACMNet[44], PENet [11], and RigNet[42], generally require massive stacked filters and layers to learn robust features. They usually have a large number of model parameters and a relatively long inference time, while the performances of the method are difficult to achieve the best of the time [7]. Even the efficiency-oriented depth completion methods, such as excellent GAENet [1] and FuseNet[28], are not able to achieve a good balance between model parameters, inference time and performance. To improve the performance of direct-regression methods, a series of SPNs are proposed to refine the output of these methods, such as CSPN [4], NLSPN [26], CSPN++ [3], and

Table 2. Quantitative comparison with state-of-the-art (SOTA) methods on the KITTI benchmark. The best and second-best results are highlighted in **red** and **blue** colors , respectively.

Methods	RMSE [mm]	MAE [mm]	iRMSE [1/km]	iMAE [1/km]
CSPN[4]	1019.64	279.46	2.93	1.15
S2D[22]	814.73	249.95	2.80	1.21
GAENet[1]	773.90	231.29	2.29	1.08
DeepLiDAR[28]	758.38	226.50	2.56	1.15
FuseNet[28]	752.88	221.19	2.34	1.14
MDANet[13]	738.23	214.99	2.12	0.99
GuideNet[30]	736.24	218.83	2.25	0.99
CSPN++[3]	743.69	209.28	2.07	0.90
NLSPN[26]	741.68	199.59	1.99	0.84
ACMNet[44]	744.91	206.09	2.08	0.90
PENet[11]	730.08	210.55	2.17	0.94
RigNet[42]	712.66	203.25	2.08	0.90
DySPN[17]	**709.12**	**192.71**	**1.88**	**0.82**
LRRU-Mini	774.43	210.87	2.21	0.90
LRRU-Tiny	738.86	200.28	2.04	0.85
LRRU-Small	717.50	197.72	1.96	0.85
LRRU-Base	**696.51**	**189.96**	**1.87**	**0.81**

DySPN[17]. The results show that the supplemental refinement modules improve the performance of the method, but also requires longer inference time.

By contrast, our proposed method (LRRUs) is more effective. Specifically, our smallest LRRU-Mini model achieves better results than some large-scale methods, such as S2D [22] and CSPN [4]. In addition, LRRU-Tiny model significantly improves the performance over LRRU-Mini by increasing only 1M parameters, which outperforms many excellent methods, such as DeepLiDAR [28]. The experimental results strongly demonstrate the effectiveness of our proposed method. It is worth noting that our largest LRRU-Base model achieves the best results of the KITTI benchmark in all evaluation metrics, including RMSE, MAE, iRMSE and iMAE, which **ranks 1st** at the time of submission. To the best of our knowledge, it is the first depth com-

Table 3. Ablation studies of the proposed LRRU on the KITTI validation dataset. "x-scale" denotes the scale of the cross-guided feature, and "←" indicates that LRRU uses 1/8-scale, 1/4-scale, 1/2-scale, and full-scale cross-guided features sequentially during the update process. The results of our four LRRU variants are highlighted with a gray background. The lower result values represent better performance.

	Models	Long-short Range Recurrent Update Strategy					Target-Dependent Update Unit					Results	
		Full-scale	1/2-scale	1/4-scale	1/8-scale	iterative	w/o kernel prediction	w/o self-guided feature	w/o Res.	w/o mean subtraction	w/o fix reference	RMSE[mm]	MAE[mm]
(a)	Baseline											1027.1	302.1
(b)	Mini	✓										819.4	228.9
(c)			✓									823.6	228.4
(d)				✓								825.2	229.4
(e)					✓							838.3	237.3
(f)		✓	✓	✓	✓							812.4	226.3
(g)		✓	✓	✓	✓	→						844.5	236.4
(h)		✓	✓	✓	✓	←						800.9	218.9
(i)		✓	✓	✓	✓	←	✓					812.2	219.7
(j)		✓	✓	✓	✓	←		✓				817.5	223.1
(k)	Mini	✓	✓	✓	✓	←			✓			803.4	221.5
(l)		✓	✓	✓	✓	←				✓		804.0	224.8
(m)		✓	✓	✓	✓	←					✓	799.4	222.2
(n)	Tiny	✓	✓	✓	✓	←						761.5	207.6
(o)	Small	✓	✓	✓	✓	←						741.3	201.8
(p)	Base	✓	✓	✓	✓	←						728.3	197.9

pletion method that obtains an RMSE value below 700 mm.
Qualitative Comparison. Fig. 7 compares the dense depth maps predicted by our LRRU-Base and several popular SPNs-based methods, such as NLSPN [26], CSPN++ [3], and DySPN [17]. These SPNs-based methods use additional refinement modules, which are generally considered to have good visualization results. However, as shown in the enlarged results, the dense depth maps predicted by these SPNs-based methods have depth discontinuity and blur effect in some areas, especially in thin structures. By contrast, our results have better structure details and are more accurate at the object boundaries.

4.3. Ablation Studies

To reduce the training time, the ablation experiments are conducted on our smallest **LRRU-Mini model**.
Network Architectures. Existing deep-learning methods based on direct regression rely heavily on the representation capability of the network. When the network size decreases, the method performance drops significantly. We select the method that employs the cross-guided feature extraction network to directly predict dense depth maps as the baseline. Table 3 (a) shows that the results of the baseline method has a large error, the RMSE is 1027.1 mm, and the MAE is 302.1 mm. Unlike direct regression, we iteratively update the initial coarse depth map by the proposed TDU to obtain an accurate depth map. As shown in Table 3 (b)-(e), even if we only employ a TDU guided by the cross-guided feature of arbitrary scales to update the initial depth map once, the proposed method achieves better performance. Specifically, compared to the baseline, the results obtained by using the full-scale cross-guided feature reduce RMSE and MAE by 207.7 mm and 73.2 mm, respec-

tively. It illustrates that our proposed network framework is more effective than direct regression when the network size is limited. In addition, As shown in Table 3 (f), when we use a TDU guided by the fusion cross-guided feature of various scales, the method performance is further improved.

Long-short Range Recurrent Update Strategy. In this paper, we propose a long-short range recurrent update strategy to manipulate the update process. By sequentially using small-to-large scale cross-guided feature to guide the TDU, our method dynamically obtains long-to-short range dependencies. The results in Table 3 (h) show that the method using our proposed strategy achieves substantially improved performance. Meanwhile, we report the results obtained by using the cross-guided features in the reverse order of ours, namely a short-to-long range recurrent update strategy. The results in Table 3 (g) show that the method using this strategy achieves poor results, which demonstrates the advantage of the proposed recurrent strategy.

Target-Dependent Update Unit. The TDU explicitly builds the cross-guided feature and self-guided feature into update unit, and updates the depth map by learned spatially-variant kernels. If the TDU predicts the residual map by direct regression rather than learning spatially-variant kernels, the results in Table 3 (i) show that the method performance will decrease significantly. Meanwhile, as shown in Table 3 (j), TDU without using the self-guided feature also heavily harms the method results. In addition, if the residual learning is not adopted, the performance shown in Table 3 (k) will decrease. As described in Sec. 3.1, we perform many operations, including subtracting the mean value from the output of the sigmoid layer and fixing the reference pixel of the update unit, to regulate the behaviors of the weights and offsets and guide their learning process. The results in Ta-

ble 3 (l) and (m) show that these restrictions are beneficial to improve the performance of the method. It is worth noting that although fixing the reference pixel is not able to boost the quantitative results greatly, we empirically observe that it effectively avoids unreliable neighboring pixels.

LRRU vs existing SPNs. To better show the advantages of our method, we employ the refinement modules of existing SPNs directly in our proposed framework. We select the refinement module of NLSPN [26] under its original setting for the comparison, which performs best among all open-source SPN-based methods. The results in Table 4 show that our framework using existing refinement module directly performs worse than our method, which illustrate the effectiveness of our proposed method.

Table 4. The performance comparison of our proposed framework using the refinement module of this paper and existing SPNs.

Methods	RMSE [mm]	MAE [mm]	iRMSE [1/km]	iMAE [1/km]
LRRU-Mini (Ours)	800.9	218.9	2.4	0.9
LRRU-Mini(NLSPN [26])	891.5	240.2	5.0	1.1

Loss Functions. The above ablation experiments are trained with $\mathcal{L}_2$ loss. As shown in Table 5, we test the combination of $\mathcal{L}_1$ and $\mathcal{L}_2$ losses that improves MAE performance while decreases RMSE performance slightly. For a balance between the RMSE and MAE, we adopt the $\mathcal{L}_1 + \mathcal{L}_2$ loss to train the model on the KITTI test dataset.

Table 5. The performance comparison of using different loss.

Methods	Loss	RMSE [mm]	MAE [mm]	iRMSE [1/km]	iMAE [1/km]
LRRU-Mini	$\mathcal{L}_2 + \mathcal{L}_1$	806.3	210.2	2.3	0.9
LRRU-Tiny	$\mathcal{L}_2 + \mathcal{L}_1$	763.8	198.9	2.1	0.8
LRRU-Small	$\mathcal{L}_2 + \mathcal{L}_1$	745.3	195.7	2.0	0.8
LRRU-Base	$\mathcal{L}_2 + \mathcal{L}_1$	729.5	188.8	1.9	0.8

Hardware cost comparison. Table 6 compares the hardware cost of LRRUs and most open-source methods on the KITTI validate dataset (the resolution of the input image is $356 * 1216$), including the model parameters, GPU memory, and runtime. The experiment is conducted on the same hardware (a GeForce RTX 3090 GPU). Although our pre-filled method (IP-BASIC [15]) needs extra 8ms CPU-time, the proposed LRRUs are more efficient. Especially, our smallest LRRU-Mini model contains only 0.3M parameters and has a fast inference speed (38.3ms).

Table 6. Hareware cost comparison of KITTI validate dataset (the resolution of input images $356 * 1216$) on a 3090 GPU. Our pre-filled method (IP-BASIC [15]) needs extra 8ms CPU-time.

Methods	Parameters	GPU Memory	GPU Time	CPU Time	Total Time
MDANet[13]	3M	2092M	89.4ms	-	89.4ms
CSPN[4]	26M	2350M	152.7ms	-	152.7ms
S2D[22]	26M	2706M	65.5ms	-	65.5ms
DeepLiDAR[28]	144M	3496M	323.6ms	-	323.6ms
NLSPN[26]	26M	2628M	76.0ms	-	76.0ms
GuideNet[30]	74M	2074M	51.3ms	-	51.3ms
PENet[11]	132M	2822M	129.6ms	-	129.6ms
LRRU-Mini	**0.3M**	**1448M**	**30.3ms**	8ms	**38.3ms**
LRRU-Tiny	1.3M	1672M	41.4ms	8ms	49.4ms
LRRU-Small	5M	2152M	59.3ms	8ms	67.3ms
LRRU-Base	21M	3170M	117.4ms	8ms	125.4ms

Table 7. Results of LRRU models on the KITTI validate dataset with test-time augmentation (TTA).

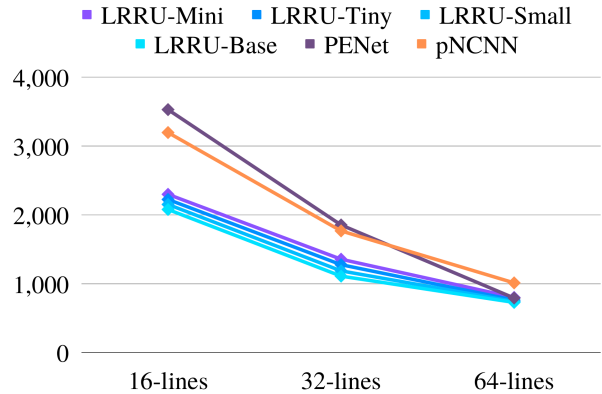| Methods	TTA	RMSE [mm]	MAE [mm]	iRMSE [1/km]	iMAE [1/km]
LRRU-Mini	False	806.3	210.2	2.3	0.9
LRRU-Tiny	False	763.8	198.9	2.1	0.8
LRRU-Small	False	745.3	195.7	2.0	0.8
LRRU-Base	False	729.5	188.8	1.9	0.8
LRRU-Mini	True	0.7943	208.5	2.3	0.9
LRRU-Tiny	True	757.8	198.0	2.1	0.8
LRRU-Small	True	739.2	195.0	2.0	0.8
LRRU-Base	True	723.4	188.1	1.9	0.8

Figure 8. The performance (RMSE [mm]) of our methods, PENet [11] and pNCNN [6] on fewer points.

Test-time augmentation (TTA). The TTA is widely used in depth completion methods (such as GuideNet [30], ACM-Net [44], *et al.*) as a performance-enhancing strategy, but is rarely described. In the inference stage, the model with TTA predicts the results of the original sample and the horizontally reversed sample at the same time, and averages their results as the final result. Table 9 shows that the TTA strategy makes the model obtain lower RMSE, especially in the smallest LRRU-Mini model.

4.4. Experiments on fewer points

The sparse depth map of the KITTI dataset is obtained by a 64-line Velodyne LiDAR. However, in many practical applications, only 32-line or 16-line LiDAR will be uesed due to cost constraints, which only provide fewer depth points. Therefore, it is crucial to analyze the performance of the method on sparse depth maps with different sparsity levels. We trained LRRUs on 64-Line LiDAR depth of KITTI validation set and tested the performance (RMSE[mm]) on fewer lines that is obtained by the method provided by [45]. Although our results shown in the Fig. 8 gradually decrease with decreasing depth points (from 64-Line to 16-Line), they still outperform competing methods at different sparse patterns, which shows that our model has better generalization capability on more sparse depth maps.

4.5. Experiments on NYUv2 Dataset

To verify the effectiveness of our proposed method in indoor scenes, we conduct comparative experiments on the

Table 8. Quantitative evaluation on the NYUv2 dataset. The best and second-best RMSE and REL results are highlighted in **red** and **blue** colors , respectively.

Methods	Params.	RMSE[m]	REL	$\delta_{1.25}$	$\delta_{1.25^2}$	$\delta_{1.25^3}$
S2D [22]	26 M	0.230	0.044	97.1	99.4	99.8
CSPN [4]	17.4 M	0.117	0.016	99.2	99.9	100.0
DeepLiDAR [28]	48 M	0.115	0.022	99.3	99.9	100.0
DepthNormal [40]	- M	0.112	0.018	99.5	99.9	100.0
FCFRNet [19]	- M	0.106	0.015	99.5	99.9	100.0
ACMNet [44]	4.9 M	0.105	0.015	99.4	99.9	100.0
GuideNet [30]	74 M	0.101	0.015	99.5	99.9	100.0
NLSPN [26]	25.8 M	0.092	**0.012**	99.6	99.9	100.0
RigNet [42]	- M	**0.090**	**0.012**	99.6	99.9	100.0
DySPN [17]	∼26 M	**0.090**	**0.012**	99.6	99.9	100.0
LRRU-Mini	0.3 M	0.101	0.013	99.4	99.9	100.0
LRRU-Tiny	1.3 M	0.096	0.012	99.5	99.9	100.0
LRRU-Small	5 M	0.093	0.012	99.5	99.9	100.0
LRRU-Base	21 M	**0.091**	**0.011**	99.6	99.9	100.0

Table 9. Quantitative evaluation on the KITTI validation dataset **under the depth-only setting**.

Methods	Params.	RMSE [mm]	MAE [mm]
S2D (Depth-only) [22]	26 M	985.1	286.5
CUNet [35]	35 M	958.8	245.3
LRRU-Mini	**0.3 M**	987.7	251.8
LRRU-Tiny	1.3 M	958.2	242.4
LRRU-Small	5 M	**950.9**	237.1
LRRU-Base	21 M	957.4	**235.9**

NYUv2 dataset. Following existing methods [17, 22, 30], we train and evaluate our model with the setting of 500 sparse LiDAR samples. The quantitative evaluation results in Table 8 demonstrate that the four variants of the proposed LRRU achieve the best performance across different parameter regimes. Specifically, our LRRU-Mini model achieves an impressive results, which employs only 0.3M parameters to obtain the same RMSE as GuideNet [30] using 74M parameters. We empirically found that if using a UNet-like model with similar amount of parameters (0.3M) to directly regress, the model will fail. Meanwhile, the LRRU-Base model outperforms state-of-the-art methods with relatively fewer parameters.

Fig. 9 shows the qualitative results of the proposed LRRU models. Since the input depth of the NYUv2 dataset is more sparse than the depth of the KITTI dataset, the pre-filled depth map obtained by a simple hand-crafted method [15] looks terrible. However, LRRUs can rectify the initial depth map by the recurrent update process, which obtains satisfactory results, even in some thin structures.

4.6. Extension to Depth-only Case

As a general framework for depth completion, our approach can be extended to scenarios where only sparse depth maps are available. In the depth-only setting, we only extract the cross-guided feature from the sparse input depth map. We report the performance of our method on the KITTI validation dataset in Table 9. The experimental results show that the method performance drops significantly compared to using RGB image input. We consider
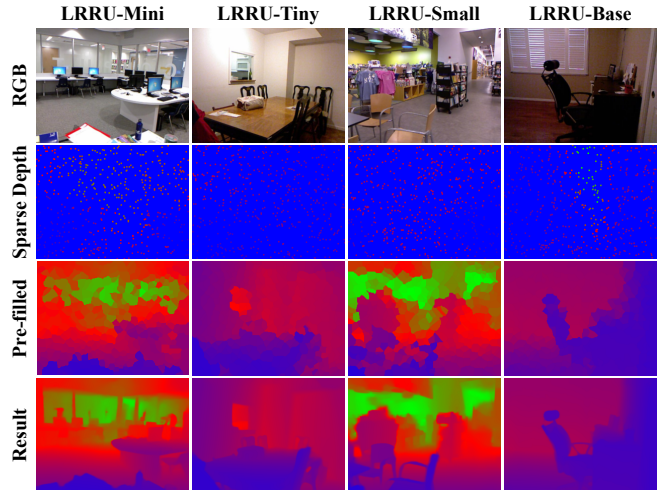

Figure 9. Qualitative results of the proposed LRRUs on the NYUv2 dataset. The sparse depth is dilated to show and "Pre-filled" denotes the pre-filled depth map. Since the sparse depth map has few available points, the pre-filled depth is very coarse. However, our LRRUs can update it to satisfactory results.

that this is because if only the sparse depth map is input, the information provided is very limited. It is difficult for the depth completion methods to recover the structural details of the depth map. However, it is worth noting that our method still achieves better results compared with S2D (Depth-only) [22] and CUNet[35], where CUNet is specifically designed for the depth-only case and achieves the best performance at present. Meanwhile, we notice that the results of LRRU-Base decrease due to over-fitting.

5. Conclusion

In this paper, we have proposed Long-short Range Recurrent Updating (LRRU) network, a novel lightweight deep network framework for depth completion. Based on the proposed target-dependent update unit and long-short range recurrent strategy, our iterative update process is content-adaptive and highly flexible. Compared with conventional direct-regression approaches, our method achieves superior performance with fewer parameters and inference time. Experimental results demonstrate that our methods outperform existing methods on both indoor and outdoor scenes, and achieve SOTA performance across different parameter regimes. In the future, we will explore our method for other dense prediction tasks, such as monocular depth estimation and semantic segmentation.

Acknowledgments

References

[1] Hu Chen, Hongyu Yang, Yi Zhang, et al. Depth completion using geometry-aware embedding. In *Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA)*, 2022. 6

[2] Zhao Chen, Vijay Badrinarayanan, Gilad Drozdov, and Andrew Rabinovich. Estimating depth from rgb and sparse sensing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2

[3] Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 2, 3, 4, 5, 6, 7

[4] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2019. 1, 2, 3, 4, 5, 6, 8, 9

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 4

[6] Abdelrahman Eldesokey, Michael Felsberg, Karl Holmquist, and Michael Persson. Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 8

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 5, 6

[8] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2022. 1

[9] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2012. 2

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[11] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Penet: Towards precise and efficient image guided depth completion. In *Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA)*, 2021. 1, 2, 3, 4, 6, 8

[12] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 2

[13] Yanjie Ke, Kun Li, Wei Yang, Zhenbo Xu, Dayang Hao, Liusheng Huang, and Gang Wang. Mdanet: Multi-modal deep aggregation network for depth completion. In *Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA)*, 2021. 1, 6, 8

[14] Beomjun Kim, Jean Ponce, and Bumsub Ham. Deformable kernel networks for joint image filtering. *International Journal of Computer Vision (IJCV)*, 2021. 4

[15] Jason Ku, Ali Harakeh, and Steven L Waslander. In defense of classical image processing: Fast depth completion on the cpu. In *2018 15th Conference on Computer and Robot Vision (CRV)*, 2018. 2, 3, 4, 8, 9

[16] Byeong-Uk Lee, Kyunghyun Lee, and In So Kweon. Depth completion using plane-residual representation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[17] Yuankai Lin, Tao Cheng, Qi Zhong, Wending Zhou, and Hua Yang. Dynamic spatial propagation network for depth completion. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022. 2, 3, 4, 5, 6, 7, 9

[18] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *Proceedings of the IEEE International Conference on 3D Vision, (3DV)*, 2021. 2

[19] 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 *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 2, 9

[20] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 3

[21] Xin Liu, Xiaofei Shao, Bo Wang, Yali Li, and Shengjin Wang. Graphcspn: Geometry-aware depth completion via dynamic gcns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2, 3, 4

[22] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA)*, 2019. 1, 2, 6, 8, 9

[23] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA)*, 2018. 1, 2

[24] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. 2, 6

[25] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew W. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE International Symposium on Mixed and Augmented Reality, (ISMAR)*, 2011. 1

[26] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2, 3, 4, 5, 6, 7, 8, 9

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming

Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 5

[28] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 6, 8, 9

[29] Kyeongha Rho, Jinsung Ha, and Youngjung Kim. Guideformer: Transformers for image guided depth completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[30] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion. *IEEE Transactions on Image Processing (TIP)*, 2020. 1, 2, 4, 6, 8, 9

[31] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[32] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *Proceedings of the IEEE International Conference on 3D Vision, (3DV)*, 2017. 1, 2

[33] Jiadai Wang, Jiajia Liu, and Nei Kato. Networking and communications in autonomous driving a survey. *IEEE Communications Surveys and Tutorials (CST)*, 2018. 1

[34] Shaoqian Wang, Bo Li, and Yuchao Dai. Efficient multiview stereo by iterative dynamic cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[35] Yufei Wang, Yuchao Dai, Qi Liu, Peng Yang, Jiadai Sun, and Bo Li. Cu-net: Lidar depth-only completion with coupled u-net. *IEEE Robotics and Automation Letters (RAL)*, 2022. 1, 2, 5, 9

[36] Alex Wong, Safa Cicek, and Stefano Soatto. Learning topology from synthetic data for unsupervised depth completion. *IEEE Robotics and Automation Letters (RAL)*, 2021. 2

[37] Alex Wong, Xiaohan Fei, Stephanie Tsuei, and Stefano Soatto. Unsupervised depth completion from visual inertial odometry. *IEEE Robotics and Automation Letters (RAL)*, 2020. 2

[38] Alex Wong and Stefano Soatto. Unsupervised depth completion with calibrated backprojection layers. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[39] Zexiao Xie, Xiaoxuan Yu, Xiang Gao, Kunqian Li, and Shuhan Shen. Recent advances in conventional and deep learning-based depth completion: A survey. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2022. 1

[40] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 9

[41] Zheyuan Xu, Hongche Yin, and Jian Yao. Deformable spatial propagation networks for depth completion. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2020. 2, 3, 4

[42] Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Baobei Xu, Jun Li, and Jian Yang. Rignet: Repetitive image guided network for depth completion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 1, 6, 9

[43] Youmin Zhang, Xianda Guo, Matteo Poggi, Zheng Zhu, Guan Huang, and Stefano Mattoccia. Completionformer: Depth completion with convolutions and vision transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[44] Shanshan Zhao, Mingming Gong, Huan Fu, and Dacheng Tao. Adaptive context-aware multi-modal network for depth completion. *IEEE Transactions on Image Processing (TIP)*, 2021. 6, 8, 9

[45] Yiming Zhao, Lin Bai, Ziming Zhang, and Xinming Huang. A surface geometry model for lidar depth completion. *IEEE Robotics and Automation Letters (RAL)*, 2021. 8

[46] Yiming Zhao, Mahdi Elhousni, Ziming Zhang, and Xinming Huang. Distance transform pooling neural network for lidar depth completion. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2021. 2

[47] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4