CFPNet: Improving Lightweight ToF Depth Completion via Cross-zone Feature Propagation

Laiyan Ding[1], Hualie Jiang[2], Rui Xu[2], Rui Huang[1]*
[1]The Chinese University of Hong Kong, Shenzhen
[2]Insta360 Research
laiyanding@link.cuhk.edu.cn, jianghualie@insta360.com, xurui@insta360.com, ruihuang@cuhk.edu.cn

Abstract

Depth completion using lightweight time-of-flight (ToF) depth sensors is attractive due to their low cost. However, lightweight ToF sensors usually have a limited field of view (FOV) compared with cameras. Thus, only pixels in the zone area of the image can be associated with depth signals. Previous methods fail to propagate depth features from the zone area to the outside-zone area effectively, thus suffering from degraded depth completion performance outside the zone. To this end, this paper proposes the CFPNet to achieve cross-zone feature propagation from the zone area to the outside-zone area with two novel modules. The first is a direct-attention-based propagation module (DAPM), which enforces direct cross-zone feature acquisition. The second is a large-kernel-based propagation module (LKPM), which realizes cross-zone feature propagation by utilizing convolution layers with kernel sizes up to 31. CFPNet achieves state-of-the-art (SOTA) depth completion performance by combining these two modules properly, as verified by extensive experimental results on the ZJU-L5 dataset. The code is available at https://github.com/denyingmxd/CFPNet.

1. Introduction

Depth completion from sparse depth measurements is an essential component of many tasks, including SLAM [23], novel view synthesis [41], and robot navigation [27]. Previous depth completion methods usually simulate sparse depth inputs randomly from depth maps acquired by RGB-D cameras, e.g., RealSense. Nevertheless, it is impractical to obtain such sparse depth as inputs for real applications. Recently, ToF sensors have been applied to depth super-resolution [13] and depth completion [21] due to their low power consumption and cost-effectiveness compared with RGB-D cameras. Readers may refer to this survey for more discussion on ToF imaging [33]. In this paper, we focus on the depth completion task, particularly implemented on a popular type of lightweight ToF sensor (e.g., ST VL53L5CX [2], denoted as L5), though our method is not limited to it. Despite its low resolution (e.g., 8×8), the low power consumption (e.g., 200mW) and low cost (e.g., \$6) of L5 allows it to be deployed in more applications. Note that we refer to this depth estimation task assisted by a lightweight ToF sensor as depth completion due to their similarity in input depth signals [36].

To get a preliminary overview of depth completion with a lightweight ToF sensor, we illustrate the sensing principle of L5 in Fig. 1 (a) as an example. Unlike conventional ToF

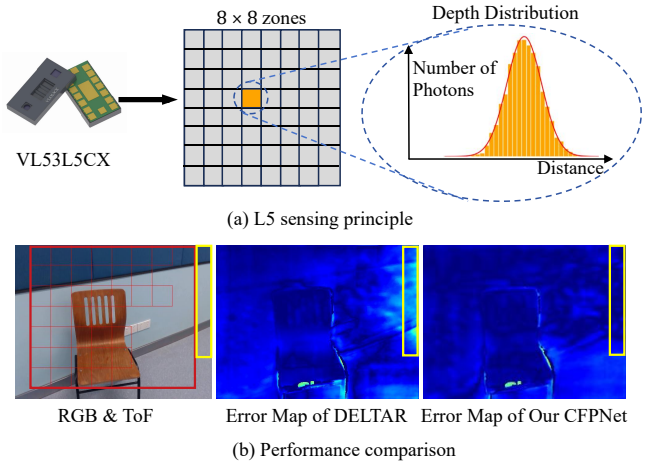


Figure 1. L5 sensing principle and performance comparison. (a) L5 would return zones of resolution 8×8, and each zone provides depth distribution information. (b) We overlay aligned zone areas on the paired RGB image and display the error maps of DELTAR and our CFPNet. The largest rectangle is the zone area, and some zones are missing due to too few received photons or inconsistency in measurement. Notice that CFPNet obtains smaller errors in outside-zone areas (the yellow rectangle).

*Corresponding author.

sensors, L5 is lightweight and has extremely low resolution (e.g., 8×8). Each zone in L5 produces a depth distribution about the corresponding 3D scene by counting the number of photons returned in discretized time intervals. DELTAR fits this distribution with a Gaussian distribution and transmits the mean and variance to save bandwidth and energy [21]. In Fig. 1 (b), we overlay the zones on the image. The zone area is the largest red rectangle, and each small red rectangle corresponds to one ToF measurement. Note that some zones are missing due to too few received photons or inconsistency in measurement.

Though depth inputs collected from lightweight ToF sensors (e.g., L5) are sparse and noisy, researchers have made depth completion from them plausible. DELTAR is a RGB-guided lightweight-ToF-based depth completion model [21]. At the core of DELTAR, an effective D-to-image module conducts feature fusion between depth and RGB features patchwisely in the zone area. Then, a self-attention layer [40] is leveraged to propagate the fused features to outside-zone areas. However, since this self-attention [40] queries values based on similarities, pixels in outside-zone areas may gather more information from outside-zone pixels which do not contain ToF information than from in-zone pixels. Thus, it is unlikely to propagate depth features effectively from zone areas to outside-zone areas. To tackle the restricted feature propagation issue, long range and steady cross-zone feature interaction are essential. To this end, we introduce a direct-attention-based module and a large-kernel-based module to establish steady and long-range dependencies between the two areas.

Direct-Attention-based Propagation Module. DAPM is based on the attention mechanism [40], which enables feature propagation regardless of pixel distances in the image. Specifically, we directly perform cross-attention from in-zone pixels to outside-zone pixels. Thus, outside-zone pixels can query information from in-zone pixels dynamically. More importantly, our DAPM avoids feature acquisition from outside-zone areas where ToF information does not exist during feature propagation. This advantage of DAPM compared with naive self-attention [40] enables more effective feature propagation.

Large-Kernel-based Propagation Module. LKPM incorporates a convolution layer of a large kernel size (e.g., 31×31) [9]. Large-kernel CNNs have been shown to have larger effective receptive fields [26] compared with small-kernel CNNs. Consequently, we use convolution layers of large kernels to establish long-range dependencies between pixels from zone areas and outside-zone areas. Moreover, the interaction between input signals in convolution depends on location rather than similarities as in attention [5]. Thus, with the large-kernel design, LKPM is not likely to fall into the situation that feature acquisition only comes from outside-zone areas, mitigating potential limitations.

Owing to the proposed DAPM and LKPM, our CFPNet can reduce the errors in outside-zone areas as in Fig. 1 (b). As a result, compared with the previous method [21], we reduce the mean absolute relative error (REL) from 0.127 to 0.103 on the ZJU-L5 dataset[21]. Notably, we decrease REL and RMSE by 46.5% and 30.8%, respectively, when the ToF is of resolution 2×2. In summary, our contributions are as follows:

1. We notice that outside-zone areas suffer from a great performance drop compared with in-zone areas. Furthermore, we propose CFPNet to alleviate this degradation with more effective cross-zone feature propagation.
2. Our CFPNet contains two feature propagation modules, namely DAPM and LKPM. DAPM allows direct feature propagation with the help of cross-attention, and LKPM propagates features using convolution layers of large kernel sizes.
3. CFPNet achieves remarkable performance gain on the ZJU-L5 dataset over previous methods. Codes will be released for peer research.

2. Related Work

2.1. Depth Completion

Early works of depth completion aim to predict pixel-wise depth maps given sparse depth [39] without RGB guidance. Yet, RGB-guided approaches generally outperform unguided ones [15]. Thus, methods that take both RGB and sparse depth as inputs have been proposed. CSPN [7] refine depth prediction within a local convolutional context. Nevertheless, fixed neighbours defined by local windows prevent information propagation in larger contexts. Consequently, NLSPN [29] allow depth refinement with non-local neighbours, CSPN++ [8] employ convolutions contexts of different kernel sizes, and PENet [16] introduce dilated CSPN++ [8] to enlarge propagation neighbourhoods. Recently, depth completion with lightweight ToF sensors has emerged. Compared with RGB-D cameras, lightweight ToF sensors (e.g., L5) are low-power and realistic [13, 21], but they only provide coarse depth distribution in each zone and have a low resolution (e.g., 8×8). DELTAR [21] is a framework that includes D-to-image cross attention to propagate depth distribution features into image features in a patch-wise way. Authors of DELTAR also notice the FOV difference between the ToF sensor and the RGB camera, and utilize self-attention [40] to propagate ToF features to global contexts. However, we notice that this self-attention is not enough and propose more effective cross-zone feature propagation modules to achieve better performance in outside-zone areas.

2.2. Long-Range Dependency

In computer vision, large receptive fields and long-range dependencies used to be acquired by stacking convolutional blocks [19]. Ever since the emergence of transformer blocks[40], long-range dependency has been modeled by these blocks and proven to be effective in semantic segmentation [11, 43], image restoration [44], depth completion [45], BEV perception [22]. These successful applications of transformer blocks [40] validate the necessity of long-range dependency modeling in deep neural networks [29]. Another way to achieve long-range dependency is to use convolutions with large kernels. Modern networks, including ResNet [12], DenseNet [17], EfficientNet [38], etc., mostly use convolution of size 3×3. Yet, models with large kernels (e.g., 9×9) are also effective [31] in image classification and localization. Recently, RepLKNet [9] prove that kernel sizes as large as 31×31 can attain very large effective receptive fields (ERFs) [26]. Based on these observations, we propose two modules that contain transformer-like blocks [40] and convolution with large kernels [9], respectively, to build long-range dependency for effective cross-zone feature propagation.

3. Method

In the task of ToF-based depth completion, we aim to infer a depth map $D \in \mathbb{R}^{H\times W\times 1}$ given RGB image $I \in \mathbb{R}^{H\times W\times 3}$ and a depth distribution map $D \in \mathbb{R}^{h\times w\times c_t}$, where $h << H$ and $w << W$ (e.g., $H = 480, W = 640, h = 8, w = 8$). Notice that c_t can vary depending on device setups and implementation details. Here, we follow DELTAR [21] to sample 16 values for each zone based on collected mean and variance from the L5 measurements, i.e., $c_t = 16$.

As discussed earlier, the performance drop in outside-zone areas can be mitigated by leveraging ToF information from zone areas. Thus, we aim to propagate features in zone areas, which contain depth information, into outside-zone areas. In this section, we first introduce the overall network architecture of our CFPNet, which is based on the DELTAR model. Then, we illustrate our newly proposed Direct-Attention-based Propagation Module (DAPM) and Large-Kernel-based Propagation Module (LKPM), both of which are designed for establishing long-range feature propagation for pixels in outside-zone areas from in-zone areas. Additionally, we empirically verify how to combine these two modules effectively to achieve better performance. Lastly, we discuss the loss function used to train the model.

3.1. Network Architecture

We build our CFPNet based on the DELTAR model, and add our DAPM and LKPM for more effective cross-zone feature propagation. As illustrated in Fig. 2, our CFPNet is composed of four branches: RGB feature extraction branch (RGB Branch), depth distribution branch (D Branch), fusion branch, and refinement branch.

RGB Branch. We use a popular convolutional encoder, i.e., EfficientNetB5 [38], to extract RGB features at multiple resolutions, which will be utilized in the following branches.

D Branch. Given ToF measurements, we first sample 16 depth values for each zone. In this way, each zone is considered as one point with sampled depth values as features. Then PointNet-like [32] structure is used to extract depth distribution features. Since the resolution of the zone area is already low, we do not conduct downsampling operations.

Fusion Branch. After acquiring RGB and depth distribution features, respectively, we aim to fuse these features effectively. This branch contains three fusion modules at different levels. Each fusion module is composed of the upsampling module, the D-to-image module [21], our DAPM, our LKPM, and the self-attention module [40]. The upsampling module upsamples the low-resolution feature map and concatenates it with RGB features from higher resolution. After upsampling, D-to-image module would conduct cross-attention from depth distribution features to RGB features in a patchwise way. Concretely, we generate keys and values from depth distribution features and generate queries from RGB features. Thus, the RGB features can dynamically retrieve information from depth distribution features. After fusing these two types of features in the zone area, we utilize DAPM (Sec. 3.2) and LKPM (Sec. 3.3) to propagate features from zone areas to outside-zone areas. Lastly, a self-attention layer is added to blend the feature maps in a global context and make it slightly smoother. Note that apart from the upsampling module, the rest modules are applied alternatively twice.

Refinement Branch. Due to the large resolution of feature maps at the highest level, we do not conduct feature fusion here. Yet, we directly upsample the output from the fusion branch and apply a refinement module to generate the predicted depth map. The refinement module is the same as in DELTAR, which is a mViT structure from Adabins [6]. This module predicts the depth in a weighted sum of adaptively predicted depth bins.

3.2. Direct-Attention-based Propagation Module

We propose to utilize the long-range dependency modeling ability of transformer-like [40] structure to directly propagate features from zone areas to outside-zone areas. Compared with using self-attention [21], our feature propagation avoids feature queries from outside-zone areas where ToF information does not exist.

Consider the toy example in Fig. 3. Given a feature map, we split the pixels into two groups depending on whether they are in or outside zone areas. This results in two sequences of tokens where each pixel is a token. Then, we

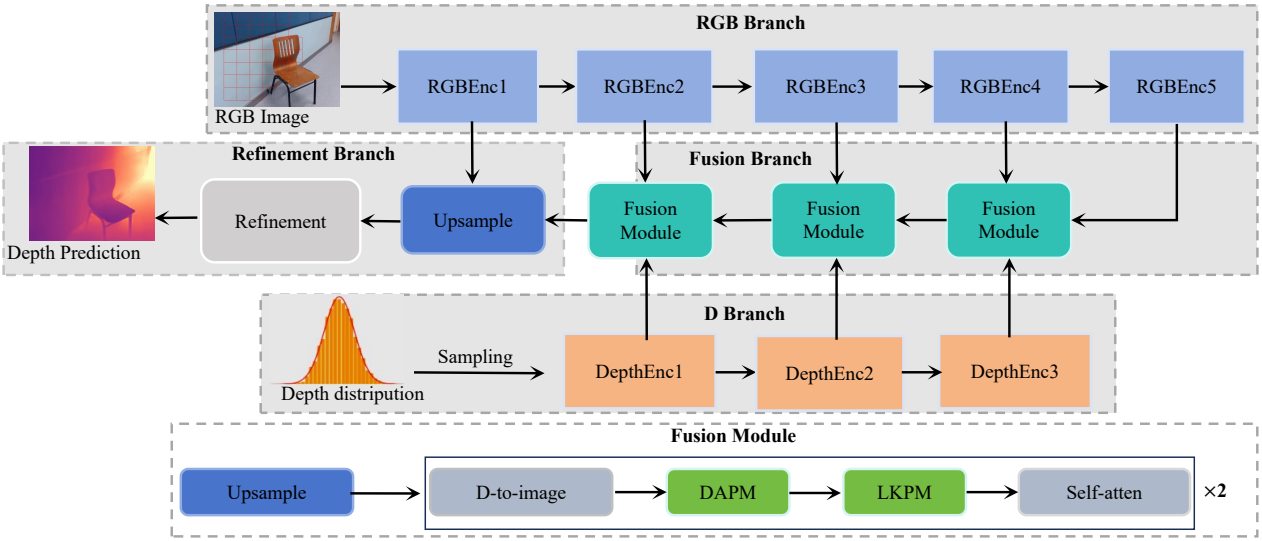


Figure 2. The architecture of our CFPNet. Our CFPNet takes RGB image and depth distribution from ToF sensors as inputs and outputs the depth completion prediction. Our newly proposed DAPM and LKPM are located in the fusion module and allow effective cross-zone feature propagation from zone areas to outside-zone areas.

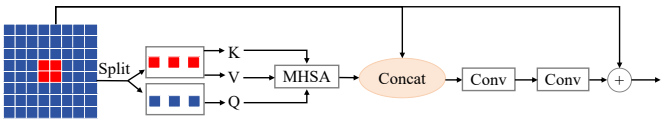


Figure 3. The pipeline of proposed DAPM. We conduct cross attention between pixels from zone areas and outside-zone areas. Additional convolution and skip connection are added to capture local contexts and promote propagation of gradients, respectively.

leverage linear cross attention [18] to propagate features efficiently. Specifically, we generate keys and values from zone areas, and queries from outside-zone areas. Then, Multi-Head Self-Attention (MHSA) [40] is used to calculate the queried values. Consequently, the features can be propagated to outside-zone areas dynamically. Notice that this operation has a global receptive field and is robust to the location of the zone areas in the image plane. Next, we concatenate the output from MHSA with the input and apply a convolution to restore the channel number as the input as in the D-to-image module. Additionally, we employ one more 3×3 convolution layer and skip connection [12] to capture local contexts with CNN in addition to transformer blocks [42] and promote gradient propagation [11], respectively. Furthermore, we validate that the convolution layer and skip connection [12] in the end are necessary in the ablation study (Table 4).

3.3. Large-Kernel-based Propagation Module

Convolution layers can also be adopted to perform feature propagation, and their receptive fields are not influenced by similarities among pixels as in self-attention [5]. In this problem, we propose to use convolution layers of very large kernels (e.g., 31) for cross-zone feature propagation. The usage of large kernels is the large distance from zone areas to outside-zone areas, especially when large portions of zones are missing as in Fig. 5 (a). Furthermore, large kernels can have larger effective receptive fields [26] compared with multiple small kernels [9]. Thus, we introduce a large-kernel-based propagation module (LKPM) based on RepLKNet [9] and ConvNeXt[24].

Fig. 4 (b) depicts our proposed LKPM. It is composed of a depthwise convolution layer of size $s \times s$ (e.g., $s = 31$), LayerNorm [4], two 1×1 convolution layers and the GELU [14] activation unit. Notably, the depthwise convolution with the same number of groups and channels allows the usage of convolution with large kernels [9]. Compared with the ConvNeXt block [24], rather than 7×7 convolution layers, we use convolution layers with much larger kernel sizes. The large kernel allows more long-range feature aggregation between pixels from zone areas and outside-zone areas. Furthermore, we heuristically set s based on the resolution of current feature maps, as we find that changing the kernel size in LKPM adaptively is better than setting $s = 31$ throughout the used blocks [9]. Specifically, given input with a size of 480×640, the fusion happens at three stages where the feature maps are of size 30×40, 60×80, 120×160. Thus, we empirically set $S = \{7, 15, 31\}$ for

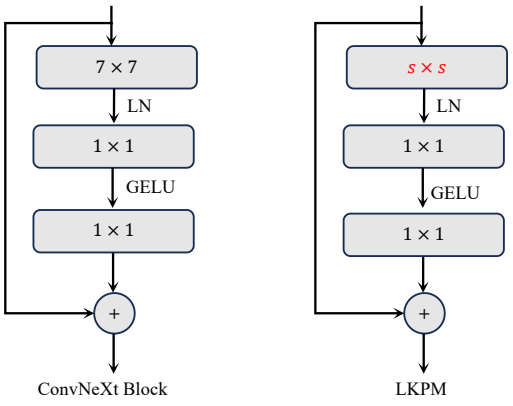
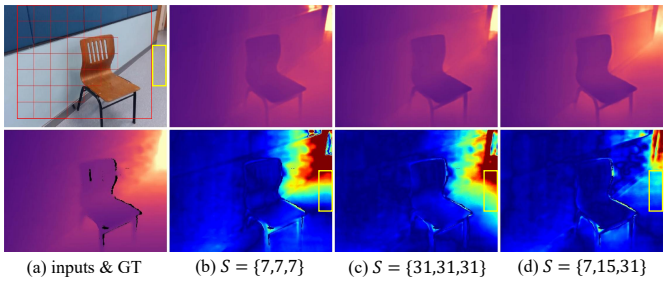


Figure 4. Designs of ConvNeXt Block and our LKPM. Different from ConvNext, we use a $s \times s$ convolution layer where s could be as large as 31 instead of fixing s as 7. Moreover, we adaptively set s based on the resolution of feature maps.



(a) inputs & GT (b) $S = \{7,7,7\}$ (c) $S = \{31,31,31\}$ (d) $S = \{7,15,31\}$

Figure 5. Qualitative results comparing different kernel designs in our LKPM on ZJU-L5 dataset. Compared with using only small or large kernel sizes, our adaptive kernel design achieves the best performance.

LKPM deployed in these stages where S is the collection of s used in three stages. This strategy to set the kernel size to be $\frac{1}{4}$ of the feature map size achieve a good trade-off between accuracy and speed. Additionally, one can always resize the input image to our used resolution to avoid performance drop from changed feature map sizes.

A visual comparison between using ConvNeXt design [24] ($S = \{7, 7, 7\}$), RepLKNet design [9] ($S = \{31, 31, 31\}$), and our kernel design ($S = \{7, 15, 31\}$) are given in Fig. 5. The error maps clearly validate that using merely small kernels [24] is not likely to propagate the features from zone areas to outside-zone areas, especially when large portions of zones are missing. Also, using only large kernels would oversmooth the predicted depth map (faraway areas are estimated nearer). Our kernel design significantly reduces the depth prediction errors in outside-zone areas in the yellow rectangles. More discussions on kernel sizes are in the ablation studies (Sec. 4.5).

3.4. Combining DAPM and LKPM

We have presented two modules, DAPM and LKPM, to propagate features from zone areas to outside-zone areas. DAPM enjoys the power of attention and has a global receptive field [40]. On the other hand, LKPM benefits from the locality of CNN, though we use large kernels. Since CNN-Transformer-like combinations have been proven to be effective in image classification [11], depth completion [45], etc., we propose to combine them for the best performance.

To this end, we test three ways of using DAPM and LKPM jointly. In model A, we apply DAPM first and then apply LKPM. In model B, we apply LKPM first and then apply DAPM. In model C, we apply LKPM and DAPM in a parallel way and fuse the two results with a summation. The rest of models A, B, and C are the same as in our baseline, which is the DELTAR[21] model. Table 2 validates that model A would perform the best on the ZJU-L5 dataset [21].

Furthermore, combining them would lead to more robust performance in various FOV difference setups. For the small FOV difference case as in the ZJU-L5 dataset, LKPM could be more effective than DAPM (see Table 2). However, in a large FOV difference scenario, DAPM can be more effective (see Table 5) thanks to the global perception ability from attention mechanism [40]. Consequently, combining them is necessary to handle different FOV setups in real-world applications.

3.5. Loss Function

In order to train a depth completion network, commonly chosen loss functions can be L1 loss, L2 loss, BerHu loss [28], Scale-Invariant (SI) loss [10] , etc. We follow previous works [6], and use a scaled version of SI loss for each sample:

$$L(d, \tilde{d}) = \alpha \sqrt{\frac{1}{N}\sum_{i=1}^{N} g_i^2 - \frac{\lambda}{N^2}\left(\sum_{i=1}^{N} \tilde{g}_i\right)^2}$$

where d and $\tilde{d}$ is the ground truth depth and predicted depth for the sample, N is the number of valid pixels, $g_i = \log \tilde{d}_i - \log d_i$. α and λ are set to 10 and 0.85.

4. Experiments

In this section, we first describe datasets and evaluation metrics. Then, quantitative and qualitative results are provided to validate the remarkable performance of our CFP-Net. Lastly, abundant ablation studies are given to verify the effectiveness of our proposed DAPM and LKPM.

4.1. Datasets and Evaluation Metrics

NYU-sim NYU-sim is a simulated dataset from NYUDepth-V2 [34], containing RGB images, simulated ToF measurements, and groundtruth depth maps. The training and test sets contain 24k and 654 samples, the same as previous works [20]. As for simulating the ToF signal, a Gaussian distribution is used to fit the depth histogram for each zone. Pixels whose depth is farther than the range the ToF sensor can measure are excluded during the histogram statistics.

ZJU-L5. DELTAR [21] provided a real-world test set, ZJU-L5, containing 527 samples from 15 different scenes.

Metrics. We follow previous works [6] to report standard metrics for depth prediction, including mean absolute relative error (REL), root mean squared error (RMSE), average $(\log_{10})$ error, threshold accuracy (δ_i).

4.2. Implementation Details

We implement our CFPNet in Pytorch [30]. For training, we use AdamW optimizer [25] with one-cycle policy [35] where maximum learning rate is of 3×10^{-4}. We train our CFPNet on four NVIDIA RTX 2080Ti GPUs with a batchsize of 16 on each GPU for 30 epochs, which takes around 12 hours.

As for training and test protocol, we follow previous works to first train on the NYU-sim dataset and select the model with the best performance on the test set of the NYU-sim dataset. Then, we report the performance of this model on the ZJU-L5 dataset [21].

4.3. Quantitative Comparisons

Table 1 lists the performance of different methods including BTS [20], Adabins [6], NLSPN [29], PENet [16], DELTAR [21], our reproduced DELTAR* [21] using their most recent codebase, and our CFPNet, on ZJU-L5 dataset [21]. The first five lines of results are quoted from DELATR [21]. Overall, Table 1 shows that specifically designed lightweight-ToF-based depth completion (DC) methods, i.e., DELTAR [21] and our CFPNet, are more effective than previous depth estimation (DE) and depth completion (DC) methods.

Note that DELTAR* is our baseline and has similar performance compared with DELTAR, except on the δ_1 and RMSE metric. Thus, we mainly compare our CFPNet with DELTAR* regarding quantitative and qualitative results. Our CFPNet outperforms DELTAR* by a notable margin. For example, our CFPNet can decrease the REL by 0.024 and increase the δ_1 by 0.021. This validates that our CFPNet can achieve SOTA performance in lightweight-ToF-based depth completion.

Methods	type	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	REL $\downarrow$	RMSE $\downarrow$	$\log_{10} \downarrow$
BTS [20]	DE	0.739	0.914	0.964	0.174	0.523	0.079
AdaBins [6]	DE	0.770	0.926	0.970	0.160	0.494	0.073
NLSPN [29]	DC	0.583	0.784	0.892	0.345	0.653	0.120
PENet [16]	DC	0.807	0.914	0.954	0.161	0.498	0.065
DELTAR [21]	DC	0.853	0.941	**0.972**	0.123	0.436	0.051
DELTAR* [21]	DC	0.862	0.943	0.970	0.127	0.461	0.058
CFPNet	DC	**0.883**	**0.949**	**0.972**	**0.103**	**0.431**	**0.047**

Table 1. Quantitative comparisons on the ZJU-L5 dataset. Compared with the most recent lightweight-ToF-based method DELTAR* (our reproduced baseline), we obtain large improvement on all metrics. Best performance is **bolded**.

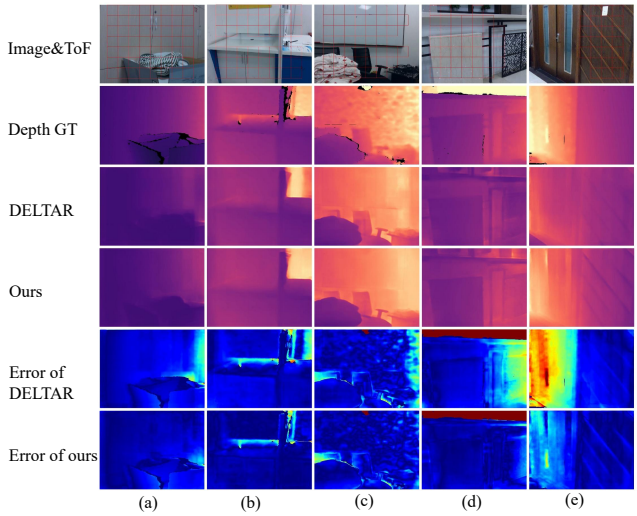


Figure 6. Qualitative comparisons between DELATR and our CFPNet on the ZJU-L5 dataset. Brighter color in error maps refers to larger errors. Errors where depth gt are missing are set to zero for visualization. Clearly, in outside-zone areas, errors are greatly reduced, and 3D properties such as planar smoothness are maintained.

4.4. Qualitative Comparisons

To validate that our CFPNet can improve depth completion performance in outside-zone areas, we provide visual comparisons between DELTAR* and our CFPNet in Fig. 6.

Obviously, our CFPNet can maintain the scene's structures better and restore more accurate depth in outside-zone regions. For example, in Fig. 6 (a), (b) and (c), the prediction on the right side of the image preserves the continuity of the wall. In contrast, DELTAR* would generate quite different predictions for in-zone pixels and outside-zone pixels of the wall. Furthermore, in more challenging scenarios, e.g., Fig. 6 (e), even if the majority of the ToF signal is lost on the left, our CFPNet can still recover the depth of the scene with higher precision. These visualizations validate that our CFPNet can effectively mitigate the performance drop in outside-zone areas. More qualitative results are provided in the supplementary material.

4.5. Ablation studies

To understand the impact of each proposed module, we conduct thorough ablation studies in this section. We first verify the improvement each module incurs and then examine the effectiveness of our designs in the module. Lastly, we discuss the differences between these two modules regarding application scenarios.

Table 2 shows that LKPM and DAPM can increase δ_1 by 0.019 and 0.010, respectively. They can also reduce RMSE by 0.022 and 0.014. We also test different combinations of DAPM and LKPM. Model A performs DAPM first and then LKPM, while model B is vice-versa. Model C conducts these two modules parallelly and sums the outputs from them. Table 2 suggests that model A can achieve the best performance. Additionally, we propose model D, which is the same as model A, except we remove the self-attention layer [40]. The better performance and fewer parameters of model D compared with DELTAR* indicates that our proposed DAPM and LKPM are more effective and efficient than self-attention [40] in this task. Still, applying self-attention [40] as in model A leads to even better performance.

Methods	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	REL $\downarrow$	RMSE $\downarrow$	$\log_{10} \downarrow$	Params(M)
DELTAR* [21]	0.862	0.943	0.970	0.127	0.461	0.058	18.545(±0.000)
+LKPM	0.881	0.947	0.971	0.104	0.439	0.049	18.996(+0.451)
+DAPM	0.872	0.945	0.970	0.112	0.447	0.050	19.837(+1.292)
Model C	0.875	0.946	0.971	0.107	0.436	0.049	20.288(+1.743)
Model B	0.877	0.943	0.969	**0.102**	0.442	0.049	20.288(+1.743)
Model A	**0.883**	**0.949**	**0.972**	0.103	**0.431**	**0.047**	20.288(+1.743)
Model D	0.874	0.946	0.973	0.111	0.435	0.050	17.285(−1.260)

Table 2. Ablation studies. We first reproduce DELTAR* as our baseline and add our DAPM or LKPM to investigate their individual effects. Then, we test different ways (Model A, B, C, D) to combine them. Both LKPM and DAPM can increase the overall performance, and Model A achieves the best performance.

Kernel Design in LKPM. As mentioned above, larger kernels allow a larger effective receptive field [9], thus suitable for cross-zone feature propagation. Yet, we still need to investigate how large the kernel should be. To this end, we show the results of using different kernel sizes in LKPM in Table 3. Overall, the performance can be improved with any configuration of LKPM, and setting the kernel sizes adaptively based on the resolution of feature maps (the fifth row) yields the best performance. However, using kernels with fixed size (7 or 31) or halving the kernel size in our design (the fourth row) can only lead to limited performance gain. Thus, these comparisons validate the effectiveness of our adaptive kernel design. Furthermore, we try adding an additional parallel convolution layer of 5×5 (the last row) as RePLKNet [9]. Nevertheless, we find such a design is not useful in this task.

Skip Connection and Convolution in DAPM. As Fig. 3 shows, DAPM includes cross-zone attention (CA) [40], the final skip connection (SC), and the final convolution (Conv). We validate the necessity of each component in DAPM. Table 4 lists the performance of different models by injecting proposed components sequentially. Obviously, cross attention [40] improves the performance. Furthermore, adding the last skip connection and convolution yields the best performance in terms of overall metrics.

Methods	Kernel sizes	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	REL $\downarrow$	RMSE $\downarrow$	$\log_{10} \downarrow$
baseline	-	0.862	0.943	0.970	0.127	0.461	0.058
+LKPM	7,7,7	0.870	0.945	0.969	0.116	0.448	0.053
+LKPM	31,31,31	0.870	0.941	0.966	0.106	0.452	0.051
+LKPM	3,7,15	0.867	0.943	0.969	0.110	0.458	0.052
+LKPM	7,15,31	**0.881**	**0.947**	**0.971**	**0.104**	**0.439**	**0.049**
+LKPM	{7,5},{15,5},{31,5}	0.878	0.945	0.969	**0.104**	0.446	0.049

Table 3. Ablation studies of kernel designs in LKPM. We investigate different combinations of kernel sizes used in LKPM in three levels. Using only small or large kernels could bring limited boost while our adaptive kernel design attains the best results.

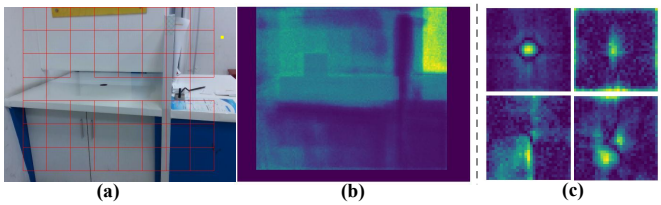


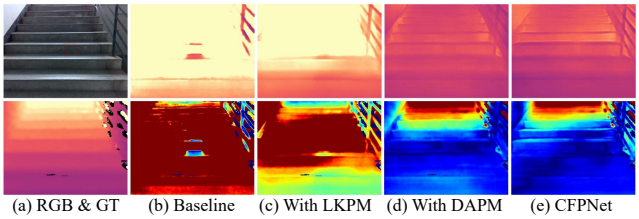
Figure 7. Visualization of learned attention map and kernel weights from our DAPM and LKPM. These results explain why DAPM and LKPM are capable of effectively propagating features from zone area to outside-zone area.

Methods	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	REL $\downarrow$	RMSE $\downarrow$	$\log_{10} \downarrow$
baseline	0.862	0.943	**0.970**	0.127	0.461	0.058
+CA	0.871	**0.945**	0.968	0.111	0.458	0.053
+CA+SC	**0.873**	**0.945**	0.967	**0.107**	0.459	0.052
+CA+SC+Conv	0.872	**0.945**	**0.970**	0.112	**0.447**	**0.050**

Table 4. Ablation studies on components in DAPM. Though using cross-attention only could already bring some improvements, adding skip connection and convolution after attention increases the prediction accuracy steadily.

Learned Attention Map and Large Kernel Weights. In Fig.7, we show the attention map (b) for the selected yellow pixel in (a). We not only obtain higher attention weight in more relevant pixels but also avoid feature aggregation from outside-zone areas where ToF feature does not exist. In Fig. 7 (c), we show some of the learned large kernel weights which indicate that our large kernel can indeed propagate information in a broad context. These visualizations verify that our proposed DAPM and LKPM indeed allow effective feature propagation,

Discussion on DAPM and LKPM. DAPM and LKPM can both improve the overall depth completion performance, and LKPM can lead to larger improvements. We argue that the small portion of outside-zone areas in the image limits the advantage of DAPM, which is designed to handle any FOV difference. To this end, we provide evi-



(a) RGB & GT (b) Baseline (c) With LKPM (d) With DAPM (e) CFPNet

Figure 8. Visual comparisons under the condition where the ToF signal is of resolution 2×2 and only one zone is valid. This extreme case validates the superior performance of DAPM against LKPM when a large FOV difference between sensors exists.

dence that DAPM can be more useful in cases of a large FOV gap between the camera and the ToF sensor.

Methods	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	REL $\downarrow$	RMSE $\downarrow$	Time(ms) $\downarrow$
baseline	0.578	0.793	0.892	0.398	0.780	37.299 ($\pm$0.000)
+LKPM	0.607	0.833	0.919	0.321	0.651	40.366(+3.067)
+DAPM	**0.701**	0.881	0.945	0.219	0.551	47.285(+9.986)
+DAPM+LKPM	0.687	**0.891**	**0.952**	**0.213**	**0.540**	50.846(+13.547)
+LKPM61	0.618	0.828	0.913	0.326	0.629	45.271(+7.972)
+LKPM101	0.638	0.845	0.921	0.302	0.612	58.821(+21.522)

Table 5. Experimental results on ZJU-L5 where we simulate the condition that ToF is of resolution 2×2, i.e., large FOV difference between cameras and ToF sensors. In this scenario, DAPM can improve the performance more than LKPM.

The FOV difference between the camera and the ToF sensor in ZJU-L5 dataset [21] is small ($45° \times 45°$ versus $55° \times 43°$). Thus, when we conduct feature fusion at the level where the feature map is of size 120×160, ToF can cover regions of size 128×128, and each zone corresponds to 16×16. This is enough to cover the color image vertically but not horizontally. Assuming that the 128×128 area lies in the center of the color image, the horizontal distance from the left border of the image to the left border of the zone area is $(160 - 128)/2 = 16$ pixels. This is why the kernel size of 31 in LKPM is enough to handle outside-zone areas even if some zones are missing. However, if we simulate that the zone area is of resolution 2×2, which is a case of large FOV difference, this horizontal distance becomes $(160 - 2 \times 16)/2 = 64$ pixels. Then LKPM should bring less performance gain than DAPM in this situation due to its limited perceptive field. This is validated by results from Table 5. More concretely, using DAPM yields more significant performance boost than using LKPM (REL is reduced by 45.0 % and 19.3%, respectively). Moreover, combining them as in model A still enjoys the advantage DAPM provides. Additionally, since kernel sizes in LKPM are specifically designed for 8×8 case, we provide two more results (LKPM61 and LKPM101) that use a maximum kernel size of 61 and 101. As the kernel size in LKPM increases, the latency increases significantly while the performance boost is limited.

A visual comparison is also provided in Fig. 8. In this extreme case where we simulate that the ToF is of resolution 2×2 and only one zone is valid, the baseline method or using only the LKPM cannot give a reasonable prediction. However, the potential of DAPM is greatly excavated in this scenario. Moreover, combining them can preserve the benefits DAPM brings. More visualization results are in the supplementary material.

From Table 2 and Table 5, we find: (1) LKPM can be efficient (only increase the number of parameters by 0.451M) and effective on ZJU-L5 where the FOV difference is small. (2) DAPM can be more useful when the FOV difference is large (e.g., a more lightweight ToF sensor VL53L3CX [1] has a FOV of 25°), though at the cost of more parameters (1.292M). Still, combining them leads to more robust performance in different conditions.

5. Conclusion and Future Work

This paper proposes CFPNet containing two novel modules to tackle the FOV difference in RGB-guided lightweight-ToF-based depth completion by effective cross-zone feature propagation. The direct-attention-based feature propagation module attains direct feature acquisition via the attention mechanism. The large-kernel-based feature propagation module utilizes convolution layers of large kernels to achieve a large receptive field. Besides, we thoroughly discuss their differences and application scenarios. Extensive experiments demonstrate that our CFPNet achieves SOTA performance on the public dataset.

However, due to the limited sensing range and the low resolution depth of lightweight ToF sensors, depth completion performance in regions that are farther than the sensing range and areas that are near object boundaries is still poor.

As for future work, though we have gained a significant performance boost, this work is about ToF-based depth completion. There exists large room for ToF-based applications, including SLAM [23], Nerf [3], AR [37], etc. It is promising to utilize the low cost and energy property of L5 to conduct more meaningful downstream tasks.

Acknowledgments This work was supported in part by Shenzhen Science and Technology Program under Grant JCYJ20220818103006012 and Longgang District Key Laboratory of Intelligent Digital Economy Security.

References

[1] Stmicroelectronics: Time-of-flight (tof) ranging sensor with multi target detection, . 8

[2] Stmicroelectronics: Time-of-flight (tof) 8x8 multizone ranging sensor with wide field of view, . 1

[3] Benjamin Attal, Eliot Laidlaw, Aaron Gokaslan, Changil Kim, Christian Richardt, James Tompkin, and Matthew O'Toole. Törf: Time-of-flight radiance fields for dynamic

scene view synthesis. *Advances in neural information processing systems*, 34:26289–26301, 2021. 8

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[5] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3286–3295, 2019. 2, 4

[6] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021. 3, 5, 6

[7] 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

[8] 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*, pages 10615–10622, 2020. 2

[9] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11963–11975, 2022. 2, 3, 4, 5, 7

[10] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 5

[11] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12175–12185, 2022. 3, 4, 5

[12] 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*, pages 770–778, 2016. 3, 4

[13] Lingzhi He, Hongguang Zhu, Feng Li, Huihui Bai, Runmin Cong, Chunjie Zhang, Chunyu Lin, Meiqin Liu, and Yao Zhao. Towards fast and accurate real-world depth super-resolution: Benchmark dataset and baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9229–9238, 2021. 1, 2

[14] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 4

[15] Junjie Hu, Chenyu Bao, Mete Ozay, Chenyou Fan, Qing Gao, Honghai Liu, and Tin Lun Lam. Deep depth completion from extremely sparse data: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[16] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Penet: Towards precise and efficient image guided depth completion. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13656–13662. IEEE, 2021. 2, 6

[17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 3

[18] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020. 4

[19] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 3

[20] 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. 6

[21] Yijin Li, Xinyang Liu, Wenqi Dong, Han Zhou, Hujun Bao, Guofeng Zhang, Yinda Zhang, and Zhaopeng Cui. Deltar: Depth estimation from a light-weight tof sensor and rgb image. In *European Conference on Computer Vision*, pages 619–636. Springer, 2022. 1, 2, 3, 5, 6, 7, 8

[22] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*, pages 1–18. Springer, 2022. 3

[23] Xinyang Liu, Yijin Li, Yanbin Teng, Hujun Bao, Guofeng Zhang, Yinda Zhang, and Zhaopeng Cui. Multi-modal neural radiance field for monocular dense slam with a light-weight tof sensor. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–11, 2023. 1, 8

[24] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 4, 5

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

[26] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29, 2016. 2, 3, 4

[27] Fangchang Ma, Luca Carlone, Ulas Ayaz, and Sertac Karaman. Sparse depth sensing for resource-constrained robots. *The International Journal of Robotics Research*, 38(8):935–980, 2019. 1

[28] Art B Owen. A robust hybrid of lasso and ridge regression. *Contemporary Mathematics*, 443(7):59–72, 2007. 5

[29] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 120–136. Springer, 2020. 2, 3, 6

[30] 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. *Advances in neural information processing systems*, 32, 2019. 6

[31] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–improve semantic segmentation by global convolutional network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4353–4361, 2017. 3

[32] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 3

[33] Xin Qiao, Matteo Poggi, Pengchao Deng, Hao Wei, Chenyang Ge, and Stefano Mattoccia. Rgb guided tof imaging system: A survey of deep learning-based methods. *International Journal of Computer Vision*, pages 1–38, 2024. 1

[34] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012. 6

[35] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, pages 369–386. SPIE, 2019. 6

[36] Wenxiu Sun, Qingpeng Zhu, Chongyi Li, Ruicheng Feng, Shangchen Zhou, Jun Jiang, Qingyu Yang, Chen Change Loy, Jinwei Gu, Dewang Hou, et al. Mipi 2022 challenge on rgb+ tof depth completion: Dataset and report. In *European Conference on Computer Vision*, pages 3–20. Springer, 2022. 1

[37] Zhanghao Sun, Wei Ye, Jinhui Xiong, Gyeongmin Choe, Jialiang Wang, Shuochen Su, and Rakesh Ranjan. Consistent direct time-of-flight video depth super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5075–5085, 2023. 8

[38] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 3

[39] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 international conference on 3D Vision (3DV)*, pages 11–20. IEEE, 2017. 2

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 3, 4, 5, 7

[41] Chen Wang, Jiadai Sun, Lina Liu, Chenming Wu, Zhelun Shen, Dayan Wu, Yuchao Dai, and Liangjun Zhang. Digging into depth priors for outdoor neural radiance fields. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 1221–1230, 2023. 1

[42] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22–31, 2021. 4

[43] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 3

[44] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5728–5739, 2022. 3

[45] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 18527–18536, 2023. 3, 5