GuideFormer: Transformers for Image Guided Depth Completion

Kyeongha Rho[1*] Jinsung Ha[2*] Youngjung Kim[1†]
[1]Agency for Defense Development (ADD), Daejeon, Korea, [2]LUXROBO, Seoul, Korea
{khrho325, read12300}@add.re.kr, jinsung@luxrobo.com

Abstract

Depth completion has been widely studied to predict a dense depth image from its sparse measurement and a single color image. However, most state-of-the-art methods rely on static convolutional neural networks (CNNs) which are not flexible enough for capturing the dynamic nature of input contexts. In this paper, we propose GuideFormer, a fully transformer-based architecture for dense depth completion. We first process sparse depth and color guidance images with separate transformer branches to extract hierarchical and complementary token representations. Each branch consists of a stack of self-attention blocks and has key design features to make our model suitable for the task. We also devise an effective token fusion method based on guided-attention mechanism. It explicitly models information flow between the two branches and captures inter-modal dependencies that cannot be obtained from depth or color image alone. These properties allow GuideFormer to enjoy various visual dependencies and recover precise depth values while preserving fine details. We evaluate GuideFormer on the KITTI dataset containing real-world driving scenes and provide extensive ablation studies. Experimental results demonstrate that our approach significantly outperforms the state-of-the-art methods.

1. Introduction

Guided depth completion is the task of converting sparse depth observations to dense depth maps with the corresponding color image. This task has been drawing more and more research attention, thanks to its wide range of applications in the computer vision field, e.g., 3D scene mapping [26] and 3D object detection [22] for robotic perception and autonomous driving. However, commercial depth-sensing cameras (e.g. LiDAR sensors) suffer from their inherent drawbacks, including specular surfaces, quantization, occlusion, and noise. These properties make depth completion a challenging problem. To tackle depth completion, a variety of methods, mostly based on deep convolutional neural networks (CNNs), have been proposed. Early works deal only with sparse depth input to estimate dense depth via sparsity invariant CNN [30] or auxiliary vision tasks [13, 18]. Recent works have shown a great success in using multi-modal information, including color images [10, 15, 20, 31, 40] and surface normal [23, 36]. These methods have achieved state-of-the-art performances over the conventional methods using only depth input. Nevertheless, CNN-based methods show fundamental limitations by their basic building block, i.e., the static convolutional layers. The interaction between convolution kernels and inputs is content-independent. Applying the same kernels to any region might not be flexible for adapting diverse and disparate spatial contexts. Furthermore, the convolution is not effective for modeling long-range dependencies. Recently, there are several attempts to develop content-adaptive CNNs for depth completion. ACMNet [40] constructs a graph propagation based network, and learns graph affinities by considering color and depth information. GuideNet [29] proposes a guided-convolution module, dynamically predicting content-adaptive convolution kernels from color images. However, these methods still use CNNs as the backbone, which leaves room for further improvement.

In this paper, we propose GuideFormer, a dual-branch architecture that takes full advantage of attention mechanisms for depth completion. Each branch, consisting of a stack of modified self-attention blocks, embeds hierarchical and complementary tokens from sparse depth and color images. This allows our model to better capture adaptive intra-modal dependencies through the whole completion process. An effective token fusion method, called guided-attention module (GAM), is also introduced by extending the standard self-attention mechanism. It models explicit information exchange among the two complementary branches and captures inter-modal dependencies from the learned cross-modal similarity. To the best of our knowledge, we are the first to apply fully transformer-based architecture for depth completion task. Experimental results on the KITTI benchmark [30] demonstrate the effectiveness of the proposed method, which outperforms the state-of-the-art methods.

Our contributions can be summarized as follows:

This research was supported by the Next Generation R&D Program (912706601), funded by the Agency for Defense Development (ADD).
* Equal contribution, † Corresponding author

- We propose a dual-branch and fully transformer-based architecture for depth completion task. It learns input-adaptive token representations from the sparse depth and color guidance images, respectively through the whole completion process. This allows us to reason about diverse intra-modal dependencies better than the existing static CNN-based methods.
- We introduce a guided-attention module (GAM) by extending the standard self-attention mechanism. It captures inter-modal dependencies and models information flow between depth and color tokens. We show that GAM is a more powerful method of fusing multi-modal information, compared to simple concatenation [10] or guided-convolution module [29].
- Our method outperforms the recent state-of-the-art approaches on the KITTI benchmark. We also provide extensive ablation studies with both quantitative and qualitative experimental analyses.

2. Related Works

2.1. CNN-based Depth Completion Methods

Depth completion methods are broadly divided into two categories: the first one takes only sparse depth measurements as input [13, 18, 30], and the second one additionally utilizes synchronized color images as guidance information [3, 8, 10, 15, 19, 20, 23, 29, 31, 37, 40]. By incorporating additional structural guidance, the second category significantly improves depth completion performance. Typically, Ma *et al.* [19] concatenate sparse depth map and its corresponding color image, and then feed them into CNNs. The works of [3, 20, 37] initially estimate dense but coarse depth image using CNNs, and then refine it with spatial propagation network (SPN) by learning local or non-local affinities. Recent methods [10, 15, 23, 29, 40] construct a dual-branch CNN to separately extract features from sparse depth and color images, and fuse them for final depth predictions. DeepLiDAR [23] and Xu *et al.* [36] impose geometric constraints between depth and intermediate surface normal to regularize the completion process and improve the robustness. However, the aforementioned methods are based on static CNN kernels that stay unchanged for input.

2.2. Content-adaptive CNNs for Depth Completion

To overcome the limitation of earlier CNN-based methods, several works [11, 29, 40] develop content-adaptive CNNs for depth completion. Huang *et al.* [11] use self-attention mechanism, implemented by gated-convolution, on each convolutional layer. They further propose a boundary consistency to produce a dense depth of the clear structure. ACMNet [40] defines a graph using k-nearest neighbor, and adaptively aggregates CNN features to encode contextual information. In the decoder, it uses a symmetric gated fusion strategy to combine depth and color information. GuideNet [29] introduce a guided-convolution module that dynamically predicts content-adaptive convolution kernels for multi-modal feature fusion.

2.3. Vision Transformer

Transformer has received considerable attention in the computer vision community since its great success in natural language processing (NLP) [32]. Recently, it has shown outstanding performance on many computer vision tasks such as image classification [6, 7, 16, 33, 34, 39], object detection [1, 6, 16, 24, 33], and semantic segmentation [28, 35, 41]. Several works [25, 38] apply transformers to monocular depth estimation, closely related to depth completion task. DPT [25] uses ViT [7] as a backbone to encode an image into token representations and reassembles them into image-like representations at multiple stages. These tokens are then progressively combined by the CNN decoder to recover depth from a single color image. TransDepth [38] extracts CNN features and further processes them with transformers, and applies attention gate decoder with independent channel- and spatial-wise attention. Unlike these methods [25, 38], we propose a dual-branch and fully transformer-based architecture to extract and fuse multi-modal token representations for depth completion.

3. Method

We elaborate GuideFormer with a detailed network structure and design choices for depth completion. It consists of three main components: (1) fully transformer-based encoder-decoder, (2) guided-attention module (GAM) to enrich the representations from (1) using a guided-attention mechanism, and (3) the final depth fusion module. The high-level architecture is illustrated in Figure 1.

3.1. Self- and Guided-attention Mechanisms

Before presenting GuideFormer, we first review the window-based self-attention mechanism in the Swin Transformer [16], and extend it to the guided-attention one. Since global self-attention [7] is computationally unaffordable for large image sizes, the Swin Transformer performs the self-attention within non-overlapping shifted local windows. Given a local window feature $\mathbf{F} \in \mathbb{R}^{W^2 \times C}$, the query, key, and value matrices are computed as follows:

$$\mathbf{Q} = \mathbf{F}\mathbf{W}^q,\ \mathbf{K} = \mathbf{F}\mathbf{W}^k,\ \mathbf{V} = \mathbf{F}\mathbf{W}^v\,, \tag{1}$$

where W and C denote the window size and the number of input channels, respectively. $\mathbf{W}^q$, $\mathbf{W}^k$, and $\mathbf{W}^v \in \mathbb{R}^{C \times d}$ are d-dimensional projection matrices which are shared across local windows. The self-attention is computed as follows:

$$\text{S-Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}} + \mathbf{B})\mathbf{V}\,, \tag{2}$$

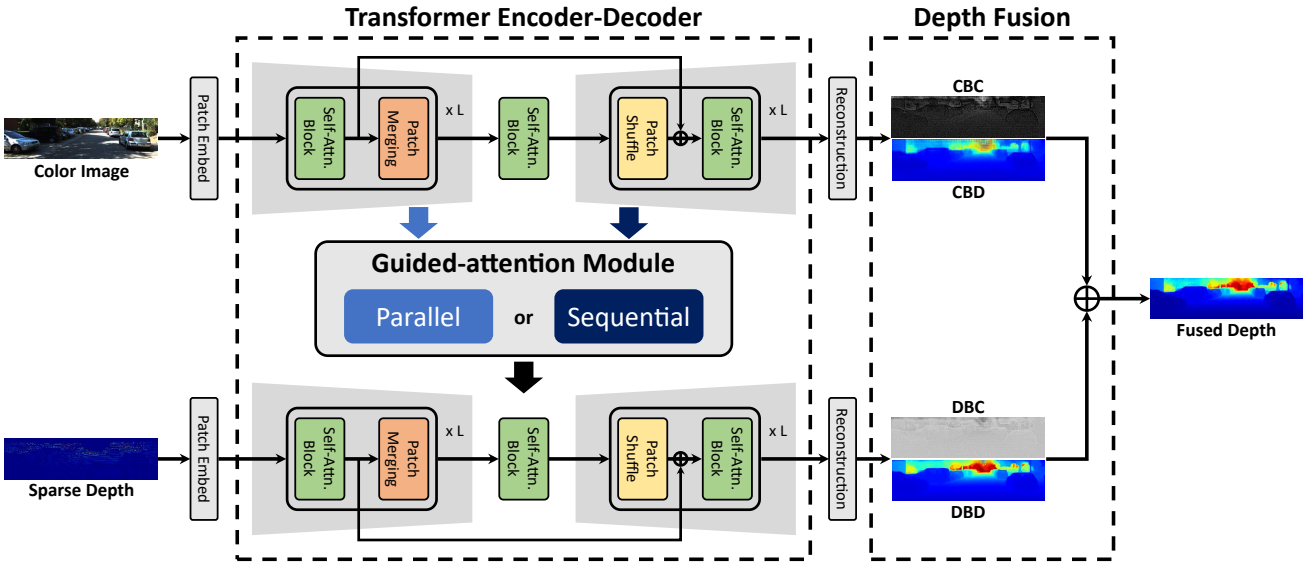


Figure 1. The overall architecture of the proposed GuideFormer. CBC, CBD, DBC, and DBD denote color-branch confidence map, color-branch depth map, depth-branch confidence map, and depth-branch depth map, respectively. The dual-branch transformer architecture extracts the token representation of multi-modal entities. The information of the tokens is fused with the guided-attention module. The output of the decoder is further processed by the reconstruction block to produce the dense depth maps with their corresponding confidence maps. Then the final complete depth map is obtained by the depth fusion module.

where $\mathbf{B}$ is the relative positional bias [16]. Equation 2 is repeated for h-times in parallel, and the outputs are concatenated to form multi-head self-attention (MSA). Note that unlike static convolutions in previous methods [10, 15, 23, 31], attention weights are dynamically calculated based on the input contents (i.e. self-similarity).

Let us consider two groups of features from input and guidance images, where the latter guides the attention learning for GuideFormer. We will use the subscripts I and G to denote the intermediate representations from each image. The guided-attention mechanism is defined as follows:

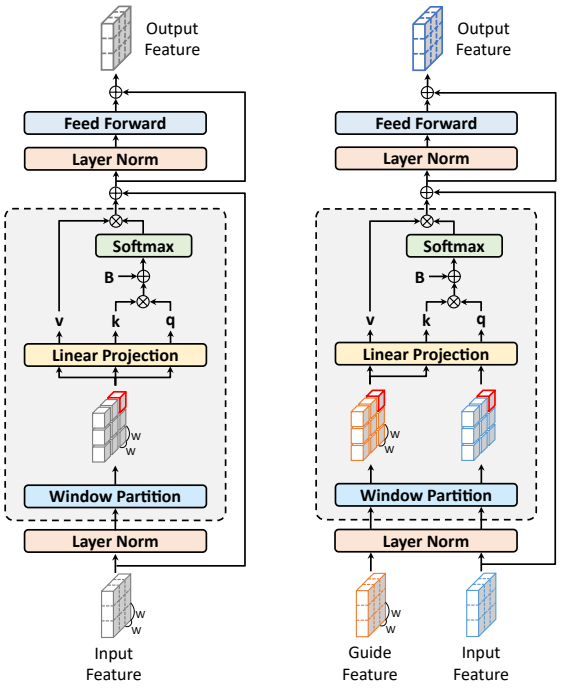
$$\text{G-Attn}_{G\rightarrow I}(\mathbf{Q}_I, \mathbf{K}_G, \mathbf{V}_G) = \text{softmax}(\frac{\mathbf{Q}_I\mathbf{K}_G^{\mathbf{T}}}{\sqrt{d}} + \mathbf{B})\mathbf{V}_G\,, \quad (3)$$

where we assume that $\mathbf{F}_I$ and $\mathbf{F}_G$ have the same spatial dimension. The form of multi-head guided-attention (MGA) is straightforward. Conceptually, the self-attention (Equation 2) takes weighted-average on $\mathbf{V}_I$ with respect to the similarity learned from $\mathbf{Q}_I$ and $\mathbf{K}_I$. In contrast, the guided-attention (Equation 3) attends $\mathbf{V}_G$ with the cross-modal similarity between $\mathbf{Q}_I$ and $\mathbf{K}_G$. By adding the attended $\mathbf{V}_G$ to $\mathbf{F}_I$, we can explicitly transfer information of color branch to depth branch by considering cross-modal dependencies.

3.2. Fully Transformer-based Encoder-Decoder

We introduce a fully transformer-based encoder-decoder that maps a sequence of tokens to a dense depth image. We basically follow the Swin Transformer [16], but make several key modifications for dense depth completion. A dual-branch encoder-decoder architecture is designed to thoroughly exploit hierarchical and complementary information from color and depth images as shown in Figure 1. Two branches have identical architecture. Sparse depth (or color) image is first split into tokens through an embedding layer. Most vision transformers implement this embedding as a single large-stride and -kernel convolution (from 4 to 16). However, we found that such convolution makes transformers difficult to optimize, and degrades the final performance for dense depth completion. We instead adopt a shallow feature extractor with two 3×3 residual blocks [9]. The stack of small convolutional layers is a good choice in the early processing pipeline, leading to more stable training and better performance.

The resulting shallow features are then flattened, and pass through three encoder stages. Each stage contains two self-attention blocks (Figure 2a) and one down-sampling layer (i.e. patch merging in [16]). A feed-forward operation in Figure 2 is composed of two linear layers with a depth-wise convolution (DWC) layer between them. We add the DWC layer to better model the local context in self-attention blocks, which is essential for preserving edges and fine-grained details. After the encoding stage, we apply consecutive self-attention blocks to extract bottleneck features without down-sampling. The decoder has a symmetric structure to its encoder counterpart, and progressively reconstructs high-resolution feature maps from the bottleneck. Each decoder stage contains one up-sampling layer and two self-attention blocks. We realize the inverse oper-



(a) Self-Attention (b) Guided-Attention

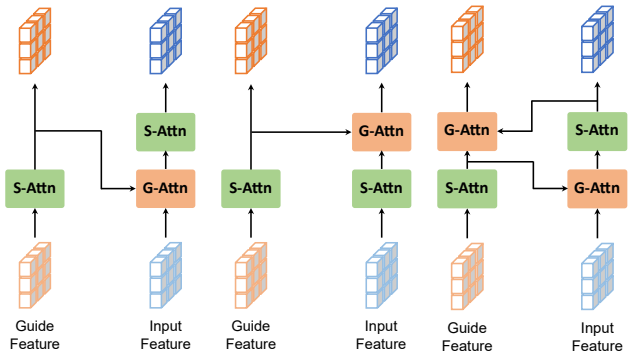
Figure 2. Two basic attention blocks of GuideFormer. $\otimes$ denotes matrix multiplication and $\oplus$ denotes addition. The self-attention block takes an input feature and outputs the feature attended by itself. On the other hand, the guided-attention block takes the input feature along with the guidance feature and outputs the attended feature guided by the guidance feature.

ation of patch merging [16] with a patch shuffle layer [27], followed by a linear layer to match the channel dimension. We add skip connections between encoder stages and their symmetric decoder stages. Comparing the architecture in [25] for dense prediction, our model captures global as well as local dependencies, and maintains the capacity to dynamically learn kernel weights both in the encoder and decoder. In the following, we explicitly model information flow between sparse depth and color branches with the proposed guided attention mechanism.

3.3. Guided-attention Module

Due to the difference in semantic information between color and sparse depth images, the features processed by each branch of transformer encoder-decoder are complementary to each other. Contrast to simple concatenation [10] or guided-convolution module [29, 40], we utilize GAM for fusing information from multi-modal inputs. Using cross-modal similarities between two features as attention weights, guided-attention provides a more straightforward way to integrate multi-modal features.

Based on the guided-attention mechanism mentioned in Section 3.1, we form a guided-attention block as shown in



(a) Pre-Guide (b) Post-Guide (c) Bidirectional-Guide

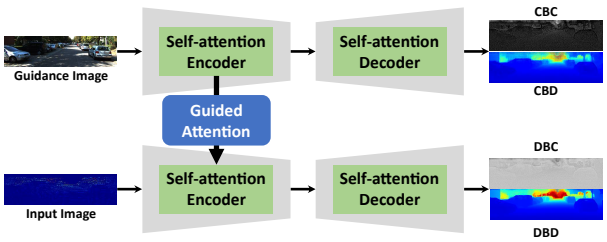
Figure 3. Flowcharts of three GAM variants for depth completion. (a) Pre-guide GAM employs guided-attention to depth feature before performing self-attention, (b) Post-guide GAM applies guided-attention to depth feature after performing self-attention, (c) Bidirectional-guide GAM first implements guided-attention to depth feature and then after performing self-attention it employs guided-attention to color feature in the reverse direction.

the Figure 2b. As in the self-attention block, the attended output of guided-attention is fed into a feed-forward layer to produce output tokens. Note that both the self-attention and the guided-attention blocks utilize the same feed-forward layer. Thus, only the attention mechanism differs among them. Next, we construct three GAM variants (Figure 3), depending on the combination of the self-attention and the guided-attention blocks. In all three GAM variants, the color feature is first processed with self-attention to capture intra-modal interaction before offering guidance to the depth feature. In pre-guide GAM (Figure 3a), the guided-attention is carried out before the implementation of the self-attention. Then the self-attention is performed on the color-guided depth feature to reorganize the information. On the other hand, post-guide GAM (Figure 3b) performs the guided-attention after modeling intra-modal interaction of the depth feature. Bidirectional-guide GAM (Figure 3c) is similar to the pre-guide GAM, but a reverse guided-attention is added to enable bidirectional information flow between two different modalities. We compare the performances of GAM variants in Section 4.2.

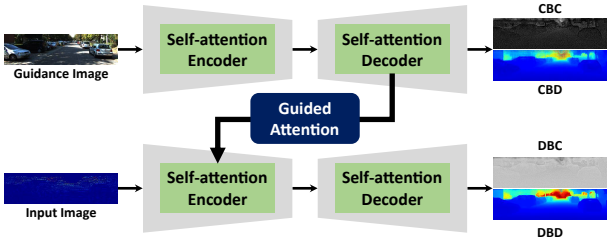
The modular GAM layers can be easily stacked in series for fusing multi-modal information in multiple scales. Based on GAM, we formulate two guidance architectures as shown in Figure 4. In the *parallel* guidance architecture, GAM processes color and depth features simultaneously. Denote the consecutive color features at the encoder as $\mathbf{F}_c^0, \mathbf{F}_c^1, \cdots, \mathbf{F}_c^L$ and the corresponding depth features as $\mathbf{F}_d^0, \mathbf{F}_d^1, \cdots, \mathbf{F}_d^L$. Then the input and output of the l-th GAM are expressed as follows:

$$[\mathbf{F}_c^l, \mathbf{F}_d^l] = \mathrm{GAM}_{parallel}^l([\mathbf{F}_c^{l-1}, \mathbf{F}_d^{l-1}]) \,, \tag{4}$$

where $l \in \{1, 2, \cdots, L\}$. In the *sequential* guidance archi-



(a) Parallel



(b) Sequential

Figure 4. Two guidance architectures based on a stack of GAMs.

tecture, the color branch is processed first and the decoded color features provide guidance to the encoded depth features. In this architecture, the final decoded color feature $\tilde{\mathbf{F}}_c^L$ provides guidance to the initial encoded depth feature $\mathbf{F}_d^0$, $\tilde{\mathbf{F}}_c^{L-1}$ provides guidance to $\mathbf{F}_d^1$, and so on.

$$\mathbf{F}_d^l = \text{GAM}_{sequential}^l([\tilde{\mathbf{F}}_c^{L-l+1}, \mathbf{F}_d^{l-1}]) \tag{5}$$

Note that the two guided-attention models have the same size with the same L.

3.4. Depth Fusion and Loss Function

After passing through the transformer encoder-decoder with GAMs, the final color feature $\tilde{\mathbf{F}}_c^L$ and the final depth feature $\tilde{\mathbf{F}}_d^L$ are obtained. Then they are taken as inputs to reconstruction blocks, consisting of two transposed convolution layers, for reconstructing the dense depth maps with corresponding confidence maps. GAMs effectively transfer the semantic information from the color images to the depth features, but not all complementary information is extracted from the color features. Thus, the final color-branch depth map is combined with the depth-branch depth map to improve the quality of the final output. As Gansbeke *et al.* [31], we utilize the confidence maps as weights for fusing the two dense depth maps. The weight values are computed from the confidence maps using the softmax function. Then, the final dense depth prediction $\hat{\mathbf{D}}_{out}$ is obtained by

$$\hat{\mathbf{D}}_{out}(p) = \frac{e^{\mathbf{C}_c(p)} \cdot \hat{\mathbf{D}}_c(p) + e^{\mathbf{C}_d(p)} \cdot \hat{\mathbf{D}}_d(p)}{e^{\mathbf{C}_c(p)} + e^{\mathbf{C}_d(p)}} , \tag{6}$$

in which p denotes a pixel and we denote the dense maps predicted from each branch by $\hat{\mathbf{D}}_c$ and $\hat{\mathbf{D}}_d$, and the confidence maps by $\mathbf{C}_c$ and $\mathbf{C}_d$.

In training the proposed network, we use the mean squared error (MSE) for computing the training loss. However, since the ground truth depth maps are often not completely dense, proper masking is required to filter valid pixels when computing the MSE. We use the following equation to compute the loss.

$$\mathcal{L}(\hat{\mathbf{D}}_{out}) = \frac{1}{|P|} \sum_{p \in P} |\hat{\mathbf{D}}_{out}(p) - \mathbf{D}_{GT}(p)|^2 , \tag{7}$$

where $\mathbf{D}_{GT}$, P, and $|P|$ denote the ground truth depth map, the set of valid pixels of $\mathbf{D}_{GT}$, and the number of valid pixels, respectively.

Moreover, during the entire training process of the proposed network, we introduce an additional supervision to the depth prediction for each branch as below:

$$\mathcal{L} = \mathcal{L}(\hat{\mathbf{D}}_{out}) + \lambda_c \mathcal{L}(\hat{\mathbf{D}}_c) + \lambda_d \mathcal{L}(\hat{\mathbf{D}}_d) , \tag{8}$$

where λ_c and λ_d are hyperparameters chosen from empirical experiments.

4. Experiments

In this section, we verify the effectiveness of our method in depth completion. We first describe the dataset used for our experiments and implementation details. Next, we provide ablation studies to analyze the importance of components of our model. Finally, we show the superior performance of our method by comparing it with other state-of-the-art approaches both quantitatively and qualitatively.

4.1. Implementation Details

Dataset We evaluate our method on the KITTI Depth Completion dataset [30], a large dataset with real-world outdoor scenes captured by a driving vehicle. It provides color images with corresponding aligned sparse depth maps for which ground truth depth is created by aggregating 11 consecutive 3D LiDAR frames and projecting them into one image frame. Nevertheless, since the GT depth contains only 16% annotated depth pixels, the sparsity of GT depth makes depth completion a challenging task. The dataset consists of 86K training samples along with 1K samples officially selected for validation and another 1K samples for testing. The validation and test set images have the resolution of 1216×352, while the training set has a slightly different image size. We use the test set to compare our model with the state-of-the-art models on the KITTI benchmark, while we use the validation set for ablation studies.

Metrics Following the criteria of the KITTI benchmark, we adopt four metrics for evaluation of performance: root mean squared error (RMSE), mean absolute error (MAE), root mean squared error of the inverse depth (iRMSE), and mean absolute error of the inverse depth (iMAE). Among these metrics, RMSE is used as the primary metric when evaluating our model.

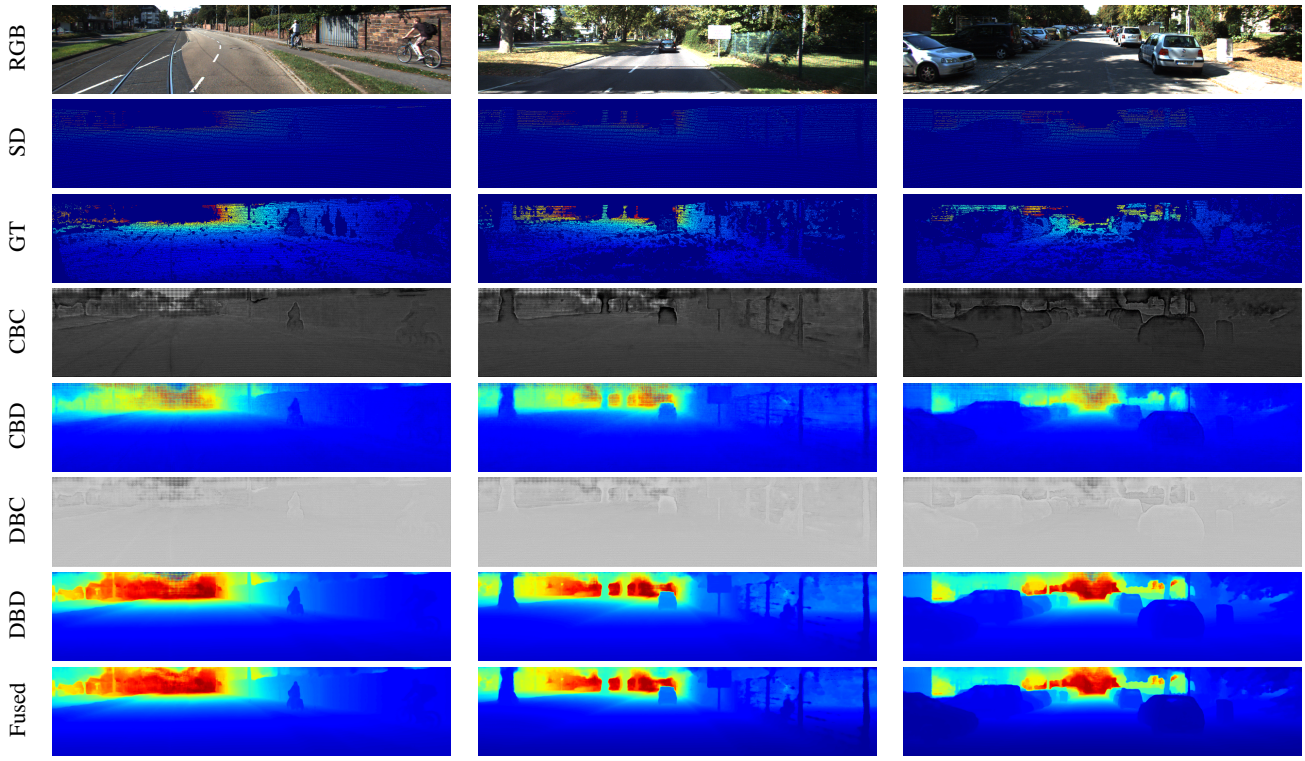


Figure 5. Visualization of the proposed method for typical examples. All the results are obtained by the GuideFormer model with the pre-guide GAMs forming the sequential guidance architecture.

Model	Size	FLOPs	Infer. time	RMSE
CNN enc - CNN dec	132M	748G	0.053s	772.78
CNN enc - Trans dec	115M	1584G	0.091s	770.13
Trans enc - CNN dec	111M	1212G	0.076s	768.41
Trans enc - Trans dec	99M	1802G	0.101s	**765.38**

Table 1. Comparison between CNN and Transformer. Concatenation is used for fusing multi-modal information by the sequential guidance architecture in all models.

Training configuration The proposed method is implemented on the PyTorch [21] framework, and the training setting is similar to PENet [10]. Since the depth maps have rare LiDAR points at the top, the training images are firstly bottom-cropped to the size of validation set images (1216×352) and then randomly cropped to 1216×320. We train our model on 8 NVIDIA V100 GPUs with a batch size of 8 for 30 epochs. We use the Adam optimizer [14] with the momentum of $\beta_1 = 0.9$, $\beta_2 = 0.99$, and the weight decay of 1×10^{-6}. Most transformer papers select the AdamW optimizer [17] with relatively large weight decay (from 10^{-2} to 10^{-4}) to train their models, but this setting makes training unstable and degrades the performance of our model. The initial learning rate is 2×10^{-4} and it is decayed by half for every 5 epochs. Moreover, the loss function in Equation 8 is configured with an initial value of 0.2 for λ_c and λ_d, decayed gradually. Note that many transformer models need to be pretrained on a large dataset like ImageNet [5] to get the expected performance. On the other hand, our model is not pretrained with any additional data and we only use the KITTI training set to train our model.

4.2. Ablation Studies

We conduct a number of experiments to verify the effectiveness of each component of GuideFormer, including the transformer encoder-decoder architecture, GAM variants, and the guidance architectures. The results are analyzed in detail with Table 1, 2, and 3.

CNN vs. Transformer The results in Table 1 show that transformer outperforms CNN with less parameters. This verifies that the input-adaptive self-attention block is more effective at extracting useful information and reconstructing dense depth maps from color and sparse depth images, compared to the static convolutional layer. However, although we employ window-based self-attention blocks to relieve the burden of computation, the computational cost of the transformer is still higher than that of CNN, which makes inference slow. Making the transformer computationally efficient is beyond the scope of this paper.

GAM Variants As shown in Table 2, the proposed GAM variants demonstrate their superior performance over simple concatenation. We can also see that pre-guide GAM

Model	Parallel	Sequential	Size
Concat	768.47	765.38	99M
Pre-guide	756.15	**754.17**	130M
Post-guide	759.98	757.52	130M
Bidirectional-guide	755.09	-	163M

Table 2. Comparison of different GAM variants. The concatenation model is selected as a baseline for comparison.

Model	Parallel	Sequential
Single-branch	867.94	
Dual-branch - DBD	759.61	758.26
Dual-branch - Fused depth	756.15	**754.17**

Table 3. Effect of dual-branch architecture and the depth fusion module. Single- and dual-branch models are of the same size. Pre-guide GAMs are adopted by dual-branch models.

outperforms post-guide GAM, meaning that performing guided-attention ahead of self-attention makes guidance more effective. Especially at the early stage of the transformer encoder, it is not easy for the transformer to learn representation from a sparse depth map. Guided-attention helps this process by delivering useful contextual information from the color features to the depth features. However, it is interesting that bidirectional-guide GAM gives better results compared to pre-guide GAM. Since color and depth features contain complementary contextual information, guiding the color feature with the depth feature augments its context.

Parallel vs. **Sequential** The results in Table 2 and 3 show that the sequential guidance architecture produces better results in general compared to its parallel counterpart. It implies that processing color features followed by depth features sequentially utilizes the guided-attention mechanism efficiently, while connecting two branches at the same stages decreases the effect of guidance. Color and depth features contain different levels of semantic information at each encoder or decoder stage. Corresponding information of each color and depth branch is more compatible in the proposed sequential guidance architecture compared to the parallel architecture, which improves the quality of guidance. Furthermore, it is interesting that the pre-guide GAMs in the sequential guidance architecture performs better than the bidirectional-guide GAMs in the parallel guidance architecture. We deduce that this is because the decoder feature of color branch is much informative than the encoder one for dense prediction. In conclusion, the nature of the guidance architecture impacts the performance more than the GAM itself.

Dual-Branch + Depth Fusion From the results in Table 3, we can see that a dual-branch encoder-decoder architecture predicts more precise dense depth maps than single branch architecture with the same model size. It verifies that dual-branch encoder-decoder architecture is optimal for extracting features from multi-modal inputs. Also as shown in Figure 5, depth-branch depth map contributes more to the final prediction in most regions. This implies that the information from color images is effectively delivered to the depth features through the guided-attention model. However, since complementary information remains in the color-branch depth map, it is needed to improve performance by the fusion module.

Method	RMSE	MAE	iRMSE	iMAE	reference
CSPN [4]	1019.64	279.46	2.93	1.15	ECCV18
IR L2 [18]	901.43	292.36	4.92	1.35	CVPR20
TWISE [12]	840.20	**195.58**	2.08	**0.82**	CVPR21
NConv [8]	829.98	233.26	2.60	1.03	PAMI20
S2D [19]	814.73	249.95	2.80	1.21	ICRA19
DepthNormal [36]	777.05	235.17	2.42	1.13	ICCV19
FusionNet [31]	772.87	215.02	2.19	0.93	MVA19
DeepLiDAR [23]	758.38	226.50	2.56	1.15	CVPR19
FuseNet [2]	752.88	221.19	2.34	1.14	ICCV19
CSPN++ [3]	743.69	209.28	2.07	0.90	AAAI20
NLSPN [20]	741.68	199.59	**1.99**	0.84	ECCV20
GuideNet [29]	736.24	218.83	2.25	0.99	TIP20
FCFR-Net [15]	735.81	217.15	2.20	0.98	AAAI21
ACMNet [40]	732.99	206.80	2.08	0.90	TIP21
PENet [10]	730.08	210.55	2.17	0.94	ICRA21
GuideFormer (ours)	**721.48**	207.76	2.14	0.97	-

Table 4. Quantitative comparisons with state-of-the-art methods on KITTI test set. Best results are shown in bold. The results of other methods are obtained from the KITTI online leaderboard, ranked by the RMSE.

4.3. Comparison with State-of-the-Art

Quantitative Analysis Taking the results of ablation studies into account, we compare our best GuideFormer, which adopts the sequential guidance architecture consisting of pre-guide GAMs, with the current state-of-the-art methods on the KITTI benchmark. As shown in Table 4, the proposed method outperforms all the peer-reviewed methods for the RMSE metric, which is considered as the primary metric for depth completion task. Although it is not on the top of the list for the other metrics, it still shows the competitive performance on these metrics. We discuss that the performance in terms of RMSE or MAE is highly related to the training loss. For example, NLSPN [20] used the combination of L_1 and L_2 losses. TWISE [12] applied the generalized L_1 loss, called asymmetric linear error. ACMNet [40] included the edge-aware smoothness using L_1 loss. As a result, these methods achieved lower MAE. However, it is non-trivial to balance RMSE and MAE. Thus, following the literature [10, 15, 29], we used only L_2 loss to train our model. We also want to emphasize that inverse metrics are inherently unstable at close objects, *i.e.*, for very large disparity values.

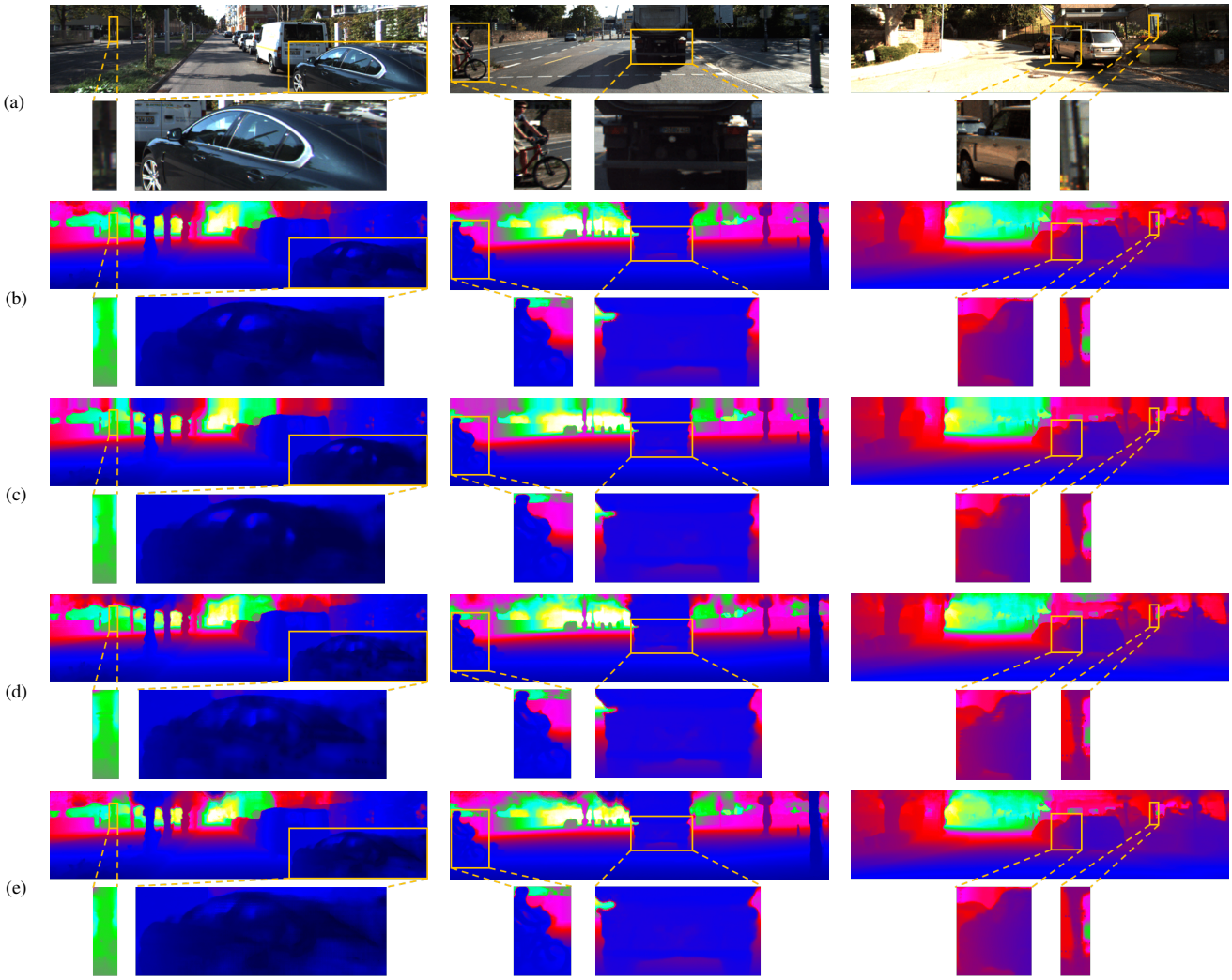


Figure 6. Qualitative comparison with state-of-the-art depth completion methods on KITTI test set. (a) color images, (b) FCFR-Net [15], (c) ACMNet [40], (d) PENet [10] and (e) GuideFormer (ours). We select two patches from each image to compare the quality of predicted depth for a variety of objects in detail.

Qualitative Analysis Figure 6 shows a visualization of several examples from the KITTI test set for evaluating the quality of our depth completion results. The state-of-the-art methods selected for comparison are FCFR-Net [15], ACMNet [40], and PENet [10]. We pick a variety of objects in the images for detailed analysis such as bikes, poles, trees, and cars. The zoomed-in patches show the considerable improvement of our method over existing state-of-the-art methods, especially at object boundaries. ACMNet [40] and PENet [10] have blurred depth prediction around boundaries, resulting in mixed depth pixels and broken edges. FCFR-Net [15] has relatively clear edges but ignores the details of objects. On the other hand, our method preserves clear edges and object details while predicting more precise depth values.

5. Conclusion

In this paper, we present GuideFormer, a dual-branch transformer architecture for color image-guided depth completion. GuideFormer consists of three main parts: (1) a fully transformer-based encoder-decoder architecture, (2) guided-attention module (GAM), and (3) the depth fusion module. Unlike existing CNN-based methods, our model utilizes the self-attention mechanism to extract internal token representations of the color and sparse depth images. Then the GAMs help fusing the information of two different modalities. By stacking GAMs to form the sequential guidance architecture, GuideFormer achieves an outstanding performance on the KITTI benchmark. We expect that our method serves as a new baseline for color image-guided depth completion and motivates further research.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[2] Yun Chen, Bin Yang, Ming Liang, and Raquel Urtasun. Learning joint 2d-3d representations for depth completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 7

[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*, pages 10615–10622, 2020. 2, 7

[4] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–119, 2018. 7

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 6

[6] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *arXiv preprint arXiv:2107.00652*, 2021. 2

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *arXiv preprint arXiv:2010.11929*, 2021. 2

[8] A. Eldesokey, M. Felsberg, and F. Khan. Confidence propagation through cnns for guided sparse depth regression. In *IEEE Transactions on Pattern Analysis & Machine Intelligence (PAMI)*, pages 2423–2436, 2020. 2, 7

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *arXiv preprint arXiv:1512.03385*, 2015. 3

[10] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Towards precise and efficient image guided depth completion. In *International Conference on Robotics and Automation (ICRA)*, 2021. 1, 2, 3, 4, 6, 7, 8

[11] Yu-Kai Huang, Tsung-Han Wu, Yueh-Cheng Liu, and Winston H. Hsu. Indoor depth completion with boundary consistency and self-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1070–1078, 2019. 2

[12] Saif Imran, Xiaoming Liu, and Daniel Morris. Depth completion with twin-surface extrapolation at occlusion boundaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 7

[13] Maximilian Jaritz, Raoul de Charette, Émilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. In *International Conference on 3D Vision (3DV)*, pages 52–60, 2018. 1, 2

[14] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2014. 6

[15] Lina Liu, Xibin Song, Xiaoyang Lyu, Junwei Diao, M. Wang, Yong Liu, and L. Zhang. Fcfr-net: Feature fusion based coarse-to-fine residual learning for monocular depth completion. In *arXiv preprint arXiv:2012.08270*, 2020. 1, 2, 3, 7, 8

[16] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *arXiv preprint arXiv:2103.14030*, 2021. 2, 3, 4

[17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. 6

[18] Kaiyue Lu, Nick Barnes, Saeed Anwar, and Liang Zheng. From depth what can you see? depth completion via auxiliary image reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 7

[19] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *International Conference on Robotics and Automation (ICRA)*, pages 3288–3295, 2019. 2, 7

[20] 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)*, pages 10615–10622, 2020. 1, 2, 7

[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 6

[22] C. Qi, W. Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 918–927, 2018. 1

[23] 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10615–10622, 2019. 1, 2, 3, 7

[24] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[25] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12179–12188, 2021. 2, 4

[26] Kourosh Sartipi, Tien Do, Tong Ke, Khiem Vuong, and Stergios I. Roumeliotis. Deep depth estimation from visual-inertial slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020. 1

[27] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[28] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *arXiv preprint arXiv:2105.05633*, 2021. 2

[29] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion. In *IEEE Transactions on Image Processing (TIP)*, pages 1116–1129, 2020. 1, 2, 4, 7

[30] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017. 1, 2, 5

[31] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. In *International Conference on Machine Vision Applications (MVA)*, pages 1–6, 2019. 1, 2, 3, 5, 7

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, page 6000–6010, 2017. 2

[33] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *arXiv preprint arXiv:2102.12122*, 2021. 2

[34] Yu-Huan Wu, Yun Liu, Xin Zhan, and Ming-Ming Cheng. P2t: Pyramid pooling transformer for scene understanding. In *arXiv preprint arXiv:2106.12011*, 2021. 2

[35] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *arXiv preprint arXiv:2105.15203*, 2021. 2

[36] 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/CVF International Conference on Computer Vision (ICCV)*, pages 2811–2820, 2019. 1, 2, 7

[37] Zheyuan Xu, Hongche Yin, and Jian Yao. Deformable spatial propagation networks for depth completion. In *IEEE International Conference on Image Processing (ICIP)*, pages 913–917, 2020. 2

[38] Guanglei Yang, Hao Tang, Mingli Ding, Nicu Sebe, and Elisa Ricci. Transformer-based attention networks for continuous pixel-wise prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[39] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. In *arXiv preprint arXiv:2107.00641*, 2021. 2

[40] Shanshan Zhao, Mingming Gong, Huan Fu, and Dacheng Tao. Adaptive context-aware multi-modal network for depth completion. In *IEEE Transactions on Image Processing (TIP)*, pages 5264–5276, 2021. 1, 2, 4, 7, 8

[41] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2