Trap Attention: Monocular Depth Estimation with Manual Traps

Chao Ning
Northwestern Polytechnical University
Xi'an 710072, China
lagarto@mail.nwpu.edu.cn

Hongping Gan[*]
Northwestern Polytechnical University
Xi'an 710072, China
ganhongping@nwpu.edu.cn

Abstract

Predicting a high quality depth map from a single image is a challenging task, because it exists infinite possibility to project a 2D scene to the corresponding 3D scene. Recently, some studies introduced multi-head attention (MHA) modules to perform long-range interaction, which have shown significant progress in regressing the depth maps. The main functions of MHA can be loosely summarized to capture long-distance information and report the attention map by the relationship between pixels. However, due to the quadratic complexity of MHA, these methods can not leverage MHA to compute depth features in high resolution with an appropriate computational complexity. In this paper, we exploit a depth-wise convolution to obtain long-range information, and propose a novel trap attention, which sets some traps on the extended space for each pixel, and forms the attention mechanism by the feature retention ratio of convolution window, resulting in that the quadratic computational complexity can be converted to linear form. Then we build an encoder-decoder trap depth estimation network, which introduces a vision transformer as the encoder, and uses the trap attention to estimate the depth from single image in the decoder. Extensive experimental results demonstrate that our proposed network can outperform the state-of-the-art methods in monocular depth estimation on datasets NYU Depth-v2 and KITTI, with significantly reduced number of parameters. Code is available at: https://github.com/ICSResearch/TrapAttention.

1. Introduction

Depth estimation is a classical problem in computer vision (CV) field and is a fundamental component for various applications, such as, scene understanding, autonomous driving, and 3D reconstruction. Estimating the depth map from a single RGB image is a challenge, since the same 2D scene can project an infinite number of 3D scenes. Therefore, the traditional depth estimation methods [27, 28, 33] are often only suitable for predicting low-dimension, sparse distances [27], or known and fixed targets [28], which obviously limits their application scenarios.

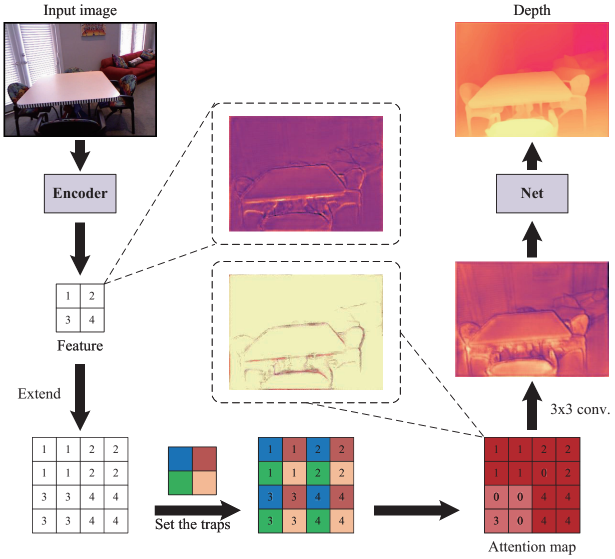


Figure 1. Illustration of trap attention for monocular deep estimation. Note that trap attention can significantly enhance depth estimation, as evidenced by the clearer depth differences between the table/chairs and the background.

To overcome these constraints, many studies [1, 8, 9, 20] have employed the deep neural networks to directly obtain high-quality depth maps. However, most of these research focuses on improving the performance of depth estimation networks by designing more complex or large-scale models. Unfortunately, such a line of research would render the depth estimation task a simple model scale problem without the trade-off between performance and computational budget.

Recently, several practitioners and researchers in monocular depth estimation [3, 17, 45] introduced the multi-head attention (MHA) modules to perform the long-range inter-

[*] Corresponding author

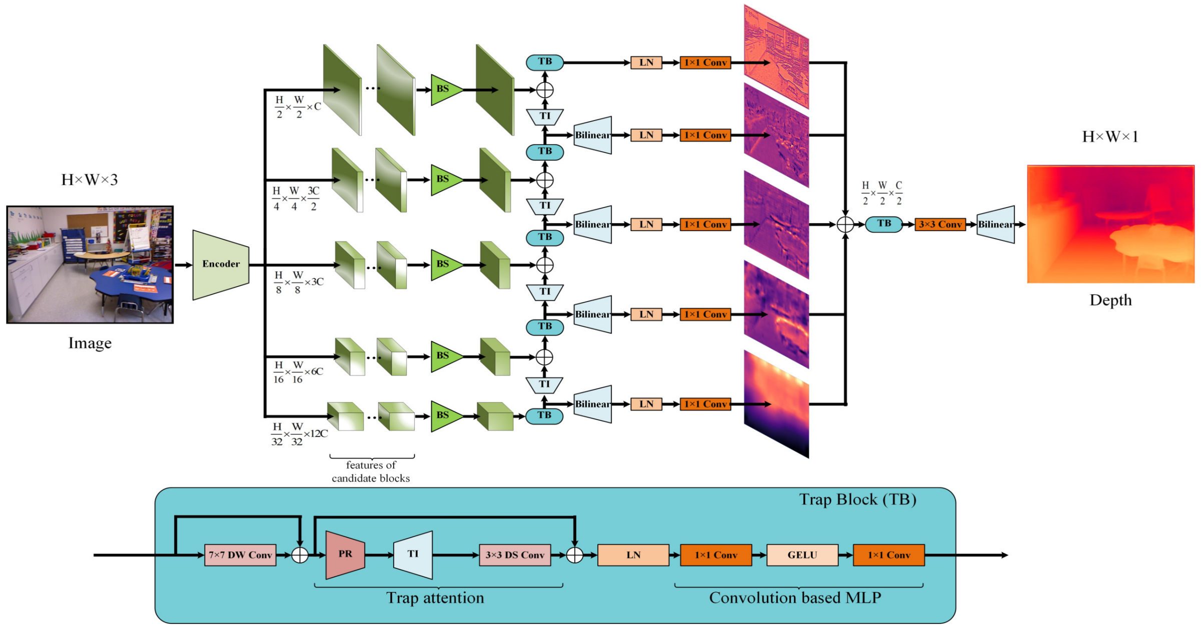
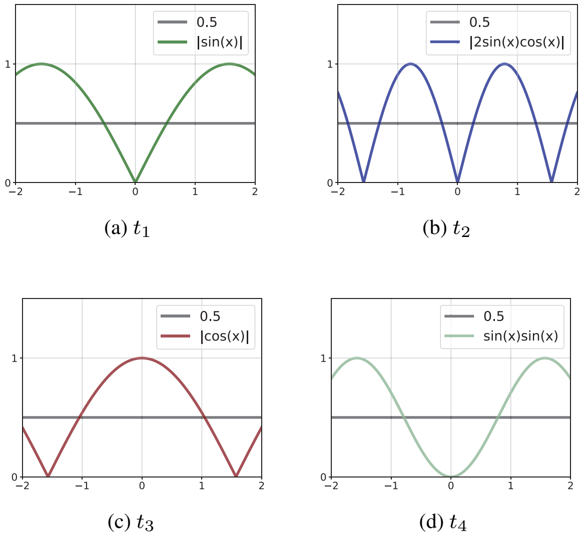

H×W×3
Image
Encoder
$\frac{H}{2}\times\frac{W}{2}\times C$
$\frac{H}{4}\times\frac{W}{4}\times\frac{3C}{2}$
$\frac{H}{8}\times\frac{W}{8}\times 3C$
$\frac{H}{16}\times\frac{W}{16}\times 6C$
$\frac{H}{32}\times\frac{W}{32}\times 12C$
features of candidate blocks
BS
TB
TI
Bilinear
LN
1×1 Conv
$\frac{H}{2}\times\frac{W}{2}\times\frac{C}{2}$
3×3 Conv
H×W×1
Depth
Trap Block (TB)
7×7 DW Conv
PR
TI
3×3 DS Conv
LN
1×1 Conv
GELU
1×1 Conv
Trap attention
Convolution based MLP

Figure 2. Overview of our trap depth estimation network, which includes an encoder and a decoder. TB, TI and BS denote the trap block, trap interpolation and block selection unit, respectively. TB is the basic block of our decoder, which decodes the depth feature from coarse to fine in five stages. TB consists of a depth-wise (DW) convolution layer, a trap attention (TA) unit and a convolution based MLP. The size of decoder depends on an arbitrary channel dimension (denoted as C). "$\oplus$" denotes the addition operation.



(a) t_1

(b) t_2

(c) t_3

(d) t_4

Figure 3. The curves for trap functions used in trap attention without the rounded operation. (a) and (d) are two similar curves. (b) has a higher frequency. (c) has a different initial phase with other curves.

action, which have shown considerable progress in regressing the depth maps. Representative works of such methods are AdaBin [3] and NeW CRFs [45]. Nevertheless, due to the quadratic computational complexity of MHA, the computational complexity of high resolution depth map for AdaBin or NeW CRFs is typically expensive, i.e., for an $h \times w$ image, its complexity is $\mathcal{O}(h^2w^2)$.

To reduce the computational complexity, in this work, we firstly exploit a deep-wise convolution layer to compute the long-distance information and then propose an attention mechanism, called trap attention, which leverages various manual traps to remove some features in extended space, and exploits a 3×3 convolution window to compute relationship and attention map. As a result, the quadratic computational complexity $\mathcal{O}(h^2w^2)$ can be converted to linear form $\mathcal{O}(hw)$. As illustrated by the example in Figure 1, the proposed trap attention is highly effective for depth estimation, which can allocate more computational resource toward the informative features, i.e., edges of table and chairs, and output a refined depth map from coarse depth map.

Based on this attention mechanism, we finally build an encoder-decoder depth estimation network, which introduces a vision transformer as the encoder, and uses the trap attention to estimate the depth from single image in the decoder. We can build our depth estimation network of different scales according to the depth estimation scene, which can obtain a balance between performance and computational budget. Experimental results show that our depth estimation network outperform previous estimation methods by remarkable margin on two most popular indoor and outdoor datasets, NYU [36] and KITTI [11], respectively.

Specifically, our model can obtain consistent predictions with sharp details on visual representations, and achieve the new state-of-the-art performance in monocular depth estimation, with only **35%** parameters of the prior state-of-the-art methods.

In summary, our main contributions are as follows:

- We use a depth-wise convolution to capture long-distance information and introduce an extra attention mechanism to compute the relationships between features, which is an efficient alternative to MHA, resulting in that the computational complexity is reduced from $\mathcal{O}(h^2w^2)$ to $\mathcal{O}(hw)$.
- We propose a novel attention mechanism that can allocate more computational resource toward the informative features, called trap attention, which is highly effective for depth estimation.
- We build an end-to-end trap network for monocular depth estimation, which can obtain the state-of-the-art performance on NYU and KITTI datasets, with significantly reduced number of parameters.

2. Related Work

2.1. Unsupervised monocular depth estimation

The unsupervised method is an approach that needs only rectified stereo image pairs to train the monocular depth estimation network. As an inspiring work, Garg et al. [10] firstly proposed to train the depth estimation network by synthesizing the rectified views, which can significantly reduce the effort to collect the training data for unseen scenes or environments. To obtain the higher quality depth prediction, several works have afterwards focused on this unsupervised setting by using photometric reconstruction error [10], or generating a probability distribution of possible disparities for each 2D point [42], or introducing a multiview consistency loss [44], or training with extra robust reprojection loss and automasking loss [13], respectively. These unsupervised methods can achieve better depth estimation results than traditional depth estimation methods. However, these unsupervised approaches always required more complex data for rectifying stereo pairs during training and testing, which may limit their generalization in new monocular scenarios.

2.2. Supervised monocular depth estimation

Supervised depth estimation methods generally use a single RGB image and take the depth map measured with range sensors device as ground truth for supervision during network training in monocular depth estimation. A notable early approach is proposed by Saxena et al. [33], which can associate multi-scaled local and global image features, and model both depths at individual nodes and the relationship between depths at different nodes via Markov random field. To learn an end-to-end mapping from RGB images to depth data, Eigen et al. [8] have pioneered a multi-scaled convolutional architecture, which firstly uses one convolutional neural network (CNN) to predict a coarse global depth, and then refine the prediction locally via another CNN. Following the success of this approach, several works have proposed different networks to estimate the depth maps by introducing strong scene priors to estimate surface normal [22], or translating the depth regression problem to depth classification [4], or by supervising via virtual planes [43], or using extra auxiliary loss function [19, 21], respectively. Our proposed depth estimation method can refine the depth features from coarse to fine in five stages, which belongs to supervised monocular depth estimation.

2.3. Attention Mechanisms

Attention mechanisms, such as SE block [16] and CBAM [41], have been introduced to direct computational resources towards more informative features. These methods generate coefficients for each channel or spatial location to emphasize important features. To focus on a specific pixel rather than a group of features, the multi-head attention (MHA) mechanism was introduced by Vaswani et al. [39] for natural language processing. MHA has been shown to be a more effective attention method than previous attention mechanisms in computer vision tasks [5, 7, 40], and has been recently leveraged for monocular depth estimation [3, 45]. However, MHA has a quadratic complexity, which makes it difficult to perform long-range interaction in high input resolutions throughout the decoder, resulting in rough prediction details or insignificant improvement for depth regression. To address this problem, we propose an efficient alternative attention mechanism, called trap attention, which can be implemented throughout the decoder to predict high-resolution depth maps with linear complexity.

3. Method

This section first details the proposed trap attention, and then displays the architecture of our depth estimation model, which exploits a vision Transformer as the encoder, and decodes the depth features via trap attention.

3.1. Trap attention

Multi-head attention (MHA) is the popular method to perform long-distance interaction in depth estimation [3, 45], which can be roughly summarized to two main operations: one is to capture long-range information, and the other is to report the feature relevance and attention map by a softmax matrix. To perform an efficient alternative to MHA, we exploit a 7 × 7 depth-wise convolution layer to capture long-range information, and present a trap attention

Table 1. Comparison of performances on the NYU dataset. "*" indicates using additional data, and "†" denotes the unsupervised method.

Method	# Params	higher is better			lower is better				
		$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Abs Rel	Sq Rel	RMSE	RMSE log	log10
Saxena et al. [34]	-	0.447	0.745	0.897	0.349	-	1.214	-	-
Li et al. [22]	-	0.621	0.886	0.968	0.232	-	0.821	-	0.094
Liu et al. [23]	-	0.650	0.906	0.974	0.213	-	0.759	-	0.087
P^{2}Net † [44]	-	0.801	0.951	0.987	0.147	-	0.553	-	0.062
Yin et al. [43]	91.6M	0.875	0.976	0.994	0.108	-	0.416	-	0.048
Long et al. [25]	81.6M	0.890	0.982	0.996	0.101	-	0.377	-	0.044
P3Depth [30]	94.2M	0.898	0.981	0.996	0.104	-	0.356	-	0.043
AdaBin [3]	78.0M	0.903	**0.984**	**0.997**	0.103	-	0.364	-	0.044
BTS [20]	47.0M	0.885	0.978	0.994	0.110	0.066	0.392	0.142	0.047
Trap-S (ours)	**28.3M**	**0.910**	**0.984**	0.996	**0.100**	**0.054**	**0.356**	**0.128**	**0.043**
Eigen et al. [8]	141.0M	0.769	0.950	0.988	0.158	-	0.641	-	-
DORN [9]	110.0M	0.828	0.965	0.992	0.115	-	0.509	-	0.051
DPT* [32]	123.1M	0.904	0.988	0.998	0.110	-	0.357	-	0.045
NeW CRFs [45]	270.5M	0.922	**0.992**	**0.998**	0.095	0.045	0.334	0.119	0.041
PackNet-SAN* [14]	110.8M	0.892	0.979	0.995	0.106	-	0.393	-	-
Trap-M (ours)	**94.2M**	0.925	0.988	0.997	0.092	0.047	0.332	0.119	0.040
Trap-L (ours)	222.0M	**0.927**	0.991	**0.998**	**0.090**	**0.044**	**0.329**	**0.117**	**0.039**

(TA) operation to report the attention map and compute the feature relationship. The trap attention first leverages manual traps to identify the importance of features, and then employs a depth-wise separable convolution layer to summarize the attention and relevance information.

Manual traps. To select the informative feature, we set some manual traps on the extended feature map and name this operation as trap interpolation (TI). In our trap design, there are 4 kinds of trap functions, namely t_1, t_2, t_3, and t_4, which can be defined as:

$$\begin{aligned} t_1(x) &= [|\sin(x)|], \\ t_2(x) &= [|2\sin(x)\cos(x)|], \\ t_3(x) &= [|\cos(x)|], \\ t_4(x) &= [\sin^2(x)], \end{aligned} \tag{1}$$

, respectively, where x is the input feature map, and "[ ]" denotes the rounding operation.

As shown in Figure 3, t_1 and t_4 are two similar functions, which are set on the diagonal regions. They can preliminarily classify, retain, and remove the input feature map. t_2 is a higher frequency function that can further classify the retained or removed features. In addition, t_3 is used to prevent the complete removal of these features during the upsampling process.

These four traps are leveraged to classify each pixel in an extended 2×2 space, which can be written as follows:

$$\mathrm{TI}\left(x_{\mathrm{i}}\right) = \begin{pmatrix} x_i \times t_1(x_i), & x_i \times t_2(x_i) \\ x_i \times t_3(x_i), & x_i \times t_4(x_i) \end{pmatrix}, \tag{2}$$

where i is the index that enumerates all possible positions along spatial dimension. Obviously, the TI operation is used to upsample the input feature map x to a larger size of 2× height and 2× width.

Attention map and feature relationship. To keep the identical resolution between input and output, we leverage a reversal operation of pixel shuffle [35] to rearrange the pixels. Afterwards, the trap interpolation is used to classify the rearranged features. Finally, we use a 3×3 depth-wise separable convolution layer with a group size of 4 to obtain the attention information and feature relevance. Suppose that the input feature map $X \in \mathbb{R}^{\hat{H} \times \hat{W} \times \hat{C}}$. Formally, the trap attention can be defined as:

$$\mathrm{TA}(X) = W_d * \mathrm{TI}(\mathrm{PR}(X)) + b_d, \tag{3}$$

where PR denotes the pixel rearrangement operator, $W_d \in \mathbb{R}^{\hat{C} \times 4 \times 3 \times 3}$ and $b_d \in \mathbb{R}^{\hat{C}}$ are the weights and biases of depth-wise separable convolutional filters, respectively, "$*$" is the convolution operation. In particular, the pixel rearrangement operator resizes X to the shape of $\frac{\hat{H}}{2} \times \frac{\hat{W}}{2} \times 4\hat{C}$, and then the TI operator upsamples it to a size of $\hat{H} \times \hat{W} \times 4\hat{C}$.

Complexity of TA. Suppose that a feature map has an $h \times w$ spatial size. We can calculate the computational complexity of MHA and TA, i.e.,

$$\begin{aligned} \Omega(\mathrm{MHA}) &= 4hwc^2 + 2(hw)^2c, \\ \Omega(\mathrm{TA}) &= 10hwc + 4 \times 3^2hwc, \end{aligned} \tag{4}$$

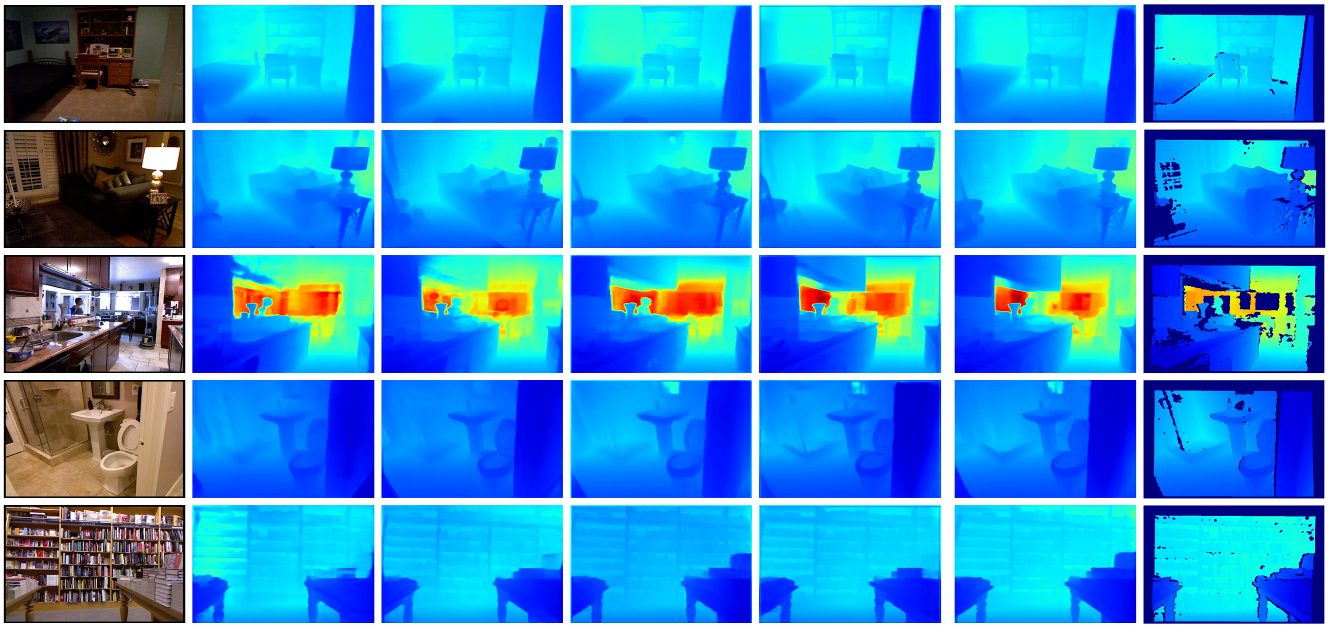


Figure 4. Qualitative comparison with the state-of-the-art methods on the NYU dataset. Compared with BTS, AdaBin and NeW CRFs, the predictions of our Trap-S and Trap-M have higher qualities. Please zoom in for more details.

where c and Ω denote the channel dimension of image and the computational complexity, respectively. Obviously, $\Omega(\text{MHA})$ is quadratic to hw, TA has the linear complexity of spatial size hw. As a result, TA can be easily implemented to predict the depth map in a high resolution, compared to MHA.

3.2. Model architecture

Overview. Our depth estimation network consists of a ViT encoder, a decoder, and a block selection (BS) unit in between (see Figure 2). The trap attention unit is embedded into the basic block of the decoder, i.e., the trap block, to compute depth features. For a single RGB image of size $H \times W$, our network has five stages of deep features for the trap attention-based decoder, ranging from $\frac{H}{32} \times \frac{W}{32}$ to $\frac{H}{2} \times \frac{W}{2}$. A TI operation is used between each two adjacent stages to upsample the lower-resolution feature map for feature fusion. Finally, the depth features of all stages are mixed in a feature map computed by a trap block and a 3×3 convolution layer for the depth map. The depth prediction is upsampled using bilinear interpolation to the same resolution as the RGB image. To adapt to different application scenarios, we have three sizes of end-to-end networks: Trap-S, Trap-M, and Trap-L. These models use XCiT-S12 [2], XCiT-M24 [2], and Swin-L [24] as encoders, respectively. Details of our Trap-S, Trap-M, and Trap-L models can be found in Section 1 of our **Supplement file**.

Block selection. During feedforward propagation in the vision Transformer encoder, the attention paid to background features can be diminished. Unlike the semantic segmentation task, which classifies background pixels into a single class, monocular depth estimation requires background pixels to be categorized into different depth levels. To address this challenge, we introduce a block selection unit that compares the feature maps of n consecutive candidate blocks pixel by pixel and selects the maximum value pixel. Formally, the output feature of the block selection unit, denoted as Y_j, is defined as:

$$Y_j = \text{Max}(B_j^1, B_j^2, ..., B_j^k ..., B_j^n), \quad (5)$$

where B_j^k denotes the pixel of output feature map in n candidate encoder blocks, k and j are the indexes of a candidate block and a position of feature map (in channel or space), respectively, Max is an operation to take the maximum value. The BS operation is used on all positions of input feature maps. Raghu et al. have demonstrated that ViT architectures own highly similar representations throughout different depths [31], hence the BS operation does not destroy the integrity of features, and can obtain the more accurate depth map.

Decoder block. The basic block of the decoder, namely trap block, which leverages a 7×7 depth-wise convolution layer and a trap attention unit to capture the spatial relationship, and obtain depth information along channel dimension by a convolution based MLP (denoted by Chan). The Chan operator exploits two 1×1 convolution layers and a GELU

Table 2. Comparison of performances on KITTI dataset. "*" indicates using additional data.

Method	# Params	higher is better			lower is better			
		$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Abs Rel	Sq Rel	RMSE	RMSE log
Saxena et al. [34]	-	0.601	0.820	0.926	0.280	3.012	8.734	0.361
Liu et al. [23]	-	0.680	0.898	0.967	0.201	1.584	6.471	0.273
Godard et al. [12]	-	0.861	0.949	0.976	0.114	0.898	4.935	0.206
Kuznietsov et al. [18]	-	0.862	0.960	0.986	0.113	0.741	4.621	0.189
Yin et al. [43]	91.6M	0.938	0.990	0.998	0.072	-	3.258	0.117
P3Depth [30]	94.2M	0.953	0.993	0.998	0.071	0.270	2.842	0.103
AdaBin [3]	78.0M	0.964	0.995	**0.999**	0.058	0.190	2.360	0.088
BTS [20]	47.0M	0.956	0.993	0.998	0.059	0.245	2.756	0.096
Trap-S (ours)	**28.3M**	**0.967**	**0.996**	**0.999**	**0.055**	**0.177**	**2.278**	**0.085**
Eigen et al. [8]	141.0M	0.702	0.898	0.967	0.203	1.548	6.307	0.282
DRON [9]	110.0M	0.932	0.984	0.994	0.072	0.307	2.727	0.120
DPT* [32]	123.1M	0.959	0.995	**0.999**	0.062	-	2.573	0.092
NeW CRFs [45]	270.5M	0.974	0.997	**0.999**	0.052	0.155	2.129	0.079
PackNet-SAN* [14]	110.8M	0.955	-	-	0.062	-	2.888	-
Trap-M (ours)	**94.2M**	0.976	**0.998**	**0.999**	0.054	0.149	1.990	0.078
Trap-L (ours)	222.0M	**0.980**	**0.998**	**0.999**	**0.050**	**0.128**	**1.869**	**0.074**

nonlinearity [15] to compute the depth information along channel dimension, which can be written as follows:

$$\mathrm{Chan}(X) = W_{c2} * (\mathrm{GELU}(W_{c1} * X + b_{c1})) + b_{c2}, \quad (6)$$

where W_{c1}, W_{c2} are the convolution weights and b_{c1}, b_{c2} are the biases of convolutions, respectively.

As detailed in Figure 2, a 7×7 depth-wise convolutions layer is employed to mix long-range depth features to input $\mathrm{z_I}$. Then a trap attention unit is used to compute the relationship and attention information $\mathrm{z_{attn}}$ from the mixed feature $\mathrm{z_{mix}}$. As a result, the trap block can be summarized as follows:

$$\begin{aligned} \mathrm{z_{mix}} &= \mathrm{DW}(\mathrm{z_I}) + \mathrm{z_I}, \\ \mathrm{z_{attn}} &= \mathrm{TA}(\mathrm{z_{mix}}) + \mathrm{z_{mix}}, \\ \mathrm{z_o} &= \mathrm{Chan}(\mathrm{LN}(\mathrm{z_{attn}})), \end{aligned} \quad (7)$$

where DW and LN denote the 7×7 depth-wise convolution layer and the LayerNorm layer, respectively, $\mathrm{z_o}$ refers to the output feature map.

Feature fusion. To fuse the depth feature maps from neighboring decoder stages, it is necessary to upsample the lower resolution features. Compared to traditional upsampling algorithms, i.e., nearest neighbor or bilinear interpolation, which linearly extends the depth features along spatial dimensions, nonlinear extending methods exhibit superior performance in feature fusion. This has been demonstrated in previous depth estimation methods, including BTS [20], and NeW CRFs [45]. In our method, the nonlinear and low-computational trap interpolation is used to upsample feature map in our depth estimation model.

Table 3. Comparison of performances on SUN RGB-D dataset.

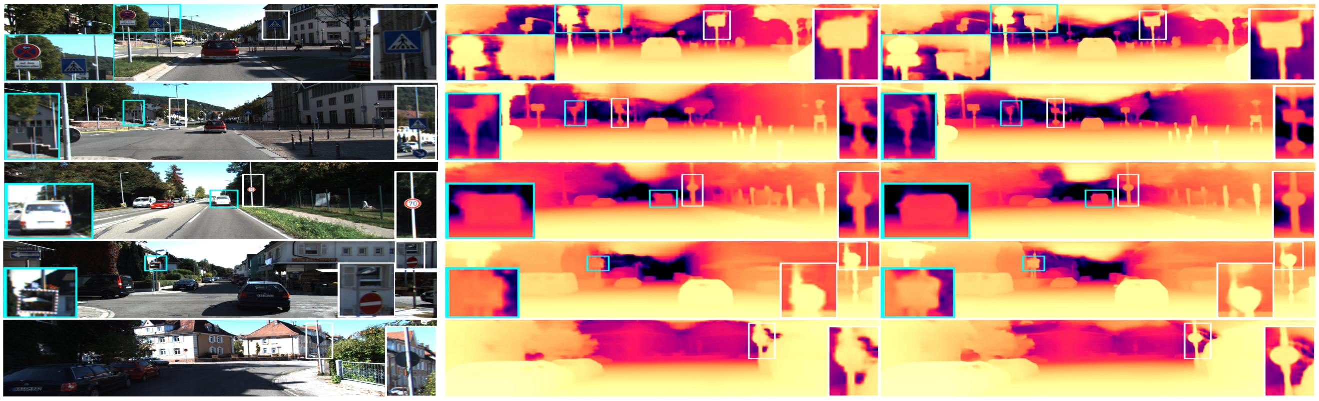
Method	higher is better			lower is better		
	1.25	1.25^2	1.25^3	Abs Rel	RMSE	log10
Chen [6]	0.757	0.943	0.984	0.166	0.494	0.071
Yin [43]	0.696	0.912	0.973	0.183	0.541	0.082
BTS [20]	0.740	0.933	0.980	0.172	0.515	0.075
AdaBin [3]	0.771	0.944	0.983	0.159	0.476	0.068
Trap-S	0.796	0.958	0.988	0.150	0.436	0.064
Trap-M	0.802	0.962	0.989	0.146	0.424	0.063
Trap-L	**0.806**	**0.964**	**0.991**	**0.141**	**0.414**	**0.061**

Loss function. Following previous studies [20, 45], scale-invariant logarithmic loss [8] is used during supervised training, which can be defined as:

$$\mathcal{L} = \alpha \sqrt{\frac{1}{N}\sum(\log \hat{d}_i - \log d_i)^2 - \frac{\lambda}{N^2}(\sum \log \hat{d}_i - \log d_i)^2}, \quad (8)$$

where $\hat{d}_i$ is the predicted depth pixel, d_i is the ground truth depth pixel, and N denotes the number of valid pixels. To keep the consistency of loss function with previous methods [3, 20, 45], we set $\lambda = 0.85$ and $\alpha = 10$.

4. Experimental Results



(a) Input

(b) NeW CRFs

(c) Trap-L(ours)

Figure 5. Qualitative comparison with the state-of-the-art method on the KITTI dataset. In comparison to NeW CRFs, our Trap-L model demonstrates stronger performance in predicting the depths of diverse targets.

4.1. Datasets

NYU dataset. There are 120K training RGB and depth pairs from 249 indoor scenes, and 654 testing samples from 215 indoor scenes in the NYU dataset [36]. We follow the official split to train and evaluate our method, with 24231 image-depth pairs for training and 654 testing images for testing.

KITTI dataset. The KITTI dataset is a outdoor benchmark with 61 scenes, which are captured from the equipment mounted on a moving vehicle. We employ the split provided by Eigen et al. [8] (23488 training samples and 697 testing samples) to evaluate our method. The capturing range of the Eigen split is 0-80m.

SUN RGB-D dataset. Following previous studies [3, 20], we evaluate the NYU pretrained models in SUN RGB-D dataset [37] without extra training. The SUN RGB-D dataset contains 10K indoor images collected from four different sensors.

4.2. Training setting and evaluation metric

Training protocols. Our models are implemented in Pytorch [29], and trained on Nvidia Tesla V100 GPUs. These networks are optimized end-to-end for 320K iterations, using batch size 8 and an AdamW [26] optimizer with an initial learning rate of 0.0001, a weight decay of 0.01, a scheduler that uses linear learning rate decay, a LayerScale [38] with initialization $\epsilon = 1$, and a linear warmup of 3.2K iterations. The output depth map of our networks is simply upscaled to the full resolution by bilinear interpolation.

Quality metrics of depth estimation. We use standard eight metrics to evaluate the performance of the predicted depth [8], including five error metrics, and three accuracy metrics. The error metrics consist of absolute relative (Abs Rel), squared relative (Sq Rel) error, Log10 error, root mean squared error (RMSE) and its log variant (RMSE log). Moreover, the thresholds (δ) for these accuracy metrics are set to 1.25, 1.25^2 and 1.25^3, respectively.

Model size. Some previous state-of-the-art methods, e.g., NeW CRFs [45], typically exploit a big scale model to obtain high performance. Obviously, it is unfair to just compare the depth estimation results of various models. Therefore, we adopt the number of parameters to measure the size of model, as done in BTS [20] and AdaBin [3].

4.3. Comparison to the state-of-the-art methods

In terms of efficiency, our main comparisons are as follows: we compare our Trap-S model with small models containing less than 100 million parameters, while we compare our Trap-M and Trap-L models with models containing over 100 million parameters.

Results on NYU. Table 1 shows the performance on NYU dataset. For small models, our Trap-S surpasses prior state-of-the-art model across five metrics with only **36.2%** parameters. For larger models, our Trap-M outperforms previous state-of-the-art method across four metrics with only **35%** parameters. As shown in Figure 4, it can be clearly observed that the predictions of our method have sharper details for various objects, e.g., bookshelves, lamps and vases. Please see the Section 2 of our **Supplement file** for more details.

Results on KITTI. The results on KITTI dataset are listed in Table 2. Our Trap-S and Trap-M can outperform previous state-of-the-art methods, and reduce about **65%** parameters. Especially, compared with similar size BTS method, Trap-S reduces the “Sq Rel” and “RMSE” error by **27.8%** and **17.3%**, respectively. Using the same Swin-L [24] encoder, our Trap-L model can surpass the prior state-of-the-art NeW CRFs model across 6 metrics. As visualized in Figure 5, compared to NeW CRFs, our method can more accurately predict for both the depth of dense objects and in-

Table 4. Ablation studies for our deep estimation networks on NYU dataset. S and M denote the Trap-S and Trap-M models without TI, TA, or BS unit, respectively. BS_i represents that BS is used to select the maximum pixels from i blocks of encoder. "†" refers to the substitution of the four trap functions in the trap attention module with diverse functions for Trap-M.

Variant	higher is better			lower is better	
	1.25	1.25^2	1.25^3	Abs Rel	RMSE
S	0.883	0.981	0.995	0.120	0.385
S + TA	0.892	0.983	0.996	0.114	0.371
S + TI + TA	0.900	0.983	0.995	0.107	0.363
S + TI + TA + BS_2	0.902	**0.984**	**0.996**	0.105	0.361
S + TI + TA + BS_4	**0.910**	**0.984**	**0.996**	**0.100**	**0.356**
M + TI + TA	0.916	0.987	0.996	0.099	0.342
M + TI + TA + BS_2	0.917	**0.988**	**0.997**	0.099	0.337
M + TI + TA + BS_4	0.917	**0.988**	**0.997**	0.098	0.335
M + TI + TA + BS_6	0.920	**0.988**	**0.997**	0.096	0.334
M + TI + TA + BS_8	**0.925**	**0.988**	**0.997**	**0.092**	**0.332**
Trap-M with different manual functions and other alternative functions					
Trap-M (baseline)	**0.925**	**0.988**	**0.997**	**0.092**	**0.332**
Without rounding	0.920	0.987	**0.997**	0.096	0.339
$t_3 = [\lvert\sin(x^2)\rvert]$	0.923	**0.988**	**0.997**	0.093	0.336
Without traps	0.900	0.986	0.996	0.112	0.359
ReLU nonlinearity†	0.902	0.986	0.996	0.111	0.358
GeLU nonlinearity†	0.913	0.986	**0.997**	0.099	0.340
SiLU nonlinearity†	0.904	0.986	0.996	0.111	0.362
4 learning weights†	0.904	0.986	**0.997**	0.109	0.354

dividual objects across various brightness scene, which can further verify the efficient performance of our model.

In addition, our proposed trap network can achieve the state-of-the-art performance among all published submissions on the **KITTI online benchmark**. Please refer to the Section 3 of our **Supplement file**.

Results on SUN RGB-D. To test the generalization of our method, we perform a cross-dataset evaluation and detail the comparison results in Table 3. Compared with the state-of-the-art method, our Trap-L can reduce the "Abs Rel", "RMSE", and "log 10" error by 11.3%, 13.0%, and 10.3%, respectively. Moreover, the visualization results are displayed in Section 2 of our **Supplement file**. And it can be clearly observed that our predictions have a stronger integrity for big targets, e.g., beds, walls, etc.

4.4. Ablation Study

We conduct several ablation studies for our proposed deep estimation network on NYU dataset. In this experiment, the trap attention (TA), trap interpolation (TI), and block selection (BS) with various blocks are optionally fitted to the proposed trap network as spare parts.

Trap attention. As shown in Table 4, the trap attention can improve the performance of "$\theta < 1.25$", "$\theta < 1.25^2$", "$\theta < 1.25^3$", "Abs Rel" and "RMSE" by 1.0%, 0.2%, 0.1% 5.0% and 3.6%, respectively. According to Table 4, we can clearly observe that trap attention is indeed effective for depth estimation. Please see Section 4 of our **Supplement file** for more details.

Trap interpolation. Compared with the well-used bilinear interpolation, our trap interpolation can gain the performance improvement, which can reduce "Abs Rel" and "RMSE" error by "6.1%" and "2.2%", respectively, as shown in Table 4. In fact, the trap interpolation and the DW convolution of trap block can be also treated as a trap attention mechanism.

Block selection. For our proposed network, the number of candidate blocks for each BS unit is an important variant. We evaluate the variant in different model sizes (Trap-S uses an encoder of 12 blocks, and Trap-M uses an encoder of 24 blocks). The results in Table 4 show that the optimal setting for the number of candidate blocks is $\frac{1}{3}$ of the total block number of encoder i.e., Trap-S and Trap-M use 4 and 8 candidate blocks, respectively.

Trap functions. According to Equation 2, we propose four manual trap functions to be used in TI or TA units. Table 4 shows that the rounding operation can slightly improve the performance of Trap-M. Figure 3 shows that t_3 differs from the other trap functions when the input x is around 0. Therefore, we replace the t_3 function with $[\lvert\sin(x^2)\rvert]$ while retaining the other three trap functions and investigate the experimental results. As indicated in Table 4, the original function of $[\lvert\cos(x)\rvert]$ outperforms the $[\lvert\sin(x^2)\rvert]$ function. Moreover, we can observe that the four manual traps can achieve significant margins over alternative functions such as ReLU nonlinearity, GELU nonlinearity, SiLU nonlinearity, or four learning weights.

5. Conclusion

This paper first exploits the depth-wise convolution to obtain long-range information, and introduces a novel trap attention mechanism to compute the relationships between features, which is an efficient alternative to MHA. As a result, the quadratic computational complexity $\mathcal{O}(h^2w^2)$ can be converted to linear form $\mathcal{O}(hw)$. Moreover, trap attention is further employed to build an end-to-end network for monocular depth estimation. Experimental results show that the proposed depth estimation network can obtain state-of-the-art scores on two most popular datasets, NYU and KITTI, with only 35% parameters of the prior state-of-the-art methods. We hope that our proposed work can inspire further research in different fields, e.g., image classification, semantic segmentation, and 3D reconstruction from multiple images.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 62101455.

References

[1] Shubhra Aich, Jean Marie Uwabeza Vianney, Md Amirul Islam, and Mannat Kaur Bingbing Liu. Bidirectional attention network for monocular depth estimation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11746–11752. IEEE, 2021. 1

[2] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *Advances in Neural Information Processing Systems (NIPS)*, 34:20014–20027, 2021. 5

[3] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4009–4018, 2021. 1, 2, 3, 4, 6, 7

[4] Yuanzhouhan Cao, Zifeng Wu, and Chunhua Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology (CSVT)*, 28(11):3174–3182, 2017. 3

[5] 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)*, pages 213–229. Springer, 2020. 3

[6] Xiaotian Chen, Xuejin Chen, and Zheng-Jun Zha. Structure-aware residual pyramid network for monocular depth estimation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 694–700, 2019. 6

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 3

[8] 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 (NIPS)*, 27, 2014. 1, 3, 4, 6, 7

[9] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2002–2011, 2018. 1, 4, 6

[10] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–756. Springer, 2016. 3

[11] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research (IJRR)*, 32(11):1231–1237, 2013. 2

[12] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 270–279, 2017. 6

[13] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3828–3838, 2019. 3

[14] Vitor Guizilini, Rares Ambrus, Wolfram Burgard, and Adrien Gaidon. Sparse auxiliary networks for unified monocular depth prediction and completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11078–11088, 2021. 4, 6

[15] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 6

[16] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7132–7141, 2018. 3

[17] Lam Huynh, Phong Nguyen-Ha, Jiri Matas, Esa Rahtu, and Janne Heikkilä. Guiding monocular depth estimation using depth-attention volume. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 581–597. Springer, 2020. 1

[18] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6647–6655, 2017. 6

[19] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV)*, pages 239–248. IEEE, 2016. 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. 1, 4, 6, 7

[21] Jae-Han Lee and Chang-Su Kim. Multi-loss rebalancing algorithm for monocular depth estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 785–801. Springer, 2020. 3

[22] Bo Li, Chunhua Shen, Yuchao Dai, Anton Van Den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1119–1127, 2015. 3, 4

[23] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 38(10):2024–2039, 2015. 4, 6

[24] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021. 5, 7

[25] Xiaoxiao Long, Cheng Lin, Lingjie Liu, Wei Li, Christian Theobalt, Ruigang Yang, and Wenping Wang. Adaptive surface normal constraint for depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12849–12858, 2021. 4

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 7

[27] Jeff Michels, Ashutosh Saxena, and Andrew Y Ng. High speed obstacle avoidance using monocular vision and reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 593–600, 2005. 1

[28] Takaaki Nagai, Takumi Naruse, Masaaki Ikehara, and Akira Kurematsu. Hmm-based surface reconstruction from single images. In *Proceedings of the International Conference on Image Processing (ICIP)*, volume 2, pages II–II. IEEE, 2002. 1

[29] 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 (NIPS)*, 32, 2019. 7

[30] Vaishakh Patil, Christos Sakaridis, Alex Liniger, and Luc Van Gool. P3depth: Monocular depth estimation with a piecewise planarity prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 4, 6

[31] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems (NIPS)*, 34:12116–12128, 2021. 5

[32] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12179–12188, 2021. 4, 6

[33] Ashutosh Saxena, Sung Chung, and Andrew Ng. Learning depth from single monocular images. *Advances in Neural Information Processing Systems (NIPS)*, 18, 2005. 1, 3

[34] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *Advances in Neural Information Processing Systems (NIPS)*, 31(5):824–840, 2008. 4, 6

[35] Wenzhe Shi, Jose Caballero, Ferenc Huszár, 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, 2016. 4

[36] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 746–760. Springer, 2012. 2, 7

[37] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576, 2015. 7

[38] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 32–42, 2021. 7

[39] 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 (NIPS)*, 30, 2017. 3

[40] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, 2018. 3

[41] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 3

[42] Junyuan Xie, Ross Girshick, and Ali Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*,

pages 842–857. Springer, 2016. 3

[43] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5684–5693, 2019. 3, 4, 6

[44] Zehao Yu, Lei Jin, and Shenghua Gao. P2net: Patchmatch and plane-regularization for unsupervised indoor depth estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 206–222. Springer, 2020. 3, 4

[45] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Newcrfs: Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 3, 4, 6, 7