Transformer-based Monocular Depth Estimation with Attention Supervision

Wenjie Chang
changwj@mail.ustc.edu.cn
Yueyi Zhang*
zhyuey@ustc.edu.cn
Zhiwei Xiong
zwxiong@ustc.edu.cn

Department of Electronic Engineering
and Information Science,
University of Science and Technology of
China, Hefei, China

Abstract

Transformer, which excels in capturing long-range dependencies, has shown great performance in a variety of computer vision tasks. In this paper, we propose a hybrid network with a Transformer-based encoder and a CNN-based decoder for monocular depth estimation. The encoder follows the architecture of classical Vision Transformer. To better exploit the potential of the Transformer encoder, we introduce the Attention Supervision to the Transformer layer, which enhances the representative ability. The down-sampling operations before the Transformer encoder lead to degradation of the details in the predicted depth map. Thus, we devise an Attention-based Up-sample Block and deploy it to compensate the texture features. Experiments on both indoor and outdoor datasets demonstrate that the proposed method achieves the state-of-the-art performance on both quantitative and qualitative evaluations. The source code and trained models can be downloaded at https://github.com/WJ-Chang-42/ASTransformer.

1 Introduction

Depth estimation plays a crucial role in contemporary computer vision tasks, such as 3D face recognition and VR/AR. Currently, there are two main ways for depth estimation. One is the active way, getting the depth via illuminating coded signals, such as ToF [18, 27] and structured light cameras [1, 35, 38]. The other follows a passive way, which is usually based on multi-view geometry, such as stereo depth estimation [16, 21, 24], structure from motion [26, 32] and depth from light field [19, 20]. Different from the mentioned methods, Monocular Depth Estimation (MDE) aims to recover the depth value of each pixel from a single RGB image.

Previously, researchers utilize structural prior knowledge to reconstruct the depth map of a scene, such as texture, object shape, edge orientations which are related with 3D information [13, 25, 31]. Recently, with the development of deep learning, the performance of MDE has been significantly improved by CNN-based models [5, 8, 9, 12, 30]. The success of CNN models is due to that the convolution operation is able to extract the structural prior

* Corresponding Author: Yueyi Zhang

knowledge from training data. The Transformer architecture, which is first introduced to natural language processing, has also demonstrated outstanding performances on high-level vision tasks, such as image classification, object detection and segmentation. Compared with CNN-based networks, Transformer-based models expand the receptive field and are able to learn global information of an image.

In this paper, we combine a Transformer-based encoder with a CNN-based decoder together to jointly solve the MDE problem. However, we notice that simply employing the Transformer-based encoder cannot provide satisfying depth results, which perhaps is due to that there are no specific objects to concentrate on for low-level vision tasks. Thus, the attention scheme of Transformer needs to be carefully tuned for low-level vision tasks. To solve this problem, we introduce Attention Supervision (AS) to the Transformer-based encoder. Specifically, we calculate an attention map of each pixel based on the ground-truth depth and add two attention loss terms in the loss function to provide more guidance for the convergence of the Tranformer-based encoder. Meanwhile, limited by computing resources, the original RGB images need to be down-sampled to fit the giant Transformer-based model. This leads to detail degradation in the predicted depth map. To this end, we propose Attention-based Up-sample Block (AUB) to compensate the texture loss. AUB utilizes the attention information learned from the Transformer layer to generate high resolution feature map without extra parameter consumption in the encoder. The contribution of this work can be summarized as follows.

1. A hybrid encoder-decoder network is proposed for MDE. The Vision Transformer architecture is utilized in the encoder, which extracts features for global 3D information. Attention Supervision is introduced to the loss function design, which provides guidance for the convergence of the Transformer layer.

2. A novel Attention-based Up-sample Block, without extra parameter consumption in the encoder, is proposed to compensate the texture loss due to image down-sampling.

3. Experimental results demonstrate that our proposed method achieves superior performances on indoor (NYU Depth V2) and outdoor (KITTI) datasets in both quantitative and qualitative ways.

2 Related work

2.1 Monocular Depth Estimation

Early works on MDE mainly focused on handcrafted features. For example, Torralba and Oliva [31] computed the mean depth of the scene from the perspective of object size. Saxena *et al.* [23] employed designed convolution filter for extracting image features at multiple scales to improve the performance on depth estimation. Recently CNN-based deep learning networks have shown great performances on vision tasks including MDE. Eigen *et al.* [8] first utilized CNN to predict depth from a single image. Chen *et al.* [5] proposed the residual pyramid-based network to learn global structural information for MDE. Huan *et al.* [9] considered the depth prediction task as a classification problem, which is further solved by a CNN-based classification network. Song *et al.* [30] designed a Laplacian pyramid-based network to precisely estimate the depth boundary. Yin *et al.* [36] introduced a geometric constraint named virtual normal to predict the depth map. Lam *et al.* [12] proposed an attention module after encoder to help the network to learn the planar structures from the scene.

2.2 Transformer Network

Transformer is one of the most important architectures in natural language processing because it could discover the relation between the input words in a global size. Recently, Transformer structure shows great potential in vision tasks. DETR [3] utilized a traditional Transformer-based encoder-decoder structure for object detection. ViT [7] deployed a giant Transformer encoder and achieved superior performance on image classification. SETR [39] combined ViT [7] with a CNN-based decoder together and got reliable image segmentation results. Wang *et al.* [34] proposed VisTR, an end-to-end model based on Transformer, for video instance segmentation. Trackformer [17] designed a new tracking-by-attention paradigm based on Transformer for multi-object tracking. Chen *et al.* [4] built a large pre-trained Transformer-based model for image restoration. Ren *et al.* [22] proposed a dense prediction network DPT, which leveraged vision transformers in place of CNN. Bhat *et al.* [2] introduced a monocular depth prediction network AdaBins, utilizing Transformers to classify the depth range of each pixel.

3 Approach

In this section, we introduce the details of our proposed approach. We first present the overview of the architecture in Sec. 3.1. Then we introduce AS in Sec. 3.2 and AUB in Sec. 3.3. The loss function we use to supervise depth prediction and guide attention learning will be introduced in Sec. 3.4.

3.1 Overview

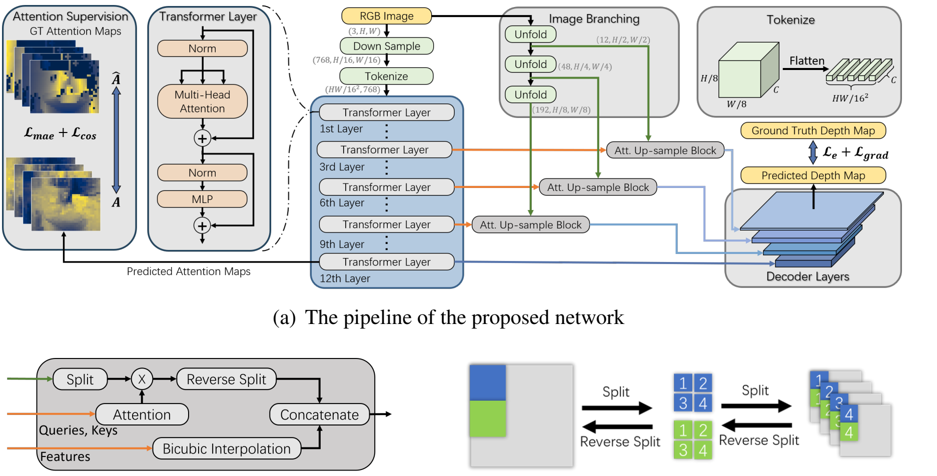
To recover depth from a single RGB image $I \in \mathbb{R}^{3\times H\times W}$, we first perform a 16× down-sampling operation via convolution. Then we convert the generated features $F \in \mathbb{R}^{768\times \frac{H}{16}\times\frac{W}{16}}$ via a flattening operation to get the tokens $T \in \mathbb{R}^{\frac{HW}{256}\times 768}$, which are fed to the Transformer encoder. Our Transformer-based encoder follows the design of ViT [7], which receives 1D sequence of token embeddings as input. There are 12 Transformer layers in the encoder, each of which consists of Normalization, Multi-head attention and MLP operations. For the 3rd, 6th, 9th and 12th layers, we output features, queries and keys. The output of the 12th layer is directly fed to the decoder. The outputs of other three layers are fed to AUB. These blocks up-sample and generate features at different scales in the decoder. The decoder of our proposed network is composed of CNN layers at four scales. In practice, our decoder design is the same with [30]. The layer at the uppermost scale outputs the final depth image. Fig. 1(a) presents the pipeline of the proposed approach.

3.2 Attention Supervision

Self-Attention (SA) is one of the key components in Transformer, which extracts the relationships among tokens in a sequence. We denote the input sequence as $\mathbf{z} \in \mathbb{R}^{N\times M}$. The basic Self-Attention operation is formulated as follows.

$$[\mathbf{q},\mathbf{k},\mathbf{v}] = MLP(\mathbf{z}) \quad \mathbf{A} = SOFTMAX(\mathbf{q}\mathbf{k}^{\top}/\sqrt{M}), \quad \mathbf{A} \in \mathbb{R}^{N\times N} \tag{1}$$

$$SA(\mathbf{z}) = \mathbf{A}\mathbf{v} \tag{2}$$



(a) The pipeline of the proposed network

(b) Attention-based Up-sample Block

(c) Split and reverse split

Figure 1: An overview of the proposed approach. (a) shows the pipeline of our proposed network. A Raw RGB image is down-sampled and resized to generate tokens. The Transformer layers generate features with the same size, thus we need to up-sample the feature maps from Transformer layers to form the multi-scale input of the decoder. (b) shows the architecture of the proposed AUB. The green route is from Image Branching. The orange route is from Transformer Layer. Queries and keys are the temporary variables which are used to calculate self-attention. (c) shows split operation and reverse split operation in AUB. The split operation firstly cuts features from Image Branching into small patches. Then pixels in the same position of patches will be combined together as small feature maps to fit the dimension of attention matrix. The reverse split operation follows opposite procedures.

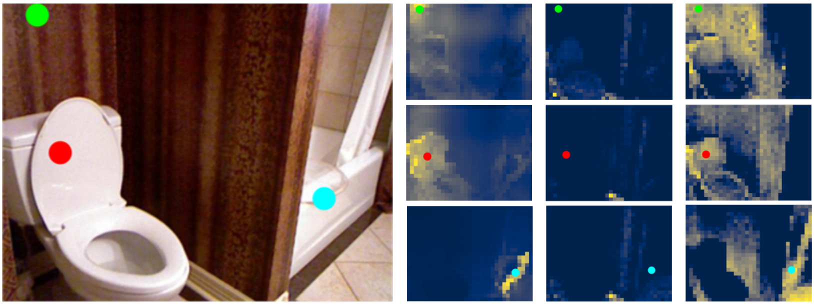
The input sequence $\mathbf{z}$ first passes a Muti-Layer Perception to get the presentation of $\mathbf{q}, \mathbf{k}, \mathbf{v} \in \mathbb{R}^{N \times M}$ which represent query, key and value respectively. Then the attention information $A_{i,j}$ is calculated from q_i and k_j which reflects the similarity between tokens in position i and j. The final output is generated by multiplying the attention matrix $\mathbf{A}$ by the value $\mathbf{v}$.

However, when directly applying the above attention mechanism to depth estimation, the attention maps extracted from the last Transformer layer do not show enough 3D structure information (Fig. 2(c)). To guide Transformer learn useful information, we propose AS under the assumption that pixels in the same depth level should have higher values in the attention map. The ground-truth of attention map $\hat{A}_{i,j}$, which is calculated from ground-truth depth map, is formulated as

$$\hat{A}_{i,j} = SOFTMAX(-\lambda |\hat{D} - \hat{d}_{i,j}|), \tag{3}$$

where $\hat{D}$ represents the depth map which is 16× down-sampled from the ground-truth depth map and normalized to $[0,1]$, $\hat{d}_{i,j}$ represents the depth value at position (i,j), λ is a hyper-parameter which controls the attention range. We set λ as 8 in our work. The supervision is only added in the last Transformer layer in the decoder. Meanwhile, the loss function needs revision after introducing AS, which will be illustrated in the following.

In Fig. 2, we visualize the attention maps extracted from the last Transformer layer and from DETR [3], a Transformer-based model for object detection, on a single image. The sub-figures demonstrate that without our AS, the network cannot get useful attention maps. After



(a) RGB Image (b) DETR (c) w/o AS (d) w/ AS

Figure 2: (a) shows an example RGB image. (b) shows the attention maps from DETR [3], a Transformer-based model for object detection. Individual instances are separated out from attention map when reference pixel is on the specific object. (c) and (d) show effects of AS on the attention maps extracted from the last Transformer layer in our architecture.

adding AS, the attention maps are able to reflect the global attention information related to the task, which are similar to the attention maps generated by DETR. We will quantitatively demonstrate the effectiveness of AS in Sec. 4.3.

3.3 Attention-based Upsample Block

From the previous introduction of our Transformer encoder, it can be seen that the output of Transform layers are 16× smaller than the input RGB image. Direct bi-cubic up-sampling may lead to texture degradation. We still need operations to perform optimized up-sampling and provide features to the corresponding layer of the decoder. Thus, we design an Image Branching module, which is shown in Fig. 1(a), to provide source information with appropriate dimensions. With this module, we further propose Attention-based Up-sample Block (AUB) to tackle the above-mentioned texture degradation problem. Fig. 1(b) shows the design of this block.

The input of AUB is the outputted features, queries and keys of the Transformer layer and the unfolded images from the Image Branching module. Specifically, features from Transformer layers are directly up-sampled to the corresponding scale in the decoder via bi-cubic interpolation. The unfolded images first passes a split operation, then are multiplied by attention maps, which are generated by the queries and keys from the Transformer layer. After a reverse split operation, the result is concatenated with the up-sampled feature. The split and reverse split operation are shown in Fig. 1(c). In this way, we make full use of the attention information learned by Transformer and compensate the missing details due to image down-sampling. It can be seen that there are no extra parameters cost in AUB.

3.4 Loss Function

Depth Loss. Usually, the depth data are dense in the nearby area but sparse in a distance. To solve the problem of uneven data distribution, we use the loss function proposed by [8] to measure the depth distance, which computes the depth errors in log space between ground-truth and predicted depth data. Mathematically, the depth loss is formulated as

$$\mathcal{L}_e(y,\hat{y}) = \sqrt{\frac{1}{n}\sum_{i\in V} e_i^2 - \frac{\alpha}{n^2}\left(\sum_{i\in V} e_i\right)^2}, \quad e_i = \log \hat{y}_i - \log y_i \tag{4}$$

where $\hat{y}$ is the ground-truth depth map, y is the predicted depth map, V is a set of valid pixels in the depth map and n is the total number of valid pixels. The factor α is set to 0.85 following [14].

Gradient Loss. The gradient loss is the L1 loss over the gradient g of the depth image. It can be denoted as

$$\mathcal{L}_{grad}(y,\hat{y}) = \frac{1}{n}\sum_{p}^{n} \left|g_{\mathbf{x}}(y_p,\hat{y}_p)\right| + \left|g_{\mathbf{y}}(y_p,\hat{y}_p)\right|, \tag{5}$$

where $g_{\mathbf{x}}$ and $g_{\mathbf{y}}$ compute the discrepancy in both x and y components for the depth image gradients of y and $\hat{y}$.

Attention loss. To assist the Transformer layer in learning depth variation information, we add two attention loss terms as follows

$$\mathcal{L}_{mae} = \frac{1}{(hw)^2}\sum \left|A_{i,j} - \hat{A}_{i,j}\right|, \quad A_{i,j} \in \mathbb{R}^{h\times w} \tag{6}$$

$$\mathcal{L}_{cos} = \frac{1}{(hw)}\sum \left[\left(1 - \mathbf{cos}\left(A_{i,j},\hat{A}_{i,j}\right)\right) + \left(1 - \mathbf{cos}\left(A_{i,j}^{\top},\hat{A}_{i,j}^{\top}\right)\right)\right], \quad A_{i,j} \in \mathbb{R}^{h\times w} \tag{7}$$

where (h,w) is the resolution of attention map, $\hat{A}_{i,j}$ and $A_{i,j}$ denote the ground-truth and predicted attention map at position (i,j), **cos** denotes cosine similarity calculation.

Overall Loss. The overall loss is the summation of the aforementioned loss terms:

$$\mathcal{L}_{total} = \begin{cases} \mathcal{L}_e + \mathcal{L}_{mae} + \mathcal{L}_{cos} & \text{epochs} \leq 10 \\ \mathcal{L}_e + \mathcal{L}_{grad} + \mathcal{L}_{mae} + \mathcal{L}_{cos} & \text{epochs} > 10 \end{cases} \tag{8}$$

In the first 10 epochs, we train the model with depth loss and attention loss. Gradient loss will be added after 10 epochs. We utilize the training strategy following [30] to avoid unstable training performance caused by the gradient loss.

4 Experiments

In this section, we describe our experimental details and compare the performance of our network with existing state-of-the-art methods. We evaluate the performance of our work on two widely-used datasets: indoor dataset NYU Depth V2 and outdoor dataset KITTI. Moreover, we perform the ablation studies to explain how the AS and AUB influence the performance.

4.1 Datasets

NYU Depth V2. The NYU Depth V2 dataset [29] consists of 120K pairs of RGB and depth images. These image pairs are captured by Microsoft Kinect sensor under 464 indoor

Method	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$	$rel \downarrow$	$log_{10} \downarrow$	$rmse \downarrow$
DORN [9]	0.828	0.965	0.992	0.115	0.051	0.509
SARPN [5]	0.878	0.977	0.994	0.111	0.048	0.514
VNL [36]	0.875	0.976	0.994	0.111	0.048	0.416
BTS [14]	0.885	0.978	0.994	0.110	0.047	0.392
Adabins [2]	0.886	0.982	0.995	0.112	0.047	0.401
DPT-Large [22]	0.886	0.980	0.994	0.114	0.047	0.398
Lapdepth [30]	0.885	0.979	0.995	0.110	0.047	0.393
Ours	**0.902**	**0.985**	**0.997**	**0.103**	**0.044**	**0.374**

Table 1: Comparisons with state-of-the-art MDE approaches on the NYU Depth v2 Dataset. The best results on each metric are marked in bold.

Method	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$	$rel \downarrow$	$rmse_{log} \downarrow$	$rmse \downarrow$
Godard [11]	0.916	0.980	0.994	0.085	0.135	3.938
VNL [36]	0.938	0.990	0.998	0.072	0.117	3.258
DORN [9]	0.932	0.984	0.994	0.072	0.120	2.727
BTS [14]	0.956	0.993	0.998	0.059	0.096	2.756
Adabins [2]	**0.964**	0.994	**0.999**	0.060	0.091	2.765
DPT-Large [22]	0.961	0.994	**0.999**	**0.058**	**0.089**	2.710
Lapdepth [30]	0.962	0.994	**0.999**	0.059	0.091	**2.446**
Ours	0.963	**0.995**	**0.999**	**0.058**	**0.089**	2.685

Table 2: Comparisons with state-of-the-art MDE approaches on KITTI Eigen split (80m cap). The best results on each metric are marked in bold.

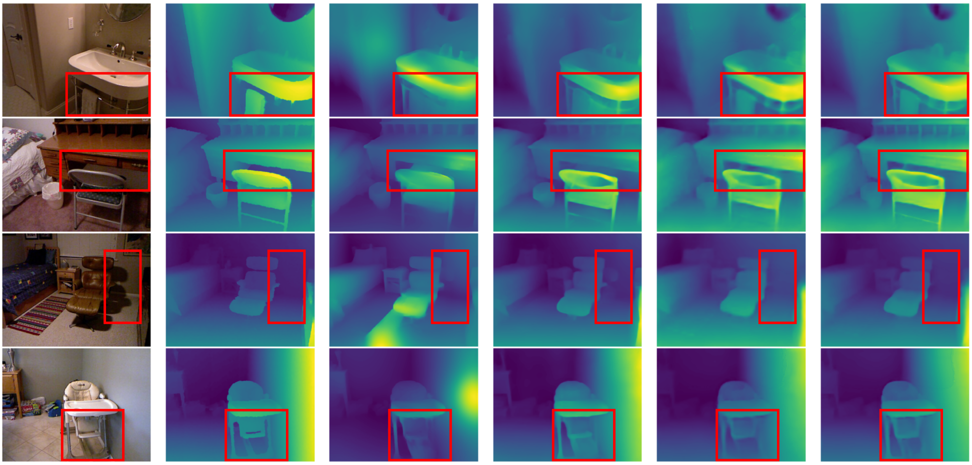
scenes with the resolution of 480 × 640 pixels. We apply the training/testing split following [8, 14]. The training set contains 36,253 images from 249 scenes. The testing set consists of 654 images from remaining 215 scenes. Training and testing samples are cropped to the resolution of 416 × 512 with the same configuration in [30].

KITTI. The KITTI dataset [10] is a large-scale outdoor dataset, which contains RGB and depth image pairs in autonomous driving scenarios. The resolution of acquired images is 375 × 1242 pixels. For a fair comparison, we adopt the split strategy introduced in [8]. According to this scheme, the training set is composed of 23,488 images from 32 scenes. The testing set contains 697 images selected from the remaining 29 scenes. The training and testing samples are cropped to the resolution 352 × 704 with the same operation as [30].

4.2 Training Details

We implement our model with the PyTorch framework. The Transformer encoder is pre-trained on the ImageNet dataset [6]. We train our model on 8 NVIDIA GeForce TitanXp Graphics cards. The weight decaying factor is set to 0.0005 for the encoder and zero for decoder with AdamW optimizer [15] where the power and momentum are set to 0.9 and 0.999. Our proposed network is first trained for 30 epochs without AUB. After 30 epochs, the AUB is deployed for training 15 epochs.

We utilize random rotation, horizontal flip and random adjustments for data augmentation. The random rotation is within the range of $[-3°, 3°]$. The horizontal flip is performed



(a) RGB (b) Ground Truth (c) SARPN [5] (d) LapDepth [30] (e) Ours w/o AUB (f) Ours

Figure 3: Qualitative comparison with other methods on the NYU Depth V2 dataset.

with 50% probability. Random adjustments in the scale range [0.9, 1.1] are applied on the brightness, color and gamma values.

4.3 Evaluation

Quantitative Results. We compare our method against state-of-the-art methods quantitatively with the following metrics: mean absolute relative error (*rel*), root mean square error (*rmse*) and threshold accuracy (δ_i). We additionally calculate absolute error in log space ($\log_{10}$) for NYU Depth V2 and root mean square error in log space ($rmse_{\log}$) for KITTI. The mathematical expressions of the evaluation metrics are presented in the following.

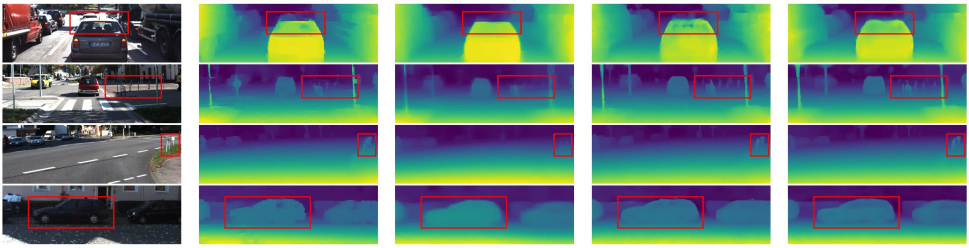
$$rel = \frac{1}{n}\sum_{p}^{n}\frac{|y_p - \hat{y}_p|}{\hat{y}_p}, \qquad rmse = \sqrt{\frac{1}{n}\sum_{p}^{n}(y_p - \hat{y}_p)^2 \Bigg)}$$

$$rmse_{log} = \sqrt{\left(\frac{1}{n}\sum_{p}^{n}(\log y_p - \log \hat{y}_p)^2\right)}, \qquad \log_{10} = \frac{1}{n}\sum_{p}^{n}\left|\log_{10}(y_p) - \log_{10}(\hat{y}_p)\right|$$

$$\delta = \ \% \text{ of } y_p \text{ s.t.} \max\left(\frac{y_p}{\hat{y}_p}, \frac{\hat{y}_p}{y_p}\right) = \delta < thr \text{ for } thr = 1.25, 1.25^2, 1.25^3$$

For SAPRN [5], we directly apply the pre-trained model [1]. For DPT [22] and AdaBins [2], we train the models with our experimental setting. In [22], there are two DPT networks, of which we choose DPT-Large with 300M parameters for comparison. Table 1 and Table 2 compare the performance of our method with the state-of-the-art methods on NYU Depth V2 and KITTI. The proposed method achieves the best performance on NYU Depth V2 in terms of all metrics and it also provides a superior evaluation result on KITTI in terms of most metrics.

[1] https://github.com/Xt-Chen/SARPN



(a) RGB (b) Ground Truth (c) VNL [36] (d) LapDepth [30] (e) Ours

Figure 4: Qualitative comparison with other methods on the KITTI dataset.

Method	Up-sample	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$	*rel* $\downarrow$	$log_{10} \downarrow$	*rmse* $\downarrow$
WR 101 [37]	bicubic	0.866	0.981	0.996	0.120	0.050	0.398
Ours w/o AS	bicubic	0.892	0.983	0.996	0.107	0.046	0.394
Ours w/ AS	bicubic	0.900	0.985	0.995	0.104	0.045	0.382
Ours w/ AS	PixelShuffle [28]	0.895	0.984	0.997	0.106	0.045	0.382
	CARAFE [33]	0.857	0.972	0.992	0.124	0.154	0.427
	DUB	0.900	0.983	0.995	0.104	0.045	0.378
	AUB	**0.902**	**0.985**	**0.997**	**0.103**	**0.044**	**0.374**

Table 3: Ablation results on the NYU Depth V2 dataset. ‘WR 101’ denotes the comparing neural network with Wide ResNet 101 as the encoder.

Qualitative Results. Fig. 3 shows visual results of SARPN [5], LapDepth [30] and our method. Fig. 3(e) shows the depth results without AUB. SARPN and LapDepth fail in reconstructing the complex corner from first and second rows. The depth results in the third row show our method has a better performance on the shadow region. From Fig. 3(e) and 3(f), it can be observed that the AUB module helps the network get sharper depth maps. In Fig. 4, we show qualitative comparisons on the KITTI dataset among our method, VNL [36] and LapDepth [30]. We can see that the overexposed region and small objects get better reconstruction by our proposed method.

Ablation Study. We perform ablation study to investigate the effectiveness of AS and AUB. To compare our Transformer-based encoder with the CNN-based encoder, we construct a similar network with a convolutional encoder ‘Wide ResNet 101’, whose parameter size is in the same level as our Transformer-based encoder (‘Wide ResNet 101’ [37], 83M; Our Encoder, 88M). To verify the effectiveness of AUB, we replace AUB with several different up-sample methods (e.g. bicubic, CARAFE [33], PixelShuffle [28], Direct Up-sample Block (DUB)). The difference between AUB and DUB is that DUB doesn’t have the attention scheme in the up-sample block. Detailed illustrations of AUB and DUB are shown in our Supplementary Material. All ablation experiments are performed on the NYU Depth V2 dataset, and share the same decoder architecture and training strategy. Table 3 shows the quantitative evaluation results of ablation experiments. The results demonstrate that the Transformer-based encoder without AS and AUB shows a similar performance to that with the CNN-based encoder. Both AS and AUB improve the performance when applied in our model. Our AUB outperforms all other up-sample methods.

To verify the effectiveness of the loss terms we use in AS, we train models using L_{cos}

Method	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$	$rel \downarrow$	$log_{10} \downarrow$	$rmse \downarrow$
Baseline	0.892	0.983	0.996	0.107	0.046	0.394
L_{mae}	0.889	0.983	0.996	0.109	0.047	0.395
L_{cos}	0.895	0.983	**0.997**	0.106	0.046	0.388
$L_{cos} + L_{mae}$	**0.900**	**0.985**	0.995	**0.104**	**0.045**	**0.382**

Table 4: Effectiveness analyses of attention losses on the NYU Depth V2 dataset. The baseline denotes our model without AS and AUB. We utilize bicubic as the up-sample strategy in the above experiments.

and L_{mae} separately. The results are shown in Table 4. Using L_{cos} has a better performance than using L_{mae}. Combining the two losses together could get the best result.

5 Conclusion

In this paper, we propose a Transformer-based deep neural network for MDE. AS is introduced to provide guidance for the convergence of the Transformer encoder. To make full use of the global depth information learned by Transformer layers, we propose AUB, which could bring the low scale attention information to the high resolution features without extra parameter cost. Experimental results demonstrate that our proposed method achieves superior performances on both indoor dataset NYU Depth V2 and outdoor dataset KITTI.

6 Acknowledgements

This work was supported by National Natural Science Foundation of China under grant No. 61901435.

References

[1] Chadi Albitar, Pierre Graebling, and Christophe Doignon. Robust structured light coding for 3d reconstruction. In *IEEE 11th International Conference on Computer Vision*, pages 1–6. IEEE, 2007.

[2] 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, June 2021.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[4] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12299–12310, 2021.

[5] Xiaotian Chen, Xuejin Chen, and Zheng-Jun Zha. Structure-aware residual pyramid network for monocular depth estimation. In *International Joint Conferences on Artificial Intelligence*, page 694–700, 2019.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[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. *International Conference on Learning Representations*, 2021.

[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*, 27:2366–2374, 2014.

[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*, pages 2002–2011, 2018.

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

[11] 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*, pages 270–279, 2017.

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

[13] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth extraction from video using non-parametric sampling. In *European Conference on Computer Vision*, pages 775–788. Springer, 2012.

[14] 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.

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

[16] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5695–5703, 2016.

[17] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021.

[18] James Noraky and Vivienne Sze. Low power depth estimation of rigid objects for time-of-flight imaging. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(6):1524–1534, 2019.

[19] Jiayong Peng, Zhiwei Xiong, Dong Liu, and Xuejin Chen. Unsupervised depth estimation from light field using a convolutional neural network. In *2018 International Conference on 3D Vision*, pages 295–303, 2018.

[20] Jiayong Peng, Zhiwei Xiong, Yicheng Wang, Yueyi Zhang, and Dong Liu. Zero-shot depth estimation from light field using a convolutional neural network. *IEEE Transactions on Computational Imaging*, 6:682–696, 2020.

[21] AN Rajagopalan, Subhasis Chaudhuri, and Uma Mudenagudi. Depth estimation and image restoration using defocused stereo pairs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1521–1525, 2004.

[22] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021.

[23] Ashutosh Saxena, Sung H Chung, Andrew Y Ng, et al. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems*, volume 18, pages 1–8, 2005.

[24] Ashutosh Saxena, Jamie Schulte, Andrew Y Ng, et al. Depth estimation using monocular and stereo cues. In *International Joint Conferences on Artificial Intelligence*, volume 7, pages 2197–2203, 2007.

[25] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2008.

[26] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[27] Sebastian Schuon, Christian Theobalt, James Davis, and Sebastian Thrun. High-quality scanning using time-of-flight depth superresolution. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–7. IEEE, 2008.

[28] 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*, pages 1874–1883, 2016.

[29] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012.

[30] M. Song, S. Lim, and W. Kim. Monocular depth estimation using laplacian pyramid-based depth residuals. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[31] Antonio Torralba and Aude Oliva. Depth estimation from image structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1226–1238, 2002.

[32] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 203(1153):405–426, 1979.

[33] Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin. Carafe: Content-aware reassembly of features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3007–3016, 2019.

[34] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2021.

[35] Zhiwei Xiong, Yueyi Zhang, Feng Wu, and Wenjun Zeng. Computational depth sensing : Toward high-performance commodity depth cameras. *IEEE Signal Processing Magazine*, 34(3):55–68, 2017.

[36] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5684–5693, 2019.

[37] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference*, pages 87.1–87.12. BMVA Press, September 2016.

[38] Yueyi Zhang, Zhiwei Xiong, Zhe Yang, and Feng Wu. Real-time scalable depth sensing with hybrid structured light illumination. *IEEE Transactions on Image Processing*, 23 (1):97–109, 2014.

[39] 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 Conference on Computer Vision and Pattern Recognition*, 2021.