SideRT: A Real-time Pure Transformer Architecture for Single Image Depth Estimation

Chang Shu, **Ziming Chen**, **Lei Chen**, **Kuan Ma**, **Minghui Wang** and **Haibing Ren**
Meituan Group
{shuchang02,chenziming02,makuan@meituan.com,chenlei90,wangminghui08,renhaibing}@meituan.com

Abstract

Since context modeling is critical for estimating depth from a single image, researchers put tremendous effort into obtaining global context. Many global manipulations are designed for traditional CNN-based architectures to overcome the locality of convolutions. Attention mechanisms or transformers originally designed for capturing long-range dependencies might be a better choice, but usually complicates architectures and could lead to a decrease in inference speed. In this work, we propose a pure transformer architecture called SideRT that can attain excellent predictions in real-time. In order to capture better global context, Cross-Scale Attention (CSA) and Multi-Scale Refinement (MSR) modules are designed to work collaboratively to fuse features of different scales efficiently. CSA modules focus on fusing features of high semantic similarities, while MSR modules aim to fuse features at corresponding positions. These two modules contain a few learnable parameters without convolutions, based on which a lightweight yet effective model is built. This architecture achieves state-of-the-art performances in real-time (51.3 FPS) and becomes much faster with a reasonable performance drop on a smaller backbone Swin-T (83.1 FPS). Furthermore, its performance surpasses the previous state-of-the-art by a large margin, improving AbsRel metric 6.9% on KITTI and 9.7% on NYU. To the best of our knowledge, this is the first work to show that transformer-based networks can attain state-of-the-art performance in real-time in the single image depth estimation field. Code will be made available soon.

1 Introduction

Single image depth estimation (SIDE) has a pivotal role in extracting 3D geometry, which has a wide range of practical applications, including automatic driving, robotics navigation, and augmented reality. The main difficulty of SIDE is that: unlike other 3D vision problems, multiple views are missing to establish the geometric relationship as 3D geometric clues can only be dug from a single image.

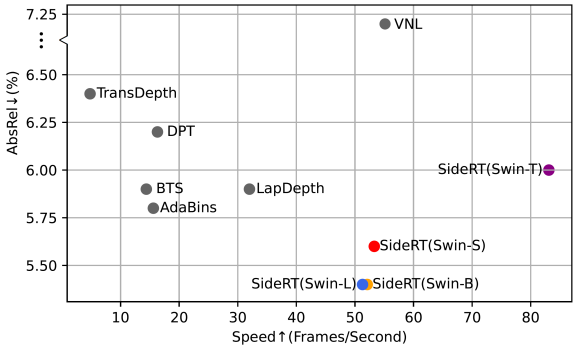

Figure 1: Inference speed versus AbsRel performance on the test set of the KITTI dataset. Previous state-of-the-art models are marked as grey points. Note that SideRT models achieve better accuracy with state-of-the-art methods at a much faster speed.

In order to solve this ill-posed problem, the ability to extract global context is paid tremendous attention, which largely relies on the powerful learning capability of modern deep neural networks. CNN-based architectures [Eigen *et al.*, 2014; Fu *et al.*, 2018; Lee *et al.*, 2019; Qiao *et al.*, 2021] once dominate the SIDE field, due to the intrinsic locality of convolution, global context is only obtained near the bottleneck. The global context is usually maintained in low-resolution feature maps. Essential clues for 3D structures like local details are lost after consecutive convolutional operations. To obtain high-resolution global context, there has been a trend in SIDE field to enlarge receptive field via large backbones [Huang *et al.*, 2017; Xie *et al.*, 2017; Sun *et al.*, 2019], feature pyramid [Lin *et al.*, 2017], spatial pyramid pooling [He *et al.*, 2015] and atrous convolution [Chen *et al.*, 2017; Yang *et al.*, 2018].

Another paradigm to extract global context is taking advantage of the long-range dependency modeling capability of the attention mechanisms. The attention module [Vaswani *et al.*, 2017; Wang *et al.*, 2018] computes the responses at each position by estimating matching scores to all positions and gathering the corresponding embeddings accordingly, so a global receptive field is guaranteed. Using attention as the main component, transformers which are initially designed for nat-

ural language processing, are found more and more applications in the computer vision field [Dosovitskiy *et al.*, 2020; Liu *et al.*, 2021; Carion *et al.*, 2020]. Thanks to the powerful ability to establish long-range dependencies of attention mechanisms and transformers, integrating them into fully convolutional architectures [Ranftl *et al.*, 2021; Bhat *et al.*, 2021; Yang *et al.*, 2021] has pushed state-of-the-art performance forward a lot.

Since the attention mechanism is usually time- and memory-consuming, inference speed has to be compromised when using transformers or attention mechanisms. Many works have been devised for more efficient implementation, but similar works are rare in the SIDE field.

This paper explores how to achieve state-of-the-art performance in real-time when using transformers and attention mechanisms. We introduce the SIDE Real-time Transformer (SideRT) based on an encoder-decoder architecture. Swin transformers are used as the encoder. The decoder is built on a novel attention mechanism named Cross-Scale Attention (CSA) and a Multi-Scale Refinement module (MSR). Both CSA and MSR modules are global operations and work collaboratively. In CSA modules, finer-resolution features are augmented by coarser-resolution features according to attention scores defined by semantic similarity. In MSR modules, coarser-resolution features are merged to spatially corresponding finer-resolution features. Since a few learnable parameters are used in the proposed modules, feature maps at different scales are fused with a fair computational overhead. Based on CSA and MSR modules, we build a lightweight decoder that conducts hierarchical depth optimization progressively to get the final prediction in a coarse-to-fine manner. Furthermore, Multi-Stage Supervision (MSS) is added at each stage to ease the training process.

As depicted in Figure 1, the proposed SideRT significantly outperforms the previous state-of-the-art at a speed of 51.3 FPS. It improves the AbsRel metric from 0.058 to 0.054 on KITTI and from 0.103 to 0.093 on NYU. Moreover, SideRT can achieve 0.060 AbsRel on KITTI, and 0.124 AbsRel on NYU on smaller backbone Swin-T [Liu *et al.*, 2021] at a speed of 83.1 FPS and 84.4 FPS respectively. To the best of our knowledge, this is the first work to show that transformer-based networks can attain state-of-the-art performance in real-time in the single image depth estimation field.

2 Related Work

SIDE is a practical vision task for 3D scene understanding. Due to the ambiguity of 3D mapping in a single view, SIDE is ill-posed. However, with the help of deep learning, considerable progress has been made in the SIDE field.

CNN-based. [Eigen *et al.*, 2014] firstly brings CNNs to the SIDE task, and subsequent researchers introduce more powerful networks, like ResNet and DenseNet, for depth estimation. They aim to encourage networks to learn the global context better, which helps the model understand the depth distribution of the image scene and draw a more reliable inference. Many global manipulations like atrous spatial pyramid pooling module (ASPP), spatial pyramid pooling and feature pyramid are adopted to enlarge receptive field [Lee *et al.*, 2019; Fu *et al.*, 2018; Song *et al.*, 2021].

Attention-based. The attention mechanism can establish associations between all the pixels, thereby overcoming the problem of establishing long-range dependencies. This capability is favored by SIDE researchers for contextual knowledge extraction. [Lee *et al.*, 2021] takes advantage of the attention mechanism to perform feature learning on the patches to obtain higher prediction accuracy. [Xu *et al.*, 2021] designs channel attention and spatial attention modules to further improve the high-level context features and low-level spatial features. [Huynh *et al.*, 2020] explores the self-attention mechanism to establish pixel's association and treats this information as depth prior. [Aich *et al.*, 2021; Hao *et al.*, 2018] apply the attention mechanism to fuse multi-level features. [Jiao *et al.*, 2018] focuses on the distribution of depth prediction data and designs an attention-driven loss to improve the quality of depth prediction in long range.

Transformer-based. Transformer [Vaswani *et al.*, 2017], first applied to the NLP field, is a type of deep neural network mainly based on the self-attention mechanism. Thanks to its strong representation capabilities, researchers are looking for ways to apply transformer in the SIDE tasks. [Yang *et al.*, 2021] introduces ViT architectures [Dosovitskiy *et al.*, 2020] to this field to compensate the intrinsic locality of convolutions. [Ranftl *et al.*, 2021] leverages ViT in place of convolutional networks as a backbone and prove the ViT-based encoder is more effective than the convolution-based for SIDE. [Bhat *et al.*, 2021] utilizes ViT to perform global processing of the scene's information and subsequently learn adaptive dividing of the depth range.

Unlike prior works which use either fully convolutional networks or combine CNNs with transformers and attention mechanisms, we explore the possibility of building the architecture without convolutions. Furthermore, we prove that the state-of-the-art can still be achieved even without convolutions. We also demonstrate that even using heavy backbones like Swin-L, our model can still run in real-time.

3 Method

In this section, we first present a general framework of our network and then introduce in detail three major components of our networks, which are cross-scale attention, multi-scale refinement and multi-stage supervision. Figure 2 illustrates our SideRT architecture.

3.1 Overview

Our proposed SideRT has a simple yet efficient encoder-decoder architecture that predicts depth from a single image. We adopt Swin Transformers [Liu *et al.*, 2021] as our backbones, each image is divided into several 4×4 non-overlapping patches. The patch, along with relative positional embedding, is embedded by the linear projection layer and the result will be processed by a series of shift window based transformer blocks. Feature maps from all four stages will be utilized during decoding.

The input to the decoder is a set of multi-scale feature maps of four stages: (I) stage 1, $\frac{H}{4} \times \frac{W}{4} \times C$, (II) stage 2, $\frac{H}{8} \times \frac{W}{8} \times 2C$, (III) stage 3, $\frac{H}{16} \times \frac{W}{16} \times 4C$, and (IV)

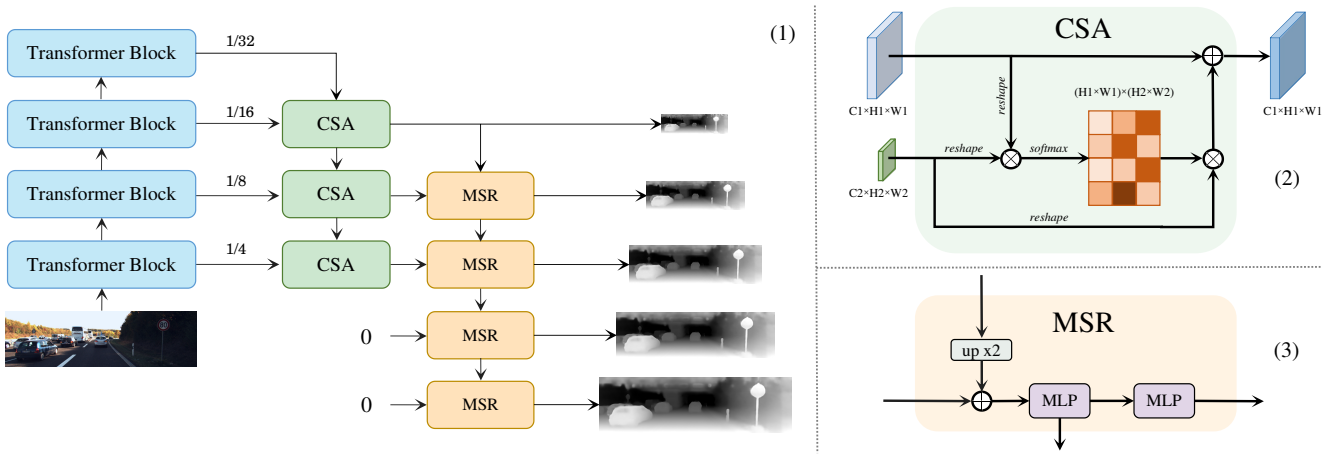

Figure 2: (1) Architecture overview. The input image is sent into a series of transformer blocks to obtain a feature pyramid. CSA and MSR modules progressively merge feature maps from adjacent scales to obtain more powerful feature representations. '0' means no input. (2) In CSA modules, finer-resolution features are augmented by coarser-resolution features with matching scores defined by semantic similarity. $\bigoplus$ and $\bigotimes$ respectively denote element-wise sum and matrix multiplication. (3) In MSR modules, coarser-resolution features are merged to spatially corresponding finer-resolution features.

stage 4, $\frac{H}{32} \times \frac{W}{32} \times 8C$. H and W respectively represent the height and width of input images, C denotes the number of channels of features from stage 1. Analogous with a standard feature pyramid, the decoder fuses feature maps progressively in a coarse-to-fine way. Our decoder consists two basic modules: cross-scale attention modules and multi-scale refinement modules (i.e. CSA and MSR modules in Figure 2). To obtain global context, CSA modules aim to fuse feature maps following the guidance of semantic similarity, while MSR modules aim to fuse feature maps according to spatial corresponding relationship. Fusion operations are conducted in a coarse-to-fine manner to get the final prediction, which keeps the same resolution as the input images.

3.2 Cross-Scale Attention

Our proposed CSA module consists of two parts: a linear layer to project each feature to the same number of channels and an attention-based fusion to fuse feature maps from adjacent scales according to semantic similarity. Denote input feature maps as $F_1 \in C_1 \times H_1 \times W_1$ and $F_2 \in C_2 \times H_2 \times W_2$, while F_1 is from the shallower stage and F_2 is from the deeper stage of the encoder. After linear projection, we propose an attention-based method to fuse these two feature maps:

$$F_{12} = L(F_1) + softmax(L(F_1) \times L(F_2)) \times L(F_2) \quad (1)$$

where $L(\cdot)$ is the linear projection operation. The linear layer will project F_1 from $C_1 \times H_1 \times W_1$ to $C \times (H_1 \times W_1)$, and F_2 from $C_2 \times H_2 \times W_2$ to $C \times (H_2 \times W_2)$ respectively.

For traditional CNN-based methods, global context information only exists near the encoder bottleneck and will be gradually weakened during the hierarchical upsampling of the decoder. Different from them, this CSA module is able to naturally capture global context dependency between feature maps from adjacent scales through calculating attention scores $softmax(L(F_1) \times L(F_2))$ across the entire feature map. Furthermore, this global context dependency contains both semantic-related and depth-related information, as illustrated in the visual analysis of Section 4.4.

3.3 Multi-Scale Refinement

Several multi-scale refinement (MSR) modules work collaboratively in a top-down hierarchy pyramid pattern. A coarser-resolution feature map from the higher MSR pyramid level and a finer-resolution feature map from the lower CSA pyramid level are fed into a MSR module. The MSR module outputs a refined feature map with the help of a low-level semantic but more accurately-localized feature map from the CSA pyramid. The output feature map of the MSR module and the next finer-resolution feature map from the lower CSA pyramid level will be fed into the next MSR module until the final feature map is generated. In a word, the whole refinement process is iterated in a coarse-to-fine way.

Each multi-scale refinement module includes three parts: a bilinear interpolation to upsample coarser-resolution feature map from MSR pyramid, an element-wise addition between the upsampled feature map and finer-resolution feature map from the CSA pyramid, and two MLP layers to reduce aliasing effect of upsampling and generate a depth map at each scale. The last two MSR modules are appended to generate the final result with the input resolution. In particular, unlike other MSR modules, they do not receive outputs from CSA modules.

Different from the fusion style of CSA modules which fuse features according to semantic similarity, MSR modules leverage upsampling to fuse features with corresponding spatial positions. Furthermore, the introduction of MLP layers facilitates information to flow globally.

3.4 Multi-Stage Supervision

In order to ease the training of the early stages of the whole architecture, we propose a multi-stage supervision (MSS) strategy at each scale to supervise the training process. In each

Method	Backbone	AbsRel↓	SqRel↓	RMSE↓	RMSE log↓	δ_1↑	δ_2↑	δ_3↑	Params↓	FPS↑
VNL	ResNext-101	0.072	-	3.258	0.117	0.938	0.990	0.998	90.4 M	54.8
DORN	ResNet-101	0.072	0.307	2.727	0.120	0.932	0.984	0.994	-	-
TransDepth	ResNet-50+ViT	0.064	0.252	2.755	0.098	0.956	0.994	**0.999**	247.4 M	4.8
DPT	VIT-Hybrid	0.062	-	2.573	0.092	0.959	0.995	**0.999**	123.0 M	16.4
BTS	ResNext-101	0.059	0.245	2.756	0.096	0.956	0.993	0.998	112.8 M	14.4
LapDepth	ResNext-101	0.059	0.212	2.446	0.091	0.962	0.994	**0.999**	73.0 M	32.0
AdaBins	EfficientNet-B5	0.058	0.190	2.360	0.088	0.964	0.995	**0.999**	78.0 M	15.6
Ours	Swin-T	0.060	0.206	2.441	0.093	0.959	0.995	**0.999**	**28.6 M**	**83.1**
	Swin-S	0.056	0.187	2.306	0.087	0.965	0.996	**0.999**	50.1 M	53.3
	Swin-B	**0.054**	**0.170**	**2.212**	0.083	0.971	0.996	**0.999**	89.2 M	52.1
	Swin-L	**0.054**	0.173	2.249	**0.082**	**0.972**	**0.997**	**0.999**	200.4 M	51.3

Table 1: Comparison to the state-of-the-art on KITTI dataset, best results are in bold.

Method	Backbone	AbsRel↓	RMSE↓	log10↓	δ_1↑	δ_2↑	δ_3↑	Params↓	FPS↑
SharpNet	ResNet-50	0.139	0.495	0.047	0.888	0.979	0.995	114.1 M	156.7
DORN	ResNet-101	0.115	0.509	0.051	0.828	0.965	0.992	-	-
LapDepth	ResNext-101	0.110	0.393	0.047	0.885	0.979	0.995	73.0 M	40.3
BTS	DenseNet-161	0.110	0.392	0.047	0.885	0.978	0.994	47.0 M	24.5
DPT	VIT-Hybrid	0.110	0.357	0.045	0.904	0.988	0.998	123.0 M	24.3
VNL	ResNext-101	0.108	0.416	0.048	0.875	0.976	0.994	90.4 M	53.6
TransDepth	ResNet-50+ViT	0.106	0.365	0.045	0.900	0.983	0.996	247.4 M	6.5
AdaBins	EfficientNet-B5	0.103	0.364	0.044	0.903	0.984	**0.997**	78.0 M	19.9
Ours	Swin-T	0.124	0.428	0.052	0.860	0.974	0.994	**28.6 M**	**84.4**
	Swin-S	0.108	0.380	0.046	0.892	0.982	0.996	50.1 M	53.2
	Swin-B	0.100	0.354	0.043	0.908	0.985	**0.997**	89.2 M	51.1
	Swin-L	**0.093**	**0.335**	**0.040**	**0.922**	**0.990**	**0.997**	200.4 M	50.3

Table 2: Comparison to the state-of-the-art on NYU dataset, best results are in bold.

stage, the loss between the prediction and the corresponding ground truth is calculated. Finally, multi-stage losses are weighted summed together.

Due to the limitation of 3D sensors, the depth data is dense at close areas whereas very sparse in the distance. To alleviate this imbalance problem, we adopt the square root loss function introduced in [Song *et al.*, 2021]. This loss calculates the difference of predicted depth value and the ground truth in the log space, as shown below:

$$L\left(y, y^{*}\right)=\sqrt{\frac{1}{n} \sum_{i \in V}^{N_V} d_{i}^{2}-\frac{\lambda}{n^{2}}\left(\sum_{i \in V}^{N_V} d_{i}\right)^{2}} \tag{2}$$

$$d_{i}=\log y_{i}-\log y_{i}^{*} \tag{3}$$

where y and y^* respectively represent the predicted depth map and the ground truth, V is the set of valid pixels in the depth map, N_V indicates the total number of valid pixels, and the balance coefficient lambda is set to 0.85.

4 Experiments

Firstly, we apply SideRT on two challenging datasets: KITTI and NYU. For both datasets, we show that SideRT can significantly outperform previous state-of-the-art methods at a much faster speed. At the end of this section, we conduct comprehensive ablations of different components and detailed visualization to verify the effectiveness of our method.

4.1 Implementation Details

The model is implemented with PyTorch. Swin transformers pretrained on ImageNet [Deng *et al.*, 2009] are used as encoders. The proposed model is trained from scratch for 160 epochs with a batch size of 6 through the AdamW optimizer. The weight decaying factor is set to $5e^{-4}$ and the learning rate is set to $1e^{-4}$. It takes 8 NVIDIA A100 SXM GPUs in the training process. Data augmentation is performed during the training phase to avoid overfitting problems. Images from KITTI and NYU are randomly cropped to 704×352 pixels and 512×416 pixels respectively. In addition, we randomly adjust the scale factor, rotate the input color images within a

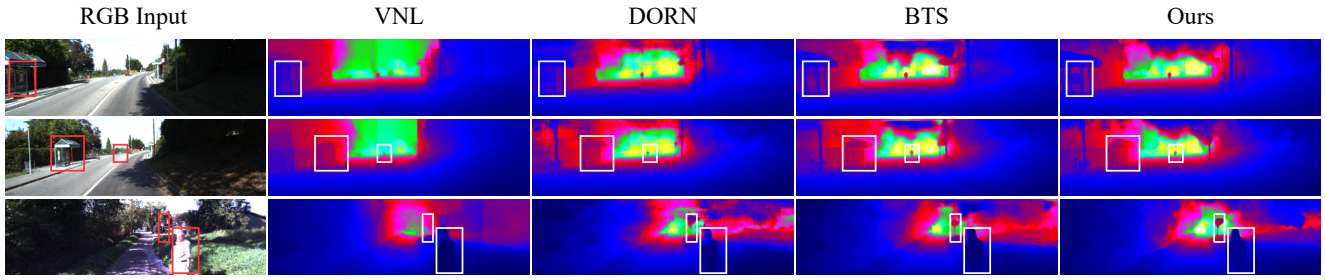

Figure 3: Visualization of depth predictions in the KITTI dataset.

AbsRel : $\frac{1}{\lvert D\rvert}\sum_{d\in D}\lvert d^* - d\rvert/d^*$
RMSE : $\sqrt{\frac{1}{\lvert D\rvert}\sum_{d\in D}\lvert\lvert d^* - d\rvert\rvert^2}$
SqRel : $\frac{1}{\lvert D\rvert}\sum_{d\in D}\lvert\lvert d^* - d\rvert\rvert^2/d^*$
RMSE log : $\sqrt{\frac{1}{\lvert D\rvert}\sum_{d\in D}\lvert\lvert logd^* - logd\rvert\rvert^2}$
$\delta_{\mathbf{t}}$: $\frac{1}{\lvert D\rvert}\lvert\{d\in D\lvert\ max(\frac{d^*}{d},\frac{d}{d^*}) < 1.25^t\}\rvert\times 100\%$

Table 3: Performance metrics for depth evaluation. d and d^* respectively denote predicted and ground truth depth, D presents a set of all the predicted depth values of an image, $|.|$ returns the number of the elements in the input set.

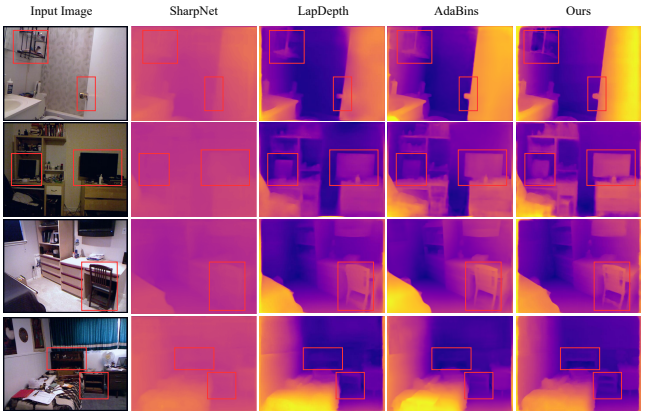

Figure 4: Visualization of depth predictions on the NYU dataset

specific range and flip input images horizontally with a probability of 0.5. The speed of all the methods is tested on a single NVIDIA GeForce RTX 2080 Ti GPU.

4.2 Benchmark Datasets

Two popular datasets (KITTI and NYU) are used for performance evaluation. The KITTI [Geiger *et al.*, 2013] dataset contains the road environment acquired in the autonomous driving scene. The resolution of the acquired images is 1242×375 pixels. We adopt the split strategy introduced by [Eigen *et al.*, 2014] for performance comparison. The test set contains 697 images from 29 scenes, and the training set contains 23488 images from 32 scenes. The maximum value of prediction depth is 80 meters.

The NYU dataset [Silberman *et al.*, 2012] contains 120K images obtained by Microsoft Kinect camera, including 464 indoor scenes with a resolution of 640×480. We also follow [Eigen *et al.*, 2014] to set train/test split, which includes 249 scenarios for training and 654 images from remaining 215 scenarios for testing. The depth map is center-cropped into 561×427 when evaluating the performance.

4.3 Comparison with State-of-the-art

In order to quantify the performance of our model, we used metrics provided by [Eigen *et al.*, 2014], which are widely applied in the performance evaluation of monocular depth estimation. The specific formulations of these metrics are shown in table 3.

We compared our method with state-of-the-art methods on the KITTI and NYU datasets on these metrics. As shown in Table 1 and 2, it is obvious that our method achieves the best performance on all metrics. On the KITTI dataset, compared with the previous state-of-the-art, AbsRel has decreased by 6.9% and SqRel has decreased by 8.9%. On the NYU dataset, compared with the previous state-of-the-art, AbsRel decreased by 9.7% and RMSE decreased by 8.0%.

Our small models like SideRT(Swin-T) and SideRT(Swin-S) outperform many state-of-the-art methods in these two challenging benchmarks. Moreover, we find that the usage of heavy backbone like Swin-L with about 200 million parameters do not slow down the inference process and shows similar inference speed with smaller models like SideRT(Swin-B) and SideRT(Swin-S).

It is worth noting that while surpassing state-of-the-art, we also achieved real-time prediction with a speed of 51.3 FPS. For a better understanding of our performance, we visualize predicted depth maps respectively in KITTI and NYU datasets, as shown in Figure 3 and 4. It can be seen that our method can also successfully predict finely detailed object boundaries that other methods cannot predict clearly.

4.4 Ablation Study

To get a better understanding of the contribution of proposed components to the overall performance, an ablation study is performed in Table 4. All the experiments are done on the KITTI dataset and use Swin-T as backbone. The training and

Method	AbsRel ↓	SqRel ↓	RMSE ↓	RMSE log ↓	δ_1↑	δ_2↑	δ_3↑	Params↓	FPS ↑
Swin-T	0.239	1.786	6.304	0.301	0.638	0.864	0.950	27.5 M	104.4
Swin-T+CSA	0.077	0.309	2.969	0.117	0.932	0.988	0.997	28.3 M	92.4
Swin-T+CSA+MSS	0.073	0.283	2.881	0.112	0.937	0.991	0.998	28.4 M	92.4
Swin-T+MSR+MSS	0.064	0.216	2.504	0.097	0.956	0.994	0.999	27.8 M	90.0
Swin-T+CSA+MSR+MSS	0.060	0.206	2.441	0.093	0.959	0.995	0.999	28.6 M	83.1

Table 4: Ablation study on different components of our work.

testing strategies are kept the same with Section 4.3.

Cross-scale Attention. To evaluate the importance of CSA modules, we remove them from the overall architecture (Row 4 and 5 in Table 4). It can be observed that the CSA module improves the performance with a small computational overhead (+0.8M Params) and a small decrease in speed (-12.0 FPS). Row 1-2 in Table 4 show that the CSA module significantly improves the depth prediction directly from the encoder Swin-T. In further visual analysis, we find that the receptive field of the encoder is relatively small, and proposed CSA module will enlarge it tremendously. It is worth mentioning that comparing performances of Row 2 in Table 4 and Row 1-2 in Table 1, the simplest architecture (Swin-T+CSA) gets similar performance with VNL [Yin *et al.*, 2019] and outperforms some metrics of DORN [Fu *et al.*, 2018].

Multi-stage Supervision. As mentioned in previous works [Tolstikhin *et al.*, 2021; Dosovitskiy *et al.*, 2020; Liu *et al.*, 2021], the training of a pure transformer architecture is much more difficult than a CNN-based counterpart. After using multi-stage supervision (MSS), we observe that the training loss curve becomes smoother and the model converges more easily. MSS brings a substantial improvement, including all the metrics, showing that this scheme is very appropriate for accelerating training and promoting the performance of pure transformer architectures.

Multi-scale Refinement. As shown in Row 3-5 in Table 4, introducing MSR modules clearly gives a boost to the performance. Theoretically speaking, CSA and MSR modules augment original feature maps from encoders in a collaborative way. CSA focuses on merging features with high similarity from a global prospect. MSR aims to fuse features with similar positions at different pyramid levels.

Inference Speed. Table 4 shows that most of the parameters come from the backbone since our lightweight decoder only contains 1.1 million parameters. After adding our proposed decoder, the AbsRel metric decreases by 74.9%, with the inference speed only decreasing by 20.4%.

Visual Analysis. In order to see whether our proposed CSA module actually enlarges the receptive field of the backbone, we follow the common practices [Yang *et al.*, 2018; Fu *et al.*, 2019] to visualize the empirical receptive field size of a CSA module, as shown in Figure 5. For an input image, we select a reference pixel (denoted by red dot) and compute its feature similarities with all the other positions.

It is obvious to see that after adding CSA, reference pixels get a stronger response from a larger scope. We find that CSA modules lead to higher responses from pixels which not only belong to similar categories but also share similar depths. This is a very beneficial attribute for depth prediction tasks.

Figure 5: Visualization of the receptive field before and after adding CSA modules. The red dot means the reference pixel. Hotter colors indicate stronger correlations.

5 Conclusion

This paper proposes two simple yet efficient mechanisms to obtain better global context. Based on those techniques, we build a lightweight pure transformer architecture that contains a few learnable parameters without convolutions. Our model shows powerful context modeling capability, leading to state-of-the-art performances on two challenging datasets. This work demonstrates that a pure transformer architecture is able to achieve a good trade-off between accuracy and running time efficiency. These findings will provide insights for future research, encouraging researchers to pay more attention to developing real-time transformer architectures for practical applications.

References

[Aich *et al.*, 2021] Shubhra Aich, Jean Marie Uwabeza Vianney, Md Amirul Islam, and Mannat Kaur Bingbing Liu. Bidirectional attention network for monocular depth estimation. In *ICRA*, 2021.

[Bhat *et al.*, 2021] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *CVPR*, 2021.

[Carion *et al.*, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[Chen *et al.*, 2017] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking

atrous convolution for semantic image segmentation. *arXiv:1706.05587*, 2017.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, and et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2020.

[Eigen *et al.*, 2014] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014.

[Fu *et al.*, 2018] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018.

[Fu *et al.*, 2019] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, 2019.

[Geiger *et al.*, 2013] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 2013.

[Hao *et al.*, 2018] Zhixiang Hao, Yu Li, Shaodi You, and Feng Lu. Detail preserving depth estimation from a single image using attention guided networks. In *3DV*, 2018.

[He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *TPAMI*, 2015.

[Huang *et al.*, 2017] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[Huynh *et al.*, 2020] Lam Huynh, Phong Nguyen-Ha, Jiri Matas, Esa Rahtu, and Janne Heikkilä. Guiding monocular depth estimation using depth-attention volume. In *ECCV*, 2020.

[Jiao *et al.*, 2018] Jianbo Jiao, Ying Cao, Yibing Song, and Rynson Lau. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In *ECCV*, 2018.

[Lee *et al.*, 2019] 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:1907.10326*, 2019.

[Lee *et al.*, 2021] Sihaeng Lee, Janghyeon Lee, Byungju Kim, Eojindl Yi, and Junmo Kim. Patch-wise attention network for monocular depth estimation. In *AAAI*, 2021.

[Lin *et al.*, 2017] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[Liu *et al.*, 2021] 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 *ICCV*, 2021.

[Qiao *et al.*, 2021] Siyuan Qiao, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation. In *CVPR*, 2021.

[Ranftl *et al.*, 2021] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *CVPR*, 2021.

[Silberman *et al.*, 2012] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[Song *et al.*, 2021] Minsoo Song, Seokjae Lim, and Wonjun Kim. Monocular depth estimation using laplacian pyramid-based depth residuals. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[Sun *et al.*, 2019] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[Tolstikhin *et al.*, 2021] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, and et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv:2105.01601*, 2021.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[Wang *et al.*, 2018] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[Xie *et al.*, 2017] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[Xu *et al.*, 2021] Yifang Xu, Chenglei Peng, Ming Li, Yang Li, and Sidan Du. Pyramid feature attention network for monocular depth prediction. In *ICME*, 2021.

[Yang *et al.*, 2018] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *CVPR*, 2018.

[Yang *et al.*, 2021] Guanglei Yang, Hao Tang, Mingli Ding, Nicu Sebe, and Elisa Ricci. Transformer-based attention networks for continuous pixel-wise prediction. In *CVPR*, 2021.

[Yin *et al.*, 2019] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *ECCV*, 2019.