EGA-Depth: Efficient Guided Attention for Self-Supervised Multi-Camera Depth Estimation

Yunxiao Shi[1] Hong Cai[1] Amin Ansari[2] Fatih Porikli[1]
[1]Qualcomm AI Research† [2]Qualcomm Technologies, Inc.
{yunxshi, hongcai, amina, fporikli}@qti.qualcomm.com

Abstract

The ubiquitous multi-camera setup on modern autonomous vehicles provides an opportunity to construct surround-view depth. Existing methods, however, either perform independent monocular depth estimations on each camera or rely on computationally heavy self attention mechanisms. In this paper, we propose a novel guided attention architecture, EGA-Depth, which can improve both the efficiency and accuracy of self-supervised multi-camera depth estimation. More specifically, for each camera, we use its perspective view as the query to cross-reference its neighboring views to derive informative features for this camera view. This allows the model to perform attention only across views with considerable overlaps and avoid the costly computations of standard self-attention. Given its efficiency, EGA-Depth enables us to exploit higher-resolution visual features, leading to improved accuracy. Furthermore, EGA-Depth can incorporate more frames from previous time steps as it scales linearly w.r.t. the number of views and frames. Extensive experiments on two challenging autonomous driving benchmarks nuScenes and DDAD demonstrate the efficacy of our proposed EGA-Depth and show that it achieves the new state-of-the-art in self-supervised multi-camera depth estimation.

1. Introduction

Depth plays a fundamental role in 3D perception, which is key to various applications including autonomous driving, AR/VR, and robotics. While it is possible to measure depth using LiDAR or Time-of-Flight (ToF) sensors, such specialized hardware can be expensive, consume a lot of power, require high-fidelity cross-sensor calibration, and fail to obtain depth for certain surfaces. On the other hand, inferring depth from camera images is more cost-efficient and can still provide promising accuracy. Traditional approaches [15, 35, 39] utilize stereoscopic vision and/or structure-from-motion to estimate depth. However, these methods have limited accuracy. By leveraging deep learning, researchers have been able to achieve significantly more accurate depth estimation [1, 12, 14, 38, 51, 54].

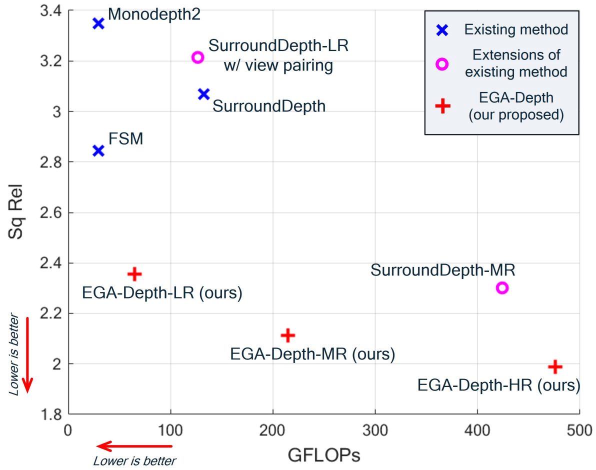

Figure 1. Accuracy (Squared Relative Error) vs. efficiency (GFLOPs). Our proposed EGA-Depth achieves the best accuracy-efficiency trade-off when comparing to baseline and latest state-of-the-art methods, including Monodepth2 [17], Full Surround Monodepth (FSM) [23], and SurroundDepth [49]. We also compare with extensions of SurroundDepth (implemented by us), e.g., using only pairs of views for self-attention, increasing feature map resolutions by replacing standard self-attention with Linformer [46]. Both of these designs under-perform our EGA-Depth. LR, MR, and HR indicate the low, median, and high resolution feature maps fed to the attention module.

Learning-based methods, however, require a massive amount of densely labeled high-quality ground-truth depth maps for training, which is costly if not impractical, to acquire at scale using range-finding sensors. To overcome this challenge, self-supervised learning [16, 17, 53] has emerged as a new paradigm to train depth estimation networks without the need for ground truth depths. Subsequent works have looked at various aspects to improve self-supervised depth estimation in terms of more advanced architectures [20,48,52] and better training procedures [4,22].

† Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

However these works only focus learning using a single camera.

More related to our work are those that consider multi-camera depth estimation, which allows one to potentially attain a 360° view of the surrounding environment. Guizilini *et al.* [23] incorporates spatial-temporal view relationships during training but still processes each camera view separately at test time. More recently, Wei *et al.* [49] proposed a transformer-based architecture to jointly process multiple camera views with standard self-attention. While this method can exploit cross-view information at test time, the expensive self-attention limits the resolution of visual features that can be processed, leading to sub-optimal results.

In this paper, we propose a novel Efficient Guided Attention architecture, EGA-Depth, in order to address the existing shortcomings and improve both the efficiency and accuracy of self-supervised multi-camera depth estimation. We first develop a guided attention mechanism that for each camera view, allows interaction between the neighboring view features. Specifically, when processing the features for each camera, we obtain queries from the current view, keys and values from the stacked features of the neighboring views. These are then fed into an efficient attention module. This allows our proposed model to achieve similar or better accuracy compared to the SOTA while using considerably less computation as can be seen in Figure 1.

Our proposed guided attention makes it possible to efficiently scale up the model complexity in order to boost depth estimation accuracy. More specifically, we can exploit higher resolutions for the visual features which allows the model to utilize more visual details for depth estimation. Furthermore, EGA-Depth can incorporate more frames from previous time steps, which allows to leverage temporal visual correlations to further improve accuracy.

Our main contributions are summarized as follows:

- We propose an efficient guided attention scheme for self-supervised multi-camera depth estimation models, where each camera view cross-references its neighboring views. Our design can significantly reduce computation costs while maintaining accuracy.
- Based on our efficient guided attention, we can efficiently exploit higher-resolution visual features as well as leverage views from previous time steps, which improves depth estimation accuracy.
- Our method acheives state-of-the-art results with optimal accuracy-efficiency trade-off on two large-scale self-supervised multi-camera depth estimation benchmarks, i.e. nuScenes [3] and DDAD [19].

2. Related Works

Self-Supervised Depth Estimation: Zhou *et al.* [53] pioneered the use of self-supervised learning for depth estimation by casting the problem as differentiable view synthesis, which employs two networks to predict depth and camera pose while masking out regions that violate the static scene assumption. Godard *et al.* [17] improved the results by proposing the per-pixel minimum reprojection loss, full-resolution multi-scale sampling, and auto-masking the pixels where no camera motion is observed. Later works seek further improvements in various aspects, e.g., improving the photometric matching [25, 40, 41], handling dynamic objects [6, 8, 18, 30], utilizing semantic segmentation [4, 21, 43], supplying temporal information to the network [13, 36, 48], etc. Recently, Guizilini *et al.* [23] extended self-supervised monocular depth estimation to the full surrounding multi-camera setting by introducing spatial-temporal contexts and pose consistency constraints in training. Wei *et al.* [49] proposed a cross-view transformer to capture interactions between cameras for multi-camera depth estimation.

Vision Transformers and Self-Attention: Inspired by the astounding success of transformers [2, 10, 45] in Natural Language Processing (NLP), the interest in exploring transformers for vision tasks [5, 11, 13, 31, 32, 44] has been soaring. At the core of transformers is the Multi-Headed Self-Attention (MHSA) mechanism [45] that computes a self-attention matrix, where the model exhaustively compares each token to every other token. This allows the model to extensively identify and exploit correlations within the input. This, however, has a quadratic complexity w.r.t. to the token sequence length that becomes a computation bottleneck. For computer vision tasks, this limitation is more pronounced since even an image of moderate resolution will result in a very long sequence length.

Linear Transformers: Recently, various techniques have been proposed to reduce the computation complexity of self-attention from quadratic to linear. Wang *et al.* [46] propose the projection of the keys and values to a fixed lower dimension. Kitaev *et al.* [29] adopts a multi-round Locality Sensitive Hashing (LSH) strategy to select the most similar pairs and only computes self-attention between them. Different methods have been proposed to approximate the softmax-based self-attention in order to achieve linear complexity [7, 26, 27, 33, 34, 37, 50]. We refer readers to recent surveys for more comprehensive coverage of linear and efficient transformer designs [28, 42].

3. Method

Here we present our novel efficient guided attention architecture for self-supervised multi-camera depth estimation (EGA-Depth). We describe the detailed design of our guided attention scheme and analyze its efficiency in Section 3.1. Section 3.2 describes how EGA-Depth enables using higher-resolution visual features as well as camera views from previous time steps, in an efficient manner, to

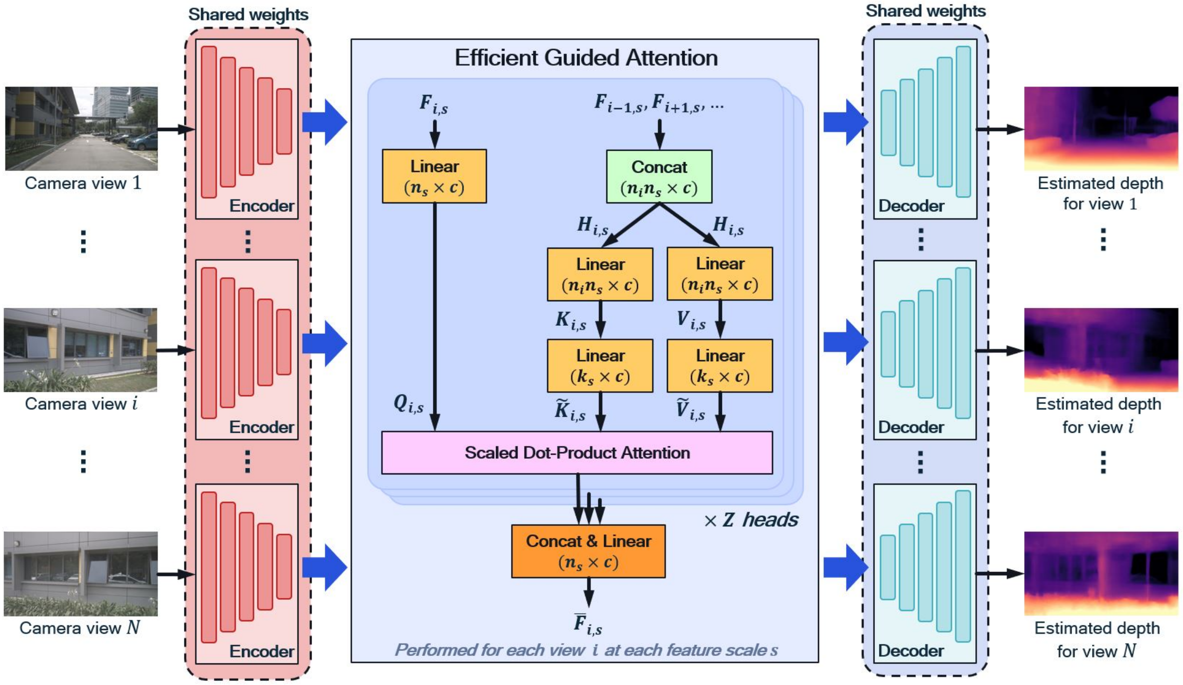
Shared weights
Camera view 1
Camera view i
Camera view N
Encoder
Encoder
Encoder
Efficient Guided Attention
$F_{i,s}$
$F_{i-1,s}, F_{i+1,s}, \dots$
Linear $(n_s \times c)$
Concat $(n_i n_s \times c)$
$H_{i,s}$
$H_{i,s}$
Linear $(n_i n_s \times c)$
Linear $(n_i n_s \times c)$
$K_{i,s}$
$V_{i,s}$
Linear $(k_s \times c)$
Linear $(k_s \times c)$
$Q_{i,s}$
$\tilde{K}_{i,s}$
$\tilde{V}_{i,s}$
Scaled Dot-Product Attention
$\times$ Z heads
Concat & Linear $(n_s \times c)$
$\bar{F}_{i,s}$
Performed for each view i at each feature scale s
Shared weights
Decoder
Decoder
Decoder
Estimated depth for view 1
Estimated depth for view i
Estimated depth for view N

Figure 2. Overview of our proposed EGA-Depth attention architecture. A single ResNet34 encoder [24] first extracts features from the input multi-camera images through weight sharing. The extracted features are then fed to our efficient guided attention module. Standard norm layers and skip connections are used in the attention module, which are not shown in the diagram for a more concise illustration. Lastly the attended features are consumed by a single decoder through shared weights to output the final estimated depth maps.

enhance the depth estimation accuracy.

3.1. Efficient Guided Attention

We first provide an overview of the meta-architecture of our multi-camera depth estimation network. Consider a setup with N cameras. For each camera $i \in \{1, ..., N\}$, the captured images are fed into a single encoder to extract multi-scale visual feature maps, $F_{i,s} \in \mathbb{R}^{n_s \times c}$, where $s \in \{1, ..., N_s\}$ denotes the feature map scale with N_s being the number of scales, $n_s = H_s W_s$ is the number of spatial elements in $F_{i,s}$, H_s and W_s denote the height and width of the feature maps at scale s, and c denotes the number of feature channels. For notation simplicity, we assume that $F_{i,s}$ is flattened. These feature maps are consumed by an attention module, which finds and utilizes their cross-correlations to refine the feature maps. We denote the output feature maps from the attention module as $\bar{F}_{i,s}$, for $i \in \{1, ..., N\}$ and $s \in \{1, ..., N_s\}$. For each camera, its updated multi-scale feature maps are then fed into a single decoder to generate the estimated depth maps.

When it comes to the attention, the existing SOTA model [49] applies the standard self-attention to the feature maps of all the views at each scale. This incurs unnecessary computations and with the attention complexity being quadratic w.r.t. both the size of the feature map as well as the number of views, a large fraction of the computation is wasted on attending across views with little or no overlaps.

Our proposed efficient guided attention does not suffer from these limitations and provides an efficient alternative to the existing methods. First, for each $F_{i,s}$, we only utilize features of the neighboring views with considerable overlaps to compute attention. This allows our model to focus on views with meaningful overlaps and avoid spending computation over non-/little-overlapping views. Next, we replace standard self-attention with our guided attention. Specifically, $F_{i,s}$ is used to compute queries and the stacked features of the neighboring views, $H_{i,s} = \text{concat}(F_{i-1,s}, F_{i+1,s}, ...) \in \mathbb{R}^{(n_i \cdot n_s) \times c}$, are used to compute keys and values, where n_i is the number of neighboring views and $F_{i-1,s}, F_{i+1,s}, ...$ denote their features. Formally, the queries, keys, and values are given by

$$Q_{i,s} = F_{i,s} W_{q,i,s},\ K_{i,s} = H_{i,s} W_{k,i,s},\ V_{i,s} = H_{i,s} W_{v,i,s}, \tag{1}$$

where $W_{q,i,s}, W_{k,i,s}, W_{v,i,s} \in \mathbb{R}^{c \times c}$ are learnable projection matrices. The attention is then calculated as follows:

$$\text{softmax}\Bigg(\underbrace{\frac{Q_{i,s} K_{i,s}^{\top}}{\sqrt{c}}}_{n_s \times (n_i \cdot n_s)} \Bigg) \cdot \underbrace{\tilde{V}_i}_{(n_i \cdot n_s) \times c} \tag{2}$$

It can be seen in Eq. (2) that when computing the attention map, our proposed guided attention only incurs a linear complexity w.r.t. the number of participating views. Moreover, as discussed, the number of participating views will be small as we only use those with meaningful overlaps. Although we require less computation, the critical functional ability to cross-reference different views is retained.

While the guided attention described above can already achieve better efficiency, its complexity is still quadratic w.r.t. the length of the features (i.e., $n_i \cdot n_s$). Inspired by the recent work of Linformer [46], we perform further projections to bring the keys and values to a prescribed, input-invariant embedding dimension. Specifically,

$$\tilde{K}_{i,s} = P_{k,i,s} K_{i,s}, \quad \tilde{V}_{i,s} = P_{v,i,s} V_{i,s}, \tag{3}$$

where $P_{k,i,s}, P_{v,i,s} \in \mathbb{R}^{k_s \times (n_i \cdot n_s)}$, $\tilde{K}_{i,s}, \tilde{V}_{i,s} \in \mathbb{R}^{k_s \times c}$, and k_s is a prescribed projection dimension for each feature scale. Using $Q_{i,s}$, $\tilde{K}_{i,s}$, and $\tilde{V}_{i,s}$, we have

$$\text{softmax}\underbrace{\left(\frac{Q_{i,s}\tilde{K}_{i,s}^{\top}}{\sqrt{c}}\right)}_{n_s \times k_s} \cdot \underbrace{\tilde{V}_{i,s}}_{k_s \times c} \tag{4}$$

which scales linearly w.r.t. the size of input feature, n_s (with an approximate choice of k_s as shown in [46]).

Figure 2 summarizes our proposed approach. Overall, it facilitates efficient attention across the multiple views, by focusing computation on meaningfully overlapping views and removing quadratic complexity from several aspects of the attention. Our efficient guided attention reaches new SOTA in accuracy with less computation, as we shall see in the experiments.

3.2. Boosting Depth Estimation Accuracy

Given the elegant design of our efficient guided attention, it is computationally feasible to scale up the attention module. We can now utilize features of higher spatial resolutions and incorporate frames from the previous time steps. Application of these new venues enables us to further improve the depth estimation accuracy.

High-Resolution Features: In the existing SOTA method [49], due to the high complexity of self-attention, tiny spatial dimensions are used for the feature maps to be fed into the attention module in order to stay computationally feasible. For instance, 11×20 feature maps are used for an input image of size 352×640 at all feature scales. This severely limits the model's ability to perform accurate depth estimation. Given the quadratic complexity, attempts to directly increase the feature resolution would result in infeasible training and inference even with multiple modern high-end GPUs (*e.g.*, 4 Nvidia A100s).

With our proposed EGA scheme, it is possible to use higher resolutions for the features, e.g., 4× larger, since the attention computation is now linear w.r.t. the feature map size. Moreover, the linear complexity enables a more fine-grained tuning capability to gradually increase the feature resolution while maintaining computational feasibility. Leveraging these higher-resolution features can effectively enhance the depth estimation accuracy, as we shall see in the experiments.

Utilizing Temporal Information: Existing works on self-supervised multi-camera depth estimation do not leverage temporal information at inference time. One key reason is the prohibitive computation and memory requirements that would be incurred when more frames were included.

Our efficient guided attention provides a viable option to incorporate more views/frames. More specifically, for each camera view, we can jointly stack features from previous frames as well as those from neighboring views, for computing keys and values in the attention. Suppose the features of the neighboring views at the current time t are $F^t_{i-1,s}, F^t_{i+1,s}, \ldots$ and features of the previous frames are $F^{t-1}_{i,s}, F^{t-2}_{i,s}, \ldots$. The new stacked reference features for view i at time t is then given by $H^t_{i,s} = \text{concat}(F^t_{i-1,s}, F^t_{i+1,s}, \ldots, F^{t-1}_{i,s}, F^{t-2}_{i,s}, \ldots) \in \mathbb{R}^{((n_i+n_t)\cdot n_s)\times c}$, where n_t is the number of previous frames to be included. Given $F^t_{i,s}$ and $H^t_{i,s}$, our guided attention can be readily computed following the steps outlined in Section 3.1. We note that when including the features of more previous frames in the attention, the complexity of computing the attention map increases linearly. In our experiments, we include the frames from the previous timestep, which has more overlapped regions with the current frames, taking into account the fact that future frames are not available at test time for real-world applications.

3.3. Loss Function

We follow standard practices [17, 49] and train our entire system by minimizing the photometric error averaged over all camera views:

$$\mathcal{L}_p = \min_{t'} \frac{\alpha}{2}(1 - \text{SSIM}(I_t, \hat{I}_{t'})) + (1-\alpha)||I_t - \hat{I}_{t'}||_1, \tag{5}$$

where $\alpha = 0.85$, $t' \in \{t-1, t+1\}$ which are the two source frames from the previous and the next time steps, $\hat{I}_{t'}$ is the synthesized target view, and $||\cdot||_1$ is the L_1 norm and SSIM$(\cdot)$ is the structural similarity measure [47].

An edge-aware smoothness loss:

$$\mathcal{L}_s = |\partial_x d^*_t| e^{-|\partial_x I_t|} + |\partial_y d^*_t| e^{-|\partial_y I_t|} \tag{6}$$

is also applied to prevent estimated depth from shrinking.

The final training loss is then given by

$$\mathcal{L} = \mathcal{L}_p + \lambda \mathcal{L}_s \tag{7}$$

where $\lambda = 0.001$ is used to balance the two loss terms.

We predict each camera pose independently instead of jointly in [49]. Also we do not perform structure-from-motion pretraining proposed in [49] as we do not attempt to recover metric depth directly. More details on training (e.g., differentiable view synthesis) can be found in the supplementary material.

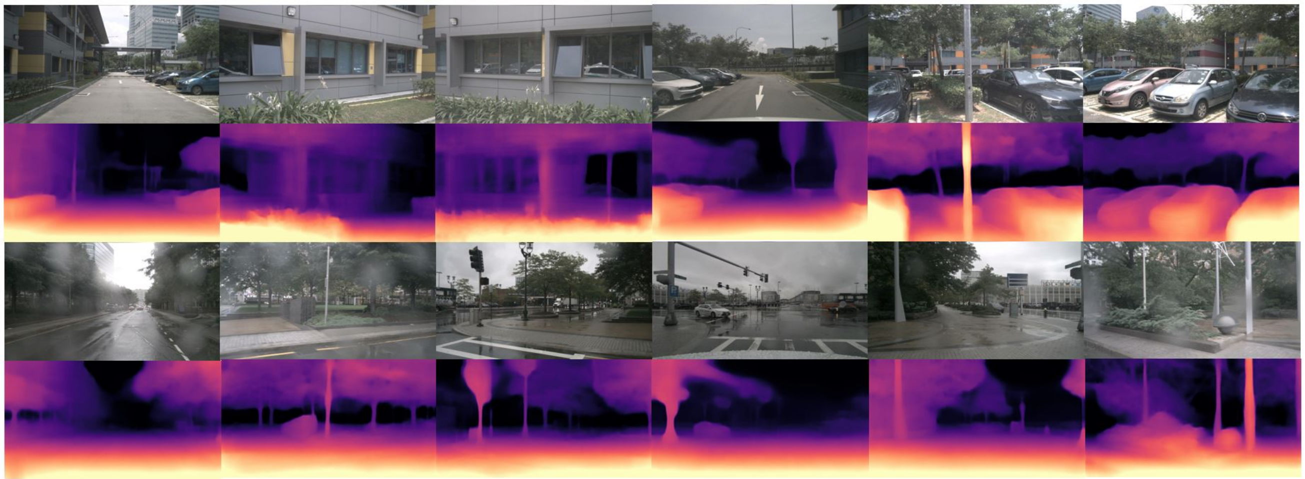

Figure 3. Qualitative results of self-supervised multi-camera depth predictions on two sample scenes from the nuScenes dataset. For each scene, we show the front, front-left, back-left, back, back-right, and front-right camera views from left to right. The first scene is under normal weather conditions and the second one is a rainy scene. We can see that even in adverse weather conditions, our EGA-Depth can generate accurate depth maps with fine details.

4. Experiments

We carry out extensive experiments to evaluate our proposed EGA-Depth on public large-scale benchmarks and compare with existing state-of-the-art solutions. We also conduct detailed ablation studies to provide insights into different aspects of our proposed approach.

4.1. Experimental Setup

Datasets: We train and evaluate EGA-Depth on two of the most popular yet challenging multi-camera autonomous driving benchmark datasets, *i.e.*, nuScenes [3] and DDAD [19].

nuScenes consists of 1.4 million images from 1,000 scenes of urban driving collected in Boston and Singapore using a synchronized camera array of 6 cameras. It is a very challenging dataset for self-supervised depth estimation since many scenes have low illumination, adverse weather conditions like rain and fog, and cluttered surroundings like construction sites. The small view overlaps across cameras further incur extra challenges for predicting consistent multi-camera depths. Following [49], we downsample the images from the original resolution of 900×1600 to 352×640. We filter out the static frames, resulting in 20,096 samples used for training and 6,019 samples for evaluation.

DDAD is an autonomous driving dataset collected in the U.S. and Japan using a synchronized 6-camera array, featuring long-range (up to 250m) and diverse urban driving scenarios. Following [49], we downsample the images from the original resolution of 1216×1936 to 384×640, resulting in 12,650 training samples and 3,950 validation samples. Following [23, 49], we use the occlusion masks to reweight the photometric loss in training.

Architecture: We use ResNet34 [24] with ImageNet [9] pretrained weights as the image encoder. For each camera view, we use the two neighboring camera views when computing our guided attention. We use $Z = 8$ attention heads. We consider 3 resolution options for the features to be fed into the attention, low resolution (LR), medium resolution (MR), and high resolution (HR). On nuScenes, for LR, we use 11×20 feature maps for all 5 scales. For MR, we use 22×40 for the 4 larger scales and 11×20 for the smallest scale. For HR, we further increase the feature map to 44×80 for the largest scale. For LR, we do not project the keys and values to a lower dimension since the input length is already small. For MR, we use $k_s = 880$ as the fixed embedding dimension in the attention for all scales. For HR, we use $k_s = 1024$ for the largest scale and $k_s = 880$ for the rest. on DDAD, we use 12×40 feature maps for all scales for LR and 24×40 for all scales for MR, with no dimension reduction for LR and $k_s = 960$ for MR.

Training: We follow the self-supervised scheme from [49] to train our network, including learning rate scheduling and data augmentation. For the pose network, we use the ResNet18-based model from [17]. We note that for pose estimation, we follow the standard practice of predicting each camera pose separately, instead of performing joint pose estimation which, based on our experiments, would give inferior results. All our experiments are conducted using 4 Nvidia A100 GPUs, each having 80GB of high-bandwidth memory.

Evaluation: We use the standard metrics to evaluate depth estimation quality [12, 17], including Absolute Relative Error (Abs Rel), Squared Relative Error (Sq Rel), Root-Mean-Squared-Error (RMSE), RMSE log, $\delta < 1.25$, $\delta <$

Method	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
Monodepth2 [17]	0.287	3.349	7.184	0.345	0.641	0.845	0.925
PackNet-SfM [19]	0.309	2.891	7.994	0.390	0.547	0.796	0.899
FSM [23]	0.299	-	-	-	-	-	-
FSM* [23]	0.334	2.845	7.786	0.406	0.508	0.761	0.894
SurroundDepth [49]	0.245	3.067	6.835	0.321	0.719	0.878	0.935
EGA-Depth-LR (ours)	0.239	2.357	6.801	0.317	0.723	0.880	0.936
EGA-Depth-MR (ours)	0.228	2.113	6.738	0.311	0.728	0.885	0.940
EGA-Depth-HR (ours)	**0.223**	**1.987**	**6.599**	**0.306**	**0.732**	**0.889**	**0.942**

Table 1. Quantitative evaluation of self-supervised multi-camera depth estimation on nuScenes. The best results are highlighted in bold. The row of FSM* shows the results of FSM reproduced by [49]. Note that all the methods use the same input image resolution. LR, MR, and HR refer to the choices of internal feature resolution in our EGA-Depth approach.

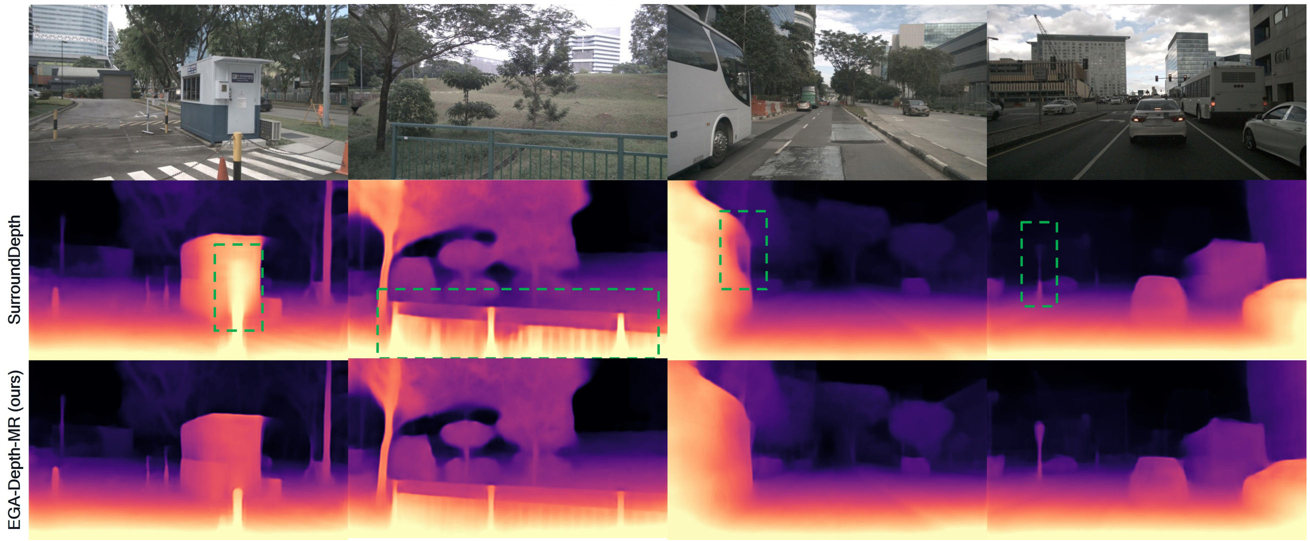

Figure 4. Qualitative comparison between EGA-Depth and SurroundDepth [49] on nuScenes. The second row shows the estimated depth maps from SurroundDepth and the third row shows the estimation from our EGA-Depth method. It can be seen that EGA-Depth provides more accurate depth estimation, with better spatial consistency and finer details. The green boxes highlight sample regions where our method considerably improves the estimation quality.

1.25^2, and $\delta < 1.25^3$.[1] Following previous works [23, 49], we apply median-scaling at test time and report numbers averaged over all cameras. The depth values are evaluated with maximum distances of 80m and 200m, for nuScenes and DDAD, respectively.

4.2. Evaluation Results

We report both quantitative and qualitative results on nuScenes and DDAD, and compare with baseline and latest methods on self-supervised multi-camera depth estimation, including Monodepth2 [17] and PackNet-SfM [19] (treating each view independently), FSM [23], and SurroundDepth [49].

nuScenes: The performance evaluation on nuScenes is shown in Table 1 and Figures 3 and 4. Results shown in Figs. 3 and 4 are generated using the EGA-Depth-MR. In Table 1, it can be seen that even our lightest model, EGA-Depth-LR, which uses low-resolution feature maps, already outperforms the latest SOTA across all metrics by a considerable margin. By leveraging higher resolutions for the features, we can further reduce depth estimation errors. Specifically, EGA-Depth-HR reduces the squared relative error by 35% when compared to the latest SOTA method of SurroundDepth.

Fig. 3 shows the multi-camera depth estimation by our proposed EGA-Depth on two sample scenes. It can be seen that EGA-Depth accurately predicts the depths (e.g., around the vehicles), even under adverse weather conditions like rain. In Fig. 4, we compare the estimated depth maps between our EGA-Depth and SurroundDepth. We see that EGA-Depth can predict depths with sharper details and more robustness, thanks to using higher-resolution feature maps ($4\times$ larger than in SurroundDepth)

DDAD: The quantitative evaluation results on DDAD

[1] We refer readers to the supplementary file for detailed mathematical definitions of these metrics.

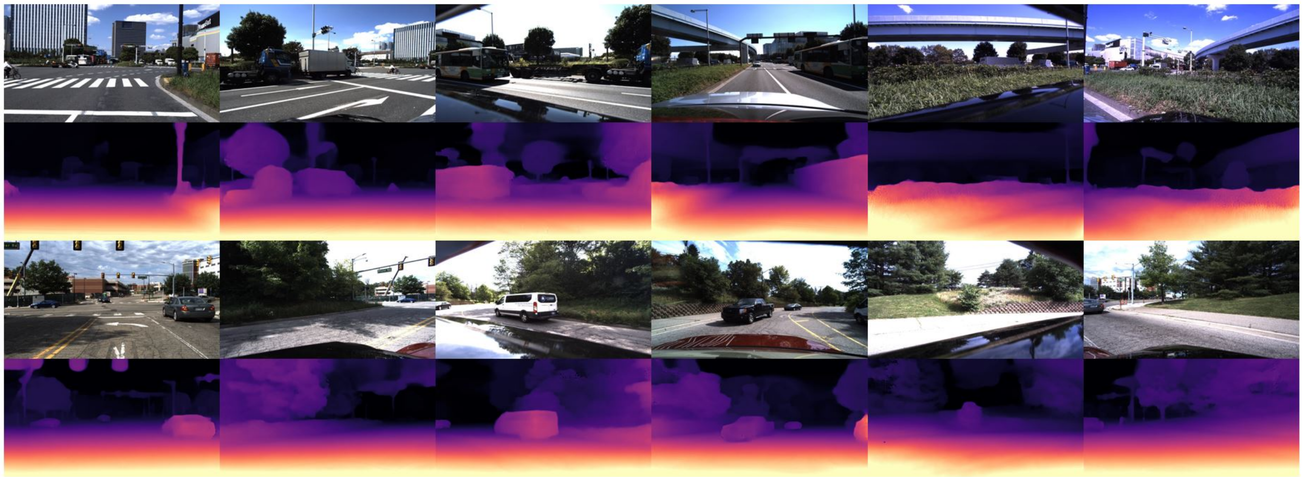

Figure 5. Qualitative results of self-supervised multi-camera depth predictions on two sample scenes from the DDAD dataset. For each scene, we show the front, front-left, back-left, back, back-right, and front-right camera views from left to right. We can see that our EGA-Depth can generate accurate depth maps with fine details.

Method	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
Monodepth2 [17]	0.217	3.641	12.962	0.323	0.699	0.877	0.939
PackNet-SfM [19]	0.234	3.802	13.253	0.331	0.672	0.860	0.931
FSM [23]	0.202	-	-	-	-	-	-
FSM* [23]	0.229	4.589	13.520	0.327	0.677	0.867	0.936
SurroundDepth [49]	0.200	3.392	12.270	0.301	0.740	0.894	0.947
EGA-Depth-LR (ours)	0.195	3.211	12.117	0.297	0.743	0.896	0.947
EGA-Depth-MR (ours)	**0.191**	**3.126**	**11.922**	**0.290**	**0.747**	**0.901**	**0.950**

Table 2. Quantitative evaluation of self-supervised multi-camera depth estimation on DDAD. The best results are highlighted in bold. The row of FSM* shows the results of FSM reproduced by [49]. Note that all the methods use the same input image resolution. LR, MR, and HR refer to the choices of internal feature resolution in our EGA-Depth approach.

Method	**Pre. Frame**	Abs Rel ↓	$\delta < 1.25$ ↑
SurroundDepth [49]		0.245	0.719
SurroundDepth-T	✓	0.368	0.501
EGA-Depth-LR		0.239	0.723
EGA-Depth-LR-T	✓	0.237	0.723
EGA-Depth-MR		0.228	0.728
EGA-Depth-MR-T	✓	0.226	0.729

Table 3. Results of leveraging features from the previous time step on nuScenes.

are shown in Table 2. We see that our EGA-Depth models consistently outperform the existing SOTA methods across all the metrics. Fig. 5 shows visual examples of predicted depth maps by using EGA-Depth-MR. It can be seen that our predicted depth maps are very accurate and preserve important details, such as object boundaries.

Using Previous Frame: As described in Section 3.2, EGA-Depth can take advantage of previous frames in our guided attention module. It can be seen in Table 3, adding frames from the previous timestep consistently improves the depth estimation accuracy of EGA-Depth. We note that the accuracy improvements are not very significant. This is due to the fact that the curated nuScenes video sequences (containing only key frames) have a low frame rate of 2Hz and hence, view overlaps between consecutive frames are small. As a result, the amount of temporal information that can be leveraged is limited by the data. As a baseline for our temporal studies, we augment our SurroundDepth implementation to take advantage of previous frames. More specifically, we feed the features of 12 views/frames (6 for the current time step and 6 for the previous time step) to its self-attention module. We note that this is consistent with the original SurroundDepth design. While this almost doubled the amount GFLOPs needed (going from 132.32 to 220.15), the accuracy becomes worse. This can be attributed to model overfitting, since much more parameters are used in the model, and attention computation over little-/non-overlapping views, causing the model to potentially learn spurious correlations. On the other hand, when using the previous frames, EGA-Depth-LR-T only incurs a small increase in GFLOPs (going from 64.94 to 91.56) while also clearly improving the depth estimation accuracy.

Attention Choice	Feature Map Resolution	k_s	$n_i \cdot n_s$	Abs Rel ↓	$\delta < 1.25$ ↑	GFLOPs ↓
SurroundDepth [49]	$\{11 \times 20\}_{\times 5}$	-	1320	0.245	0.719	132.32
SurroundDepth-MR	$\{22 \times 40\}_{\times 4}$	880	5280	0.236	0.721	424.01
	$\{11 \times 20\}_{\times 1}$	-	1320			
EGA-Depth-LR	$\{11 \times 20\}_{\times 5}$	-	440	0.239	0.723	64.94
EGA-Depth-MR	$\{22 \times 40\}_{\times 4}$	880	1760	0.228	0.728	214.81
	$\{11 \times 20\}_{\times 1}$	-	440			
EGA-Depth-HR	$\{44 \times 80\}_{\times 1}$	1024	7040	0.223	0.732	475.79
	$\{22 \times 40\}_{\times 3}$	880	1760			
	$\{11 \times 20\}_{\times 1}$	-	440			

Table 4. Results of computing attention over feature maps of different resolutions on nuScenes. The first column shows the two attention choices of SurroundDepth and our EGA-Depth. The second column shows the resolutions of the 5-scale (from top to bottom) feature maps going into the attention module. k_s is the projection dimension that is used to map the length of the flattened and concatenated neighboring-view features, $n_i \cdot n_s$, to a constant value. SurroundDepth-MR is our extension of SurroundDepth using higher feature resolutions; note that it is infeasible to scale up the feature maps of the original SurroundDepth, even when using 4 Nvidia A100 GPUs. On the other hand, EGA-Depth can process features with a resolution as large as 44×80 for 352×640 input images.

4.3. Ablation Studies

Feature Map Resolution: Table 4 shows details of the feature map resolution and projection dimension (k_s) choices of our EGA-Depth models, as well as their accuracies and computational costs. When using low-resolution features in EGA-Depth-LR, we do not reduce the key and value dimensions since the input length (i.e., $H_s \times W_s$ of the feature map) is already small with $n_s = 440$. EGA-Depth-LR already outperforms SurroundDepth and uses 51% less computation. Our EGA-Depth-MR and EGA-Depth-HR models employ higher resolutions for the feature maps, leading to improved depth estimation accuracy with increased computational costs.

In order to investigate the performance of SurroundDepth with higher resolution features, we implement a baseline, SurroundDepth-MR. In order to make the computation feasible, we replace the standard self-attention with Linformer and increase the feature map resolution by 4× except for the smallest scale. Although SurroundDepth-MR has better accuracy than the original SurroundDepth, it requires significantly more computations. In particular, SurroundDepth-MR can only achieve slightly better accuracy than EGA-Depth-LR, but requires 4.5× GFLOPs.

Our proposed EGA-Depth provides much more favorable accuracy-efficiency trade-offs as compared to using the SurroundDepth attention scheme, as also shown in Figure 1. Moreover, even when using 4 Nvidia A100 GPUs, further scaling up the feature maps of SurroundDepth (using linear transformer) results in infeasible memory costs.

Projection Dimension k_s: In Table 5, we show how the choice of the projection dimension k_s impacts the final depth estimation performance. For EGA-Depth-MR, we set $k_s = 880$, which reduces the query and key dimensions by half. We also experiment with other values of k_s. It can be seen that all these k_s values allow our model to generate more accurate depth estimation as compared to SurroundDepth. However, we observe that increasing k_s does not necessarily map to monotonically improving results.

Method	k_s	$n_i \cdot n_s$	Abs Rel ↓	$\delta < 1.25$ ↑
SurroundDepth [49]	-	1320	0.245	0.719
EGA-Depth-MR	512	1760	0.233	0.721
	880		0.228	0.728
	1024		0.229	0.728
	1536		0.235	0.723

Table 5. Results on nuScenes of using different values of k_s when projecting keys and values to a lower dimension.

5. Conclusion

In this paper, we presented a novel, efficient guided attention approach, EGA-Depth, for self-supervised multi-camera depth estimation. In contrast to existing SOTA that applied standard self-attention to all the camera views, we proposed to utilize attention across each camera view and its neighboring views, which significantly reduces computational costs while improving accuracy. We further incorporated an operation to project the keys and values of the attention to lower, prescribed dimensions, which removed the quadratic complexity with respect to the input feature map resolution. Based on our inherently efficient design, we leveraged higher feature resolutions in order to further improve our depth estimation accuracy. Our EGA-Depth framework can also readily incorporate frames from previous time steps for attention. As we have shown in our experiments on challenging benchmark datasets of nuScenes and DDAD, our proposed EGA-Depth sets the new SOTA accuracy while our design points lie on the Pareto frontier of accuracy-efficiency trade-off.

References

[1] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021. 1

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 5

[4] Hong Cai, Janarbek Matai, Shubhankar Borse, Yizhe Zhang, Amin Ansari, and Fatih Porikli. X-distill: Improving self-supervised monocular depth via cross-task distillation. In *British Machine Vision Conference*, 2021. 1, 2

[5] 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. 2

[6] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019. 2

[7] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020. 2

[8] Qi Dai, Vaishakh Patil, Simon Hecker, Dengxin Dai, Luc Van Gool, and Konrad Schindler. Self-supervised object motion and depth estimation from video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1004–1005, 2020. 2

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ACL*, 2018. 2

[11] 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. *International Conference on Learning Representations*, 2021. 2

[12] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 1, 5

[13] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021. 2

[14] 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. 1

[15] Yasutaka Furukawa, Brian Curless, Steven M Seitz, and Richard Szeliski. Towards internet-scale multi-view stereo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 1434–1441. IEEE, 2010. 1

[16] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 1

[17] 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*, pages 3828–3838, 2019. 1, 2, 4, 5, 6, 7

[18] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8977–8986, 2019. 2

[19] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 5, 6, 7

[20] Vitor Guizilini, Rareș Ambruș, Dian Chen, Sergey Zakharov, and Adrien Gaidon. Multi-frame self-supervised depth with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 160–170, 2022. 1

[21] V. Guizilini, R. Hou, J. Li, R. Ambrus, and A. Gaidon. Semantically-guided representation learning for self-supervised monocular depth. In *Proceedings of the International Conference on Learning Representations*, 2020. 2

[22] Vitor Guizilini, Kuan-Hui Lee, Rareş Ambruş, and Adrien Gaidon. Learning optical flow, depth, and scene flow without real-world labels. *IEEE Robotics and Automation Letters*, 7(2):3491–3498, 2022. 1

[23] Vitor Guizilini, Igor Vasiljevic, Rares Ambrus, Greg Shakhnarovich, and Adrien Gaidon. Full surround monodepth from multiple cameras. *IEEE Robotics and Automation Letters*, 7(2):5397–5404, 2022. 1, 2, 5, 6, 7

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3, 5

[25] Hualie Jiang, Laiyan Ding, Zhenglong Sun, and Rui Huang. Dipe: Deeper into photometric errors for unsupervised learn-

ing of depth and ego-motion from monocular videos. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 10061–10067, 2020. 2

[26] Jungo Kasai, Hao Peng, Yizhe Zhang, Dani Yogatama, Gabriel Ilharco, Nikolaos Pappas, Yi Mao, Weizhu Chen, and Noah A Smith. Finetuning pretrained transformers into rnns. *arXiv preprint arXiv:2103.13076*, 2021. 2

[27] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020. 2

[28] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Computing Surveys*, 2022. 2

[29] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020. 2

[30] M. Klingner, J.-A. Termöhlen, J. Mikolajczyk, and T. Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *Proceedings of the European Conference on Computer Vision*, 2020. 2

[31] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022. 2

[32] 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, October 2021. 2

[33] Jiachen Lu, Jinghan Yao, Junge Zhang, Xiatian Zhu, Hang Xu, Weiguo Gao, Chunjing Xu, Tao Xiang, and Li Zhang. Soft: softmax-free transformer with linear complexity. *Advances in Neural Information Processing Systems*, 34:21297–21309, 2021. 2

[34] Sachin Mehta and Mohammad Rastegari. Separable self-attention for mobile vision transformers. *arXiv preprint arXiv:2206.02680*, 2022. 2

[35] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011. 1

[36] Vaishakh Patil, Wouter Van Gansbeke, Dengxin Dai, and Luc Van Gool. Don't forget the past: Recurrent depth estimation from monocular video. *IEEE Robotics and Automation Letters*, 5(4):6813–6820, 2020. 2

[37] Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A Smith, and Lingpeng Kong. Random feature attention. *arXiv preprint arXiv:2103.02143*, 2021. 2

[38] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021. 1

[39] Ashutosh Saxena, Jamie Schulte, Andrew Y Ng, et al. Depth estimation using monocular and stereo cues. In *IJCAI*, volume 7, pages 2197–2203, 2007. 1

[40] Yunxiao Shi, Jing Zhu, Yi Fang, Kuochin Lien, and Junli Gu. Self-supervised learning of depth and ego-motion with differentiable bundle adjustment. *arXiv preprint arXiv:1909.13163*, 2019. 2

[41] C. Shu, K. Yu, Z. Duan, and K. Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *Proceedings of the European Conference on Computer Vision*, 2020. 2

[42] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 2022. 2

[43] Fabio Tosi, Filippo Aleotti, Pierluigi Zama Ramirez, Matteo Poggi, Samuele Salti, Luigi Di Stefano, and Stefano Mattoccia. Distilled semantics for comprehensive scene understanding from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2

[44] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 2

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

[46] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 1, 2, 4

[47] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 4

[48] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1164–1174, 2021. 1, 2

[49] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Yongming Rao, Guan Huang, Jiwen Lu, and Jie Zhou. Surrounddepth: Entangling surrounding views for self-supervised multi-camera depth estimation. *arXiv preprint arXiv:2204.03636*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[50] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14138–14148, 2021. 2

[51] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE/CVF Con-*

ference on Computer Vision and Pattern Recognition, pages 3916–3925, 2022. 1

[52] Chaoqiang Zhao, Youmin Zhang, Matteo Poggi, Fabio Tosi, Xianda Guo, Zheng Zhu, Guan Huang, Yang Tang, and Stefano Mattoccia. Monovit: Self-supervised monocular depth estimation with a vision transformer. *arXiv preprint arXiv:2208.03543*, 2022. 1

[53] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017. 1, 2

[54] Jing Zhu, Yunxiao Shi, Mengwei Ren, and Yi Fang. Mdanet: memorable domain adaptation network for monocular depth estimation. In *British Machine Vision Conference 2020*, 2020. 1