Monocular Depth Estimation with Adaptive Geometric Attention

Taher Naderi*, Amir Sadovnik*, Jason Hayward†, Hairong Qi*
*Department of Electrical Engineering and Computer Science
†Department of Nuclear Engineering
University of Tennessee, Knoxville, TN 37996, USA
{tnaderi, asadovnik, jhayward, hqi}@utk.edu

Abstract

Single image depth estimation is an ill-posed problem. That is, it is not mathematically possible to uniquely estimate the 3rd dimension (or depth) from a single 2D image. Hence, additional constraints need to be incorporated in order to regulate the solution space. In this paper, we explore the idea of constraining the model by taking advantage of the similarity between the RGB image and the corresponding depth map at the geometric edges of the 3D scene for more accurate depth estimation. We propose a general light-weight adaptive geometric attention module that uses the cross-correlation between the encoder and the decoder as a measure of this similarity. More precisely, we use the cosine similarity between the local embedded features in the encoder and the decoder at each spatial point. The proposed module along with the encoder-decoder network is trained in an end-to-end fashion and achieves superior and competitive performance in comparison with other state-of-the-art methods. In addition, adding our module to the base encoder-decoder model adds only an additional 0.03% (or 0.0003) parameters. Therefore, this module can be added to any base encoder-decoder network without changing its structure to address any task at hand.

1. Introduction

Depth estimation is an important step in understanding the geometry of a 3D scene. In addition, many downstream applications, such as 3D modeling, robotics, autonomous driving, etc., rely on accurate depth estimation. The minimal sensory setup for depth estimation is to use a single monocular image. However, recovering the scene's depth from a single image is an ill-posed problem that requires additional priors, often referred to as monocular depth cues, like perspective, occlusion, object size, texture, etc. These cues can be exploited through learning-based methods with prior knowledge to disambiguate different 3D interpretations.

Figure 1. Comparison of edges and gradients in an RGB image and the corresponding depth map. Top-left: RGB image. Top-right: the corresponding depth map of the RGB image. Bottom-left: Laplacian of the RGB image. Bottom-right: Laplacian of the depth map.

Existing deep learning methods can usually estimate accurate 2D depth maps. However, they lack local details and are often highly distorted when the maps are projected into 3D. This is due to the usage of down-sampling in the pre-trained fully convolutional encoders, mostly designed for classification purpose. While feature resolution and granularity may not be important in performing tasks like image classification, they are crucial for dense prediction, where the architecture of the model should ideally be able to deliver features at or close to the resolution of the input image.

Various techniques have been proposed to mitigate the above-mentioned issues. One way is using dilated convolutions [55] to rapidly increase the receptive field without down-sampling. Another way is using skip connections from multiple stages of the encoder to the decoder [40]. By the same token, the problem has been addressed in [44] by connecting multi-resolution representations in parallel throughout the network. While all these techniques

can solve the issue to some extent, they are subject to the problem of washed out information in deeper convolutional networks [19]. To mitigate the effect of these convolutions, some researchers have suggested to replace the building blocks entirely or in some places in networks by attention-based blocks [29, 34] or transformers [52, 39] which are themselves, based on attention mechanisms.

Even given that we can find a way to produce a high resolution depth map with many details by using skip connections, we still run into an additional problem. We explain this using the example shown in Fig. 1 that compares an RGB image and the corresponding depth map. The cabinet on the left and the table surface are almost texture-less in the RGB image and have gradient only at geometric edge locations in the depth map. On the other hand, the wall with the brick texture mainly shows a gradient-less area in the depth map but a lot of gradient in the RGB image. Looking at the high-pass filtered RGB image and the depth map suggests that most of the information needed to extract a depth map from a scene is near the geometric edges, i.e., edges in the RGB image which come from the geometric structure of the 3D scene. However, to extract the geometric edges, we need to first remove the edges in the RGB image which mainly come from texture and color changes.

For this reason we wish to give the convolutional neural network the ability to deduce the local geometric structure of the RGB image using guidance from the corresponding depth map. However, the depth map is not available at evaluation time. Instead, we explore the idea of constraining the model by taking advantage of the similarity of the RGB features and the corresponding depth map features at geometric edges of the 3D scene for more accurate depth estimation. We hence propose a light-weight attention module that uses the cross-correlation between the encoder and the decoder. The functionality of this module can be interpreted as a guiding tool for an efficient feature extraction in the encoder and it can be used to merge the same size feature maps from the encoder to the decoder efficiently in any encoder-decoder structure with minimum added weight and computation burden to the base encoder-decoder network to address any task at hand.

The proposed module along with the encoder-decoder network is trained in an end-to-end fashion on both the indoor NYUDV2 dataset [37] and the outdoor KITTI dataset [12] and achieves superior and competitive performance in comparison with state-of-the-art.

2. Related Works

The ability of CNNs to work as a regressor has made them a good candidate for depth estimation. However, compared to estimation of the exact depth of a single point, it is easier to estimate its depth range [2, 10] and formulate the depth estimation as a pixel-wise classification task instead. Our work benefits from both methods.

Depth estimation with (geometric) constraints. Deep learning methods have been proven to be effective in depth map estimation. However, they lack local details in 2D and they are often highly distorted when the maps are projected into 3D. In this case, One can also improve depth estimation using some kind of (geometric) constraint. While [17] tried to solve these issues by fusing multi-scale features, [54] exploited the virtual normals of virtual surfaces to estimate the depth map in 3D scene robustly. By the same token, [30] proposed a two-streamed CNN that predicts both depth and depth gradients and then fusing the outputs together into a detailed depth map. Another example of two-streamed CNN is GeoNet [38], which jointly predicts depth and surface normal maps from a single image. Similar to [7, 54, 38] which exploit geometric constraints, [27] assumed local planar for every local patch to guide depth prediction more effectively.

Intuitively, neighboring pixels with similar appearances should have similar depth estimation and major depth changes usually lie in the vertical direction in outdoor scenes. This constraint was utilized in [11] for single image depth estimation.

Depth estimation in relation to segmentation. Depth estimation and semantic segmentation symbiosis represents one of the closest relationship in deep learning tasks. Some works have tried to exploit one to help improve the performance of the other or both at the same time [31, 46, 13, 7, 21, 16, 25, 46]. However, the performance is not the only incentive for this symbiosis. For example, [24] exploits semantic guidance to solve the dynamic object problem in monocular depth estimation.

Improving depth estimation using semantic segmentation can be interpreted as attending to the objects and their borders instead of all pixels just like in [43, 21]. While pixel-wise visual attention maps have shown their effectiveness [23, 43] suggested an object-level attention model for autonomous driving.

Depth estimation based on attention and transformers. Attention mechanisms have been used in depth estimation works previously. Most of the works are based on [48, 22] which in turn borrowed the idea from natural language processing (NLP) [42]. The suggested dot products and matrix multiplications usually try to find correlation between different spatial parts of tensor features [41, 22, 42, 53, 5]. The problem with these operations is they are computationally expensive where optimization is made more difficult due to lots of multiplication operations involved. Similar to [33, 50, 2], the authors in [51] employed a continuous CRF to fuse multi-scale information derived from a CNN. Different from the past works, they imposed structural constraints on an estimated attention map to estimate depth. Attention fusion was also used in [15]. In [1] the

authors, inspired by neural machine translation, introduced a CNN scheme which exploits forward and backward attention mechanisms. [22] used a self-attention context module to explore the inference of similar disparity values at non-contiguous regions of the image. Exactly the same mechanism was also adopted in [36].

While attention and geometric constraint are beneficial for depth estimation, combination of both can be exploited to improve depth estimation [20, 45]. [14] tried to use attention mechanism to improve monocular depth estimation as well. Different from our work, their spatial attention mechanism is separated from their global context module while ours combines the two stages in one light-weight and local module with different operations, i.e., sensitivity-enhanced geometric similarity in embedded Euclidean space.

Attention can be easily exploited in loss function since the ground truth depth is available when training the network. Having this in mind, [21] has tried to benefit from an attention-driven Loss that adjusts the backpropagation flow accordingly.

3. Proposed Method

In this section we discuss the structure of our model and the optimization as well as an in-depth discussion about our proposed module. As discussed in Sec. 2, depth estimation can be defined as a regression problem or a classification problem. We choose to adopt the model along with its cost functions from [54] as our base model which uses both classification and regression at the same time. We then introduce our proposed attention module into the base model for performance improvement. The addition of our module imposes a minimal change to the base model in the sense of computational cost and only adding few additional parameters to the network model.

Like any other regression and/or classification problems, there are two aspects of the method which contribute to the quality of the estimation, namely, the model and its structure, and the cost function and the optimization method. In the following, we elaborate on both aspects.

3.1. Model

We would like to guide the encoder to shape the RGB features using the depth map for better depth estimation at each spatial point. However, at prediction time the depth map is not available. Instead, we use the local cross-correlation of the embedded encoder and decoder features as the local similarity measure at each spatial point. The eventual criteria for this guidance is the sensitivity enhanced absolute value of the cosine similarity between the local embedded features at every spatial point of the encoder and the decoder. By enhancing the sensitivity, we try to make any non-zero correlation between the encoder and the decoder features at each spatial point more effective. The similarity

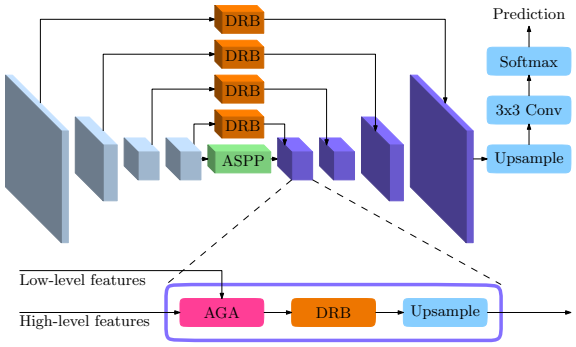

Figure 2. An overall structure of the model.

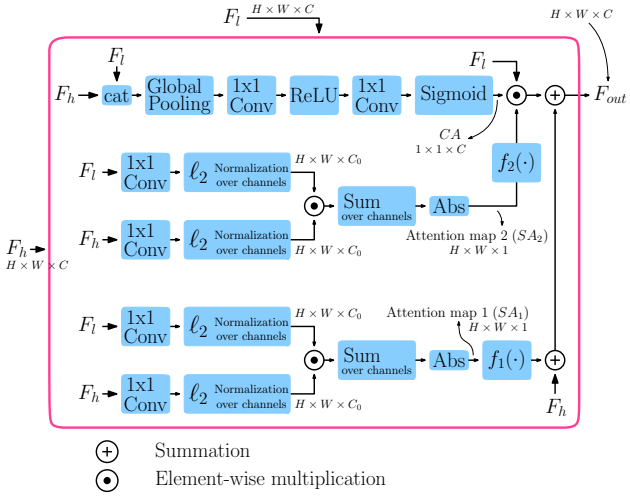

Figure 3. The internal structure of AGA module in its most general settings. $\mathcal{S}$, $\mathcal{C}$ and $\mathcal{A}$ stand for spatial, channel-wise and attention, respectively. F_l and F_h are low-level and high-level features from the encoder and the previous stage of the decoder, respectively. The attention maps SA_1 and SA_2 are discussed in Sec. 3.1 and are equivalent to (1).

measure is absolute and normalized version of dot product (dot product is the conventional attention technique) which means more constraints are imposed on the network to regulate the solution space better.

We adopt the model and cost functions from [54] as our base model. We then add the proposed adaptive geometric attention (AGA) module to the base model as well as adding an ℓ_2 term to the cost function, as shown in Fig. 3. The overall structure of the model is depicted in Fig. 2.

The model mainly consists of two parts, an encoder which extracts features from the input RGB image at different spatial resolutions, and a decoder which reconstructs the depth map from the features extracted by the encoder. In addition, the encoder and the decoder are connected to each other using an Astrous Spatial Pyramid Pooling (ASPP) module [4] to increase the receptive field of the entire model. All upsampling operations in the model are based on the bilinear resizing method. The whole encoder-decoder structure in the base model [54], itself had been

borrowed from [32]. The decoder in [32, 54] comprises of several adaptive merging blocks (AMB) to fuse features from different levels of the encoder and the decoder, and dilated residual blocks (DRB) modules to increase the receptive field of the encoder and transform the encoder features. AMB blocks, in [32, 54], merge the encoder's features into the decoder's features adaptively which can be considered a channel-wise attention mechanism. The operations in the AMB are nothing but concatenation of both the encoder and the decoder features, followed by the squeeze and excitation operations using the squeeze and excitation networks (SE Networks) [18].

Instead of the AMB block we use our own improved AGA module as shown in Figs. 2 and 3 in its most general form. Our AGA block benefits from both spatial and channel-wise attention integrated into one module. The first row of operations in Fig. 3 is in fact from the AMB module. The novel part of the module is the spatial attention operations which are mixed with the channel-wise operation in an additive and multiplicative fashion. For the spatial attention, the module uses the local cross-correlation of the encoder and the decoder features at each spatial points to shape the encoder features spatially.

At first, our AGA module uses 1×1 convolutions as a bottleneck to go from hyper space (feature space) to embedded Euclidean space for both the encoder and the decoder features. Then the module uses cross-correlation of the embedded features from the encoder and the decoder. More precisely, the module uses absolute value of cosine similarity of the embedded features of the encoder and the decoder at each spatial point as a measure of structural similarity between the depth map features and the RGB features. Since this similarity measure is absolute and normalized, the module can put more constraints on the solution space. As a result, it can shape the RGB features in a better way, both spatially and channel-wise, using the decoder as the representation of the depth features for better depth estimation. See Fig. 7 for visual effect of the spatial attention. The operations in the second row and the third row of Fig. 3 which calculate the spatial attention (attention maps SA_1 and SA_2) are equivalent to

$$SA_i = |cossim\left(E_{l,i}, E_{h,i}\right)|\,,\ i = 1, 2 \qquad (1)$$

where $E_{l,i}$ and $E_{h,i}$ denote the embedded features of low level features, i.e., F_l or the encoder features, and high level features, i.e., F_h or the decoder features, respectively. The operations in Fig. 3 are depicted in this way to facilitate the comprehension of their extension to the non-local AGA module in Fig. 8 which will be discussed in Sec. 4.5.

Not only the channel-wise attention and the spatial attention are different in the above-mentioned implementation details, but also the purposes of the two are different. The channel-wise attention provides the encoder feature with

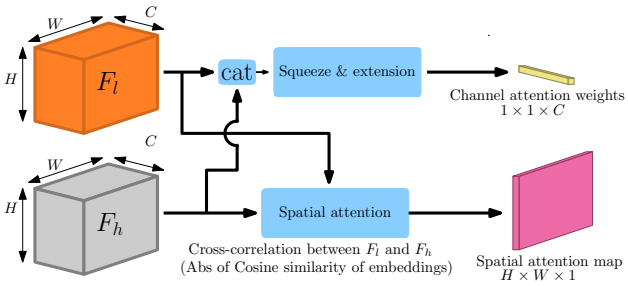

Figure 4. Illustration of the differences between the spatial attention and the channel-wise attention discussed in Sec. 3.1.

one scalar multiplicative weight for the entire of each single channel of size $H \times W$. So for the entire encoder's $H \times W \times C$ feature it provides a vector of size $1 \times C$. The vector is scaled before added to the decoder's feature. Spatial attention, instead, is an $H \times W$ attention map that each feature vector at each spatial point of the encoder feature will be multiplied by the corresponding spatial value of the attention map. See Fig. 4.

The AGA module uses the sensitivity-enhanced absolute value of the above-mentioned cosine similarity. The absolute value enforces the correlation between two feature vectors at each spatial point of the encoder and the decoder features independent of the direction. That is, what we wish is to compare the presence of any spatial cross-correlation between the depth map features and the RGB features. If there is a correlation between the depth map features and the RGB features, then they carry information about each other regardless of the sign of the correlation. The AGA module in the most general form has been depicted in Fig. 3. To go from hyperspace, C, to the embedded space, C_0, at each spatial point, we utilize a 1×1 convolution with bottlenecking $C_0 < C$. In this way, we are able to avoid permutations of the information in different channels since the 2D convolution operation is fully connected in channel direction of the input and summation is permutation indifferent. This bottleneck will give us the structure of the features in that spatial point in the embedded space. This operation is local. Being local and bottle-necked, it is light. The output of this operation is an $H \times W$ spatial attention map. As shown in Fig. 3, the AGA module's output is an $H \times W \times C$ tensor of features

$$F_{out} = [f_1(\mathcal{SA}_1) + f_2(\mathcal{SA}_2) \times \mathcal{CA}] \times F_l + F_h. \qquad (2)$$

where $\mathcal{S}$, $\mathcal{C}$ and $\mathcal{A}$ stand for spatial, channel-wise and attention, respectively. F_l and F_h are low-level and high-level features from the encoder and the previous stage of the decoder, respectively. The first spatial attention map, $\mathcal{SA}_1$, is additive while the second one, $\mathcal{SA}_2$, is multiplied by the channel-wise attention weights. Element-wise summation and multiplication of tensors of sizes $H \times W \times 1$ and $1 \times 1 \times C$ and $H \times W \times C$ are possible since these op-

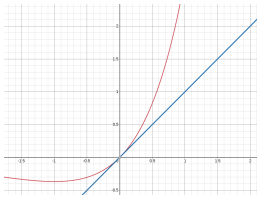

Figure 5. Comparing $x \exp(x)$, in red color, and x, in blue color.

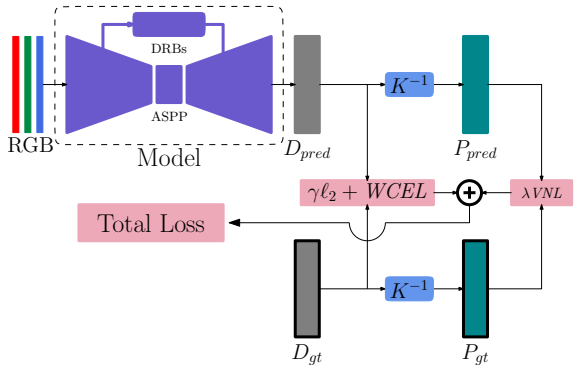

Figure 6. The overall training pipeline. Total loss consists of three terms ℓ_{WCEL}, ℓ_{VN} and ℓ_2. ℓ_{WCEL} and ℓ_2 compare the absolute predicted depth map, D_{pred}, and the ground truth depth map, D_{gt}. ℓ_{VN} compares the virtual normals using the predicted point cloud, P_{pred}, and the ground truth point cloud P_{gt}. γ and λ are scaling constants tuned to give an appropriate effect to each term in the total cost function. ℓ_{WCEL}, ℓ_{VN} are from the base model [54].

erations broadcast the operand tensors. $f_1(\cdot)$ and $f_2(\cdot)$ are introduced to enhance the sensitivity to any non-zero correlation between the high-level features and low-level features in each spatial point. They are chosen either of

- $f(\mathcal{SA}) = \mathcal{SA}$
- $f(\mathcal{SA}) = \mathcal{SA} \exp(\mathcal{SA})$

The first one means spatial attention without enhancing sensitivity. The second one means spatial attention with enhanced sensitivity. See Fig. 5 for comparison between them. We experienced that enhancing the sensitivity around 1 helps. One explanation is that the gradients in a normalized output are suppressed. To completely turn off the sensitivity to the additive spatial attention and multiplicative spatial attention, we set $f_1(\mathcal{SA}) = 0$ and $f_2(\mathcal{SA}) = 1$.

Our AGA module merges the attended low-level features from each level of the encoder to the corresponding decoder's high-level features. The AGA module will learn the merging parameters, during optimization, to merge the information for all elements of the $H \times W \times C$ encoder tensors weights, i.e., $[f_1(\mathcal{SA}_1) + f_2(\mathcal{SA}_2) \times \mathcal{CA}]$.

3.2. Loss functions

We utilize a 3-term cost function. The virtual normal loss and the weighted cross-entropy loss were already used in the base model [54]. We add a third term ℓ_2, based on the $\mathbf{L}_2$ norm of the error.

Virtual Normal Loss (VNL). The surface normal is an important local feature for 3D reconstruction and depth estimation. However, calculating surface normals in a small area is prone to noise. To remove the effect of noise, [54] suggests to calculate the normals of virtual surfaces built by triangles which their constructing points have been chosen far from each other in 3D scene at random. If $\mathbf{n}^i_{pred}$ and $\mathbf{n}^i_{gt}$ are the predicted normal and ground truth normal at the point i respectively, then the computed Virtual Normal loss is:

$$\ell_{VN} = \frac{1}{N}\left(\sum_{i=1}^{N} \left\|\mathbf{n}^i_{pred} - \mathbf{n}^i_{gt}\right\|_1\right) \tag{3}$$

where N is total number of valid sampled triangles. See [54] for details. Similar results can be achieved if one uses the virtual slope in 3D scene instead of virtual normals. See ablation study in [54]. ℓ_{VN} helps with relative pixel-wise depth values of the predicted depth map and its structure in regression fashion.

Pixel-wise Absolute Depth Supervision. In addition to VNL, we have two terms which enforce pixel-wise absolute depth supervision. Similar to [10, 54], we have used quantized real-valued depth. We formulated the depth prediction as a classification problem instead of regression by employing the cross entropy loss. More precisely, we borrowed the weighted cross-entropy loss (WCEL) from [2, 54], with the weight being the information gain. See [2] for details. Combination of these two above-mentioned terms has been already utilized in our based model [54]. In addition to these two terms, we decided to use the $\mathbf{L}_2$ norm of the difference between the ground truth and the predicted depth map, to decrease the root mean square error (RMSE) of the predicted depth map. We combine WCEL and VNL and $\mathbf{L}_2$ loss together to gain an overall supervision in the training phase. The total loss is

$$\ell = \ell_{WCEL} + \lambda\ell_{VN} + \gamma\ell_2 \tag{4}$$

where the weights λ and γ define the contribution of each term. We have set λ to 6 and γ to 25 based on extensive empirical studies. The overall training pipeline is illustrated in Fig. 6.

4. Experiments and Results

We perform our experiments on the NYUDV2 dataset [37] and the KITTI dataset [12] to evaluate the performance of our proposed algorithm in comparison with state-of-the-art methods. We also perform ablation studies to better understand the contribution of the different settings of our attention module.

4.1. Datasets

NYUDV2. The NYUDV2 dataset consists of 464 different indoor scenes, which are divided into 249 scenes for training and 215 for testing. Similar to [10], we use the training scenes after synchronization using the tool provided by [37] to train our model for our main results and ablation study on NYUDV2. We refer to this dataset as large NYUDV2. Moreover, we use a subset of the Raw NYUDV2 dataset. which is split to 249/215 train/test split scenes for our ablation study. We refer to the subset as small NYUDV2.

KITTI. The KITTI dataset contains over 93K outdoor images and depth maps with an approximate resolution of 1240×374. All images are captured on driving cars by stereo cameras and a Lidar. We test on 697 images from 29 scenes split by Eigen et al. [8]. We remove all the images from the scenes in which one of them is in the test scenes and use the remaining RGB images and corresponding ground truth for training.

4.2. Implementation details

Similar to [54], the ResNeXt-101(32 × 4d) [49] encoder pre-trained on ImageNet [6] is used as the encoder in our model. The base model is exactly as decribed in [54] but we replace all AMB modules with our AGA modules. See Fig. 2. In our main results (Sec. 4.4), we use the AGA module as described in Sec. 3.1 with the additional ℓ_2 loss term. All 1×1 bottle-necks in the AGA modules are $\frac{1}{16}$ times of their input channel size.

In all of our experiments our base learning rate is 0.003 used along with a learning rate scheduling going from 1 to 0 linearly for all training procedures on the large NYUDV2 and KITTI and the same learning rate scheduling with power 0.9, is chosen on the small NYUDV2. Stochastic gradient descent is applied as the optimization method with a batch size of 16 on the large NYUDV2 and the KITTII and a batch size of 8 on the small NYUDV2. The weight decay and momentum are set to 0.0005 and 0.9 respectively. The model is trained for 99300 iterations on large NYUDV2 and KITTI and 5000 iterations on small NYUDV2.

We conduct data augmentation on the training samples using the following methods. On small and large NYUDV2 the RGB image and the corresponding depth map are randomly resized with ratio [1, 0.92, 0.86, 0.8, 0.75, 0.7, 0.67], randomly flipped horizontally, and finally randomly cropped to 384×384. A similar process is applied for KITTI but resizing with the ratio [1, 1.1, 1.2, 1.3, 1.4, 1.5] and cropping with 384 × 512. Note that the depth map should be scaled to the corresponding resizing ratio [8].

It is worth mentioning that the overall model is similar to what has been used in [54] except the AGA module in magenta in Fig. 2. The base model in [54] has exactly 90436054 parameters and we are adding only 28672 parameters to it which is around 0.03% (or 0.0003) of the total parameters of the base model. In addition, all added operations are light since they are local.

Method	Err(lower is better)			Acc(higher is better)		
	rel	lg10	rms	σ_1	σ_2	σ_3
Eigen [7]	0.158	-	0.641	0.769	0.950	0.988
Chakrab [3]	0.149	-	0.620	0.806	0.958	0.987
Li [30]	0.143	0.063	0.635	0.788	0.958	0.991
Su [41]	0.137	0.058	0.498	0.826	0.967	0.995
Qi [38]	0.128	0.057	0.569	0.834	0.960	0.990
Wang[46]	0.128	-	0.497	0.845	0.966	0.990
Wang [47]	0.128	-	0.493	0.844	0.964	0.991
Laina [26]	0.127	0.055	0.573	0.811	0.953	0.988
Xu [50]	0.121	0.052	0.586	0.811	0.954	0.987
Lee [28]	0.119	0.050	0.430	0.870	0.974	0.993
Wang [45]	0.115	0.049	0.519	0.871	0.975	0.993
Fu [10]	0.115	0.051	0.509	0.828	0.965	0.992
Hu [17]	0.115	0.050	0.530	0.866	0.975	0.993
Liu [35]	0.113	0.049	0.523	0.872	0.975	0.993
Lee [27]	0.110	0.047	<u>0.392</u>	0.885	0.978	0.994
Huynh [20]	0.108	-	0.412	0.882	0.980	**0.996**
Fang [9]	0.101	-	0.412	0.868	0.958	0.986
Yang [52]	0.106	0.045	**0.365**	**0.900**	**0.983**	**0.996**
base [54]	0.108	0.048	0.416	0.875	0.976	0.994
ours	**0.097**	**0.042**	0.444	<u>0.897</u>	<u>0.982</u>	**0.996**

Table 1. Results on large NYUDV2 as compared to other state-of-the-art methods. The best result in each column (measure) is depicted in bold text. The second best is underlined.

Method	Err(lower is better)			Acc(higher is better)		
	rel	rms	rmslog	σ_1	σ_2	σ_3
Su [41]	0.117	4.251	0.174	0.894	0.971	0.984
Fang [9]	0.098	4.075	0.174	0.889	0.963	0.985
Wang [45]	0.096	4.327	0.171	0.893	0.963	0.983
Fu [10]	0.072	**2.727**	0.120	0.932	0.984	0.994
Liu [35]	<u>0.070</u>	2.912	0.121	0.942	0.986	0.992
Lee [27]	**0.059**	<u>2.756</u>	**0.096**	**0.956**	**0.993**	**0.998**
base [54]	0.072	3.258	0.117	0.938	0.990	**0.998**
ours	<u>0.070</u>	3.223	<u>0.113</u>	<u>0.944</u>	<u>0.991</u>	**0.998**

Table 2. Results on KITTI dataset as compared with state-of-the-art methods. The best result in each column (measure) is depicted in bold text. The second best is underlined. Our model consistently beats the base model [54] in all measures and shows comparable performance with other state state-of-the-art methods.

4.3. Evaluation metrics

Similar to [26] we evaluate the performance of our depth prediction quantitatively based on mean absolute relative error (rel), mean log 10 error (lg10), root mean squared error (rms), root mean squared log of error (rmslog) and the accuracy under threshold ($\sigma_i < 1.25^i\ ,\ i = 1, 2, 3$).

4.4. Comparison with state-of-the-art

A comparison of our results with state-of-the-art methods is shown in table 1 for large NYUDV2 and in Table 2 for the KITTI dataset. As shown in Table 1, our suggested method achieves best or comparable results in all the measures except one among all state-of-the-art meth-

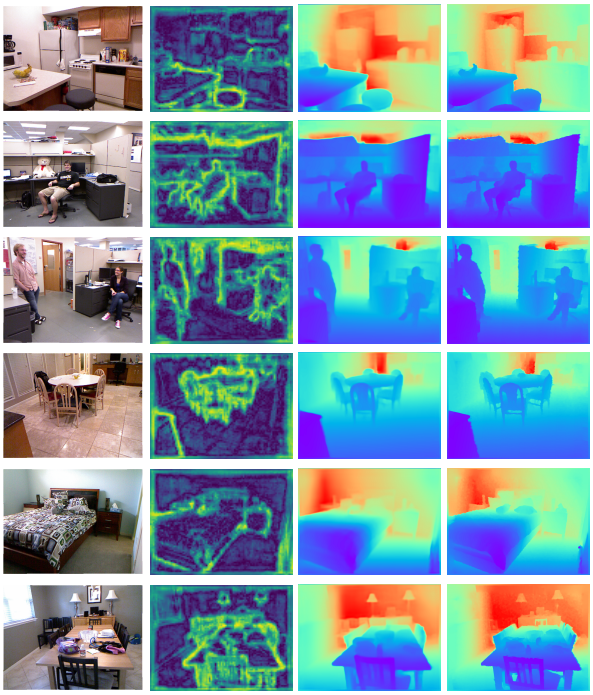

Figure 7. Qualitative results. From left to right: RGB image, attention map, predicted depth map, ground truth depth map. As the attention map depicts, the attention is higher at the geometric boundary of the 3D scene. This attention is strongest at occlusion boundaries which is an important subset of geometric boundaries.

ods. Examples of our trained model's outputs, largest attention map, ground truth depth map and RGB images are depicted in Fig. 7. The attention map shows stronger response around geometric boundary of the 3D scene as expected by our method. This attention is the strongest at the occlusion boundaries which is an important subset of geometric boundaries. The clear separation of the objects with emphasized geometric boundaries around them suggests that the AGA module is performing as expected by reducing the effect of texture edges and focusing on geometric ones. The performance on the KITTI datset in comparison with state-of-the-art shows that the methodology is effective on KITTI dataset as well. As it is shown in Table 2 our model outperforms the base model [54] in all measures and shows comparable results in comparison with the state-of-the-art in all other measures.

4.5. Ablation study

In this section we conduct two sets of experiments.

Effectiveness of the proposed AGA module over the base model [54] and added ℓ_2 loss term. We compare the effect of different internal settings for our suggested general AGA module depicted in Fig. 3 as well as the added ℓ_2 term in the total cost function. The settings which are referred to by first column of Table 3 in this section are the settings for general coefficients of low-level features, i.e., $[f_1(\mathcal{SA}_1) + f_2(\mathcal{SA}_2) \times \mathcal{CA}]$, in Eq. (2). The first row in Table 3 is the base model[54] with its cost functions, i.e., VNL and WCEL. Other than the base model and its cost functions in [54] we have added ℓ_2 to the total cost functions. So we provide different settings with and without ℓ_2 to study the effect of the term. As Table 3 suggests, the structure with first spatial attention with sensitivity enhanced added to the channel attention works the best for the NYUDV2 dataset. However, it is possible that on other data distributions setting $S5$ might be an option because we noticed that setting $S5$ has less spikes during training in our experiments which is a desirable trait. The root mean square measure (rms) in Table 3 is lower for the settings with the added term ℓ_2 in the total cost function, i.e., $S7$ (lowest) and $S4$ (second lowest). This shows the effectiveness of ℓ_2 term in the total cost function. Note that $S6$ and $S7$ are the same in their AGA setting but the later has the added ℓ_2 term in the total cost function.

Set.	Error (lower is better)			Acc (higher is better)		
	rel	log10	rms	σ_1	σ_2	σ_3
[54]	0.1408	0.0590	0.5951	0.8217	0.9635	0.9907
S2	0.1385	0.0581	0.5856	0.8269	0.9644	0.9915
S3	0.1388	0.0586	0.5980	0.8232	0.9641	0.9912
S4	0.1381	0.0578	<u>0.5702</u>	0.8277	0.9643	0.9919
S5	<u>0.1361</u>	0.0577	0.5832	0.8283	0.9658	0.9916
S6	**0.1345**	<u>0.0574</u>	0.5904	<u>0.8302</u>	<u>0.9670</u>	<u>0.9921</u>
S7	0.1364	**0.0568**	**0.5567**	**0.8319**	**0.9671**	**0.9929**

Table 3. Ablation study for different settings in our suggested general AGA module in Fig. 3 compared to the base model [54]. The settings which are referred to by first column of this table are the settings for general coefficients of low-level features, i.e. $[f_1(\mathcal{SA}_1) + f_2(\mathcal{SA}_2) \times \mathcal{CA}]$, in Eq. (2). The results are the last iteration of each experiment which are filtered using a moving average with length 15. In this table the first row represents $\mathcal{CA}$ is the base model [54]. $S2 = \mathcal{SA}_2 \times \mathcal{CA}$. $S3 = \mathcal{SA}_2$. $S4 = \mathcal{SA}_2 \times \mathcal{CA}$ as well as added ℓ_2 in total cost-function. $S5 = \mathcal{SA}_1 \times exp(\mathcal{SA}_1) + \mathcal{SA}_2 \times \mathcal{CA}$. $S6 = \mathcal{SA}_1 \times exp(\mathcal{SA}_1) + \mathcal{CA}$. $S7 = \mathcal{SA}_1 \times exp(\mathcal{SA}_1) + \mathcal{CA}$ as well as added ℓ_2 in total cost-function. $\mathcal{S}$, $\mathcal{C}$ and $\mathcal{A}$ stand for spatial, channel-wise and attention respectively. All settings have been trained with $\ell = \ell_{WCEL} + \lambda\ell_{VN}$, but the ones with added ℓ_2 trained using $\ell = \ell_{WCEL} + \lambda\ell_{VN} + \gamma\ell_2$. The best value in each column is bold type and the second best is underlined.

Effectiveness of the proposed methodology over the conventional attention. Second, we show the novelty of the AGA module's implementation, (i.e. sensitivity enhanced absolute value of cosine similarity of the features in embedded space) as a measure of similarity between the encoder's and the decoder's features at each spatial point in comparison with the traditional attention mechanisms. The second set of experiments aims at showing the effectiveness of the above-mentioned cross-correlation measure between the low-level features (the encoder features) as representation of the RGB image and the high-level features (the decoder

Set.	Error (lower is better)			Acc (higher is better)		
	rel	log10	rms	σ_1	σ_2	σ_3
DS7	0.102	0.045	0.452	0.881	0.974	0.994
NS7	0.101	0.045	0.450	0.882	0.976	0.994
S7	**0.097**	**0.042**	**0.444**	**0.897**	**0.982**	**0.996**

Table 4. Comparison between our AGA module and the conventional attention techniques, dot product and costly matrix multiplication. The $S7$ is just like the setting of main results of table 1, 2 and table 4. In this table the first row, $DS7$, is the same setting in $S7$ but using dot product as spactial attention mechanism instead of our similarity measure. $NS7$ is extension of $S7$ to non-local operation and compare each local feature vector with all other points in other spatial points.

features) as the representation of the depth map by comparing it to the conventional attention techniques, i.e., dot product and matrix multiplication (non-local operations). We compare three settings, $DS7$, $NS7$ and $S7$. The $S7$ setting has been described in Table 3 which is the same for Tables 1 and 2 as well. The $DS7$ setting is the same as $S7$ but using dot product as coefficients for spatial attention mechanism instead of formula (1) and (2). $NS7$ is the extension of $S7$ to non-local operations. It compares each spatial points with all points in all other spatial points. The details of implementation of non-local AGA module has been depicted in Fig. 8. It is important to note that all three models have exactly the same number of parameters. The only difference is whether we have used formula (1) and (2) or dot product and attention is local or non-local.

As Table 4 suggests, our local AGA module works the best in comparison with the conventional attention mechanisms. The reason that non-local AGA is showing inferior performance in comparison with our suggested (local) AGA module is the introduction of lots of multiplication in forward pass in non-local AGA module in comparison with the local counterparts in the overall model. Those multiplications create complications in gradients, as a result the optimization process become less efficient. In addition, absolute value of cosine similarity is normalized and sign indifferent and measures similarity as far as there is a cross-correlation between the two source of information while dot product does not consider these two. In other words, the absolute value of cosine similarity is absolute and normalized version of dot product which means imposing more constraint on the network to regulate the solution space better. We also enhance the sensitivity at any non-zero correlation. The matrix multiplication create the non-local version of operations which are computationally costly and not much effective as well.

5. Discussion and Conclusion

The main idea of this paper was taking advantage of the similarity of the RGB image and the depth map in the area of the 3D scene close to geometric edges. In other words,

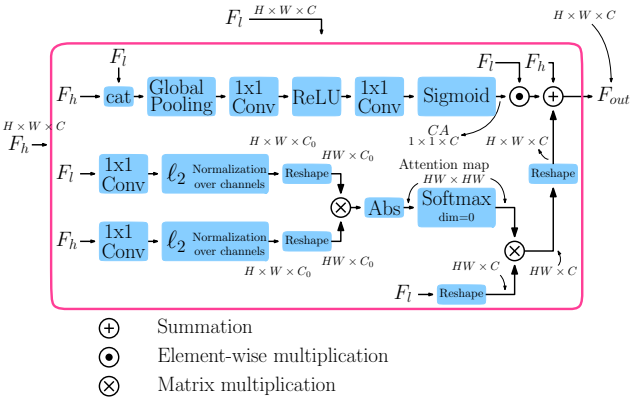

Figure 8. The internal structure of the Non-local AGA module as the natural extension from the setting for the AGA modules for the main results of the paper.

we want to guide the encoder to shape better RGB features using the depth map for better depth estimation at each spatial point. The eventual criteria for this guidance is the sensitivity enhanced absolute value of cosine similarity between the local embedded features at every spatial point of the encoder and the decoder. We are able to do so since the features in the decoder are close to the end of the model and closer to the cost function in training phase.

The benefits of using absolute value of local cosine similarity in embedded space in comparison with the conventional attention techniques, i.e., dot product, is that it is absolute and normalized so it puts stronger constraints on the network to regulate solution space better. It is also local so it does not create difficulty in optimization with matrix multiplications in non-local versions. It is important to note that for designing our suggested AGA module which uses the guidance of the depth map features to shape the RGB features, one might be able to assign more time and hardware resources to find more effective complex operations instead of $f_1(\mathcal{SA}_1) + f_2(\mathcal{SA}_2) \times \mathcal{CA}$ in Fig. 3 and (2). However, fine tuning the structure and parameters of such a module would be difficult. Hence, we decided to use the divide-and-conquer strategy, where we divide the guidance to additive and multiplicative spatial attention weights, $f_1(\mathcal{SA}_1)$ and $f_2(\mathcal{SA}_2)$, and channel-wise attention weights $\mathcal{CA}$.

Acknowledgments

This paper is supported in part by the Defense Threat Reduction Agency under grant number HDTRA 1-18-1-005 and in part by the Department of Energy National Nuclear Security Administration through the Nuclear Science and Security Consortium under Award Number(s) DE-NA0003180 and/or DE-NA0000979.

References

[1] Shubhra Aich, Jean Marie Uwabeza Vianney, Md Amirul Islam, Mannat Kaur, and Bingbing Liu. Bidirectional attention network for monocular depth estimation. *arXiv preprint arXiv:2009.00743*, 2020.

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

[3] Ayan Chakrabarti, Jingyu Shao, and Gregory Shakhnarovich. Depth from a single image by harmonizing overcomplete local network predictions. *arXiv preprint arXiv:1605.07081*, 2016.

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[5] Yuru Chen, Haitao Zhao, Zhengwei Hu, and Jingchao Peng. Attention-based context aggregation network for monocular depth estimation. *International Journal of Machine Learning and Cybernetics*, pages 1–14, 2021.

[6] 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.

[7] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[8] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[9] Zhicheng Fang, Xiaoran Chen, Yuhua Chen, and Luc Van Gool. Towards good practice for cnn-based monocular depth estimation. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1091–1100, 2020.

[10] 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.

[11] Yukang Gan, Xiangyu Xu, Wenxiu Sun, and Liang Lin. Monocular depth estimation with affinity, vertical pooling, and label enhancement. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 224–239, 2018.

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[13] Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. Semantically-guided representation learning for self-supervised monocular depth. *arXiv preprint arXiv:2002.12319*, 2020.

[14] Xiaoyan Guo and Wen Zheng. Improving monocular depth estimation by leveraging structural awareness and complementary datasets. *ECCV, Lecture Notes in Computer Science*, 2020.

[15] Zhixiang Hao, Yu Li, Shaodi You, and Feng Lu. Detail preserving depth estimation from a single image using attention guided networks. In *2018 International Conference on 3D Vision (3DV)*, pages 304–313. IEEE, 2018.

[16] Lei He, Jiwen Lu, Guanghui Wang, Shiyu Song, and Jie Zhou. Sosd-net: Joint semantic object segmentation and depth estimation from monocular images. *Neurocomputing*, 440:251–263, 2021.

[17] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1043–1051. IEEE, 2019.

[18] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[19] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[20] 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.

[21] Jianbo Jiao, Ying Cao, Yibing Song, and Rynson Lau. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In *Proceedings of the European conference on computer vision (ECCV)*, pages 53–69, 2018.

[22] Adrian Johnston and Gustavo Carneiro. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4756–4765, 2020.

[23] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles. In *Proceedings of the European conference on computer vision (ECCV)*, pages 563–578, 2018.

[24] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *European Conference on Computer Vision*, pages 582–600. Springer, 2020.

[25] Varun Ravi Kumar, Marvin Klingner, Senthil Yogamani, Stefan Milz, Tim Fingscheidt, and Patrick Mader. Syndistnet: Self-supervised monocular fisheye camera distance estimation synergized with semantic segmentation for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 61–71, 2021.

[26] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE, 2016.

[27] 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.

[28] Jae-Han Lee and Chang-Su Kim. Multi-loss rebalancing algorithm for monocular depth estimation. In *Proceedings of the 2020 European Conference on Computer Vision (ECCV), Glasgow, UK*, pages 23–28, 2020.

[29] Zeyu Lei, Yan Wang, Zijian Li, and Junyao Yang. Attention based multilayer feature fusion convolutional neural network for unsupervised monocular depth estimation. *Neurocomputing*, 423:343–352, 2021.

[30] Jun Li, Reinhard Klein, and Angela Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3372–3380, 2017.

[31] Rui Li, Qing Mao, Pei Wang, Xiantuo He, Yu Zhu, Jinqiu Sun, and Yanning Zhang. Semantic-guided representation enhancement for self-supervised monocular trained depth estimation. *arXiv preprint arXiv:2012.08048*, 2020.

[32] Ruibo Li, Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, and Lingxiao Hang. Deep attention-based classification network for robust depth prediction. In *Asian Conference on Computer Vision*, pages 663–678. Springer, 2018.

[33] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5162–5170, 2015.

[34] Jing Liu, Xiaona Zhang, Zhaoxin Li, and Tianlu Mao. Multi-scale residual pyramid attention network for monocular depth estimation. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5137–5144. IEEE, 2021.

[35] Peng Liu, Zonghua Zhang, Zhaozong Meng, and Nan Gao. Joint attention mechanisms for monocular depth estimation with multi-scale convolutions and adaptive weight adjustment. *IEEE Access*, 8:184437–184450, 2020.

[36] Alwyn Mathew, Aditya Prakash Patra, and Jimson Mathew. Self-attention dense depth estimation network for unrectified video sequences. *arXiv preprint arXiv:2005.14313*, 2020.

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

[38] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 283–291, 2018.

[39] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *arXiv preprint arXiv:2103.13413*, 2021.

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[41] Wen Su, Haifeng Zhang, Quan Zhou, Wenzhen Yang, and Zengfu Wang. Monocular depth estimation using information exchange network. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[43] Dequan Wang, Coline Devin, Qi-Zhi Cai, Fisher Yu, and Trevor Darrell. Deep object-centric policies for

autonomous driving. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8853–8859. IEEE, 2019.

[44] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[45] Jianrong Wang, Ge Zhang, Mei Yu, Tianyi Xu, and Tao Luo. Attention-based dense decoding network for monocular depth estimation. *IEEE Access*, 8:85802–85812, 2020.

[46] Lijun Wang, Jianming Zhang, Oliver Wang, Zhe Lin, and Huchuan Lu. Sdc-depth: Semantic divide-and-conquer network for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 541–550, 2020.

[47] Lijun Wang, Jianming Zhang, Yifan Wang, Huchuan Lu, and Xiang Ruan. Cliffnet for monocular depth estimation with hierarchical embedding loss. In *European Conference on Computer Vision*, pages 316–331. Springer, 2020.

[48] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[49] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[50] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5354–5362, 2017.

[51] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3917–3925, 2018.

[52] Guanglei Yang, Hao Tang, Mingli Ding, Nicu Sebe, and Elisa Ricci. Transformers solve the limited receptive field for monocular depth prediction. *arXiv preprint arXiv:2103.12091*, 2021.

[53] Xinchen Ye, Shude Chen, and Rui Xu. Dpnet: Detail-preserving network for high quality monocular depth estimation. *Pattern Recognition*, 109:107578, 2021.

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

[55] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.