# HQDec: Self-Supervised Monocular Depth Estimation Based on a High-Quality Decoder

Fei Wang, *Student Member, IEEE*, and Jun Cheng, *Member, IEEE*

***Abstract*— Decoders play significant roles in recovering scene depths. However, the decoders used in previous works ignore the propagation of multilevel lossless fine-grained information, cannot adaptively capture local and global information in parallel, and cannot perform sufficient global statistical analyses on the final output disparities. In addition, the process of mapping from a low-resolution (LR) feature space to a high-resolution (HR) feature space is a one-to-many problem that may have multiple solutions. Therefore, the quality of the recovered depth map is low. To this end, we propose a high-quality decoder (HQDec), with which multilevel near-lossless fine-grained information, obtained by the proposed adaptive axial-normalized position-embedded channel attention sampling module (AdaAxialNPCAS), can be adaptively incorporated into a LR feature map with high-level semantics utilizing the proposed adaptive information exchange scheme. In the HQDec, we leverage the proposed adaptive refinement module (AdaRM) to model the local and global dependencies between pixels in parallel and utilize the proposed disparity attention module to model the distribution characteristics of disparity values from a global perspective. To recover fine-grained HR features with maximal accuracy, we adaptively fuse the high-frequency information obtained by constraining the upsampled solution space utilizing the local and global dependencies between pixels into the HR feature map generated from the nonlearning method. Extensive experiments demonstrate that each proposed component improves the quality of the depth estimation results over the baseline results, and the developed approach achieves state-of-the-art results on the KITTI and DDAD datasets. The code and models will be publicly available at HQDec.**

Manuscript received 5 June 2023; revised 11 August 2023; accepted 2 September 2023. Date of publication 7 September 2023; date of current version 5 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grant U21A20487, in part by the Shenzhen Technology Project under Grant JCYJ20220818101206014 and Grant JSGG20140703092631382, in part by the Shenzhen Engineering Laboratory for 3D Content Generating Technologies under Grant [2017]476, and in part by the Chinese Academy of Sciences (CAS) Key Technology Talent Program. This article was recommended by Associate Editor D. Gragnaniello. *(Corresponding author: Jun Cheng.)*

Fei Wang is with the Guangdong-Hong Kong-Macao Joint Laboratory of Human–Machine Intelligence-Synergy Systems, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China, also with the Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: fei.wang2@siat.ac.cn).

Jun Cheng is with the Guangdong-Hong Kong-Macao Joint Laboratory of Human–Machine Intelligence-Synergy Systems, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: jun.cheng@siat.ac.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2023.3312721.

Digital Object Identifier 10.1109/TCSVT.2023.3312721

***Index Terms*— Depth estimation, high-quality decoder, self-supervised learning.**

## I. Introduction

DEPTH plays a crucial role in mobile robot vision and navigation [1], smart medical information technology [2], and industrial robots. Although the existing traditional methods [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15] can achieve competitive results, these methods require expensive depth sensors and considerable labor to obtain sufficient data labeled with pixel-level depth information or even require stereo video sequences [13], [14] for network training. Labeled depth data are expensive to acquire and can only be applied to limited scenarios. To alleviate this constraint, researchers have recently attempted to directly infer depths from large amounts of easily accessible unlabeled monocular videos in a self-supervised fashion, resulting in various objective functions [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28]. Such functions are used to seek the global optimal solution for the depth estimation network (DepthNet), and various network architectures [23], [25], [26], [29], [30], [31], [32], [33], [34], [35], [36], [37] have been designed to build robust DepthNet variants.

However, the existing DepthNet architectures still have many shortcomings. First, to compensate for the loss of fine-grained information caused by plain downsampling operations (e.g., max pooling), the existing methods either ignore (e.g., methods based on plain skip connections [38]) or fail to exploit (e.g., methods [23], [31], [39], [40] based on both dense connections and lossy downsampling) the fine-grained information contained in lower-level features. To this end, we propose a multilevel near-lossless fine-grained information fusion scheme. Second, the existing methods fail to adaptively utilize global and local information in parallel during the decoder stage to accurately infer depths. To this end, we propose AdaRM. Third, most current methods [16], [17], [18], [19], [21], [22], [29], [31], [33], [41], [42] fail to sufficiently and globally analyze the disparity values. To this end, we propose a global disparity attention module (AttDisp).

Specifically, the proposed multilevel near-lossless fine-grained information fusion scheme adaptively incorporates multilevel near-lossless fine-grained information with saliency information, obtained by the proposed AdaAxialNPCAS from

a HR feature map, into a LR feature map with high-level semantics and adaptively fuses the recovered high-frequency information (obtained by constraining the solution space utilizing both the global and local dependencies between pixels) into the HR feature map obtained from the nonlearning method. The proposed AdaRM (a) efficiently captures local information by utilizing a local filter and models long-range dependency-based transformer mechanisms in parallel during the decoding stage, and it (b) adaptively fuses the extracted local and global information into the original feature map. In the proposed AttDisp, we reweight the local disparity map by utilizing the global attention weights generated by computing the global correlation of the decoded feature map.

Finally, instead of scaling the prediction results based on median information, to solve the inherent scale ambiguity problem encountered by self-supervised monocular methods [16], [17], [18], [19], [21], [31], we propose an adaptive scale alignment strategy (AdaSearch) to scale the obtained estimation results to the ground truths measured via light detection and ranging (LiDAR) by considering both median and mean information.

Our main contributions are summarized as follows: 1) We propose a multilevel near-lossless fine-grained information fusion scheme to compensate for the loss of fine-grained information. 2) We propose an AdaRM to efficiently capture local and global information in parallel during the decoding stage and adaptively fuse the extracted local and global information into the original feature map. 3) We propose an AttDisp to model the distribution characteristics of the disparity values from a global perspective.

## II. Related Work

Recently, depth estimation algorithms [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [16], [17], [18], [19], [21], [22], [23], [24], [25], [26], [29], [30], [31], [32], [33], [34], [35], [36], [37], [42], [43], [44], [45], [46], [47], [48] based on deep learning have attracted considerable attention. These methods can be divided into supervised and self-supervised depth estimation approach depending on whether ground truths are needed. Supervised approaches require expensive depth sensors and considerable labor to obtain sufficient data labeled with pixel-level depth information for network training, while unsupervised methods alleviate this constraint and can directly utilize photometric differences as supervision signals to train networks for estimating depths and camera poses from unlabeled monocular videos.

### A. Supervised Depth Estimation

Supervised approaches require much manually labeled sparse point cloud data to guide the learning process of DepthNet.

Eigen et al. [3] first predicted a depth map from a single image by stacking coarse-scale and fine-scale networks. Subsequently, the depth estimation task was cast as a deep continuous conditional random field (CRF) learning problem [49], an ordinal regression problem [4], [50] or a detail transfer problem [5]. To restore local details, the Laplacian pyramid [15] was incorporated into the decoder architecture. However, ground-truth data were needed.

### B. Self-Supervised Depth Estimation

Despite their superior performance, supervised models are not universally applicable and heavily depend on the acquired ground truth-data. Moreover, the data annotation process is often slow and costly. The obtained annotations also suffer from structural artifacts. All these challenges strongly motivate us to infer depth in an unsupervised manner.

Zhou et al. [42] first proposed a completely unsupervised monocular estimation method. Subsequently, to explicitly address dynamic objects and occlusions, additional subnetworks [16], [19], more geometric prior knowledge [17], [18], [21], [27], [28], semantic information [22], [29], [30] and multiframe inputs [37], [43] were introduced.

Different from the above methods that either optimize objective functions or utilize multitask or mining time information to improve the quality of the resulting depth maps, good DepthNet architectures have also been designed to fit the functions mapped from images to the desired depth levels.

To predict high-quality depth maps, especially depth maps with sharp details, dense connections [23], [31], cross-scale feature fusion techniques [51], [52], [53], improved sampling operators (e.g., improved interpolation operators [32] based on a subpixel convolutional layer [54], the packed and unpacked operations [33] ) were developed. In addition, Zhang et al. [24] extended the perceptual area of the depth map produced over the source image by utilizing a multiscale scheme.

### C. Depth Estimation Based on Attention/Transformer

In recent years, efforts have been made to leverage attention mechanisms [55], in particular, that of Vision Transformer (ViT) [56], to enhance DepthNet to achieve improved depth estimation [6], [7], [8], [9], [10], [11], [12], [25], [26], [34], [35], [36], [37], [57]. Examples include depth attention volumes [6], multi modal fusion techniques [7], the AdaBins approach [8] based on ViT, and a depth value discretization strategy [34]/3D spatial correlations model [25] based on self-attention.

Instead of transferring details [5], ViT can be used as an encoder [9], [36] or directly plugged into the end of ResNet [10] to achieve more globally coherent predictions. Using a different approach from [9], [10], and [36], Lee et al. [11] modeled the relationships among neighboring pixels using the attention maps of each local patch. Based on [11], the structural information learned from image patches by exploiting graph convolution was also employed in [12] to attain improved performance.

To obtain per-pixel depth maps with sharper boundaries and richer details, Song et al. [35] modeled all of the position information contained in a given feature map based on the attention weights calculated for each channel. Kaushik et al. [26] forced their model to obtain rich contextual information by directly adding an attention map to the corresponding feature map. Meng et al. [50] fused the details

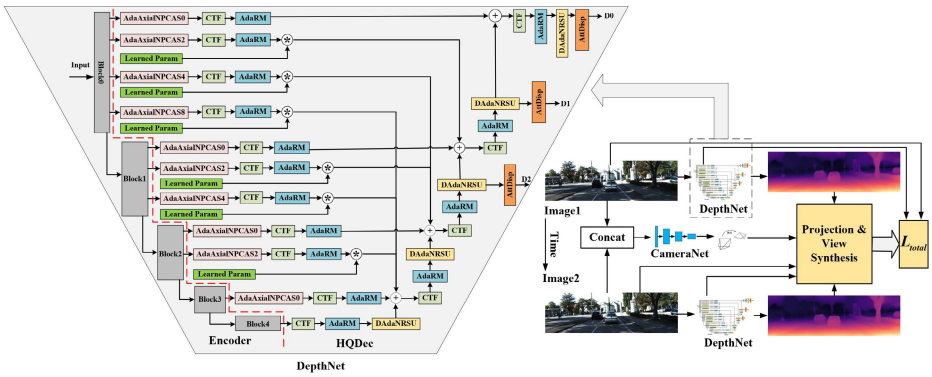



Fig. 1. Overview diagram of the connection between the encoder and the proposed HQDec and training architecture figure.

obtained by directly max pooling the HR feature map at one scale into the LR feature map. Han et al. [57] captured long-range dependencies via a Transformer backbone in the encoder stage while enhancing fine details utilizing pixelwise attention in the decoder stage. Guizilini et al. [37] utilized attention mechanisms to refine the per-pixel matching probabilities, resulting in improvements over the standard similarity metrics.

However, the above methods either ignore (e.g., methods [9], [11], [23], [32], [33] based on plain skip connections [38], in which only the information contained in same-level features is used.) or fail to exploit (e.g., the methods [35] and [50], which fuse low-level features obtained through lossy downsampling of the source image or by directly max pooling an HR feature map at one scale, respectively, and the method of [53], which only concatenates outputs at the same level from all intermediate stages) the multilevel near-lossless fine-grained information obtained from different HR feature maps, or they cannot adaptively capture global and local information in parallel during the decoding stage and adaptively fuse this information into the original feature map (e.g., methods based on pure ViT/CNNs, which model only global [9] or only local [23] information within a feature extraction layer). In addition, to our knowledge, the existing techniques (e.g., methods [16], [17], [18], [19], [21], [22], [29], [31], [33], [41], [42], which utilize only a local filter to regress the disparity map) do not sufficiently and globally analyze the output disparity values. In this paper, we propose (a) a multilevel near-lossless fine-grained information fusion scheme, (b) an AdaRM, and (c) an AttDisp to address these shortcomings.

## III. Method

The overview of the proposed HQDec is shown in Fig. 1.

### A. Problem Description and Optimization Objective

In the self-supervised monocular depth estimation task, our goal is to infer the corresponding scene depth from the given image. Due to the lack of ground-truth depth values, following a previously developed method [21], the loss function shown in formula (1) is used for the optimization objective to jointly train DepthNet and CameraNet on unlabeled monocular videos.

$$L_{total} = \lambda_p * \hat{L}_p + \lambda_d * L_d + \lambda_f * L_{feat} + L_s \quad (1)$$

where the hyperparameter settings in formula (1) are the same as those in the previous work [21]. $\hat{L}_p$ represents the bidirectional weighted photometric loss proposed in [21], $L_d$ represents the bidirectional depth structure consistency loss proposed in [21], and $L_s$ represents the smoothness loss used in [21]. $L_{feat}$ represents the bidirectional feature perception loss [21], which is weighted by both the adaptive weights and the bidirectional camera occlusion masks proposed in [21].

### B. AdaRM

It has been the consensus that the local feature in an image can be efficiently extracted by a CNNs, but such a network cannot model the long-range dependencies between pixels. On the other hand, although pure transformer mechanisms [55], [56] have unique advantages in terms of modeling global dependencies, they lack some of the inductive biases that are inherent to CNNs, such as translation equivariance and locality. These disadvantages aren't conducive to dense prediction tasks.

Different from previously developed approaches [9], [55], [56], [57], which draw long-term dependencies between sequences of input words [55] for machine translation tasks or sequences of linear patch embeddings split from images [56] for image classification tasks by utilizing pure multihead self-attention (MHA) mechanisms or model global relations by utilizing existing ViT as backbone [9], [57]. We adaptively exploit the strengths of both CNNs and MHA in parallel during the decoding stage and propose an AdaRM, as shown in Fig. 2. Concretely, the information of $X_{AdaRM}$ comes from the sum of the information of the three subbranches. The information in the left branch is the original information contained in $X_{in}$. This branch ensures that the information in $X_{in}$ can be fully propagated to the next stage ($X_{AdaRM}$). In the middle branch, we utilize CNNs to further model the local context information of the pixels in $X_{in}$ and adaptively propagate it to $X_{AdaRM}$ by employing a learnable parameter tensor, which guarantees that the information of each pixel in $X_{AdaRM}$ is obtained by performing weighted fusion on

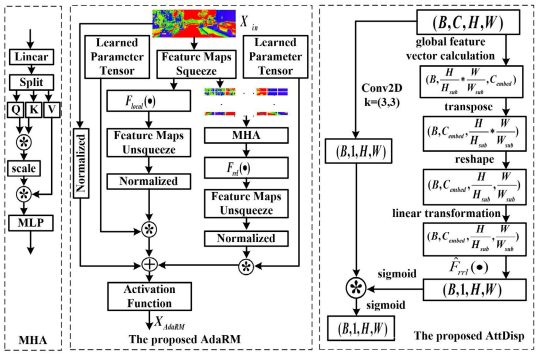



Fig. 2. The proposed AdaRM and AttDisp.

TABLE I
PARAMETER SETTINGS

| | Fine-tuning Module | | | | | Disp Module | | |
|---|---|---|---|---|---|---|---|---|
| | $s_0$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_0$ | $s_1$ | $s_2$ |
| $N_{sq}$ | 2 | 4 | 4 | 8 | 8 | - | - | - |
| $H_{sub}$ | $\frac{H_{in}}{16}$ | $\frac{H_{in}}{16}$ | $\frac{H_{in}}{16}$ | $\frac{H_{in}}{16}$ | $\frac{H_{in}}{32}$ | $\frac{H_{in}}{16}$ | $\frac{H_{in}}{16}$ | $\frac{H_{in}}{16}$ |
| $W_{sub}$ | $\frac{W_{in}}{16}$ | $\frac{W_{in}}{16}$ | $\frac{W_{in}}{16}$ | $\frac{W_{in}}{16}$ | $\frac{W_{in}}{32}$ | $\frac{W_{in}}{16}$ | $\frac{W_{in}}{16}$ | $\frac{W_{in}}{16}$ |
| $C_{embed}$ | 24 | 48 | 64 | 160 | 256 | 12 | 24 | 32 |
| $N$ | 2 | 4 | 4 | 8 | 8 | 2 | 4 | 4 |

the pixel information at the corresponding position and the surrounding positions in $X_{in}$. In the right branch, we utilize MHA to draw global dependencies between the sequences of subfeatures split from $X_{in}$ and adaptively propagate them to $X_{AdaRM}$ by employing a learnable parameter tensor, which ensures that the information in each subfeature map depends on the information in all remaining subfeature maps. As a result, each pixel in $X_{AdaRM}$ has the ability to perceive both local and global information without downsampling. For example, given pixel $x_{i,j}$ in row i and column j of $X_{in}$, we can calculate its corresponding output $x^o_{i,j}$ according to formula (2a). Formulas (2b) and (2c) make $x_{i,j}$ interact with its surrounding pixels and those pixels contained in the remaining subfeature maps, respectively. Therefore, the local and global context information of $x_{i,j}$ can be modeled.

$$x^o_{i,j} = x_{i,j} + \lambda_1 * x^{local}_{i,j} + \lambda_2 * x^{global}_{i,j} \tag{2a}$$

$$x^{local}_{i,j} = \sum w_{m,n} * x_{m,n} \tag{2b}$$

$$x^{global}_{i,j} = F_{rrl}(w^{corr}_l * f_l)[i,j] \tag{2c}$$

where $m \in \{i-1, i, i+1\}$ and $n \in \{j-1, j, j+1\}$. The $w_{m,n}$ denote the learnable weights. $f_l$ denotes the subfeature vector obtained by embedding the subfeature map containing pixel $x_{i,j}$ generated by sampling $X_{in}$. $w^{corr}_l$, which ensures that $f_l$ interacts with the remaining vectors, represents the global correlation weights between all subfeature vectors.

Specifically, suppose we are a feature map $X \in \Re^{C\times H\times W}$, where $C$, $H$, and $W$ are the number of channels, height, and width of the feature map, respectively. A feature map $X_3$ with local context information can be obtained by the function $F_{local}(\cdot)$ with a square filter (e.g., $3\times 3$) according to formula (3). To make the information denser, we first squeeze $X$ by employing the $F_{squeeze}(\cdot)$ function (e.g., a $1\times 1$ filter) before extracting local context information and then unsqueeze $X_2$ to its original dimensionality by utilizing $F_{unsqueeze}(\cdot)$ (e.g., a $1\times 1$ filter) for the subsequent feature fusion process.

$$X_1 = F_{squeeze}(X), \quad X_1 \in \Re^{\frac{C}{N_{sq}}\times H\times W} \tag{3a}$$

$$X_2 = F_{local}(X_1), \quad X_2 \in \Re^{\frac{C}{N_{sq}}\times H\times W} \tag{3b}$$

$$X_3 = F_{unsqueeze}(X_2), \quad X_3 \in \Re^{C\times H\times W} \tag{3c}$$

where $N_{sq}$ represents the squeezing ratio.

To obtain a feature map with global context information, we sample $X_1$ into a nonoverlapping subfeature map by employing the sample function $F_{sample}(\cdot)$, which can be implemented with a filter whose shape is the same as that of the subfeature map and whose cross-correlation stride is controlled according to the size of the filter. The subfeature map can be embedded into a feature vector $x$, which is employed to calculate the global correlation of the subfeature map via a learnable embedding function $F_{embed}(\cdot)$, as shown in formula (4).

$$x = F_{embed}(F_{sample}(X_1)), \quad x \in \Re^{\frac{H}{H_{sub}}*\frac{W}{W_{sub}}\times C_{embed}} \tag{4}$$

where $H_{sub}$, $W_{sub}$, and $C_{embed}$ denote the height and width of the subfeature map and the number of feature vectors, respectively. In $F_{embed}(\cdot)$, we first utilize a $1\times 1$ filter followed by an ELU action function to embed the sampled feature map into the desired dimension $C_{embed}$, flatten the embedded feature map into a one-dimensional tensor, and finally transpose it into the desired feature vector x.

To compute the global correlation weights of $x$, we first triple $x$ by utilizing a learnable function $F_{linear}(\cdot)$ (e.g., a linear layer), and then we reshape these feature vectors to $N$ subspaces via the function $F_{reshape}(\cdot)$ for the purpose of jointly focusing on the information derived from different representation subspaces at different positions. Thereafter, the feature vectors of each subspace are split into three parts, as shown in formula (5a); one of these parts serves as the query vector, and the other two parts serve as the key-value pairs. We can compute the global correlation weights by rescaling the product of the query vector and the corresponding key via the softmax function in each subspace. These weights are assigned to the corresponding value vectors, resulting in corresponding global feature vectors in the corresponding subspace. Then, the global feature vector $x_1$ containing the information from different representation subspaces at different positions can be obtained by concatenating and linearly transforming different subfeatures in different subspaces. Note that we adopt residual connections [58] to mitigate the degradation problem and utilize a multilayer perception function $F_{mlp}(\cdot)$, implemented by nonlinear transformation layers consisting of two linear layers followed by a Gaussian error linear unit function, to enhance the nonlinear fitting ability of the model. The final global feature vector $x_2$ in formula (5c) can then be

obtained.

$$Q, K, V = F_{split}(F_{reshape}(F_{linear}(x))) \tag{5a}$$

$$x_1 = F_{linear}(F_{reshape}(Softmax(QK^T * s)V)) \tag{5b}$$

$$x_2 = x + x_1 + F_{mlp}(x + x_1) \tag{5c}$$

where $Q, K, V \in \Re^{N \times \frac{H}{H_{sub}} * \frac{W}{W_{sub}} \times \frac{C_{embed}}{N}}$. $s = (\frac{C_{embed}}{N})^{-0.5}$ denotes the scale factor.

For information fusion purposes, the dimensions of $x_2$ and $X$ must be equal. To this end, $x_2$ is first mapped to subfeature map $X_4$ in formula (6a) by employing the learned function $F_{rrl}(\cdot)$, which can be implemented by a transpose convolution whose filter shape and stride are the same as those of the subfeature map. Then, the channels of $X_4$ in formula (6b) are aligned by a learned linear function $F_{unsqueeze}(\cdot)$ (e.g., a $1 \times 1$ filter). Finally, $X_{AdaRM}$ is obtained according to formula (6c).

$$X_4 = F_{rrl}(x_2), X_4 \in \Re^{\frac{C}{Nsq} \times H \times W} \tag{6a}$$

$$X_5 = F_{unsqueeze}(X_4), X_5 \in \Re^{C \times H \times W} \tag{6b}$$

$$X_{AdaRM} = X + P_1 * X_3 + P_2 * X_5, X_{AdaRM} \in \Re^{C \times H \times W} \tag{6c}$$

where $P_1, P_2 \in \Re^{H \times W}$ denotes learnable parameter tensors with initial values of zero.

### C. Multilevel Near-Lossless Fine-Grained Information Fusion

Fine-grained information has a significant impact on the performance achieved in dense prediction tasks. However, the loss of high-frequency information occurs as downsampling proceeds. Once high-frequency information is lost, it is difficult to recover fine-grained HR feature maps from LR maps after performing noninvertible low-pass filtering and subsampling operations [54], [59]. The existing methods design either plain skip connections [38] or dense connections [23], [31], [40] to reduce the degree of information loss. However, these schemes either ignore or fail to exploit the fine-grained information contained in lower-level features. To this end, we propose a multilevel near-lossless fine-grained information fusion scheme to compensate for the loss of fine-grained information. In the proposed scheme, we address this challenge by (1) improving the downsampling strategy to propagate more fine-grained information to LR feature maps, (2) adaptively incorporating multilevel fine-grained information into a high-level feature map, and (3) improving the upsampling strategy to recover as much high-frequency information as possible.

*1) Drive the Low-Resolution Feature Map to Retain More Fine-Grained Information:* Compared to an LR feature map with high-level semantics output by the encoder, a feature map with a higher resolution output in a shallower layer preserves more details. This motivates us to propagate more of the spatial structure information contained in the HR feature maps to the LR features with high-level semantics. Inspired by [33], [54], and [60], we propose AdaAxialNPCAS shown in Fig. 3. Instead of directly performing max pooling or

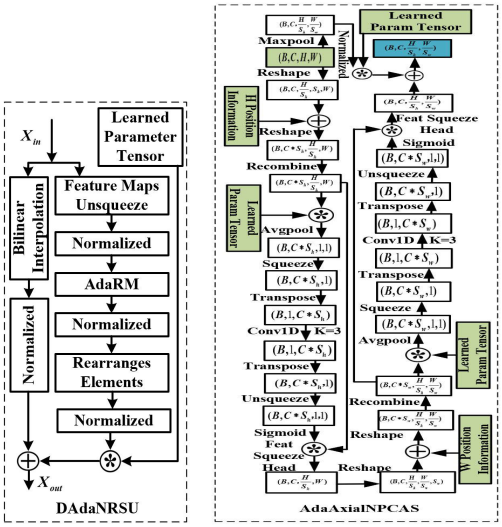



Fig. 3. The proposed AdaAxialNPCAS and DAdaNRSU.

stride downsampling, the proposed AdaAxialNPCAS directly folds the spatial dimensions of convolutional feature maps into extra feature channels along the axial direction to obtain the corresponding LR feature maps. The fact that the extra pixels in the HR feature maps are directly put into extra channels in the LR feature maps during transformation ensures that information is not lost because this transformation is reversible; only the positions where the pixel values are stored change, whereas the number of pixels and their values remain unchanged during the transformation. Different from a previously developed approach [33], which compresses the concatenated feature maps to a desired number of output channels via 3D convolutions, we first add the position information of each element in the original feature map to record the relative positions between the transformed pixels before executing recombination, and we then compute a channel head to represent each channel by utilizing learnable global average pooling in order to focus on the contribution of each pixel to the global information, considering that each pixel in a feature map does not contribute to the global information at exactly the same level. To give greater weights to the required channels, we reweight each channel by employing the weights calculated by the channel head and compress the reweighted feature maps to a desired shape.

Concretely, given a feature map $X \in \Re^{C \times H \times W}$, we first rearrange the elements in $X$ along the height direction and add the position information of each element in $X$ to the rearranged tensor for the purpose of losslessly squeezing the spatial information into the channel dimension, resulting in a new tensor $X_h \in \Re^{C * S_h \times \frac{H}{S_h} \times W}$, where $S_h$ denotes the number of downsampling operations along the height dimension. Then, we recombine the information contained in $X_h$ by utilizing group convolution. To model the dependencies between the channels of the recombined feature map, we first weight the elements by utilizing a learned parameter tensor, which is initialized to one, because different elements in the same channel play different roles in representing global channel information. We then perform global average pooling on the

spatial information of the weighted feature map to represent each channel and employ a Conv1D to learn correlations. Finally, the attention weights between the different channels can be obtained with a sigmoid function. The recombined feature map, whose channels contain the spatial information of $X$, is multiplied by the attention weights to let the model focus on the more useful spatial information of $X_h$. We then squeeze the channel to the desired shape by utilizing a squeezing head (e.g., $3\times3$ convolutions followed by an ELU). Similarly, we process the obtained feature map along the width direction. To preserve the most salient information contained in the LR feature map and adaptively fuse it into the feature map obtained by the above processing scheme, the sampled feature map with the salient information, obtained by max pooling, is normalized and multiplied by a learned parameter tensor, which is initialized to zero, before being fused.

*2) Adaptively Incorporate Multilevel Fine-Grained Information Into a High-Level Feature Map:* Among the existing works [31], [40], reference [40] directly fused lower-level detailed information with high-level semantic feature maps; however, the existing semantic gaps between different levels of information impede information fusion. Reference [31] fused different-level semantic information into higher-resolution feature maps. However, inaccurate semantic information has a negative impact on higher-resolution feature maps.

To recover scene depth as accurately as possible by utilizing multilevel hierarchical information, we propose AdaIE, which can adaptively incorporate multilevel fine-grained information into a high-level semantic feature map and let the model itself decide what fine-grained information needs to be fused and to what extent. Concretely, before the spatial structure details contained in the HR feature map are incorporated into the high-level semantic feature map, $P$ is multiplied by the corresponding feature map, as shown in formula (7). For example, the encoded feature map $X^3_{enc}$ is not only embedded with higher-level semantic information from $X^4_{dec}$ but also adaptively fused with the spatial structure details contained in the lower-level feature maps (e.g., $X^0_{enc}$, $X^1_{enc}$, etc.) before being decoded. Consequently, higher-level semantic information and more accurate spatial details can achieve complementary advantages, resulting in sharper scene depths.

$$X^k_{dec} \sim \begin{cases} X^{k+1}_{dec} + X^k_{enc}, & k = 0 \\ X^{k+1}_{dec} + X^k_{enc} + \sum_{i=0}^{k-1} P^i X^i_{enc}, & k > 0 \end{cases} \tag{7}$$

where $P^i$ is a learnable parameter tensor whose initial value is set to zero. $k \in \{0, 1, 2, 3\}$ represents the number of stages.

*3) Recover as Much High-Frequency Information as Possible by Constraining the Solution Space:* Upsampling, which directly affects the quality of the predicted depth maps, is an important component of DepthNet that is based on an encoder and a decoder. However, it is difficult to recover fine-grained HR feature maps from LR maps by utilizing nonlearning methods because the process of mapping from a LR feature space to a HR feature space is a one-to-many problem that may have multiple solutions. To this end, different from the existing methods that recover HR feature maps with only bilinear or nearest interpolation, in which only the local context information can be considered, we propose DAdaNRSU (shown in Fig. 3), which recovers more high-frequency information by adaptively modeling the local and global dependencies between pixels to restrict the solution space and adaptively fuse this information into the coarse-grained HR feature maps obtained by traditional upsampling methods.

Specifically, given a feature map $X_{in} \in \Re^{C\times H\times W}$, on the one hand, the coarse-grained HR feature map $X^1_{high}$, which helps with providing a good initial value for each layer of the decoder at the beginning of the training process, can be recovered via traditional upsampling methods, which are commonly used in decoders [16], [17], [42]. On the other hand, $X_{in}$ can be expanded by a learned function $F_{expand}(\cdot)$ with a $1\times1$ filter into a higher-dimensional feature subspace to recover the HR feature map in formula (8). To seek additional constraints that limit the solution space, we model the local and long-range dependencies between the pixels in $X^1_{low}$ by utilizing formula (6c). Then, the HR feature map $X^2_{high}$ can be obtained by rearranging the elements in $X^2_{low}$ to form a new feature space $\Re^{C\times2H\times2W}$ by employing the pixel shuffling function $F_{pixelshuffle}(\cdot)$ from [61]. Finally, we adaptively fuse $X^2_{high}$, normalized by formula (8e), into a normalized version $X^1_{high}$ by utilizing a learned parameter tensor $P$ that is initialized to zero.

$$X^1_{low} = F_{expand}(X_{in}), \quad X^1_{low} \in \Re^{4C\times H\times W} \tag{8a}$$
$$X^2_{low} = F_{refine}(X^1_{low}), \quad X^2_{low} \in \Re^{4C\times H\times W} \tag{8b}$$
$$X^2_{high} = F_{pixelshuffle}(X^2_{low}), \quad X^2_{high} \in \Re^{C\times2H\times2W} \tag{8c}$$
$$X_{high} = X^1_{high} + P * X^2_{high}, \ P, X_{high} \in \Re^{C\times2H\times2W} \tag{8d}$$
$$Y = (X - E(X))/\sigma(X) \tag{8e}$$

where $E(\cdot)/\sigma(\cdot)$ are the mean/standard deviation.

### D. Disparity Attention Module

The disparity output layer, which plays an important role in the process of transforming the decoded feature map into the desired disparity information, is an important part of the depth estimation network. Most current methods [16], [17], [18], [19], [21], [22], [29], [31], [33], [41], [42] directly use local 2D convolution followed by a sigmoid function to regress the decoded feature map to the disparity values, however, this technique is incapable of supporting a sufficient global analysis to infer the disparity value of the current pixel, causing inaccurate depths to be regressed in some scenes (for example, when objects exist in a scene that have identical surfaces but different actual depths, some areas of these objects may be inferred to have the same depth because these regions have highly similar or even identical pixel values). Intuitively, sufficient context information can provide better semantic pixel information, and long-range dependencies outside similar objects in the scene can provide additional constraints to help accurately predict the depth information of the current pixel. To this end, we propose an AttDisp (shown in Fig. 2) to infer the disparity value from the corresponding decoded feature map by utilizing sufficient context information.

Specifically, given the decoded feature map $X_{dec} \in \Re^{C\times H\times W}$, we can compute the corresponding global feature vectors $x_{dec} \in \Re^{\frac{H}{H_{sub}}*\frac{W}{W_{sub}}\times C_{embed}}$ according to formulas (4) and (5). Different from the purpose of the global feature vectors in Section III-B, we expect to obtain the global correlation matrix of the disparity information from $x_{dec}$. To this end, we first reshape $x_{dec}^T \in \Re^{C_{embed}\times\frac{H}{H_{sub}}*\frac{W}{W_{sub}}}$ into a matrix and then utilize learnable linear functions (e.g., $1\times 1$ convolution functions) to linearly transform this matrix into the corresponding subfeature map $X_{gc} \in \Re^{C_{embed}\times\frac{H}{H_{sub}}\times\frac{W}{W_{sub}}}$. We then map $X_{gc}$ into the attention feature map $X_{att} \in \Re^{1\times H\times W}$ by utilizing the learned function $\hat{F}_{rrl}(\cdot)$ implemented by a transpose convolution, whose filter shape and stride are the same as those of the subfeature map. In parallel, we map $X_{dec}$ into the corresponding disparity map $D_{local} \in \Re^{1\times H\times W}$ by utilizing a 2D convolution kernel. To endow $D_{local}$ with a global visual field, $D_{local}$ is multiplied by the global attention weights generated by activating $X_{att}$ using a sigmoid function.

### E. Adaptive Scale Alignment Strategy

The depth scales predicted directly from monocular videos are unknown. Instead of calculating the scale factor based on median information [16], [17], [18], [19], [21], [31], we propose an AdaSearch, shown in formula (9), where both the median and mean information can be considered to obtain the scale factor.

$$scale_{ada} = \frac{\zeta * D^{gt}_{median} + (1-\zeta) * D^{gt}_{mean}}{\zeta * D^{pred}_{median} + (1-\zeta) * D^{pred}_{mean}} \tag{9}$$

The median scale is a special case in which $\zeta = 1$. We divide the interval from 0 to 1 into 10 equal pieces. For each value, we calculate the corresponding relative absolute value error, and the scale factor corresponding to the minimum error is used as the scale factor of the current frame depth.

### F. Network Design

*1) DepthNet:* EfficientNetV2-s [62], without a classifier, is used as an encoder. To build DepthNet, the encoder is divided into five blocks according to the resolution of the output feature map. The encoded feature map output by each block is refined utilizing formula (10). The overall connection diagram between the encoder and the HQDec is shown in Fig. 1.

$$X^{(i,j)}_{enc} = F_{refine}(F_{ct}(F^{(i,j)}_{down}(X^{i}_{enc}))) \tag{10}$$

where $i \in \{0,\ldots,3\}$, $j \in \{0,\ldots,3-i\}$, $F^{(i,j)}_{down}(\cdot)$ denotes AdaAxialNPCAS operator, and $F^{(i,j)}_{down}(X^{i}_{enc})$ means that the encoded feature map $X^{i}_{enc}$ at stage $i$ is downsampled $2^j$ times. $F_{ct}(\cdot)$ represents the channel transformation function that is implemented by a $3\times 3$ filter followed by an ELU function. $F_{refine}(\cdot)$ denotes AdaRM operator.

Therefore, the decoded feature map $X^k_{dec}$ at stage $k$ can be obtained by rewriting formula (7), and the calculation process is shown in formula (11).

Following previous work [21], the decoded feature map $X^k_{dec}, k \in \{0,1,2\}$ is used as the corresponding candidate feature to generate disparity information. According to formula (12), we map the decoded feature map $X^k_{dec}$ to the desired disparity. Finally, as is common practice [16], [17], [21], [42], the estimated disparity value is mapped into an actual distance between 0.1 and 100 meters according to formula (13).

$$X^k_{dec} = \begin{cases} F_{up}(F_{refine}(X^4_{enc})) + X^{(3,0)}_{enc} \\ \quad + \sum_{n=0}^{2} P^{(i,3-n)} X^{(i,3-n)}_{enc}, & k=3 \\ F_{up}(F_{refine}(X^{k+1}_{dec})) + X^{(k,0)}_{enc} \\ \quad + \sum_{n=0}^{k-1} P^{(i,k-n)} X^{(i,k-n)}_{enc}, & 0<k\le 2 \\ F_{up}(F_{refine}(X^1_{dec})) + X^{(0,0)}_{enc}, & k=0 \end{cases} \tag{11}$$

$$D^k_{disp} = F_{disp}(F_{up}(F_{refine}(F_{ct}(X^k_{dec})))) \tag{12}$$

$$\hat{D}^k = 1/(10 * D^k_{disp} + 0.01) \tag{13}$$

where $F_{up}(\cdot)$ denotes the porposed DAdaNRSU operator. $F_{disp}(\cdot)$ denotes AttDisp operator. $\hat{D}^k$ and $D^k_{disp}$ denote the corresponding estimated depth and disparity, respectively.

*2) CameraNet:* The high-level semantic features, encoded by FBNetV3-B [63], are first squeezed by a $1\times 1$ filter with a stride of 1 and then transformed by employing two $3\times 3$ convolution layers followed by a rectified linear unit (ReLU) function. Finally, the relative pose, which is parameterized with 6 degrees of freedom (DOFs) whose first three dimensions denote translation and whose last three dimensions denote the rotation vector, is regressed by aggregating the estimation values at all spatial locations via global average pooling.

## IV. Experiments

### A. Dataset

We conducted experiments on the KITTI RAW dataset [64], the KITTI Odometry dataset [65] and the DDAD dataset [33]. Similar to previous related works [16], [42], the KITTI RAW dataset was split as in [3], with approximately 40k frames used for training and 5k frames used for validation. We evaluated DepthNet on test data consisting of 697 test frames in accordance with Eigen's testing split, and we evaluated CameraNet on sequences $09-10$ of the KITTI Odometry dataset. The DDAD dataset was split as in [33], where 150 scene videos from the front camera were used for training and 50 scene videos were used for evaluation purposes.

### B. Training Details

The proposed learning framework was implemented using the PyTorch Library [61]. The training set was augmented using ColorJitter by randomly changing the brightness, contrast, saturation, and hue of each image, as well as performing random scaling, random horizontal flipping, and random cropping. During the training process, three consecutive video frames were used as a training sample and fed to the model. The same hyperparameter setting as that in [21] was adopted

TABLE II
PARAMETER SETTINGS

| Dataset | DepthNet | CameraNet | Batchsize | IterNum | LR |
|---|---|---|---|---|---|
| KITTIRAW | $128 \times 416$ | $128 \times 416$ | 32 | 100,000 | $1e^{-4}$ |
| KITTIRAW | $192 \times 640$ | $128 \times 416$ | 24 | 50,000 | $0.5e^{-4}$ |
| KITTIRAW | $320 \times 1024$ | $128 \times 416$ | 10 | 50,000 | $0.3e^{-4}$ |
| KITTIRAW | $384 \times 1280$ | $128 \times 416$ | 8 | 50,000 | $0.2e^{-4}$ |
| DDAD | $384 \times 640$ | $384 \times 640$ | 12 | 100,000 | $0.5e^{-4}$ |

for the loss function. Rather than storing all intermediate activations of the entire computed graph for backward computation, we instead recomputed them during the backward pass to save memory. DepthNet and CameraNet were coupled by the loss function from [21] and jointly trained on an RTX 3090Ti GPU from scratch using the AdamW [66] optimizer. The model for HR input images was fine-tuned by utilizing the weight obtained from the adjacent LR experiments. With increasing resolution, the batch size and learning rate were gradually reduced. The details are shown in Table II.

### C. Comparison With the State-of-the-Art Methods

In Table III, we quantitatively compare the proposed HQDec with the previously developed state-of-the-art methods at different resolutions. The results show that the HQDec outperformed these state-of-the-art completely unsupervised monocular estimation methods [17], [18], [21], [23], [24], [31], [33], [34], [35], [68], [69], [70], [74] and monocular estimation methods [22], [29], [30], [75] guided by semantic labels at the same resolution. The HQDec, which only utilized a single frame to estimate the corresponding depth during inference, could achieve better performance that was similar to or even better than that of the multiframe approaches [37], [43]. In Table IV, we also provide the results evaluated on the improved KITTI ground-truth data [72]. These results also confirmed that the proposed HQDec outperformed all previously published self-supervised methods that estimate the corresponding depth map by utilizing only a single frame during inference. Benefitting from the estimated high-quality depth, we also achieved competitive pose estimation errors, although CameraNet and DepthNet were only trained on the KITTI RAW dataset, as shown in Table VII. To further verify the HQDec's generalization ability, we also evaluated the HQDec on the more challenging DDAD dataset [33]. The results in Table V show that the HQDec outperformed all previously published self-supervised methods, including the approaches that utilize multiple frames to predict the corresponding depth map and methods that use semantic labels as supervision signals.

In Fig. 6, we conduct a qualitative comparison with the previously developed methods [18], [21], [31], [33] on some challenging samples selected from the KITTI dataset. In the left/middle samples, the previously developed methods [18], [21], [31], [33] tended to inaccurately estimate the depths in some areas (e.g., the white wall in the left image and the white tent in the middle image). In the right sample, although the methods from [18], and [31] accurately estimated the depth of the white car, they failed to accurately predict the depth of the roadside lawn. The methods in [21], and [33] exhibited the opposite phenomenon. However, the proposed HQDec could accurately predict the corresponding depths in these scenarios. We attribute this phenomenon to the fact that the HQDec could utilize multilevel features with global and local context information to infer the current depth. The model was told that the same object (e.g., the white wall, the white tent, etc.) should have a similar depth by establishing long-range dependencies between the pixels. In addition, compared with the existing methods, the HQDec could accurately predict not only the depth of the moving car by modeling the relationships between the moving car and its surrounding pixels but also the depth of the lawn in the right sample by modeling the long-range dependencies between the pixels in the lawn area.

Fig. 4 shows that areas in the scene that should theoretically have similar depths were given similar weights by AttDisp.

In Fig. 10, we present the qualitative results obtained by the different methods on the DDAD dataset. A similar phenomenon as that produced on the KITTI dataset can be observed. Examples include the trucks in the left/middle samples, the bus in the right sample, and the moving white car in the middle sample. The previous approaches [21], [33] inaccurately inferred the depths in these regions, while the proposed HQDec accurately estimated the corresponding depths.

Table VI shows the complexity levels of the tested models. Our method exhibited relatively low computational complexity and can satisfy real-time requirements.

### D. Ablation Studies

To better understand the contribution of each component proposed in Section III to the overall performance of our method, we performed ablation studies on the baseline and different variants of each component in Tables IX, X, XI, XII, XIII, and XIV. The differences between the different variants are as follows. Compared with 'AdaRM', 'RM' in Table XII directly fuses both the local and global extracted information into the original feature map. Compared with 'DAdaNRSU', 'AdaNRSU' in Table IX models local and long-range dependencies between pixels by utilizing 'RM' instead of 'AdaRM'. 'NRCU' directly fuses the high-frequency information recovered by 'RM' into a coarse-grained HR feature map (obtained by bilinear interpolation) by concatenating along the channel dimension. Compared with 'NRCU', 'RCU' directly fuses the corresponding feature maps without normalizing them. In Table XIII, compared with 'AdaAxialNPCAS', 'AdaNPCAS' rearranges the elements in the feature map in a nonaxial way, and recombines the information contained in the rearranged tensor (where the position information of each element in the original feature map is not added) via standard convolution. Compared with 'AdaNPCAS', 'AdaNCAS' represents each channel by treating each element in the same channel equally. Compared with 'AdaNCAS', 'NCAS' performs information fusion directly by concatenating along the channel dimension. Compared with 'NCAS', 'CAS' only utilizes the channel attention sampling module to obtain a LR feature map.

TABLE III

PERFORMANCE COMPARISON CONDUCTED ON THE KITTI DATASET WITH MONOCULAR DEPTH ESTIMATION CAPPED AT 80 M. THE PREDICTION RESULTS WERE ALIGNED BY THE MEDIAN GROUND-TRUTH LIDAR INFORMATION. 'M'/'S': SELF-SUPERVISED MONO/STEREO SUPERVISION. 'MULTI-FR.' INDICATES THAT THE DEPTH MAP WAS PREDICTED BY UTILIZING MULTIPLE FRAMES DURING THE INFERENCE PROCESS. '‡' INDICATES THAT THE PREDICTION RESULTS WERE ALIGNED BY THE PROPOSED ADASEARCH. 'DE/PE' REFERS TO THE BACKBONE OF THE ENCODER USED IN THE DEPTH/POSE ESTIMATION NETWORK

| Method | DE | PE | Data | RES | Sup + Multi-Fr? | Error↓ | | | | Accuracy↑ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | AbsRel | SqRel | RMSE | RMSE log | $\delta_1$ | $\delta_2$ | $\delta_3$ |
| Watson et al. [43] | RN18 | RN18 | K+CS | 192×640 | M+Multi-Fr | 0.098 | 0.770 | 4.459 | 0.176 | 0.900 | 0.965 | 0.983 |
| Guizilini et al. [37] | Depthformer | RN18 | K+CS | 192×640 | M+Multi-Fr | **0.090** | 0.661 | 4.149 | 0.175 | 0.905 | 0.967 | 0.984 |
| Zhou et al. [67] | Swin | - | K | 384×1280 | S | **0.090** | **0.538** | **3.896** | **0.169** | **0.906** | **0.969** | **0.985** |
| Li et al. [68] | DN | RN18 | K | 128×416 | M | 0.130 | 0.950 | 5.138 | 0.209 | 0.843 | 0.948 | 0.978 |
| Godard et al. [18] | RN18 | RN18 | K | 128×416 | M | 0.128 | 1.087 | 5.171 | 0.204 | 0.855 | 0.953 | 0.978 |
| Yan et al. [69] | RN50 | RN50 | K | 128×416 | M | 0.116 | 0.893 | 4.906 | 0.192 | 0.874 | 0.957 | 0.981 |
| **HQDec (Ours)** | EffV2s | FBv3 | K | 128×416 | M | 0.103 | 0.706 | 4.569 | 0.176 | 0.882 | 0.962 | 0.985 |
| **HQDec (Ours)**‡ | EffV2s | FBv3 | K | 128×416 | M | **0.099** | **0.693** | **4.494** | **0.173** | **0.887** | **0.963** | **0.985** |
| Lyu et al. [31] | RN18 | RN18 | K | 192×640 | M | 0.109 | 0.792 | 4.632 | 0.185 | 0.884 | 0.962 | 0.983 |
| Petrovai et al. [70] | RN50 | RN18 | K+CS | 192×640 | M | 0.100 | 0.661 | 4.264 | 0.172 | 0.896 | 0.967 | **0.985** |
| Zhao et al. [51] | MPVit | RN18 | K | 192×640 | M | 0.099 | 0.708 | 4.372 | 0.175 | 0.900 | 0.967 | 0.984 |
| He et al. [52] | HR18 | RN18 | K | 192×640 | M | 0.096 | **0.632** | **4.216** | 0.171 | **0.903** | **0.968** | **0.985** |
| Han et al. [57] | Swin | PN7 | K | 192×640 | M | 0.098 | 0.728 | 4.458 | 0.176 | 0.898 | 0.966 | 0.984 |
| **HQDec (Ours)** | EffV2s | FBv3 | K | 192×640 | M | 0.096 | 0.654 | 4.281 | 0.169 | 0.896 | 0.965 | **0.985** |
| **HQDec (Ours)**‡ | EffV2s | FBv3 | K | 192×640 | M | **0.092** | 0.642 | 4.233 | **0.167** | 0.899 | 0.966 | **0.985** |
| Godard et al. [18] | RN18 | RN18 | K | 320×1024 | M | 0.115 | 0.882 | 4.701 | 0.190 | 0.879 | 0.961 | 0.982 |
| Lyu et al. [31] | RN18 | RN18 | K | 320×1024 | M | 0.106 | 0.755 | 4.472 | 0.181 | 0.892 | 0.966 | 0.984 |
| Masoumian et al. [71] | RN50 | RN18 | K | 320×1024 | M | 0.104 | 0.720 | 4.494 | 0.181 | 0.888 | 0.965 | 0.984 |
| Zhou et al. [53] | HR18 | RN18 | K | 320×1024 | M | 0.097 | 0.722 | 4.345 | 0.174 | 0.907 | 0.967 | 0.984 |
| **HQDec (Ours)** | EffV2s | FBv3 | K | 320×1024 | M | 0.093 | 0.654 | 4.102 | 0.165 | 0.906 | **0.968** | **0.986** |
| **HQDec (Ours)**‡ | EffV2s | FBv3 | K | 320×1024 | M | **0.088** | **0.638** | **4.052** | **0.163** | **0.909** | **0.968** | 0.985 |
| Guizilini et al. [33] | PackNet | PN7* | K | 384×1280 | M | 0.107 | 0.802 | 4.538 | 0.186 | 0.889 | 0.962 | 0.981 |
| Lyu et al. [31] | RN18 | RN18 | K | 384×1280 | M | 0.104 | 0.727 | 4.410 | 0.179 | 0.894 | 0.966 | 0.984 |
| Wang et al. [21] | RN50 | PN7 | K+CS | 384×1280 | M | 0.104 | 0.798 | 4.501 | 0.184 | 0.889 | 0.961 | 0.982 |
| Yan et al. [69] | RN50 | RN50 | K | 384×1280 | M | 0.102 | 0.715 | 4.312 | 0.176 | 0.900 | 0.968 | 0.984 |
| **HQDec (Ours)** | EffV2s | FBv3 | K | 384×1280 | M | 0.092 | 0.634 | 4.079 | **0.164** | 0.908 | **0.968** | **0.986** |
| **HQDec (Ours)**‡ | EffV2s | FBv3 | K | 384×1280 | M | **0.088** | **0.624** | **4.031** | **0.162** | 0.911 | 0.968 | **0.986** |

The results in Table IX show the impacts of different upsampling schemes on the resulting depth estimation performance. The results demonstrate that the performance could be improved by replacing the bilinear interpolation-based upsampling (BIU) process in the decoder with the proposed refined upsampling (RCU) method. However, its parameters also increased. To this end, we increased the model capacity of 'BIU' by stacking more convolutions performing after bilinear interpolation in the 'BIU' scheme, resulting in the 'BIU*' scheme. Although scheme 'BIU*' performed better than the 'BIU' scheme, it was weaker than the 'NRCU' scheme. We believe that this may be because the HR feature maps obtained by BIU and formula (8c) had different data distribution characteristics and semantic levels, making information fusion difficult. The results of the 'AdaNRSU' scheme in Table IX show that the depth predictions could be further improved by allowing the model itself to adaptively decide what information contained in the HR feature map (obtained by rearranging the elements in a LR tensor to a HR tensor) needs to be fused into the HR feature map obtained by BIU. Benefitting from the proposed 'AdaRM' that could adaptively fuse local and global information, the depth was further improved by replacing the 'RM' module with 'AdaRM'. Although 'DAdaNRSU' and 'BIU' had similar numbers of parameters, 'DAdaNRSU' outperformed 'BIU*'.

Table X shows the impacts of different schemes for exchanging information between the multilevel HR feature maps derived from the shallower layers and the feature maps with higher-level semantics on the predicted depth. Compared with using only the UNet-based [38] style connection ('w/o IE' in Table X) between the encoder and decoder, the 'plain IE' could incorporate multilevel spatial information from the shallower layers of different stages into a feature map with higher-level semantics, resulting in better depth predictions. The results of the 'AdaIE' in Table X indicate that the performance can be further improved if the model is allowed to adaptively decide which levels of spatial information from the different stages of the encoder in the shallower layers should be introduced to the feature map with higher-level semantics and how much spatial information should be incorporated into these feature maps. Moreover, compared with 'BIU*', which is equivalent to stacking convolutions directly after performing bilinear interpolation in the 'w/o IE' scheme with a larger number of parameters, the 'AdaIE' scheme outperformed 'BIU*'.

TABLE IV

MONOCULAR DEPTH ESTIMATION PERFORMANCE COMPARISON CONDUCTED ON THE IMPROVED KITTI DATASET [72]

| Method | DE | PE | Data | RES | Sup + Multi-Fr? | Error↓ | | | | Accuracy↑ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | AbsRel | SqRel | RMSE | RMSE log | $\delta_1$ | $\delta_2$ | $\delta_3$ |
| Watson et al. [43] | RN18 | RN18 | K+CS | 192×640 | M+Multi-Fr | 0.064 | 0.320 | 3.187 | 0.104 | 0.946 | 0.990 | 0.995 |
| Guizilini et al. [37] | Depthformer | RN18 | K+CS | 192×640 | M+Multi-Fr | **0.055** | 0.271 | 2.917 | 0.095 | 0.955 | 0.991 | **0.998** |
| Guizilini et al. [37] | Depthformer | RN18 | K+CS | 352×1216 | M+Multi-Fr | **0.055** | **0.265** | **2.723** | **0.092** | **0.959** | **0.992** | **0.998** |
| Godard et al. [18] | RN18 | RN18 | K | 192×640 | M | 0.090 | 0.545 | 3.942 | 0.137 | 0.914 | 0.983 | 0.995 |
| Guizilini et al. [33] | PackNet | PN7* | K+CS | 192×640 | M | 0.078 | 0.420 | 3.485 | 0.121 | 0.931 | 0.986 | 0.996 |
| Zhou et al. [53] | HR18 | RN18 | K | 192×640 | M | 0.076 | 0.414 | 3.493 | 0.119 | 0.936 | 0.988 | 0.997 |
| Zhao et al. [51] | MPVit | RN18 | K | 192×640 | M | 0.075 | 0.389 | 3.419 | 0.115 | 0.938 | 0.989 | 0.997 |
| He et al. [52] | HR18 | RN18 | K | 192×640 | M | 0.074 | 0.362 | 3.345 | 0.114 | 0.940 | 0.990 | 0.997 |
| Wang et al. [21] | RN50 | PN7 | K+CS | 256×832 | M | 0.077 | 0.425 | 3.537 | 0.121 | 0.934 | 0.985 | 0.996 |
| **HQDec (Ours)** | EffV2s | FBv3 | K | 128×416 | M | 0.074 | 0.403 | 3.746 | 0.121 | 0.930 | 0.986 | 0.997 |
| **HQDec (Ours)**‡ | EffV2s | FBv3 | K | 128×416 | M | 0.071 | 0.384 | 3.632 | 0.117 | 0.935 | 0.987 | 0.997 |
| **HQDec (Ours)** | EffV2s | FBv3 | K | 192×640 | M | 0.065 | 0.328 | 3.289 | 0.107 | 0.945 | 0.990 | 0.997 |
| **HQDec (Ours)**‡ | EffV2s | FBv3 | K | 192×640 | M | 0.062 | 0.318 | 3.231 | 0.105 | 0.948 | 0.990 | 0.997 |
| **HQDec (Ours)** | EffV2s | FBv3 | K | 256×832 | M | 0.060 | 0.298 | 3.004 | 0.100 | 0.955 | 0.991 | **0.998** |
| **HQDec (Ours)**‡ | EffV2s | FBv3 | K | 256×832 | M | **0.058** | **0.289** | **2.953** | **0.098** | **0.958** | **0.992** | **0.998** |
| Godard et al. [18] | RN18 | RN18 | K | 320×1024 | M | 0.086 | 0.462 | 3.577 | 0.127 | 0.924 | 0.986 | 0.996 |
| Lyu et al. [31] | RN18 | RN18 | K | 384×1280 | M | 0.075 | 0.357 | 3.239 | 0.113 | 0.937 | 0.991 | **0.998** |
| Guizilini et al. [33] | PackNet | PN7* | K+CS | 384×1280 | M | 0.071 | 0.359 | 3.153 | 0.109 | 0.944 | 0.990 | 0.997 |
| Wang et al. [21] | RN50 | PN7 | K+CS | 384×1280 | M | 0.072 | 0.377 | 3.358 | 0.115 | 0.939 | 0.987 | 0.996 |
| **HQDec (Ours)** | EffV2s | FBv3 | K | 320×1024 | M | 0.061 | 0.296 | 2.944 | 0.099 | 0.956 | **0.992** | **0.998** |
| **HQDec (Ours)**‡ | EffV2s | FBv3 | K | 320×1024 | M | **0.058** | 0.286 | **2.896** | **0.097** | **0.959** | **0.992** | **0.998** |
| **HQDec (Ours)** | EffV2s | FBv3 | K | 384×1280 | M | 0.061 | 0.292 | 2.976 | 0.099 | 0.956 | **0.992** | **0.998** |
| **HQDec (Ours)**‡ | EffV2s | FBv3 | K | 384×1280 | M | **0.058** | **0.284** | 2.921 | **0.097** | **0.959** | **0.992** | **0.998** |

TABLE V

MONOCULAR DEPTH ESTIMATION PERFORMANCE COMPARISON CONDUCTED ON THE DDAD DATASET [33]

| Method | DE | PE | Sup + Multi-Fr? | RES | Error↓ | | | | Accuracy↑ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | AbsRel | SqRel | RMSE | RMSElog | $\delta_1$ | $\delta_2$ | $\delta_3$ |
| Guizilini et al. [33] | PackNet | PN7* | M | 384×640 | 0.162 | 3.917 | 13.452 | 0.269 | 0.823 | - | - |
| Han et al. [57] | Swin | PN7 | M | 384×640 | 0.151 | 3.591 | 14.350 | 0.244 | - | - | - |
| Guizilini et al. [73] | RN101 | RN18 | M+Semantic | 384×640 | 0.147 | 2.922 | 14.452 | - | 0.809 | - | - |
| Watson et al. [43] | RN18 | RN18 | M+Multi-Fr. | - | 0.146 | 3.258 | 14.098 | - | 0.822 | - | - |
| Guizilini et al. [37] | Depthformer | RN18 | M+Multi-Fr. | - | 0.135 | 2.953 | 12.477 | - | 0.836 | - | - |
| Wang et al. [21] | RN50 | PN7 | M | 384×640 | 0.121 | 1.229 | 6.721 | 0.187 | 0.853 | 0.958 | **0.985** |
| **HQDec (Ours)** | EffV2s | FBv3 | M | 384×640 | 0.113 | 1.214 | 6.286 | 0.176 | 0.876 | 0.963 | **0.985** |
| **HQDec (Ours)**‡ | EffV2s | FBv3 | M | 384×640 | **0.107** | **1.173** | **6.109** | **0.171** | **0.885** | **0.964** | **0.985** |

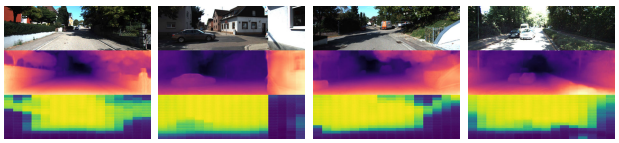


Fig. 4. Visualization of the attention weights. Areas of similar colors or the same color in the third row of Fig. 4 represent similar or even identical weight values. The farther away a position is, the closer the corresponding weight value is to one. The smaller the distance is, the closer the corresponding weight value is to zero. As seen from the RGB images, the depth values of the regions corresponding to these areas should be similar or even the same. These include the center areas and the leaf areas in the first column of attention maps, the surface areas of the car and the wall on the right in the second column of attention maps, the tent on the right and the leaf areas in the upper right corner in the third column of attention maps, and the leaf areas in the right and center areas in the fourth column of attention maps.

Table XI shows the influence of the disparity module on the resulting model performance. The results demonstrate that the performance could be improved if the traditional disparity module (corresponding to the results of 'Conv2D Disp' in Table XI) was assisted with sufficient global statistical properties (corresponding to the results of 'AttDisp' in Table XI). We believe that this may be because sufficient contextual information can provide more semantics for the current pixel, resulting in the model becoming capable of inferring more accurate depths. For example, the learned semantic knowledge can tell the model that the surfaces of the objects in a scene should have more similar depths. Furthermore, compared with 'BIU*', which utilizes the traditional disparity module but has a larger number of parameters, 'AttDisp' still achieved better depth prediction results.

Table XII shows the impacts of different refinement schemes on the depth estimation results. Compared with 'w/o IE'

TABLE VI

COMPLEXITY AND OFFLINE INFERENCE TIME COMPARISON AMONG THE DIFFERENT DEPTHNETS WITH BATCH SIZE = 1. ALL RESULTS WERE EVALUATED ON AN RTX 4090 GPU WITH THE SAME SETTINGS. THE RESOLUTION WAS SET TO 128 × 416. 'GPU-UTILI' REPRESENTS GPU UTILIZATION OVER MOST OF THE TIME RANGE DURING THE INFERENCE PROCESS. A SMALLER VALUE INDICATES A LONGER WAIT TIME FOR DATA DURING INFERENCE

| Method | Param(M) | GFLOPs(G) | FPS | GPU-Utili |
|---|---|---|---|---|
| Zhao et al. [51] | 27.87 | 13.02 | 73 | 64% |
| Zhou et al. [53] | 10.87 | 6.84 | 69 | 67% |
| He et al. [52] | 9.98 | 4.67 | 68 | 65% |
| Guizilini et al. [33] | 128.29 | 89.04 | 28 | 99% |
| Godard et al. [18] | 14.84 | 3.49 | 212 | 72% |
| Johnston et al. [34] | 64.45 | 56.47 | 63 | 80% |
| Lyu et al. [31] | 14.61 | 7.84 | 172 | 66% |
| Masoumian et al. [71] | 73.85 | 13.84 | 142 | 71% |
| Wang et al. [21] | 32.52 | 7.22 | 145 | 70% |
| Yan et al. [69] | 58.34 | 17.66 | 129 | 86% |
| Ours | 29.29 | 5.90 | 36 | 51% |

TABLE VII

CAMERA POSE PREDICTION PERFORMANCE COMPARISON

| Method | RES | Seq. 09 | Seq. 10 |
|---|---|---|---|
| Jia et al. [74] | 128 × 416 | 0.011±0.006 | 0.009±0.007 |
| Godard et al. [18] | 192 × 640 | 0.021±0.009 | 0.014±0.010 |
| Ambrus et al. [76] | 192 × 640 | 0.009±0.004 | **0.008±0.007** |
| Guizilini et al. [33] | 192 × 640 | 0.011±0.006 | 0.009±0.007 |
| He et al. [52] | 192 × 640 | 0.021±0.009 | 0.014±0.010 |
| Zhou et al. [53] | 192 × 640 | 0.020±0.009 | 0.014±0.010 |
| Bian et al. [17] | 256 × 832 | 0.016±0.007 | 0.016±0.015 |
| Ranjan et al. [16] | 256 × 832 | 0.012±0.007 | 0.012±0.008 |
| Wang et al. [21] | 256 × 832 | 0.012±0.007 | 0.012±0.008 |
| Jia et al. [74] | 256 × 832 | 0.009±0.007 | 0.009±0.007 |
| Wang et al. [21] | 384 × 1280 | 0.008±0.005 | **0.008±0.006** |
| **Ours** | 128 × 416 | **0.007±0.004** | **0.008±0.007** |

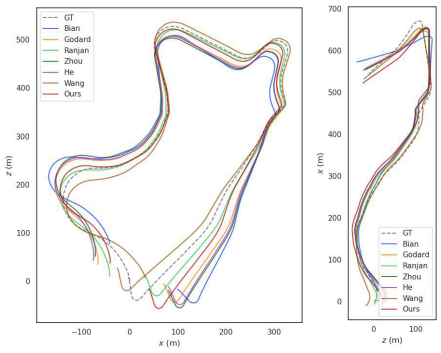



Fig. 5. Qualitative results obtained on the odometry dataset.

in Table X, 'w/o refine' in Table XII (namely, 'plain IE' in Table X) could yield improved performance, but this improvement was limited because the features derived from different stages had different semantics. To this end, we designed different modules for refining the feature map to mitigate the above effect before performing direct fusion. The results shown in Table XII indicate that this method could help with the information fusion process and improve the prediction results if the feature maps derived from different stages were refined before being fused. 'PRB*' shows that increasing the capacity of the refinement module could further improve the depth predictions, which occurred because 'PRB*' has a larger capacity and receptive field, which could transform feature maps derived from different stages into feature maps with more similar semantics than those of 'PRB'. Benefitting from its ability to capture local information and model long-range dependencies in parallel, the proposed 'RM' scheme outperformed 'PRB' and even 'PRB*', which has a larger capacity. Moreover, because the feature maps generated by the local filter and transformer had different semantics, the proposed 'AdaRM', in which the feature maps could be adaptively fused by utilizing the learned parameter tensor, was more beneficial for information fusion than 'RM'.

Table XIII shows the impacts of different downsampling modules on the resulting model performance. The results indicate that the traditional downsampling schemes (e.g., max pooling and strided convolution) achieved the worst performance among the downsampling schemes in Table XIII, which may be due to the loss of information induced by downsampling. Combining max pooling-based downsampling with strided convolution could mitigate these effects. Compared with max pooling and/or strided convolution, the 3D packing [33] scheme could propagate and preserve more details, resulting in the recovery of accurate depths. The proposed 'CAS' method directly modeled the dependencies between the expanded structure information by learning cross-channel interactions via the utilization of lightweight 1D convolutions, resulting in better depth predictions than those of 3D packing. It was found that the depth prediction performance could be further improved by fusing the saliency information obtained via max pooling into the above fine-grained representations, namely, the 'NCAS' and 'AdaNCAS' methods shown in Table XIII. Moreover, compared with the proposed 'NCAS' method, where the saliency information and the fine-grained representations were fused by implementing concatenation in the channel dimension after performing normalization, the proposed 'AdaNCAS' is more conducive to information fusion because this scheme allows the model to decide what information needs to be propagated to the decoder. Compared with the 'AdaNCAS' scheme that directly calculated each piece of channel information by treating each element of the feature map equally, the proposed 'AdaNPCAS', which could focus on the most important elements and give these elements greater weights through an elementwise learnable parameter tensor with an initial value of one, could achieve better performance. The proposed 'AdaAxialNPCAS' approach could achieve performance that was similar to or even better than that of the proposed 'AdaNPCAS' method in terms of some metrics, with 37 times fewer parameters. The proposed 'AdaAxialNPCAS' (1.61 M parameters) outperformed the 3D packing [33] (8.47 M parameters) and 'PRB*' (15.12 M parameters) methods that utilize residual blocks to process the downsampled ('max pooling+stride') feature maps from different stages, which is attributed to the fact that the 'AdaAxialNPCAS' architecture could propagate and preserve more details.

TABLE VIII
QUANTITATIVE COMPARISON AMONG THE CAMERA POSES FOR THE FULL TRAJECTORIES

| Method | RES | Snippet Length | Seq. 09 | | | Seq. 10 | | |
|---|---|---|---|---|---|---|---|---|
| | | | RMSE (m) | Rel. trans. (%) | Rel. rot. (deg/m) | RMSE (m) | Rel. trans. (%) | Rel. rot. (deg/m) |
| Ranjan et al. [16] | 256 × 832 | 5 | 22.82 | 5.96 | 0.017 | 13.95 | **8.06** | 0.032 |
| Godard et al. [18] | 192 × 640 | 3 | 39.83 | 8.45 | 0.018 | **12.60** | 8.46 | 0.033 |
| He et al. [52] | 192 × 640 | 3 | 44.82 | 9.71 | 0.021 | 13.85 | 9.19 | 0.033 |
| Wang et al. [21] | 256 × 832 | 5 | **16.83** | 6.07 | 0.020 | 16.86 | 10.77 | 0.027 |
| Zhou et al. [53] | 192 × 640 | 3 | 46.66 | 9.83 | 0.021 | 14.19 | 9.83 | 0.033 |
| Bian et al. [17] | 256 × 832 | 3 | 56.86 | 12.57 | 0.033 | 20.71 | 10.05 | 0.049 |
| **Ours** | 128 × 416 | 3 | 25.11 | **5.20** | **0.014** | 21.46 | 12.31 | **0.026** |

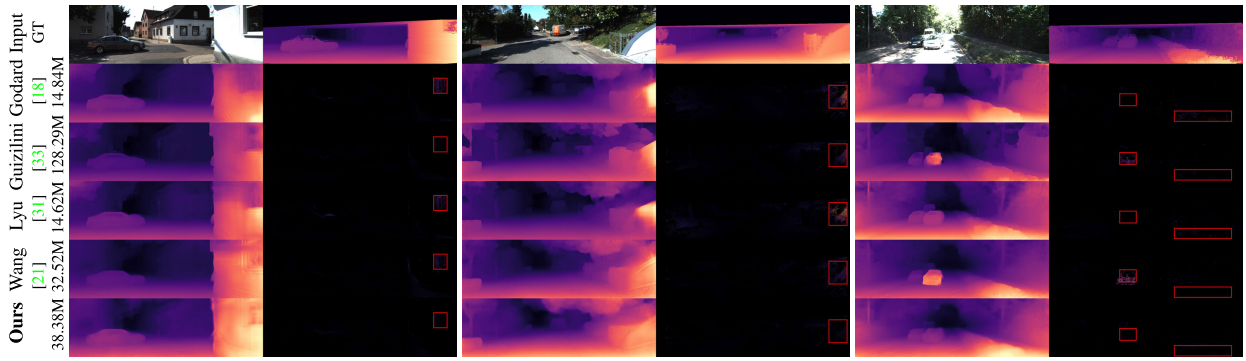



Fig. 6. Qualitative comparison among the example results (384 × 1280 or similar) obtained on the KITTI dataset. The points without valid LIDAR measurements in the error maps were masked out by using the improved ground-truth maps [72].

TABLE IX
ABLATION STUDIES INVOLVING THE UPSAMPLE MODULE CONDUCTED ON KITTI

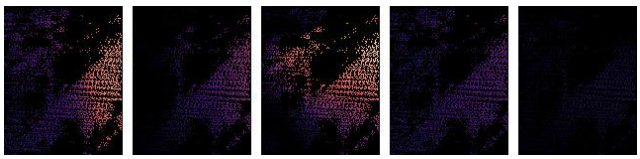
| Scheme | Resolutions | Total Param | Enc Param | Dec Param | Depth Error↓ | | | | Depth Accuracy↑ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | AbsRel | SqRel | RMSE | RMSE log | $\delta_1$ | $\delta_2$ | $\delta_3$ |
| BIU | 128×416 | 20.22M | 19.85M | 0.37M | 0.1164 | 0.8187 | 5.0042 | 0.1915 | 0.8519 | 0.9524 | 0.9822 |
| RCU | 128×416 | 24.09M | 19.85M | 4.24M | 0.1145 | 0.8165 | 4.9564 | 0.1903 | 0.8589 | 0.9533 | 0.9818 |
| BIU* | 128×416 | 28.58M | 19.85M | 8.73M | 0.1129 | 0.7948 | 5.0990 | 0.1932 | 0.8574 | 0.9533 | 0.9814 |
| NRCU | 128×416 | 24.10M | 19.85M | 4.25M | 0.1108 | 0.7531 | 4.8238 | 0.1853 | 0.8649 | 0.9562 | 0.9831 |
| AdaNRSU | 128×416 | 26.38M | 19.85M | 6.53M | 0.1055 | 0.7299 | 4.7259 | 0.1812 | 0.8729 | 0.9584 | 0.9837 |
| DAdaNRSU | 128×416 | 28.45M | 19.85M | 8.60M | 0.1060 | 0.7209 | 4.6367 | 0.1800 | 0.8745 | 0.9592 | 0.9841 |

TABLE X
ABLATION STUDIES INVOLVING THE INFORMATION EXCHANGE MODULE

| Scheme | Resolutions | Total Param | Enc Param | Dec Param | Depth Error↓ | | | | Depth Accuracy↑ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | AbsRel | SqRel | RMSE | RMSE log | $\delta_1$ | $\delta_2$ | $\delta_3$ |
| w/o IE | 128×416 | 20.22M | 19.85M | 0.37M | 0.1164 | 0.8187 | 5.0042 | 0.1915 | 0.8519 | 0.9524 | 0.9822 |
| plain IE | 128×416 | 20.50M | 19.85M | 0.65M | 0.1153 | 0.8110 | 4.9428 | 0.1905 | 0.8526 | 0.9522 | 0.9820 |
| AdaIE | 128×416 | 20.68M | 19.85M | 0.83M | 0.1125 | 0.7885 | 4.8540 | 0.1867 | 0.8633 | 0.9556 | 0.9828 |

Godard [18] Guizilini [33] Lyu [31] Wang [21] **Ours**

Fig. 7. Enlarged error map corresponding to the position of the red rectangular box in the left sample of Fig. 6.



Godard [18] Guizilini [33] Lyu [31] Wang [21] **Ours**

Fig. 8. Enlarged error map corresponding to the position of the red rectangular box in the middle sample of Fig. 6.

Godard [18] Guizilini [33] Lyu [31] Wang [21] **Ours**

Fig. 9. Enlarged error map corresponding to the position of the red rectangular box in the right sample of Fig. 6.

In Table XIV, we investigate the effects of different scale alignment strategies on the obtained absolute depths. The results show that the absolute depths obtained with the median information concerning some indices (e.g., 'AbsRel', 'SqRel', and '$\delta_1$') were better than those obtained with mean information. For the other indices (e.g., 'RMSE', RMSElog', '$\delta_2$, and '$\delta_3$'), the opposite phenomenon was observed. As a simple compromise in which the mean and median information were equally considered, namely, 'fuse', the obtained absolute depths for the above indices except 'AbsRel' were better than

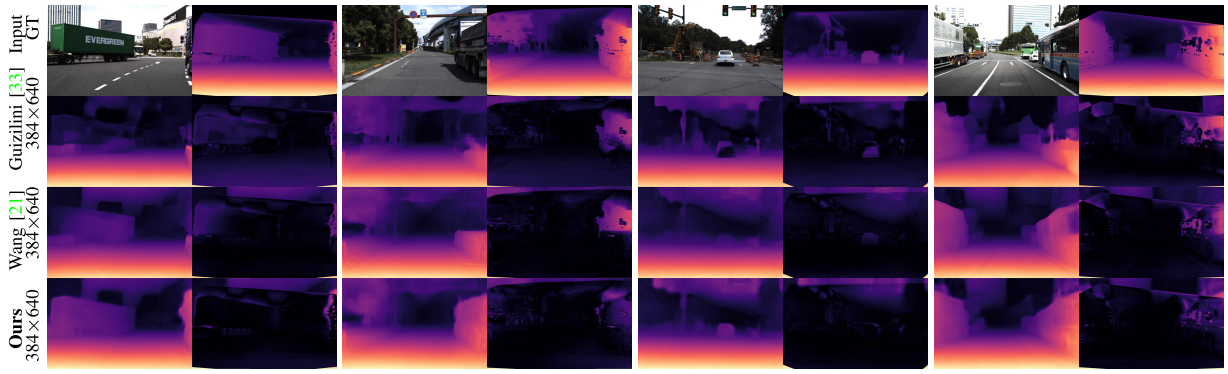



Fig. 10. Qualitative comparison among examples of the results obtained on the DDAD [33] dataset.

TABLE XI
ABLATION STUDIES CONCERNING THE DISPARITY ATTENTION MODULE

| Scheme | Resolutions | Total Param | Enc Param | Dec Param | Depth Error↓ | | | | Depth Accuracy↑ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | AbsRel | SqRel | RMSE | RMSE log | $\delta_1$ | $\delta_2$ | $\delta_3$ |
| Conv2D disp | 128×416 | 20.22M | 19.85M | 0.37M | 0.1164 | 0.8187 | 5.0042 | 0.1915 | 0.8519 | 0.9524 | 0.9822 |
| AttDisp | 128×416 | 20.51M | 19.85M | 0.66M | 0.1104 | 0.7643 | 4.8104 | 0.1844 | 0.8649 | 0.9565 | 0.9833 |

TABLE XII
ABLATION STUDIES INVOLVING THE FEATURE MAP REFINEMENT MODULE

| Scheme | Resolutions | Total Param | Enc Param | Dec Param | Depth Error↓ | | | | Depth Accuracy↑ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | AbsRel | SqRel | RMSE | RMSE log | $\delta_1$ | $\delta_2$ | $\delta_3$ |
| w/o refine | 128×416 | 20.50M | 19.85M | 0.65M | 0.1153 | 0.8110 | 4.9428 | 0.1905 | 0.8526 | 0.9522 | 0.9820 |
| PRB | 128×416 | 21.19M | 19.85M | 1.34M | 0.1136 | 0.8153 | 5.0026 | 0.1909 | 0.8569 | 0.9522 | 0.9812 |
| PRB* | 128×416 | 34.97M | 19.85M | 15.12M | 0.1120 | 0.7755 | 4.8556 | 0.1881 | 0.8628 | 0.9556 | 0.9820 |
| RM | 128×416 | 27.68M | 19.85M | 7.83M | 0.1092 | 0.7714 | 4.8258 | 0.1867 | 0.8675 | 0.9553 | 0.9819 |
| AdaRM | 128×416 | 29.08M | 19.85M | 9.23M | 0.1066 | 0.7247 | 4.7484 | 0.1816 | 0.8693 | 0.9575 | 0.9841 |

TABLE XIII
ABLATION STUDIES CONCERNING DOWNSAMPLING

| Scheme | Resolutions | Total Param | Enc Param | Dec Param | Depth Error↓ | | | | Depth Accuracy↑ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | AbsRel | SqRel | RMSE | RMSElog | $\delta_1$ | $\delta_2$ | $\delta_3$ |
| maxpooling | 128×416 | 20.34M | 19.85M | 0.49M | 0.1158 | 0.8051 | 4.9501 | 0.1909 | 0.8513 | 0.9520 | 0.9818 |
| stride | 128×416 | 20.97M | 19.85M | 1.12M | 0.1157 | 0.8170 | 5.0172 | 0.1909 | 0.8528 | 0.9518 | 0.9820 |
| maxpooling+stride | 128×416 | 20.50M | 19.85M | 0.65M | 0.1153 | 0.8110 | 4.9428 | 0.1905 | 0.8526 | 0.9522 | 0.9820 |
| 3D packing | 128×416 | 28.32M | 19.85M | 8.47M | 0.1129 | 0.8024 | 4.9460 | 0.1893 | 0.8587 | 0.9531 | 0.9817 |
| CAS | 128×416 | 79.10M | 19.85M | 59.25M | 0.1080 | 0.7491 | 4.7819 | 0.1833 | 0.8701 | 0.9575 | 0.9831 |
| NCAS | 128×416 | 35.04M | 19.85M | 15.19M | 0.1062 | 0.7590 | 4.7246 | 0.1832 | 0.8723 | 0.9574 | 0.9829 |
| AdaNCAS | 128×416 | 79.27M | 19.85M | 59.42M | 0.1051 | 0.7481 | 4.7107 | 0.1819 | 0.8758 | 0.9580 | 0.9832 |
| AdaNPCAS | 128×416 | 79.28M | 19.85M | 59.43M | 0.1048 | 0.7332 | 4.6057 | 0.1792 | 0.8775 | 0.9600 | 0.9839 |
| AdaAxialNPCAS | 128×416 | 21.46M | 19.85M | 1.61M | 0.1043 | 0.7269 | 4.6585 | 0.1802 | 0.8760 | 0.9591 | 0.9836 |

TABLE XIV
ABLATION STUDIES INVOLVING DIFFERENT SCALE ALIGNMENT STRATEGIES. THE SAME DECODER CONSISTING OF 'DADANRSU', 'ADAIE', 'ATTDISP', 'ADARM', AND 'ADAAXIALNPCAS' WAS USED

| Scheme | Resolutions | Total Param | Enc Param | Dec Param | Depth Error↓ | | | | Depth Accuracy↑ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | AbsRel | SqRel | RMSE | RMSElog | $\delta_1$ | $\delta_2$ | $\delta_3$ |
| median ($\zeta = 1$) | 128×416 | 29.29M | 19.85M | 9.44M | 0.1026 | 0.7063 | 4.5689 | 0.1763 | 0.8822 | 0.9615 | 0.9845 |
| mean ($\zeta = 0$) | 128×416 | 29.29M | 19.85M | 9.44M | 0.1158 | 0.7079 | 4.3250 | 0.1743 | 0.8813 | 0.9649 | 0.9862 |
| fuse ($\zeta = 0.5$) | 128×416 | 29.29M | 19.85M | 9.44M | 0.1051 | 0.6870 | 4.3986 | 0.1720 | 0.8874 | 0.9644 | 0.9857 |
| AdaSearch | 128×416 | 29.29M | 19.85M | 9.44M | 0.0994 | 0.6926 | 4.4937 | 0.1732 | 0.8866 | 0.9627 | 0.9849 |

those of the techniques that only considered median information. Compared with 'fuse', 'AdaSearch' could adaptively select the appropriate scale factor from the scale search space, resulting in more accurate absolute depths.

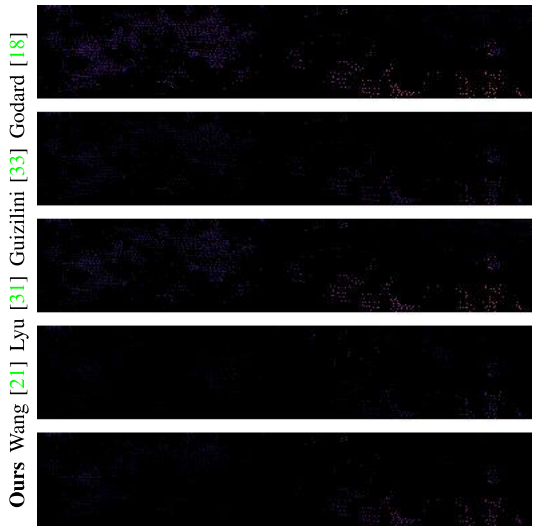



Fig. 11. Enlarged error map corresponding to the position of the red rectangular box in the right sample of Fig. 6.

TABLE XV
PERFORMANCE COMPARISON ON THE DIFFERENT BACKBONE WITH THE PROPOSED DECODER HQDEC

| BackBone | Resolutions | AbsRel | SqRel | RMSE | RMSElog | $\delta_1$ |
|---|---|---|---|---|---|---|
| RN50+PN7 | $128 \times 416$ | 0.1052 | 0.7797 | 4.7137 | 0.1848 | 0.8813 |
| Effv2s+FBv3 | $128 \times 416$ | 0.1026 | 0.7063 | 4.5689 | 0.1763 | 0.8822 |

## V. CONCLUSION

In this paper, we design an HQDec for DepthNet. Our experimental results indicate that the proposed method outperforms previously developed state-of-the-art single-frame depth estimation methods, including those that use semantic labels as guidance signals. Although we only use single-frame information during the inference process, our method equals or even exceeds methods that use multiframe image information during inference. However, we observe that the 'black hole' problem, which arises from the presence of moving objects in a scene that violates the static scene assumption, still exists. Additionally, during training, our method still requires known camera intrinsics, which forbids the use of random internet videos with unknown camera types. Slight grid artifacts, which are derived from the adopted global dependence modeling scheme, may occur in some areas of the predicted depth. The parameters of the model vary slightly depending on the resolution of the input image. We plan to address these problems in future work.

## REFERENCES

[1] Y. D. V. Yasuda, L. E. G. Martins, and F. A. M. Cappabianco, "Autonomous visual navigation for mobile robots: A systematic literature review," *ACM Comput. Surv.*, vol. 53, no. 1, pp. 1–34, Jan. 2021.

[2] X. Liu et al., "Dense depth estimation in monocular endoscopy with self-supervised learning methods," *IEEE Trans. Med. Imag.*, vol. 39, no. 5, pp. 1438–1447, May 2020.

[3] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 2366–2374.

[4] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2002–2011.

[5] S. M. H. Miangoleh, S. Dille, L. Mai, S. Paris, and Y. Aksoy, "Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9680–9689.

[6] L. Huynh, P. Nguyen-Ha, J. Matas, E. Rahtu, and J. Heikkilä, "Guiding monocular depth estimation using depth-attention volume," in *Proc. ECCV*, Aug. 2020, pp. 581–597.

[7] K. K. Parida, S. Srivastava, and G. Sharma, "Beyond image to depth: Improving depth prediction using echoes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8264–8273.

[8] S. F. Bhat, I. Alhashim, and P. Wonka, "Adabins: Depth estimation using adaptive bins," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jan. 2021, pp. 4009–4018.

[9] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12179–12188.

[10] G. Yang, H. Tang, M. Ding, N. Sebe, and E. Ricci, "Transformer-based attention networks for continuous pixel-wise prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16269–16279.

[11] S. Lee, J. Lee, B. Kim, E. Yi, and J. Kim, "Patch-wise attention network for monocular depth estimation," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 1873–1881.

[12] M. Lee, S. Hwang, C. Park, and S. Lee, "EdgeConv with attention module for monocular depth estimation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 2364–2373.

[13] Z. Li et al., "Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6177–6186.

[14] K. Park, S. Kim, and K. Sohn, "High-precision depth estimation using uncalibrated LiDAR and stereo fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 321–335, Jan. 2020.

[15] M. Song, S. Lim, and W. Kim, "Monocular depth estimation using Laplacian pyramid-based depth residuals," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 11, pp. 4381–4393, Nov. 2021.

[16] A. Ranjan et al., "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12240–12249.

[17] J.-W. Bian et al., "Unsupervised scale-consistent depth and ego-motion learning from monocular video," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 35–45.

[18] C. Godard, O. M. Aodha, M. Firman, and G. Brostow, "Digging into self-supervised monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3828–3838.

[19] G. Wang, C. Zhang, H. Wang, J. Wang, Y. Wang, and X. Wang, "Unsupervised learning of depth, optical flow and pose with occlusion from 3D geometry," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 308–320, Jan. 2022.

[20] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," in *Proc. 33rd AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8001–8008.

[21] F. Wang, J. Cheng, and P. Liu, "CbwLoss: Constrained bidirectional weighted loss for self-supervised learning of depth and pose," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 6, pp. 5808–5821, Jun. 2023, doi: 10.1109/TITS.2023.3250744.

[22] M. Klingner, J.-A. Termöhlen, J. Mikolajczyk, and T. Fingscheidt, "Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2020, pp. 582–600.

[23] Y. Zhang, S. Xu, B. Wu, J. Shi, W. Meng, and X. Zhang, "Unsupervised multi-view constrained convolutional network for accurate depth estimation," *IEEE Trans. Image Process.*, vol. 29, pp. 7019–7031, 2020.

[24] Y. Zhang, M. Gong, J. Li, M. Zhang, F. Jiang, and H. Zhao, "Self-supervised monocular depth estimation with multiscale perception," *IEEE Trans. Image Process.*, vol. 31, pp. 3251–3266, 2022.

[25] P. Ruhkamp, D. Gao, H. Chen, N. Navab, and B. Busam, "Attention meets geometry: Geometry guided spatial–temporal attention for consistent self-supervised monocular depth estimation," in *Proc. Int. Conf. 3D Vis. (3DV)*, Dec. 2021, pp. 837–847.

[26] V. Kaushik, K. Jindgar, and B. Lall, "ADAADepth: Adapting data augmentation and attention for self-supervised monocular depth estimation," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7791–7798, Oct. 2021.

[27] S. Chen, Z. Pu, X. Fan, and B. Zou, "Fixing defect of photometric loss for self-supervised monocular depth estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1328–1338, Mar. 2022.

[28] F. Tian et al., "Depth estimation using a self-supervised network based on cross-layer feature fusion and the quadtree constraint," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 1751–1766, Apr. 2022.

[29] V. Guizilini, R. Hou, J. Li, R. Ambrus, and A. Gaidon, "Semantically guided representation learning for self-supervised monocular depth," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–14.

[30] H. Jung, E. Park, and S. Yoo, "Fine-grained semantics-aware representation enhancement for self-supervised monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12622–12632.

[31] X. Lyu et al., "HR-Depth: High resolution self-supervised monocular depth estimation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 3, 2021, pp. 2294–2301.

[32] S. Pillai, R. Ambrus, and A. Gaidon, "SuperDepth: Self-supervised, super-resolved monocular depth estimation," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 9250–9256.

[33] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, "3D packing for self-supervised monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2482–2491.

[34] A. Johnston and G. Carneiro, "Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4756–4765.

[35] X. Song et al., "MLDA-Net: Multi-level dual attention-based network for self-supervised monocular depth estimation," *IEEE Trans. Image Process.*, vol. 30, pp. 4691–4705, 2021.

[36] A. Varma, H. Chawla, B. Zonooz, and E. Arani, "Transformers in self-supervised monocular depth estimation with unknown camera intrinsics," in *Proc. 17th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2022, pp. 1–12.

[37] V. Guizilini, R. Ambruş, D. Chen, S. Zakharov, and A. Gaidon, "Multi-frame self-supervised depth with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 160–170.

[38] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.

[39] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *Proc. Int. Workshop Deep Learn. Med. Image Anal.*, 2018, pp. 3–11.

[40] H. Huang et al., "UNet 3+: A full-scale connected UNet for medical image segmentation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 1055–1059.

[41] J. Watson, M. Firman, G. Brostow, and D. Turmukhambetov, "Self-supervised monocular depth hints," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2162–2171.

[42] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1851–1858.

[43] J. Watson, O. M. Aodha, V. Prisacariu, G. Brostow, and M. Firman, "The temporal opportunist: Self-supervised multi-frame monocular depth," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1164–1174.

[44] A. Masoumian, H. A. Rashwan, J. Cristiano, M. S. Asif, and D. Puig, "Monocular depth estimation using deep learning: A review," *Sensors*, vol. 22, no. 14, p. 5353, Jul. 2022.

[45] P. Vyas, C. Saxena, A. Badapanda, and A. Goswami, "Outdoor monocular depth estimation: A research review," 2022, *arXiv:2205.01399*.

[46] J. L. G. Bello and M. Kim, "PLADE-Net: Towards pixel-level accuracy for self-supervised single-view depth estimation with neural positional encoding and distilled matting loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6847–6856.

[47] J. L. GonzalezBello and M. Kim, "Forget about the LiDAR: Self-supervised depth estimators with med probability volumes," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 12626–12637.

[48] J. S. Martin, C. Russell, S. Hadfield, and R. Bowden, "Deconstructing self-supervised monocular reconstruction: The design decisions that of matter," Dec. 2022, *arXiv:2208.01489*. [Online]. Available: https://openreview.net/forum?id=GFK1FheE7F

[49] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, Oct. 2015.

[50] X. Meng, C. Fan, Y. Ming, and H. Yu, "CORNet: Context-based ordinal regression network for monocular depth estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4841–4853, Jul. 2022.

[51] C. Zhao et al., "MonoViT: Self-supervised monocular depth estimation with a vision transformer," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2022, pp. 668–678.

[52] M. He, L. Hui, Y. Bian, J. Ren, J. Xie, and J. Yang, "RA-depth: Resolution adaptive self-supervised monocular depth estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 565–581.

[53] H. Zhou, D. Greenwood, and S. Taylor, "Self-supervised monocular depth estimation with internal feature fusion," in *Proc. Brit. Mach. Vis. Conf.*, 2021, pp. 1–13.

[54] W. Shi et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1874–1883.

[55] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30. Long Beach, CA, USA, Dec. 2017, pp. 5998–6008.

[56] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–22.

[57] D. Han, J. Shin, N. Kim, S. Hwang, and Y. Choi, "Trans-DSSL: Transformer based depth estimation via self-supervised learning," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10969–10976, Oct. 2022.

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[59] Z. Wang, J. Chen, and S. C. H. Hoi, "Deep learning for image super-resolution: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3365–3387, Oct. 2021.

[60] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[61] A. Paszke et al., "Automatic differentiation in PyTorch," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017.

[62] M. Tan and Q. V. Le, "EfficientNetv2: Smaller models and faster training," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10096–10106.

[63] X. Dai et al., "FBNetV3: Joint architecture-recipe search using predictor pretraining," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 16271–16280.

[64] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

[65] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[66] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–19.

[67] Z. Zhou and Q. Dong, "Self-distilled feature aggregation for self-supervised monocular depth estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 709–726.

[68] H. Li, A. Gordon, H. Zhao, V. Casser, and A. Angelova, "Unsupervised monocular depth learning in dynamic scenes," in *Proc. Conf. Robot. Learn.*, vol. 155, 16–18, Nov. 2021, pp. 1908–1917.

[69] J. Yan, H. Zhao, P. Bu, and Y. Jin, "Channel-wise attention-based network for self-supervised monocular depth estimation," in *Proc. Int. Conf. 3D Vis. (3DV)*, Dec. 2021, pp. 464–473.

[70] A. Petrovai and S. Nedevschi, "Exploiting pseudo labels in a self-supervised learning framework for improved monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1578–1588.

[71] A. Masoumian, H. A. Rashwan, S. Abdulwahab, J. Cristiano, M. S. Asif, and D. Puig, "GCNDepth: Self-supervised monocular depth estimation based on graph convolutional network," *Neurocomputing*, vol. 517, pp. 81–92, Jan. 2023.

[72] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant CNNs," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 11–20.

[73] V. Guizilini, J. Li, R. Ambrus, and A. Gaidon, "Geometric unsupervised domain adaptation for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8517–8527.

[74] S. Jia, X. Pei, X. Jing, and D. Yao, "Self-supervised 3D reconstruction and ego-motion estimation via on-board monocular video," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7557–7569, Jul. 2022.

[75] C. Shu, K. Yu, Z. Duan, and K. Yang, "Feature-metric loss for self-supervised learning of depth and egomotion," in *Proc. Eur. Conf. Comput. Vis.*, Nov. 2020, pp. 572–588.

[76] R. Ambrus, V. Guizilini, J. Li, and S. P. A. Gaidon, "Two stream networks for self-supervised ego-motion estimation," in *Proc. Conf. Robot. Learn.*, 2020, pp. 1052–1061.

**Fei Wang** (Student Member, IEEE) is currently pursuing the Ph.D. degree with the Shenzhen Institute of Advanced Technology, University of Chinese Academy of Sciences. His current research interests include computer vision, structure from motion, robotics, and deep learning.

**Jun Cheng** (Member, IEEE) received the B.Eng. and M.Eng. degrees from the University of Science and Technology of China, Hefei, China, in 1999 and 2002, respectively, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2006. He is currently with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, as a Professor and the Director of the Laboratory for Human Machine Control. His current research interests include computer vision, robotics, machine intelligence, and control.