Self-distilled Feature Aggregation for Self-supervised Monocular Depth Estimation

Zhengming Zhou[1,2] and Qiulei Dong[1,2,3(✉)]

[1] National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing 100190, China
zhouzhengming2020@ia.ac.cn, qldong@nlpr.ia.ac.cn
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences,
Beijing 100049, China
[3] Center for Excellence in Brain Science and Intelligence Technology,
Chinese Academy of Sciences, Beijing 100190, China

Abstract. Self-supervised monocular depth estimation has received much attention recently in computer vision. Most of the existing works in literature aggregate multi-scale features for depth prediction via either straightforward concatenation or element-wise addition, however, such feature aggregation operations generally neglect the contextual consistency between multi-scale features. Addressing this problem, we propose the Self-Distilled Feature Aggregation (SDFA) module for simultaneously aggregating a pair of low-scale and high-scale features and maintaining their contextual consistency. The SDFA employs three branches to learn three feature offset maps respectively: one offset map for refining the input low-scale feature and the other two for refining the input high-scale feature under a designed self-distillation manner. Then, we propose an SDFA-based network for self-supervised monocular depth estimation, and design a self-distilled training strategy to train the proposed network with the SDFA module. Experimental results on the KITTI dataset demonstrate that the proposed method outperforms the comparative state-of-the-art methods in most cases. The code is available at https://github.com/ZM-Zhou/SDFA-Net_pytorch.

Keywords: Monocular depth estimation · Self-supervised learning · Self-distilled feature aggregation

1 Introduction

Monocular depth estimation is a challenging topic in computer vision, which aims to predict pixel-wise scene depths from single images. Recently, self-supervised methods [12,14,16] for monocular depth learning have received much attention, due to the fact that they could be trained without ground truth depth labels.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-19769-7_41.

S. Avidan et al. (Eds.): ECCV 2022, LNCS 13661, pp. 709–726, 2022.
https://doi.org/10.1007/978-3-031-19769-7_41

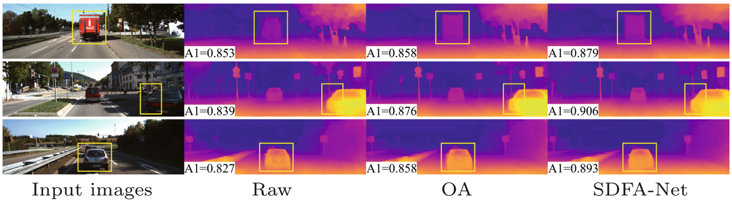

Fig. 1. Depth map comparison by the networks (described in Sect. 4.3) with different feature aggregation techniques (the straightforward concatenation (Raw), the Offset-based Aggregation (OA) [25], and the proposed SDFA module) on KITTI [13], where 'A1' is an accuracy metric.

The existing methods for self-supervised monocular depth estimation could be generally categorized into two groups according to the types of training data: the methods which are trained with monocular video sequences [30,50,57] and the methods which are trained with stereo pairs [12,14,16]. Regardless of the types of training data, many existing methods [15,16,40–42,45,53] employ various encoder-decoder architectures for depth prediction, and the estimation processes could be considered as a general process that sequentially learns multi-scale features and predicts scene depths. In most of these works, their encoders extract multi-scale features from input images, and their decoders gradually aggregate the extracted multi-scale features via either straightforward concatenation or element-wise addition, however, although such feature aggregation operations have demonstrated their effectiveness to some extent in these existing works, they generally neglect the contextual consistency between the multi-scale features, i.e., the corresponding regions of the features from different scales should contain the contextual information for similar scenes. This problem might harm a further performance improvement of these works.

Addressing this problem, we firstly propose the Self-Distilled Feature Aggregation (SDFA) module for simultaneously aggregating a pair of low-scale and high-scale features and maintaining their contextual consistency, inspired by the success of a so-called 'feature-alignment' module that uses an offset-based aggregation technique for handling the image segmentation task [1,24,25,34,35]. It has to be pointed out that such a feature-alignment module could not guarantee its superior effectiveness in the self-supervised monocular depth estimation community. As shown in Fig. 1, when the Offset-based Aggregation (OA) technique is straightforwardly embedded into an encoder-decoder network for handling the monocular depth estimation task, although it performs better than the concatenation-based aggregation (Raw) under the metric 'A1' which is considered as an accuracy metric, it is prone to assign inaccurate depths into pixels on occlusion regions (e.g., the regions around the contour of the vehicles in the yellow boxes in Fig. 1) due to the calculated inaccurate feature alignment by the feature-alignment module for these pixels. In order to solve this problem, the proposed SDFA module employs three branches to learn three feature offset maps respectively: one offset map is used for refining the input low-scale feature, and the other two are jointly used for refining the input high-scale feature under a designed self-distillation manner.

Then, we propose an SDFA-based network for self-supervised monocular depth estimation, called SDFA-Net, which is trained with a set of stereo image pairs. The SDFA-Net employs an encoder-decoder architecture, which uses a modified version of tiny Swin-transformer [37] as its encoder and an ensemble of multiple SDFA modules as its decoder. In addition, a self-distilled training strategy is explored to train the SDFA-Net, which selects reliable depths by two principles from a raw depth prediction, and uses the selected depths to train the network by self-distillation.

In sum, our main contributions include:

- We propose the Self-Distilled Feature Aggregation (SDFA) module, which could effectively aggregate a low-scale feature with a high-scale feature under a self-distillation manner.
- We propose the SDFA-Net with the explored SDFA module for self-supervised monocular depth estimation, where an ensemble of SDFA modules is employed to both aggregate multi-scale features and maintain the contextual consistency among multi-scale features for depth prediction.
- We design the self-distilled training strategy for training the proposed network with the explored SDFA module. The proposed network achieves a better performance on the KITTI dataset [13] than the comparative state-of-the-art methods in most cases as demonstrated in Sect. 4.

2 Related Work

Here, we review the two groups of self-supervised monocular depth estimation methods, which are trained with monocular videos and stereo pairs respectively.

2.1 Self-supervised Monocular Training

The methods trained with monocular video sequences aim to simultaneously estimate the camera poses and predict the scene depths. An end-to-end method was proposed by Zhou et al. [57], which comprised two separate networks for predicting the depths and camera poses. Guizilini et al. [20] proposed PackNet where the up-sampling and down-sampling operations were re-implemented by 3D convolutions. Godard et al. [15] designed the per-pixel minimum reprojection loss, the auto-mask loss, and the full-resolution sampling in Monodepth2. Shu et al. [45] designed the feature-metric loss defined on feature maps for handling less discriminative regions in the images.

Additionally, the frameworks which learnt depths by jointly using monocular videos and extra semantic information were investigated in [4,21,31,33]. Some works [2,28,55] investigated jointly learning the optical flow, depth, and camera pose. Some methods were designed to handle self-supervised monocular depth estimation in the challenging environments, such as the indoor environments [27,56] and the nighttime environments [36,49].

2.2 Self-supervised Stereo Training

The methods trained with stereo image pairs generally estimate scene depths by predicting the disparity between an input pair of stereo images. A pioneering work was proposed by Garg et al. [12], which used the predicted disparities and one image of a stereo pair to synthesize the other image at the training stage. Godard et al. [14] proposed a left-right disparity consistency loss to improve the robustness of monocular depth estimation. Tosi et al. [47] proposed monoResMatch which employed three hourglass-structure networks for extracting features, predicting raw disparities and refining the disparities respectively. FAL-Net [16] was proposed to learn depths under an indirect way, where the disparity was represented by the weighted sum of a set of discrete disparities and the network predicted the probability map of each discrete disparity. Gonzalez and Kim [18] proposed the ambiguity boosting, which improved the accuracy and consistency of depth predictions.

Additionally, for further improving the performance of self-supervised monocular depth estimation, several methods used some extra information (e.g., disparities generated with either traditional algorithms [47,53,59] or extra networks [3,6,22], and semantic segmentation labels [59]). For example, Watson et al. [53] proposed depth hints which calculated with Semi Global Matching [23] and used to guide the network to learn accurate depths. Other methods employed knowledge distillation [19] for self-supervised depth estimation [40,41]. Peng et al. [40] generated an optimal depth map from the multi-scale outputs of a network, and trained the same network with this distilled depth map.

3 Methodology

In this section, we firstly introduce the architecture of the proposed network. Then, the Self-Distilled Feature Aggregation (SDFA) is described in detail. Finally, the designed self-distilled training strategy is given.

3.1 Network Architecture

The proposed SDFA-Net adopts an encoder-decoder architecture with skip connections for self-supervised monocular depth estimation, as shown in Fig. 2.

Encoder. Inspired by the success of vision transformers [5,10,44,51,54] in various visual tasks, we introduce the following modified version of tiny Swin-transformer [37] as the backbone encoder to extract multi-scale features from an input image $I^l \in \mathbb{R}^{3 \times H \times W}$. The original Swin-transformer contains four transformer stages, and it pre-processes the input image through a convolutional layer with stride $= 4$, resulting in 4 intermediate features $[C_1, C_2, C_3, C_4]$ with the resolutions of $[\frac{H}{4} \times \frac{W}{4}, \frac{H}{8} \times \frac{W}{8}, \frac{H}{16} \times \frac{W}{16}, \frac{H}{32} \times \frac{W}{32}]$. Considering rich spatial information is important for depth estimation, we change the convolutional layer with stride $= 4$ in the original Swin-transformer to that with stride $= 2$ in order to keep more high-resolution image information, and accordingly, the resolutions of the

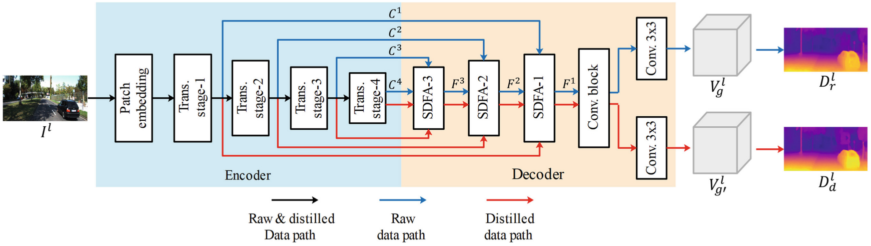

Fig. 2. Architecture of SDFA-Net. SDFA-Net is used to predict the volume-based depth representation V_g^l and the depth map D^l from the input image I^l. The features extracted by the encoder could be passed through the raw and distilled data paths to predict raw and distilled depth maps D_r^l, D_d^l respectively.

features extracted from the modified version of Swin-transformer are twice as they were, i.e. $[\frac{H}{2} \times \frac{W}{2}, \frac{H}{4} \times \frac{W}{4}, \frac{H}{8} \times \frac{W}{8}, \frac{H}{16} \times \frac{W}{16}]$.

Decoder. The decoder uses the multi-scale features $\{C^i\}_{i=1}^4$ extracted from the encoder as its input, and it outputs the disparity-logit volume V_g^l as the scene depth representation. The decoder is comprised of three SDFA modules (denoted as SDFA-1, SDFA-2, SDFA-3 as shown in Fig. 2), a convolutional block and two convolutional output layers. The SDFA module is proposed for adaptively aggregating the multi-scale features with learnable offset maps, which would be described in detail in Sect. 3.2. The convolutional block is used for restoring the spatial resolution of the aggregated feature to the size of the input image, consisting of a nearest up-sampling operation and two 3×3 convolutional layers with the ELU activation [7]. For training the SDFA modules under the self-distillation manner and avoiding training two networks, the encoded features could be passed through the decoder via two data paths, and be aggregated by different offset maps. The two 3×3 convolutional output layers are used to predict two depth representations at the training stage, which are defined as the raw and distilled depth representations respectively. Accordingly, the two data paths are defined as the raw data path and the distilled data path. Once the proposed network is trained, given an arbitrary test image, only the outputted distilled depth representation $V_{g'}^l$ is used as its final depth prediction at the inference stage.

3.2 Self-Distilled Feature Aggregation

The Self-Distilled Feature Aggregation (SDFA) module with learnable feature offset maps is proposed to adaptively aggregate the multi-scale features and maintain the contextual consistency between them. It jointly uses a low-scale decoded feature from the previous layer and its corresponding feature from the encoder as the input, and it outputs the aggregated feature as shown in Fig. 3. The two features are inputted from either raw or distilled data path.

As seen from Fig. 3, under the designed module SDFA-i ($i = 1, 2, 3$), the feature F^{i+1} (specially $F^4 = C^4$) from its previous layer is passed through a

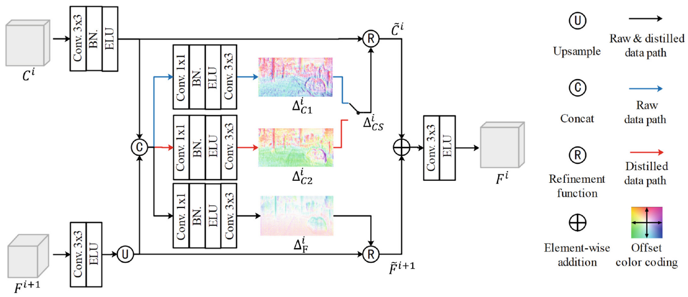

Fig. 3. Self-Distilled Feature Aggregation. Different offset maps are chosen in different data paths.

3 × 3 convolutional layer with the ELU activation [7] and up-sampled by the standard bilinear interpolation. Meanwhile, an additional 3 × 3 convolutional layer with the Batch Normalization (BN) [26] and ELU activation is used to adjust the channel dimensions of the corresponding feature C^i from the encoder to be the same as that of F^{i+1}. Then, the obtained two features via the above operations are concatenated together and passed through three branches for predicting the offset maps Δ^i_F, Δ^i_{C1}, and Δ^i_{C2}. The offset map Δ^i_F is used to refine the up-sampled F^{i+1}. According to the used data path P, a switch operation '$\mathcal{S}(\Delta^i_{C1}, \Delta^i_{C2}|P)$' is used to select an offset map from Δ^i_{C1} and Δ^i_{C2} for refining the adjusted C^i, which is formulated as:

$$\Delta^i_{CS} = \mathcal{S}(\Delta^i_{C1}, \Delta^i_{C2}|P) = \begin{cases} \Delta^i_{C1}, & P = \text{ raw} \\ \Delta^i_{C2}, & P = \text{ distilled} \end{cases} \quad . \tag{1}$$

After obtaining the offset maps, a refinement function '$\mathcal{R}(F, \Delta)$' is designed to refine the feature F by the guidance of the offset map Δ, which is implemented by the bilinear interpolation kernel. Specifically, to generate a refined feature $\tilde{F}(p)$ on the position $p = [x, y]^\top$ from $F \in \mathbb{R}^{C \times H \times W}$ with an offset $\Delta(p)$, the refinement function is formulated as:

$$\tilde{F}(p) = \mathcal{R}(F, \Delta(p)) = \langle F(p + \Delta(p)) \rangle \quad , \tag{2}$$

where '$\langle \cdot \rangle$' denotes the bilinear sampling operation. Accordingly, the refined features $\tilde{F}^{i+1}$ and $\tilde{C}^i$ are generated with:

$$\tilde{F}^{i+1} = \mathcal{R}\left(\mathcal{U}\left(\text{conv}\left(F^{i+1}\right)\right), \Delta^i_F\right) \quad , \tag{3}$$

$$\tilde{C}^i = \mathcal{R}\left(\text{convb}\left(C^i\right), \Delta^i_{CS}\right) \quad , \tag{4}$$

where '$\mathcal{U}(\cdot)$' denotes the bilinear up-sample, 'conv$(\cdot)$' denotes the 3 × 3 convolutional layer with the ELU activation, and 'convb$(\cdot)$' denotes the 3 × 3 convolutional layer with the BN and ELU. Finally, the aggregated feature F^i is obtained with the two refined features as:

$$F^i = \text{conv}\left(\tilde{C}^i \oplus \tilde{F}^{i+1}\right) \quad , \tag{5}$$

where '$\oplus$' denotes the element-wise addition.

Considering that (i) the offsets learnt with self-supervised training are generally suboptimal because of occlusions and image ambiguities and (ii) more effective offsets are expected to be learnt by utilizing extra clues (e.g., some reliable depths from the predicted depth map in Sect. 3.3), SDFA is trained in the following designed self-distillation manner.

3.3 Self-distilled Training Strategy

In order to train the proposed network with a set of SDFA modules, we design a self-distilled training strategy as shown in Fig. 4, which divides each training iteration into three sequential steps: the self-supervised forward propagation, the self-distilled forward propagation and the loss computation.

Self-supervised Forward Propagation. In this step, the network takes the left image I^l in a stereo pair as the input, outputting a left-disparity-logit volume $V_g^l \in \mathbb{R}^{N\times H\times W}$ via the raw data path. As done in [16], I^l and V_g^l are used to synthesize the right image $\hat{I}^r$, while the raw depth map D_r^l is obtained with V_g^l.

Specifically, given the minimum and maximum disparities $d_{\min}$ and $d_{\max}$, we firstly define the discrete disparity level d_n by the exponential quantization [16]:

$$d_n = d_{\max}\left(\frac{d_{\min}}{d_{\max}}\right)^{\frac{n}{N-1}}, \quad n = 0, 1, ..., N-1 \quad , \tag{6}$$

where N is the total number of the disparity levels. For synthesizing the right images, each channel of V_g^l is shifted with the corresponding disparity d_n, generating the right-disparity-logit volume $\hat{V}_g^r$. A right-disparity-probability volume $\hat{V}_p^r$ is obtained by passing $\hat{V}_g^r$ through the softmax operation and is employed for stereoscopic image synthesis as:

$$\hat{I}^r = \sum_{n=0}^{N-1} \hat{V}_{p|n}^r \odot I_n^l \quad , \tag{7}$$

where $\hat{V}_{p|n}^r$ is the n^{th} channel of $\hat{V}_p^r$, '$\odot$' denotes the element-wise multiplication, and I_n^l is the left image shifted with d_n. For obtaining the raw depth map, V_g^l is passed through the softmax operation to generate the left-disparity-probability volume V_p^l. According to the stereoscopic image synthesis, the n^{th} channel of V_p^l approximately equals the probability map of d_n. And a pseudo-disparity map d_+^l is obtained as:

$$d_+^l = \sum_{n=0}^{N-1} V_{p|n}^l \cdot d_n \quad . \tag{8}$$

Given the baseline length B of the stereo pair and the horizontal focal length f_x of the left camera, the raw depth map D_r^l for the left image is calculated via $D_r^l = \frac{Bf_x}{d_+^l}$.

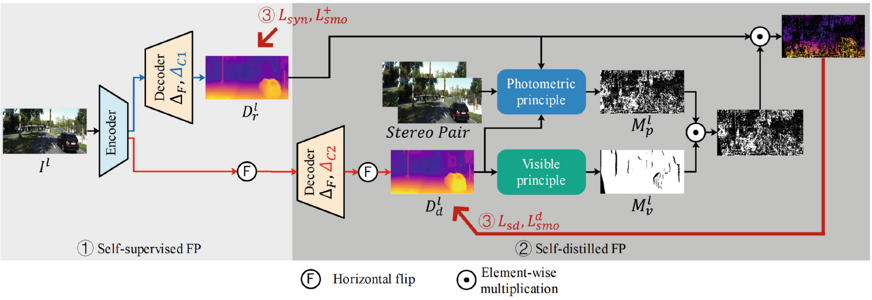

Fig. 4. Self-distilled training strategy. It comprises three steps: the self-supervised Forward Propagation (FP), the self-distilled Forward Propagation (FP), and the loss computation.

Self-distilled Forward Propagation. For alleviating the influence of occlusions and learning more accurate depths, the multi-scale features $\{C^i\}_{i=1}^4$ extracted by the encoder are horizontally flipped and passed through the decoder via the distilled data path, outputting a new left-disparity-logit volume $V^l_{g'}$. After flipping $V^l_{g'}$ back, the distilled disparity map d^l_d and distilled depth map D^l_d are obtained by the flipped $V^l_{g'}$ as done in Eq. (8). For training the SDFA-Net under the self-distillation manner, we employs two principles to select reliable depths from the raw depth map D^l_r: the photometric principle and the visible principle. As shown in Fig. 4, we implement the two principles with two binary masks, respectively.

The photometric principle is used to select the depths from D^l_r in pixels where they make for a superior reprojected left image. Specifically, for an arbitrary pixel coordinate p in the left image, its corresponding coordinate p' in the right image is obtained with a left depth map D^l as:

$$p' = p - \left[\frac{Bf_x}{D^l(p)}, 0\right]^\top \quad . \tag{9}$$

Accordingly, the reprojected left image $\hat{I}^l$ is obtained by assigning the RGB value of the right image pixel p' to the pixel p of $\hat{I}^l$. Then, a weighted sum of the L_1 loss and the structural similarity (SSIM) loss [52] is employed to measure the photometric difference between the reprojected image $\hat{I}^l$ and raw image I^l:

$$l(D^l) = \left(\alpha \left\|\hat{I}^l - I^l\right\|_1 + (1-\alpha)\mathrm{SSIM}(\hat{I}^l, I^l)\right) \quad , \tag{10}$$

where α is a balance parameter. The photometric principle is formulated as:

$$M^l_p = \left[\left(l\left(D^l_r\right) - l\left(D^l_d\right) < \epsilon\right) \cap l\left(D^l_r\right) < t_1\right] \quad , \tag{11}$$

where '[·]' is the iverson bracket, ϵ and t_1 are predefined thresholds. The second term in the iverson bracket is used to ignore the inaccurate depths with large photometric errors.

The visible principle is used to eliminate the potential inaccurate depths in the regions that are visible only in the left image, e.g., the pixels occluded in the right image or out of the edge of the right image. The pixels which are occluded in the right image could be found with the left disparity map d^l. Specifically, it is observed that for an arbitrary pixel location $p = [x, y]^\top$ and its horizontal right neighbor $p_i = [x + i, y]^\top (i = 1, 2, ..., K)$ in the left image, if the corresponding location of p is occluded by that of p_i in the right image, the difference between their disparities should be close or equal to the difference of their horizontal coordinates [59]. Accordingly, the occluded mask calculated with the distilled disparity map d^l_d is formulated as:

$$M^l_{occ}(p) = \left[\min_i \left(\left|d^l_d(p_i) - d^l_d(p) - i\right|\right) < t_2\right] \quad , \tag{12}$$

where t_2 is a predefined threshold. Additionally, an out-of-edge mask M^l_{out} [38] is jointly utilized for filtering out the pixels whose corresponding locations are out of the right image. Accordingly, the visible principle is formulated as:

$$M^l_v = M^l_{occ} \odot M^l_{out} \quad . \tag{13}$$

Loss Computation. In this step, the total loss is calculated for training the SDFA-Net. It is noted that the raw depth map D^l_r is learnt by maximizing the similarity between the real image I^r and the synthesized image $\hat{I}^r$, while the distilled depth map D^l_d is learnt with self-distillation. The total loss function is comprised of four terms: the image synthesis loss L_{syn}, the self-distilled loss L_{sd}, the raw smoothness loss L^+_{smo}, and the distilled smoothness loss L^d_{smo}.

The image synthesis loss L_{syn} contains the L_1 loss and the perceptual loss [29] for reflecting the similarity between the real right image I^r and the synthesized right image $\hat{I}^r$:

$$L_{syn} = \left\|\hat{I}^r - I^r\right\|_1 + \beta \sum_{i=1,2,3} \left\|\phi_i(\hat{I}^r) - \phi_i(I^r)\right\|_2 \quad , \tag{14}$$

where '$\|\cdot\|_1$' and '$\|\cdot\|_2$' represent the L_1 and L_2 norm, $\phi_i(\cdot)$ denotes the output of i^{th} pooling layer of a pretrained VGG19 [46], and β is a balance parameter.

The self-distilled loss L_{sd} adopts the L_1 loss to distill the accurate depths from the raw depth map D^l_r to the distilled depth map D^l_d, where the accurate depths are selected by the photometric and visible masks M^l_p and M^l_v:

$$L_{sd} = M^l_p \odot M^l_v \odot \left\|D^l_d - D^l_r\right\|_1 \quad . \tag{15}$$

The edge-aware smoothness loss L_{smo} is employed for constraining the continuity of the pseudo disparity map d^l_+ and the distilled disparity map d^l_d:

$$L^+_{smo} = \left\|\partial_x d^l_+\right\|_1 e^{-\gamma\left\|\partial_x I^l\right\|_1} + \left\|\partial_y d^l_+\right\|_1 e^{-\gamma\left\|\partial_y I^l\right\|_1} \quad , \tag{16}$$

$$L^d_{smo} = \left\|\partial_x d^l_d\right\|_1 e^{-\gamma\left\|\partial_x I^l\right\|_1} + \left\|\partial_y d^l_d\right\|_1 e^{-\gamma\left\|\partial_y I^l\right\|_1} \quad , \tag{17}$$

where '∂_x', '∂_y' are the differential operators in the horizontal and vertical directions respectively, and γ is a parameter for adjusting the degree of edge preservation.

Accordingly, the total loss is a weighted sum of the above four terms, which is formulated as:

$$L = L_{syn} + \lambda_1 L^{+}_{smo} + \lambda_2 L_{sd} + \lambda_3 L^{d}_{smo} \quad , \tag{18}$$

where $\{\lambda_1, \lambda_2, \lambda_3\}$ are three preseted weight parameters. Considering that the depths learnt under the self-supervised manner are unreliable at the early training stage, λ_2 and λ_3 are set to zeros at these training iterations, while the self-distilled forward propagation is disabled.

4 Experiments

In this section, we evaluate the SDFA-Net as well as 15 state-of-the-art methods and perform ablation studies on the KITTI dataset [13]. The Eigen split [11] of KITTI which comprises 22600 stereo pairs for training and 679 images for testing is used for network training and testing, while the improved Eigen test set [48] is also employed for network testing. Additionally, 22972 stereo pairs from the Cityscapes dataset [8] are jointly used for training as done in [16]. At both the training and inference stages, the images are resized into the resolution of 1280 × 384. We utilize the crop proposed in [12] and the standard depth cap of 80 m in the evaluation. The following error and accuracy metrics are used as done in the existing works [15,16,33,40,45]: Abs Rel, Sq Rel, RMSE, logRMSE, A1 $= \delta < 1.25$, A2 $= \delta < 1.25^2$, and A3 $= \delta < 1.25^3$. Please see the supplemental material for more details about the datasets and metrics.

4.1 Implementation Details

We implement the SDFA-Net with PyTorch [39], and the modified version of tiny Swin-transformer is pretrained on the ImageNet1K dataset [9]. The Adam optimizer [32] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ is used to train the SDFA-Net for 50 epochs with a batch size of 12. The initial learning rate is firstly set to 10^{-4}, and is downgraded by half at epoch 30 and 40. The self-distilled forward propagation and the corresponding losses are used after training 25 epochs. For calculating disparities from the outputted volumes, we set the minimum and the maximum disparities to $d_{\min} = 2, d_{\max} = 300$, and the number of the discrete levels are set to $N = 49$. The weight parameters for the loss function are set to $\lambda_1 = 0.0008, \lambda_2 = 0.01$ and $\lambda_3 = 0.0016$, while we set $\beta = 0.01$ and $\gamma = 2$. For the two principles in the self-distilled forward propagation, we set $\alpha = 0.15, \epsilon = 1e - 5, t_1 = 0.2, t_2 = 0.5$ and $K = 61$. We employ random resizing (from 0.75 to 1.5) and cropping (192 × 640), random horizontal flipping, and random color augmentation as the data augmentations.

Table 1. Quantitative comparison on both the raw and improved KITTI Eigen test sets. ↓ / ↑ denotes that lower/higher is better. The best and the second best results are in **bold** and underlined in each metric.

Method	PP	Sup.	Data.	Resolution	Abs Rel ↓	Sq Rel ↓	RMSE ↓	logRMSE ↓	A1 ↑	A2 ↑	A3 ↑
Raw Eigen test set											
R-MSFM6 [58]		M	K	320 × 1024	0.108	0.748	4.470	0.185	0.889	0.963	0.982
PackNet [20]		M	K	384 × 1280	0.107	0.802	4.538	0.186	0.889	0.962	0.981
SGDepth [33]		M(s)	K	384 × 1280	0.107	0.768	4.468	0.186	0.891	0.963	0.982
FeatureNet [45]		M	K	320 × 1024	0.104	0.729	4.481	0.179	0.893	0.965	0.984
monoResMatch [47]	✓	S(d)	K	384 × 1280	0.111	0.867	4.714	0.199	0.864	0.954	0.979
Monodepth2 [15]	✓	S	K	320 × 1024	0.105	0.822	4.692	0.199	0.876	0.954	0.977
DepthHints [53]	✓	S(d)	K	320 × 1024	0.096	0.710	4.393	0.185	0.890	0.962	0.981
DBoosterNet-e [18]		S	K	384 × 1280	0.095	0.636	4.105	0.178	0.890	0.963	0.984
SingleNet [3]	✓	S	K	320 × 1024	0.094	0.681	4.392	0.185	0.892	0.962	0.981
FAL-Net [16]	✓	S	K	384 × 1280	0.093	0.564	3.973	0.174	0.898	0.967	**0.985**
Edge-of-depth [59]	✓	S(s,d)	K	320 × 1024	0.091	0.646	4.244	0.177	0.898	0.966	0.983
PLADE-Net [17]	✓	S	K	384 × 1280	**0.089**	0.590	4.008	0.172	0.900	0.967	**0.985**
EPCDepth [40]	✓	S(d)	K	320 × 1024	0.091	0.646	4.207	0.176	0.901	0.966	0.983
SDFA-Net (Ours)		S	K	384 × 1280	0.090	0.538	3.896	0.169	0.906	**0.969**	**0.985**
SDFA-Net (Ours)	✓	S	K	384 × 1280	**0.089**	**0.531**	**3.864**	**0.168**	**0.907**	**0.969**	**0.985**
PackNet [20]		M	CS+K	384 × 1280	0.104	0.758	4.386	0.182	0.895	0.964	0.982
SemanticGuide [21]		M(s)	CS+K	384 × 1280	0.100	0.761	4.270	0.175	0.902	0.965	0.982
monoResMatch [47]	✓	S(d)	CS+K	384 × 1280	0.096	0.673	4.351	0.184	0.890	0.961	0.981
DBoosterNet-e [18]	✓	S	CS+K	384 × 1280	0.086	0.538	3.852	0.168	0.905	0.969	**0.985**
FAL-Net [16]	✓	S	CS+K	384 × 1280	0.088	0.547	4.004	0.175	0.898	0.966	0.984
PLADE-Net [17]	✓	S	CS+K	384 × 1280	0.087	0.550	3.837	0.167	0.908	**0.970**	**0.985**
SDFA-Net (Ours)		S	CS+K	384 × 1280	0.085	0.531	3.888	0.167	0.911	0.969	**0.985**
SDFA-Net (Ours)	✓	S	CS+K	384 × 1280	**0.084**	**0.523**	**3.833**	**0.166**	**0.913**	**0.970**	**0.985**
Improved Eigen test set											
DepthHints [53]	✓	S(d)	K	320 × 1024	0.074	0.364	3.202	0.114	0.936	0.989	0.997
WaveletMonoDepth [43]	✓	S(d)	K	320 × 1024	0.074	0.357	3.170	0.114	0.936	0.989	0.997
DBoosterNet-e [18]		S	K	384 × 1280	0.070	0.298	2.916	0.109	0.940	0.991	0.998
FAL-Net [16]	✓	S	K	384 × 1280	0.071	0.281	2.912	0.108	0.943	0.991	0.998
PLADE-Net [17]	✓	S	K	384 × 1280	**0.066**	0.272	2.918	0.104	0.945	0.992	0.998
SDFA-Net (Ours)	✓	S	K	384 × 1280	0.074	**0.228**	**2.547**	**0.101**	**0.956**	**0.995**	**0.999**
PackNet [20]		M	CS+K	384 × 1280	0.071	0.359	3.153	0.109	0.944	0.990	0.997
DBoosterNet-e [18]	✓	S	CS+K	384 × 1280	**0.062**	0.242	2.617	**0.096**	0.955	0.994	0.998
FAL-Net [16]	✓	S	CS+K	384 × 1280	0.068	0.276	2.906	0.106	0.944	0.991	0.998
PLADE-Net [17]	✓	S	CS+K	384 × 1280	0.065	0.253	2.710	0.100	0.950	0.992	0.998
SDFA-Net (Ours)	✓	S	CS+K	384 × 1280	0.069	**0.207**	**2.472**	**0.096**	**0.963**	**0.995**	**0.999**

4.2 Comparative Evaluation

We firstly evaluate the SDFA-Net on the raw KITTI Eigen test set [11] in comparison to 15 existing methods listed in Table 1. The comparative methods are trained with either monocular video sequences (M) [20,21,33,45,58] or stereo image pairs (S) [3,15–18,40,43,47,53,59]. It is noted that some of them are trained with additional information, such as the semantic segmentation label (s) [21,33,59], and the offline computed disparity (d) [40,43,47,53,59]. Additionally, we also evaluate the SDFA-Net on the improved KITTI Eigen test set [48]. The corresponding results are reported in Table 1.

As seen from Lines 1–23 of Table 1, when only the KITTI dataset [13] is used for training (K), SDFA-Net performs best among all the comparative methods

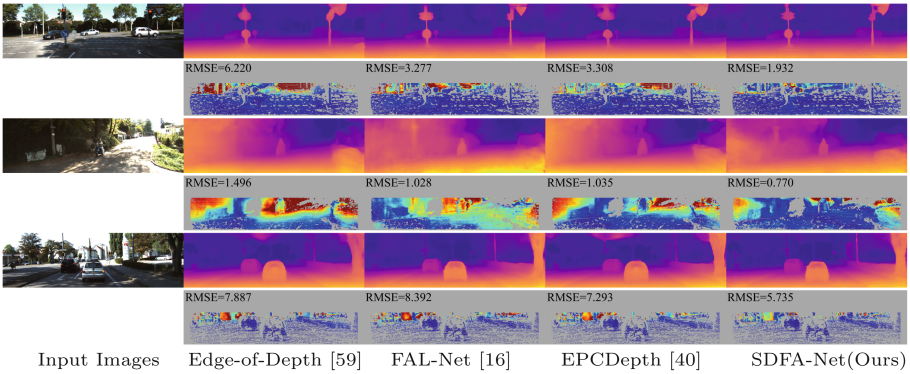

Fig. 5. Visualization results of Edge-of-Depth [59], FAL-Net [16], EPCDepth [40] and our SDFA-Net on KITTI. The input images and predicted depth maps are shown in the odd rows, while we give the corresponding 'RMSE' error maps calculated with the improved Eigen test set in the even rows. For the error maps, red indicates larger error, and blue indicates smaller error. (Color figure online)

in most cases even without the post processing (PP.), and its performance is further improved when a post-processing operation step [14] is implemented on the raw Eigen test set. When both Cityscapes [8] and KITTI [13] are jointly used for training (CS + K), the performance of SDFA-Net is further improved, and it still performs best among all the comparative methods on the raw Eigen test set, especially under the metric 'Sq. Rel'.

As seen from Lines 24–34 of Table 1, when the improved Eigen test set is used for testing, the SDFA-Net still performs superior to the comparative methods in most cases. Although it performs relatively poorer under the metric 'Abs Rel.', its performances under the other metrics are significantly improved. These results demonstrate that the SDFA-Net could predict more accurate depths.

In Fig. 5, we give several visualization results of SDFA-Net as well as three typical methods whose codes have been released publicly: Edge-of-Depth [59], FAL-Net [16], and EPCDepth [40]. It can be seen that SDFA-Net does not only maintain delicate geometric details, but also predict more accurate depths in distant scene regions as shown in the error maps.

4.3 Ablation Studies

To verify the effectiveness of each element in SDFA-Net, we conduct ablation studies on the KITTI dataset [13]. We firstly train a baseline model only with the self-supervised forward propagation and the self-supervised losses (i.e. the image synthesis loss L_{syn} and the raw smoothness loss L^{+}_{smo}). The baseline model comprises an original Swin-transformer backbone (Swin) as the encoder and a raw decoder proposed in [15]. It is noted that in each block of the raw decoder, the multi-scale features are aggregated by the straightforward concatenation. Then we replace the backbone by the modified version of the Swin-transformer

Table 2. (a) Quantitative comparison on the raw Eigen test set in the ablation study. (b) The average norms (in pixels) of the offset vectors on the raw Eigen test set.

Method	Abs Rel↓	Sq Rel↓	RMSE↓	logRMSE↓	A1↑	A2↑	A3↑
Swin+Raw	0.102	0.615	4.323	0.184	0.887	0.963	0.983
Swin† +Raw	0.100	0.595	4.173	0.180	0.893	0.966	0.984
Swin† +SDFA-1	0.097	0.568	3.993	0.175	0.898	0.968	0.985
Swin† +SDFA-2	0.099	0.563	3.982	0.174	0.900	0.968	0.985
Swin† +SDFA-3	0.094	0.553	3.974	0.172	0.896	0.966	0.985
Swin† +OA	0.091	0.554	4.082	0.174	0.899	0.967	0.984
SDFA-Net	**0.090**	**0.538**	**3.896**	**0.169**	**0.906**	**0.969**	**0.985**

(a)

SDFA-i	Δ_F^i	Δ_{C1}^i	Δ_{C2}^i
1	1.39	0.14	0.59
2	5.85	0.53	0.56
3	14.89	0.65	0.65

(b)

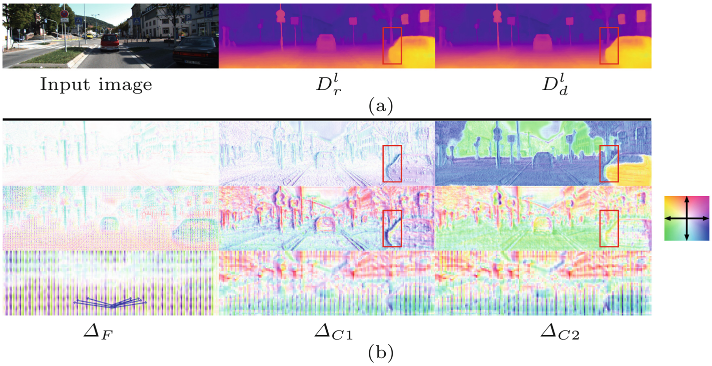

Fig. 6. (a) Visualization of the raw and distilled depth maps predicted with the SDFA-Net on KITTI. (b) Visualization of the corresponding offset maps learnt in SDFA modules. From the top to bottom, the offset maps are learnt in SDFA-1, SDFA-2, and SDFA-3. The color coding is visualized on the right part, where the hue and saturation denote the norm and direction of the offset vector respectively. The blue arrows in the bottom left offset map show the offset vectors in a 3 × 3 patch. (Color figure online)

(Swin†). We also sequentially replace the i^{th} raw decoder block with the SDFA module, and our full model (SDFA-Net) employs three SDFA modules. The proposed self-distilled training strategy is used to train the models that contain the SDFA module(s). Moreover, an Offset-based Aggregation (OA) module which comprises only two offset branches and does not contain the 'Distilled data path' is employed for comparing to the SDFA module. We train the model which uses the OA modules in the decoder with the self-supervised losses only. The corresponding results are reported in Table 2(a).

From Lines 1–5 of Table 2(a), it is noted that using the modified version ('Swin†') of Swin transformer performs better than its original version for depth prediction, indicating that the high-resolution features with rich spatial details are important for predicting depths. By replacing different decoder blocks with the proposed SDFA modules, the performances of the models are improved in most cases. It demonstrates that the SDFA module could aggregate features more effectively for depth prediction. As seen from Lines 6–7 of Table 2(a), although ‘Swin$^{\dagger}+$OA’ achieves better quantitative performance compared to ‘Swin$^{\dagger}+$Raw’, our full model performs best under all the metrics, which indicates that the SDFA module is benefited from the offset maps learnt under the self-distillation.

Additionally, we visualize the depth maps predicted with the models denoted by ‘Swin$^{\dagger}+$Raw’, ‘Swin$^{\dagger}+$OA’, and our SDFA-Net in Fig. 1. Although the offset-based aggregation improves the quantitative results of depth prediction, ‘Swin$^{\dagger}+$OA’ is prone to predict inaccurate depths on occlusion regions. SDFA-Net does not only alleviate the inaccurate predictions in these regions but also further boost the performance of depth estimation. It demonstrates that the SDFA module maintains the contextual consistency between the learnt features for more accurate depth prediction. Please see the supplemental material for more detailed ablation studies.

To further understand the effect of the SDFA module, in Fig. 6, we visualize the predicted depth results of SDFA-Net, while the feature offset maps learnt by each SDFA module are visualized with the color coding. Moreover, the average norms of the offset vectors in these modules are shown in Table 2(b), which are calculated on the raw Eigen test set [13]. As seen from the first column in Table 2(b) and Fig. 6(b), the offsets in $\varDelta_F$ reflect the scene information, and the norms of these offset vectors are relatively long. These offsets are used to refine the up-sampled decoded features with the captured information. Since the low-resolution features contain more global information and fewer scene details, the directions of the offset vectors learnt for these features (e.g., $\varDelta_F^3$ in bottom left of Fig. 6(b)) vary significantly. Specifically, considering that the features in an arbitrary 3×3 patch are processed by a convolutional kernel, the offsets in the patch could provide the adaptive non-local information for the process as shown with blue arrows in Fig. 6(b). As shown in the last two columns in Table 2(b) and Fig. 6(b), the norms of the offsets in $\varDelta_{C1}$ and $\varDelta_{C2}$ are significantly shorter than that in $\varDelta_F$. Considering that the encoded features maintain reliable spatial details, these offsets are used to refine the features in little local areas.

Moreover, it is noted that there are obvious differences between the $\varDelta_{C1}$ and $\varDelta_{C2}$, which are learnt under the self-supervised and self-distillation manners respectively. Meanwhile, the depth map obtained with the features aggregated by $\varDelta_F$ and $\varDelta_{C2}$ are more accurate than that obtained with the features aggregated by $\varDelta_F$ and $\varDelta_{C1}$, especially on occlusion regions (marked by red boxes in Fig. 6(a)). These visualization results indicate that the learnable offset maps are helpful for adaptively aggregating the contextual information, while the self-distillation is helpful for further maintaining the contextual consistency between the learnt feature, resulting in more accurate depth predictions.

5 Conclusion

In this paper, we propose the Self-Distilled Feature Aggregation (SDFA) module to adaptively aggregate the multi-scale features with the learnable offset maps. Based on SDFA, we propose the SDFA-Net for self-supervised monocular depth estimation, which is trained with the proposed self-distilled training strategy. Experimental results demonstrate the effectiveness of the SDFA-Net. In the future, we will investigate how to aggregate the features more effectively for further improving the accuracy of self-supervised monocular depth estimation.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (Grant Nos. U1805264 and 61991423), the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB32050100), the Beijing Municipal Science and Technology Project (Grant No. Z211100011021004).

References

1. Cardace, A., Ramirez, P.Z., Salti, S., Di Stefano, L.: Shallow features guide unsupervised domain adaptation for semantic segmentation at class boundaries. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1160–1170 (2022)
2. Chen, Y., Schmid, C., Sminchisescu, C.: Self-supervised learning with geometric constraints in monocular video: connecting flow, depth, and camera. In: ICCV, pp. 7063–7072 (2019)
3. Chen, Z., et al.: Revealing the reciprocal relations between self-supervised stereo and monocular depth estimation. In: ICCV, pp. 15529–15538 (2021)
4. Cheng, B., Saggu, I.S., Shah, R., Bansal, G., Bharadia, D.: S^3Net: semantic-aware self-supervised depth estimation with monocular videos and synthetic data. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12375, pp. 52–69. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58577-8_4
5. Cheng, Z., Zhang, Y., Tang, C.: Swin-depth: using transformers and multi-scale fusion for monocular-based depth estimation. IEEE Sens. J. **21**(23), 26912–26920 (2021)
6. Choi, H., et al.: Adaptive confidence thresholding for monocular depth estimation. In: ICCV, pp. 12808–12818 (2021)
7. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (ELUs). arXiv preprint arXiv:1511.07289 (2015)
8. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. In: CVPR, pp. 3213–3223 (2016)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR, pp. 248–255 (2009)
10. Dosovitskiy, A., et al.: An image is worth 16×16 words: transformers for image recognition at scale. In: ICLR (2021)
11. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Advances in Neural Information Processing Systems, pp. 2366–2374 (2014)
12. Garg, R., B.G., V.K., Carneiro, G., Reid, I.: Unsupervised CNN for single view depth estimation: geometry to the rescue. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 740–756. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_45
13. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: CVPR, pp. 3354–3361 (2012)

14. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: CVPR, pp. 270–279 (2017)
15. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: ICCV, pp. 3828–3838 (2019)
16. GonzalezBello, J.L., Kim, M.: Forget about the lidar: self-supervised depth estimators with med probability volumes. Adv. Neural. Inf. Process. Syst. **33**, 12626–12637 (2020)
17. GonzalezBello, J.L., Kim, M.: PLADE-Net: towards pixel-level accuracy for self-supervised single-view depth estimation with neural positional encoding and distilled matting loss. In: CVPR, pp. 6851–6860 (2021)
18. GonzalezBello, J.L., Kim, M.: Self-supervised deep monocular depth estimation with ambiguity boosting. IEEE TPAMI (2021)
19. Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: a survey. IJCV **129**(6), 1789–1819 (2021)
20. Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., Gaidon, A.: 3d packing for self-supervised monocular depth estimation. In: CVPR, pp. 2485–2494 (2020)
21. Guizilini, V., Hou, R., Li, J., Ambrus, R., Gaidon, A.: Semantically-guided representation learning for self-supervised monocular depth. In: International Conference on Learning Representations (ICLR) (2020)
22. Guo, X., Li, H., Yi, S., Ren, J., Wang, X.: Learning monocular depth by distilling cross-domain stereo networks. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11215, pp. 506–523. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01252-6_30
23. Hirschmuller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: CVPR, vol. 2, pp. 807–814. IEEE (2005)
24. Huang, S., Lu, Z., Cheng, R., He, C.: FaPN: feature-aligned pyramid network for dense image prediction. In: ICCV, pp. 864–873 (2021)
25. Huang, Z., Wei, Y., Wang, X., Liu, W., Huang, T.S., Shi, H.: AlignSeg: feature-aligned segmentation networks. IEEE TPAMI **44**(1), 550–557 (2021)
26. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456 (2015)
27. Ji, P., Li, R., Bhanu, B., Xu, Y.: MonoIndoor: towards good practice of self-supervised monocular depth estimation for indoor environments. In: ICCV, pp. 12787–12796 (2021)
28. Jiao, Y., Tran, T.D., Shi, G.: EffiScene: efficient per-pixel rigidity inference for unsupervised joint learning of optical flow, depth, camera pose and motion segmentation. In: CVPR, pp. 5538–5547 (2021)
29. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
30. Johnston, A., Carneiro, G.: Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In: CVPR, pp. 4756–4765 (2020)
31. Jung, H., Park, E., Yoo, S.: Fine-grained semantics-aware representation enhancement for self-supervised monocular depth estimation. In: ICCV, pp. 12642–12652 (2021)
32. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

33. Klingner, M., Termöhlen, J.-A., Mikolajczyk, J., Fingscheidt, T.: Self-supervised monocular depth estimation: solving the dynamic object problem by semantic guidance. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12365, pp. 582–600. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58565-5_35
34. Li, X., et al.: Improving semantic segmentation via decoupled body and edge supervision. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12362, pp. 435–452. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58520-4_26
35. Li, X., et al.: Semantic flow for fast and accurate scene parsing. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 775–793. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_45
36. Liu, L., Song, X., Wang, M., Liu, Y., Zhang, L.: Self-supervised monocular depth estimation for all day images using domain separation. In: ICCV, pp. 12737–12746 (2021)
37. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. In: ICCV, pp. 10012–10022 (2021)
38. Mahjourian, R., Wicke, M., Angelova, A.: Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In: CVPR, pp. 5667–5675 (2018)
39. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. Adv. Neural. Inf. Process. Syst. **32**, 8026–8037 (2019)
40. Peng, R., Wang, R., Lai, Y., Tang, L., Cai, Y.: Excavating the potential capacity of self-supervised monocular depth estimation. In: ICCV, pp. 15560–15569 (2021)
41. Pilzer, A., Lathuiliere, S., Sebe, N., Ricci, E.: Refine and distill: exploiting cycle-inconsistency and knowledge distillation for unsupervised monocular depth estimation. In: CVPR, pp. 9768–9777 (2019)
42. Poggi, M., Aleotti, F., Tosi, F., Mattoccia, S.: On the uncertainty of self-supervised monocular depth estimation. In: CVPR, pp. 3227–3237 (2020)
43. Ramamonjisoa, M., Firman, M., Watson, J., Lepetit, V., Turmukhambetov, D.: Single image depth prediction with wavelet decomposition. In: CVPR, pp. 11089–11098 (2021)
44. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: ICCV, pp. 12179–12188 (2021)
45. Shu, C., Yu, K., Duan, Z., Yang, K.: Feature-metric loss for self-supervised learning of depth and egomotion. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12364, pp. 572–588. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58529-7_34
46. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
47. Tosi, F., Aleotti, F., Poggi, M., Mattoccia, S.: Learning monocular depth estimation infusing traditional stereo knowledge. In: CVPR, pp. 9799–9809 (2019)
48. Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A.: Sparsity invariant CNNs. In: 2017 International Conference on 3D Vision (3DV), pp. 11–20 (2017)
49. Wang, K., et al.: Regularizing nighttime weirdness: efficient self-supervised monocular depth estimation in the dark. In: ICCV, pp. 16055–16064 (2021)
50. Wang, L., Wang, Y., Wang, L., Zhan, Y., Wang, Y., Lu, H.: Can scale-consistent monocular depth be learned in a self-supervised scale-invariant manner? In: ICCV, pp. 12727–12736 (2021)

51. Wang, W., et al.: Pyramid vision transformer: a versatile backbone for dense prediction without convolutions. In: ICCV, pp. 568–578 (2021)
52. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE TIP **13**(4), 600–612 (2004)
53. Watson, J., Firman, M., Brostow, G.J., Turmukhambetov, D.: Self-supervised monocular depth hints. In: ICCV (2019)
54. Yang, G., Tang, H., Ding, M., Sebe, N., Ricci, E.: Transformer-based attention networks for continuous pixel-wise prediction. In: ICCV, pp. 16269–16279 (2021)
55. Yin, Z., Shi, J.: GeoNet: unsupervised learning of dense depth, optical flow and camera pose. In: CVPR, pp. 1983–1992 (2018)
56. Zhou, J., Wang, Y., Qin, K., Zeng, W.: Moving indoor: unsupervised video depth learning in challenging environments. In: ICCV, pp. 8618–8627 (2019)
57. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: CVPR, pp. 1851–1858 (2017)
58. Zhou, Z., Fan, X., Shi, P., Xin, Y.: R-MSFM: recurrent multi-scale feature modulation for monocular depth estimating. In: ICCV, pp. 12777–12786 (2021)
59. Zhu, S., Brazil, G., Liu, X.: The edge of depth: explicit constraints between segmentation and depth. In: CVPR, pp. 13116–13125 (2020)