

Two-in-One Depth: Bridging the Gap Between Monocular and Binocular Self-supervised Depth Estimation

Zhengming Zhou and Qiulei Dong 

State Key Laboratory of Multimodal Artificial Intelligence Systems, CASIA
School of Artificial Intelligence, UCAS

zhouzhengming2020@ia.ac.cn qldong@nlpr.ia.ac.cn

Abstract

Monocular and binocular self-supervised depth estimations are two important and related tasks in computer vision, which aim to predict scene depths from single images and stereo image pairs respectively. In literature, the two tasks are usually tackled separately by two different kinds of models, and binocular models generally fail to predict depth from single images, while the prediction accuracy of monocular models is generally inferior to binocular models. In this paper, we propose a Two-in-One self-supervised depth estimation network, called TiO-Depth, which could not only compatibly handle the two tasks, but also improve the prediction accuracy. TiO-Depth employs a Siamese architecture and each sub-network of it could be used as a monocular depth estimation model. For binocular depth estimation, a Monocular Feature Matching module is proposed for incorporating the stereo knowledge between the two images, and the full TiO-Depth is used to predict depths. We also design a multi-stage joint-training strategy for improving the performances of TiO-Depth in both two tasks by combining the relative advantages of them. Experimental results on the KITTI, Cityscapes, and DDAD datasets demonstrate that TiO-Depth outperforms both the monocular and binocular state-of-the-art methods in most cases, and further verify the feasibility of a two-in-one network for monocular and binocular depth estimation. The code is available at <https://github.com/ZM-Zhou/TiO-Depth-pytorch>.

1. Introduction

With the development of deep learning techniques, deep-neural-network-based methods have shown their effectiveness for handling both the monocular and binocular depth estimation tasks, which pursue depths from single images and stereo image pairs respectively [5, 12, 14, 57]. Since

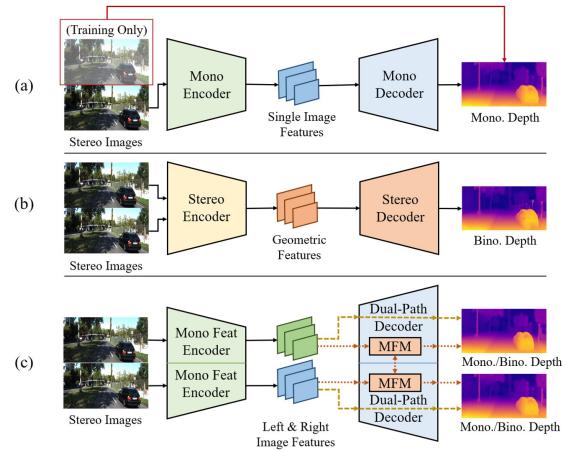


Figure 1. Diagrams of three kinds of self-supervised depth estimation models trained with stereo pairs: (a) Monocular model is tested with a single image but needs stereo pairs during training. (b) Binocular model is trained and tested with stereo pairs, but could not predict depths from a single image; (c) TiO-Depth could be tested with both single images and stereo pairs.

it is time-consuming and labor-intensive to obtain abundant high-quality ground truth scene depths, monocular and binocular self-supervised depth estimation methods, which do not require ground truth depths for training, have attracted increasing attention in recent years [15, 18, 51, 55].

It is noted that the above two tasks are closely related, as shown in Fig. 1: both the monocular and binocular methods output the same type of results (*i.e.*, depth maps), and some self-supervised monocular methods [7, 17, 53] use the same type of training data (*i.e.*, stereo pairs) as the binocular models. Their main difference is that the monocular task is to predict depths from a single image, while the binocular task is to predict depths from a stereo pair. Due to this difference, the two tasks have been handled separately by two different kinds of models (*i.e.*, monocular and binocular models) in literature. Compared with the monocular

 corresponding author

ular models that learn depths from single image features, the binocular models focus on learning depths from the geometric features (*e.g.* cost volumes [55]) generated with stereo pairs, and consequently, they generally perform better than the monocular models but could not predict depth from a single image. Moreover, it is found in [7] that although the whole performances of the monocular models are poorer than the binocular ones, the monocular models still perform better on some special local regions, *e.g.*, the occluded regions around objects which could only be seen at a single view. Inspired by this finding, some monocular (or binocular) models employed a separate binocular (or monocular) model to boost their performances in their own task [1, 7, 9, 13, 36, 38, 45]. All the above issues naturally raise the following problem: **Is it feasible to explore a general model that could not only compatibly handle the two tasks, but also improve the prediction accuracy?**

Obviously, a general model has the following potential advantages in comparison to the separate models: **(1) Flexibility:** This model could compatibly deal with both the monocular and binocular tasks, and it would be of great benefit to the platforms with a binocular system in the real application, where one camera in the binocular system might be occasionally occluded or even broken down. **(2) High Efficiency:** This model has the potential to perform better than both monocular and binocular models, while the number of its parameters is less than that of two separate models.

Addressing the aforementioned problem and potential advantages of a general depth estimation model, in this paper, we propose a Two-in-One model for both monocular and binocular self-supervised depth estimations, called TiO-Depth. TiO-Depth employs a monocular model as a sub-network of a Siamese architecture, so that the whole architecture could take stereo images as input. Considering that the two sub-networks extract image features independently, we design a monocular feature matching module to fuse features from the two sub-networks for binocular prediction. Then, a multi-stage joint-training strategy is proposed for training TiO-Depth in a self-supervised manner and boosting its accuracy in the two tasks by combining their relative advantages and alleviating their disadvantages.

In sum, our main contributions include:

- We propose a novel self-supervised depth estimation model called TiO-Depth, which could handle both the monocular and binocular depth estimation tasks.
- We design a dual-path decoder with the monocular feature matching modules for aggregating the features from either single images or stereo pairs, which may provide new insights into the design of the self-supervised depth estimation network.
- We propose a multi-stage joint-training strategy for

training TiO-Depth, which is helpful for improving the performances of TiO-Depth in the two tasks.

2. Related work

2.1. Self-supervised monocular depth estimation

Self-supervised monocular depth estimation methods take multi-view images as training data and learn to estimate the depth from a single input image with the image reconstruction. The existing methods could be categorized into two groups according to the training data: video training methods and stereo training methods.

The methods trained with video sequences [6, 8, 18, 24, 29, 31, 42, 46, 56, 58, 25] needed to estimate scene depths and camera poses simultaneously. Zhou *et al.* [58] proposed an end-to-end framework which is comprised of two separate networks for predicting depths and camera poses. Godard *et al.* [18] designed a per-pixel minimum reprojection loss with an auto-mask and a full-resolution sampling for training the model to learn more accurate depths. SD-SSMDE [42] utilized a self-distillation framework where a student network was trained by the absolute depth pseudo labels generated with a teacher network. Several methods [8, 24, 29, 31] used extra semantic information for improving the performance, and the frameworks explored in [6, 56] jointly learnt depth, camera pose and optical flow. Additionally, the multi-frame monocular depth estimation was handled in [23, 54], which predicted more accurate depths by taking two frames of a monocular video as input.

The methods trained with stereo image pairs [3, 7, 9, 15, 17, 19, 41, 43, 47, 53, 62, 60, 59] generally predicted scene depths by estimating the disparity between the stereo pair. Godard *et al.* [17] designed a left-right disparity consistency loss to improve its robustness. Zhu *et al.* [62] proposed an edge consistency loss between the depth map and the semantic segmentation map, while a stereo occlusion mask was proposed for alleviating the influence of the occlusion problem during training. An indirect way of learning depths was proposed in [3, 19, 20], where the model outputted a probability volume of a set of discrete disparities for depth prediction. The self-distillation technique [21] was incorporated in [41, 60] to boost the performance of the model by using the reliable results predicted by itself. Considering that the stereo pairs were available at the training stage, Watson *et al.* [53] proposed to utilize the disparities generated with Semi Global Matching [26] as the ‘Depth Hints’ to improve the accuracy. The frameworks that trained a monocular depth estimation network with the pseudo labels selected from the results of a binocular depth estimation network were proposed in [9, 7].

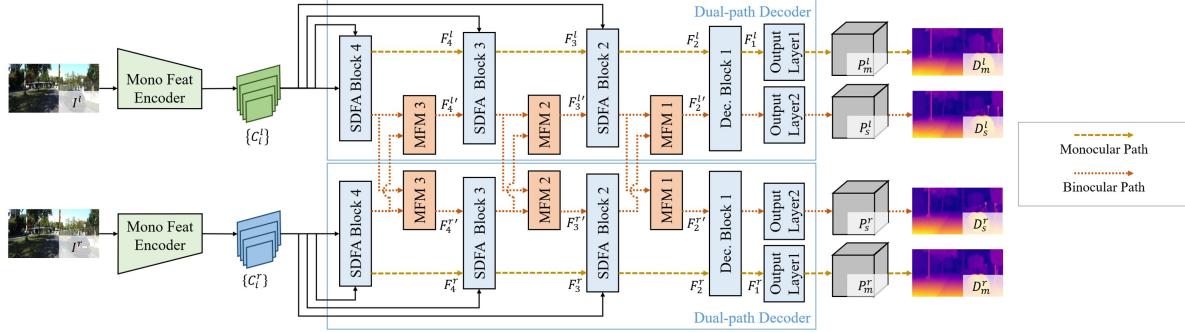


Figure 2. Architecture of TiO-Depth. TiO-Depth employs a Siamese architecture and each sub-network is comprised of a **Monocular Feature Encoder** and a dual-path decoder. The features extracted by the encoder are passed through the decoder via different paths for handling different tasks. $\{P_m, P_s\}$ denote the probability volumes predicted by the monocular and binocular paths respectively, while $\{D_m, D_s\}$ are the corresponding depth maps. The superscripts ‘l’ and ‘r’ denote the left and right views respectively.

2.2. Self-supervised binocular depth estimation

Binocular depth estimation (so called as stereo matching) aims to estimate depths by taking stereo image pairs as input [4, 5, 26, 57]. Recently, self-supervised binocular depth estimation methods [58, 55, 51, 34, 50, 28, 1] were proposed for overcoming the limitation of the ground truth. Zhou *et al.* [58] proposed a framework for learning stereo matching in an iterative manner, which was guided by the left-right check. UnOS [51] and Flow2Stereo [34] were proposed for predicting optical flow and binocular depth simultaneously, where the geometrical consistency between the two types of the predicted results was used to improve the accuracy of them. Wang *et al.* [50] proposed a parallax-attention mechanism to learn the stereo correspondence. H-Net [28] was proposed to learn binocular depths with a Siamese network and an epipolar attention mechanism.

3. Methodology

In this section, we firstly introduce the architecture of the proposed TiO-Depth, including the details of the dual-path decoder and the Monocular Feature Matching (MFM) module. Then, we describe the multi-stage joint-training strategy and the loss functions for training TiO-Depth.

3.1. Overall architecture

Since TiO-Depth is to handle both monocular and binocular depth estimation tasks, it should be able to predict depths from both single image features and geometric features, while the binocular and monocular models could only estimate depths from one type of the features respectively. To this end, TiO-Depth utilizes a Siamese architecture as shown in Fig. 2, and each of the two sub-networks is used as a monocular model. They predict the monocular depth D_m from a single image $I \in \mathbb{R}^{3 \times H \times W}$ for avoiding the model learning depths only based on the geometric fea-

tures, where $\{H, W\}$ denote the height and width of the image. The parameters of the two sub-networks are shared, and they consist of a monocular feature encoder and a decoder. For effectively extracting geometric features from available stereo pairs for the binocular task, the dual-path decoder is proposed as the decoder part of the sub-networks, where a binocular path is added to the path for the monocular task (called monocular path). In the binocular path, the MFM modules are added to learn the geometric features by matching the monocular features extracted by the two sub-networks from a stereo pair and integrate them into the input features. Accordingly, the full TiO-Depth is used to predict binocular depths $\{D_s^l, D_s^r\}$.

Specifically, a modified Swin-transformer [35] is adopted as the encoder as done in [60], which extracts 4 image features $\{C_i\}_{i=1}^4$ with the resolutions of $\{\frac{H}{2^i} \times \frac{W}{2^i}\}_{i=1}^4$. We detail the dual-path decoder and the MFM module as following.

3.2. Dual-path decoder

As shown in Fig. 2, the dual-path decoder is used to gradually aggregate the extracted image features for depth prediction, which consists of three Self-Distilled Feature Aggregation (SDFA) blocks [60], one decoder block [18], three monocular feature matching (MFM) modules, and two 3×3 convolutional layers used as the output layers. The features could be passed through different modules via different paths for the monocular and binocular tasks.

For monocular depth estimation, the multi-scale features $\{C_i\}_{i=1}^4$ are gradually aggregated by the SDFA blocks and the decoder block, which is defined as the monocular path. The SDFA block was proposed in [60] for aggregating the features with two resolutions and maintaining the contextual consistency, which takes a low resolution decoder feature F_{i+1} (Specifically, $F_5 = C_4$) and a high resolution encoder feature C_{i-1} , outputting a new decoder feature with

the same shape as C_{i-1} . The decoder block is comprised of two 3×3 convolutional layers with the ELU activation [10] and an upsample operation for generating a high resolution feature F_i from the output of the last block. The output layer is to generate a discrete disparity volume $V \in \mathbb{R}^{N \times H \times W}$ from the last decoder feature F_1 , where N is the number of the discrete disparity levels.

It is noted that two volumes (defined as the auxiliary volume V_a and the final volume V_m) could be generated for monocular depth estimation by using different offset learning branches in SDFA blocks at the training stage, which would be trained with the photometric loss and the distilled loss at different steps respectively. More details would be described in Sec. 3.4. Accordingly, the branches in SDFA used to generate the two volumes are called auxiliary branch and the final branch. Since V_a is only used at the training stage, it is not illustrated in Fig. 2, and the depth calculated based on V_m is the final monocular result.

For binocular depth estimation, the dual-path decoders in the two sub-networks are utilized for processing left and right image features via the binocular path. In this path, MFM modules take the decoder features $\{F_i^l, F_i^r\}_{i=2}^4$ outputted by the SDFA blocks (where the auxiliary branch is used) for generating the corresponding stereo features $\{F_i^{l'}, F_i^{r'}\}_{i=2}^4$ by incorporating the stereo knowledge. The left and right stereo discrete disparity volumes $\{V_s^l, V_s^r\}$ are obtained by passing the last decoder features $\{F_1^l, F_1^r\}$ to another output layer in each decoder.

For obtaining the depth map from the discrete disparity volume V , as done in [2, 60], a set of discrete disparity levels $\{b_n\}_{n=0}^{N-1}$ is generated with the mirrored exponential disparity discretization by given the maximum and minimum disparities $[b_{\min}, b_{\max}]$. Then, a probability volume P is obtained by normalizing V through a softmax operation along the first (*i.e.* channel) dimension, and a disparity map is calculated by weighted summing of $\{b_n\}_{n=0}^{N-1}$ with the corresponding channels in P :

$$d = \sum_{n=0}^{N-1} P_n \odot b_n , \quad (1)$$

where P_n denotes the n^{th} channel of P and ‘ \odot ’ is the element-wise multiplication. Given the baseline length B of the stereo pair and the horizontal focal length f_x of the camera, the depth map is calculated via $D = \frac{B f_x}{d}$.

3.3. Monocular Feature Matching (MFM) module

Given the features $\{F^l, F^r\} \in \mathbb{R}^{C \times H' \times W'}$ obtained from the two decoders of the two sub-networks, MFM utilizes the cross-attention mechanism [49] for generating the cost volume at the left (or right) view and integrates it into the corresponding feature for outputting a stereo feature that has the same shape of input the feature. $\{C, H', W'\}$ are

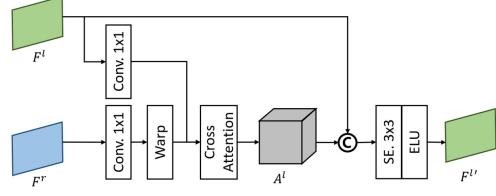


Figure 3. Architecture of Monocular Feature Matching (MFM) module. ‘ \odot ’ denotes the concatenation operation and ‘SE.’ is the SE convolutional layer [27].

the channel, height, and width of the features. Without loss of generality, as shown in Fig. 3, for obtaining the stereo feature at the left-view $F^{l'}$, MFM firstly applies two 1×1 convolutional layers to generate the left-view query feature Q^l and the right-view key feature K^r from $\{F^l, F^r\}$ respectively. As done in [23], the left-view cost volume is generated based on the attention scores between Q^l and a set of shifted K^r , where each score map $S_n^l \in \mathbb{R}^{1 \times H' \times W'}$ is calculated between Q^l and K^r shifted with b'_n , which is formulated as:

$$S_n^l = \frac{\text{sum}(Q^l \odot K_n^r)}{\sqrt{C}} , \quad (2)$$

where K_n^r denotes the K^r shifted with b'_n , and ‘ $\text{sum}(\cdot)$ ’ is a sum operation along the first dimension. Then, the cost volume $A^l \in \mathbb{R}^{N \times H' \times W'}$ is obtained by concatenating S_n^l generated with all the disparity levels $\{b'_n = \frac{W'}{W} b_n\}_{n=0}^{N-1}$ and normalizing it with a softmax operation along the first dimension:

$$A^l = \text{softmax}([S_n^l]_{n=0}^{N-1}) , \quad (3)$$

where ‘ $[\cdot]$ ’ denotes the concatenation operation. For integrating the stereo knowledge in the cost volume into the decoder feature to obtain the stereo feature $F^{l'}$, F^l and A^l are concatenated and passed through a 3×3 SE convolutional layer [27] with the ELU activation:

$$F^{l'} = \text{SE}([A^l, F^l]) . \quad (4)$$

3.4. Multi-stage joint-training strategy

TiO-Depth is trained with stereo image pairs in a self-supervised manner. Considering the motivation of the architecture of TiO-Depth and the different advantages and constraints of the two tasks, we design the multi-stage training strategy as shown in Fig. 4. There are three stages in the strategy, where the training iterations are divided into one, two and three steps respectively. At the last two stages, the training at the current step could be benefited from the results generated at the previous steps. We detail the three steps as following.

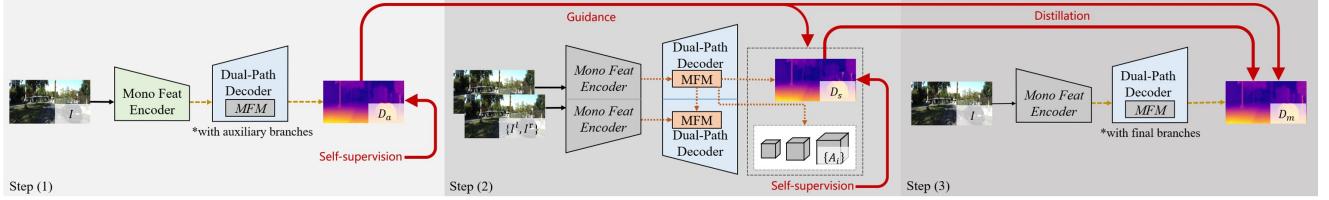


Figure 4. Multi-stage joint-training strategy. There are three steps in each training iteration, where TiO-Depth is trained for different tasks. The training at the current step could be benefited from the results generated at the previous steps. The modules that do not optimized in each step are denoted by grey and the *italic font*.

Step (1). TiO-Depth is trained for learning monocular depth estimation under monocular constraints at this step. The discrete depth constraint [59, 2] is used to generate a left-view reconstructed image \hat{I}_a^l with the right-view auxiliary volume V_a^r (generated with the auxiliary branches in SDFAs as mentioned in Sec. 3.2) and the right-view real image I^r . As done in [2, 60], the monocular loss L_M for training TiO-Depth contains a reconstruction loss L_{rec1} for reflecting the difference between \hat{I}_a^l and I^l , and an edge-aware smoothness loss L_{smo1} :

$$L_M = L_{rec1} + \lambda_1 L_{smo1}, \quad (5)$$

where λ_1 is a preset weight parameters. All the parameters in TiO-Depth except MFMs are optimized at this step.

Step (2). TiO-Depth is trained for learning binocular depth estimation under binocular constraints and some monocular results obtained at step (1). The continuous depth constraint [59, 7] is used to reconstruct a left-view image \tilde{I}_s^l by taking the right-view image I^r and the predicted left-view depth map D_s^l as the input. Then, a stereo loss is adopted to train the network, which consists of the following terms:

The stereo reconstruction loss term L_{rec2} is formulated as a weighted sum of the L_1 loss and the structural similarity (SSIM) loss [52] as done in [7, 18]. Considering the relative advantage of the monocular results on the occluded regions, the occluded pixels in I^l are replaced by the corresponding pixels in a monocular reconstructed image \tilde{I}_a^l calculated with the auxiliary monocular depth map D_a^l :

$$L_{rec2} = \alpha \|\tilde{I}_s^l - I'^l\|_1 + (1 - \alpha) \text{SSIM}(\tilde{I}_s^l, I'^l), \quad (6)$$

$$I'^l = M_{occ}^l \odot I^l + (1 - M_{occ}^l) \odot \tilde{I}_a^l, \quad (7)$$

where α is a balance parameter and ' $\|\cdot\|_1$ ' denotes the L_1 norm. M_{occ}^l is an occlusion mask generated with the auxiliary monocular disparity d_a^l as done in [62], where the values are zeros in the occluded regions, and ones otherwise.

The cost volume loss term L_{cos} is adopted to guide the cost volumes $\{A_i^l\}_{i=1}^3$ generated in MFMs through the aux-

iliary monocular probability volume P_a^l , which is formulated as:

$$L_{cos} = \sum_{i=1}^3 \frac{1}{\Omega_i} \sum_{\|A_i^l(x) - P_a^l(x)\|_1 > t_1} \|A_i^l(x) - P_a^l(x)\|_1, \quad (8)$$

where Ω_i denotes the number of the valid coordinates x in A_i , and t_1 is a predefined threshold. ' $\langle \cdot \rangle$ ' denotes the bilinear sampling operation for getting the element at the corresponding coordinate of x in a different resolution volume.

The disparity guidance loss term L_{gui} leverages both the gradient information and the edge region values in the auxiliary monocular disparity map d_a^l for improving the quality of the binocular result:

$$L_{gui} = \|\partial_x d_a^l - \partial_x d_s^l\|_1 + \|\partial_y d_a^l - \partial_y d_s^l\|_1 + M_{out}^l \odot \|d_a^l - d_s^l\|_1, \quad (9)$$

where ' ∂_x ', ' ∂_y ' are the differential operators in the horizontal and vertical directions respectively, M_{out}^l denotes a binary mask [37] where the pixels whose reprojected coordinates are out of the image are ones, and zeros otherwise. Accordingly, the stereo loss is formulated as:

$$L_S = L_{rec2} + \lambda_2 L_{smo2} + \lambda_3 L_{cos} + \lambda_4 L_{gui}, \quad (10)$$

where $\{\lambda_2, \lambda_3, \lambda_4\}$ are preset weight parameters, and L_{smo2} is the edge-aware smoothness loss [18]. At this step, only the parameters in the dual-path decoder are optimized.

Step (3). TiO-Depth is trained in a distilled manner by utilizing the results obtained at step (1)&(2) as the teacher for further improving monocular prediction. A distilled loss L_{dis} is used to constrain the final monocular probability volume P_m^l (generated with the final branches in SDFAs) with the stereo probability volume P_s^l and the auxiliary monocular probability volume P_a^l . Considering the relative advantages of the monocular and stereo results, a hybrid probability volume P_h^l is generated by fusing them weighted by a half-object-edge map M_{hoe}^l :

$$P_h^l = (1 - M_{hoe}^l) \odot P_s^l + M_{hoe}^l \odot P_a^l. \quad (11)$$

| Method | PP. | Sup. | Resolution | Abs. Rel. ↓ | Sq. Rel. ↓ | RMSE ↓ | logRMSE ↓ | A1 ↑ | A2 ↑ | A3 ↑ |
|--------------------------|-----|-------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| R-MSFM6 [61] | | M | 320×1024 | 0.108 | 0.748 | 4.470 | 0.185 | 0.889 | 0.963 | 0.982 |
| PackNet [22] | | M | 384×1280 | 0.107 | 0.802 | 4.538 | 0.186 | 0.889 | 0.962 | 0.981 |
| SGDepth [31] | | M(Se.) | 384×1280 | 0.107 | 0.768 | 4.468 | 0.186 | 0.891 | 0.963 | 0.982 |
| SD-SSMDE [42] | | M | 320×1024 | 0.098 | 0.674 | 4.187 | 0.170 | 0.902 | 0.968 | 0.985 |
| monoResMatch [47] | ✓ | S(SGM) | 384×1280 | 0.111 | 0.867 | 4.714 | 0.199 | 0.864 | 0.954 | 0.979 |
| Monodepth2 [18] | ✓ | S | 320×1024 | 0.105 | 0.822 | 4.692 | 0.199 | 0.876 | 0.954 | 0.977 |
| DepthHints [53] | ✓ | S(SGM) | 320×1024 | 0.096 | 0.710 | 4.393 | 0.185 | 0.890 | 0.962 | 0.981 |
| SingleNet [7] | ✓ | S(S.T.) | 320×1024 | 0.094 | 0.681 | 4.392 | 0.185 | 0.892 | 0.962 | 0.981 |
| FAL-Net [19] | ✓ | S | 384×1280 | 0.093 | 0.564 | 3.973 | 0.174 | 0.898 | 0.967 | 0.985 |
| Edge-of-depth [62] | ✓ | S(SGM, Se.) | 320×1024 | 0.091 | 0.646 | 4.244 | 0.177 | 0.898 | 0.966 | 0.983 |
| PLADE-Net [20] | ✓ | S | 384×1280 | 0.089 | 0.590 | 4.008 | 0.172 | 0.900 | 0.967 | 0.985 |
| EPCDepth [41] | ✓ | S(SGM) | 320×1024 | 0.091 | 0.646 | 4.207 | 0.176 | 0.901 | 0.966 | 0.983 |
| OCFD-Net [59] | ✓ | S | 384×1280 | 0.090 | 0.563 | 4.005 | 0.172 | 0.903 | 0.967 | <u>0.984</u> |
| SDFA-Net [60] | ✓ | S | 384×1280 | 0.089 | <u>0.531</u> | 3.864 | <u>0.168</u> | 0.907 | <u>0.969</u> | 0.985 |
| <i>TiO-Depth</i> | | S | 384×1280 | <u>0.085</u> | 0.544 | 3.919 | 0.169 | 0.911 | <u>0.969</u> | 0.985 |
| <i>TiO-Depth</i> | ✓ | S | 384×1280 | 0.083 | 0.521 | 3.864 | 0.167 | 0.912 | 0.970 | 0.985 |
| DepthFormer (2F.) [23] | M | 192×640 | 0.090 | 0.661 | 4.149 | 0.175 | 0.905 | 0.967 | 0.984 | |
| ManyDepth (2F.) [54] | M | 320×1024 | 0.087 | 0.685 | 4.142 | 0.167 | 0.920 | 0.968 | 0.983 | |
| H-Net (Bino.) [28] | S | 192×640 | 0.076 | 0.607 | 4.025 | 0.166 | 0.918 | 0.966 | 0.982 | |
| <i>TiO-Depth (Bino.)</i> | S | 384×1280 | 0.063 | 0.523 | 3.611 | 0.153 | 0.943 | 0.972 | 0.985 | |

Table 1. Quantitative comparison on the KITTI Eigen test set. ↓ / ↑ denotes that lower / higher is better. The best and the second best results are in **bold** and underlined under each metric. The methods marked with ‘2F.’ predict depths by taking 2 frames from a monocular video as input, while the methods with ‘Bino.’ predict depths by taking stereo pairs as input. ‘PP’ means using the post-processing step. The methods marked with ‘Se., ‘SGM’, and ‘S.T.’ are trained with the semantic segmentation label, the depth generated with SGM [26], and the depth predicted by a binocular teacher network respectively.

M_{hoe}^l is a grayscale map for indicating the flat areas and the areas on one side of the object, where the binocular results are more accurate experimentally:

$$M_{hoe}^l = M_{occ'}^l \odot \min\left(\frac{\text{maxpool}(\|k * D_s^l\|_1)}{t_2}, 1\right) \quad , \quad (12)$$

where ‘maxpool(·)’ denotes a 3×3 max pooling layer with stride 1, ‘ $*$ ’ denotes the convolutional operation, k is a 3×3 Laplacian kernel, and t_2 is a predefined threshold. $M_{occ'}^l$ is an opposite occlusion mask obtained by treating the left-view disparity map as the right-view one during calculating the occlusion mask. KL divergence is employed to reflect the similarity between the final monocular probability volume P_m^l and P_h^l , which is formulated as:

$$L_{dis} = \text{KL}(P_h^l || P_m^l) \quad . \quad (13)$$

Only the parameters in the SDFA blocks, the decoder block and the output layer are optimized at this step. Please see the supplemental material for more details about the training strategy and losses.

4. Experiments

In this section, we train TiO-Depth on the KITTI dataset [16], and the evaluations are conducted on the KITTI, Cityscapes [11], and DDAD [22] datasets. For monocular depth estimation, the Eigen split [12] of KITTI is utilized, which consists of a training set with 22600 stereo pairs and a test set with 697 images. For binocular depth

estimation, a training set with 28968 stereo pairs collected from KITTI is used for training as done in [7, 33, 51], while the training set of the KITTI 2015 stereo benchmark [39] is used for the evaluation, which consists of 200 image pairs. For exploring the generation ability of TiO-Depth, Cityscapes and DDAD are used for conducting an additional evaluation. Please see the supplemental material for more details about the datasets and metrics.

4.1. Implementation details

TiO-Depth is implemented with the PyTorch [40] framework. The tiny size modified Swin-transformer [35, 60] used as the monocular feature encoder is pretrained on the ImageNet dataset [44]. We set the minimum and the maximum disparities to $b_{\min} = 2$, $b_{\max} = 300$ for the discrete disparity volume, and the number of the discrete disparity levels is set to $N = 49$. The weight parameters for the loss function are set to $\lambda_1 = 0.0008$, $\lambda_2 = 0.008$, $\lambda_3 = 0.01$, and $\lambda_4 = 0.01$, while we set $\alpha = 0.15$, $t_1 = 1$, and $t_2 = 0.13$. The Adam optimizers [30] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ are used to train TiO-Depth for 50 epochs. The learning rate is firstly set to 10^{-4} , and is downgraded by half at the 20, 30, 40, 45 epochs. At both the training and testing stages, the images are resized into the resolution of 384×1280 , while we assume that the intrinsics of all the images are identical. The on-the-fly data augmentations are performed in training, including random resizing (from 0.67 to 1.5) and cropping (256×832), random horizontal flipping, and random color augmentation.

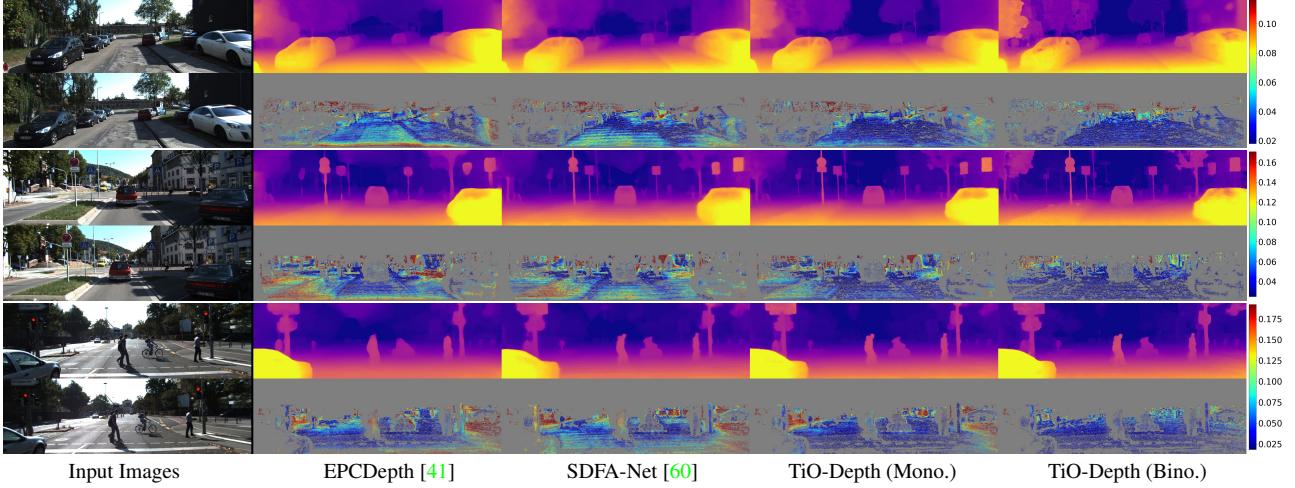


Figure 5. Visualization results of EPCDepth [41], SDFA-Net [60] and our TiO-Depth on KITTI. The input stereo pairs are shown in the first column, where the left-view images are used for monocular depth estimation. The predicted depth maps with the corresponding ‘Abs. Rel.’ error maps calculated on an improved Eigen test set [48] are shown in the following columns. For the error maps, red indicates larger error, and blue indicates smaller error as shown in the color bars.

| Method | Sup. | Resolution | Abs. Rel. ↓ | Sq. Rel. ↓ | RMSE ↓ | logRMSE ↓ | A1 ↑ | A2 ↑ | A3 ↑ | EPE-all ↓ | D1-all ↓ |
|-------------------------|---------|------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|---------------|
| MonoDepth [17] | S | 256×512 | 0.068 | 0.835 | 4.392 | 0.146 | 0.942 | 0.978 | 0.989 | - | 9.194 |
| UnOS (Stereo-only) [51] | S | 256×832 | 0.060 | 0.833 | 4.187 | 0.135 | 0.955 | 0.981 | 0.990 | - | 7.073 |
| UnOS (Full) [51] | MS | 256×832 | <u>0.049</u> | 0.515 | 3.404 | 0.121 | 0.965 | 0.984 | <u>0.992</u> | - | 5.943 |
| Liu <i>et al.</i> [33] | S | 256×832 | 0.051 | 0.532 | 3.780 | 0.126 | 0.957 | 0.982 | 0.991 | 1.520 | 9.570 |
| Flow2Stereo [34] | MS | 384×1280 | - | - | - | - | - | - | - | 1.340 | 6.130 |
| StereoNet [7] | S | 320×1024 | 0.052 | 0.558 | 3.733 | 0.123 | 0.961 | 0.984 | 0.992 | - | - |
| StereoNet-D [7] | S* | 320×1024 | 0.048 | <u>0.482</u> | <u>3.393</u> | <u>0.105</u> | 0.969 | 0.989 | 0.994 | - | - |
| TiO-Depth | S | 384×1280 | 0.050 | 0.434 | 3.239 | 0.104 | 0.967 | 0.987 | 0.994 | 1.282 | 6.647 |
| SingleNet (Mono.) [7] | S(S.T.) | 320×1024 | 0.083 | 0.688 | 4.464 | 0.154 | 0.904 | 0.972 | 0.990 | - | - |
| TiO-Depth (Mono.) | S | 384×1280 | 0.075 | 0.458 | 3.717 | 0.130 | 0.925 | 0.979 | 0.992 | 2.203 | 17.860 |
| TiO-Depth (Mono.)+PP. | S | 384×1280 | 0.073 | 0.439 | 3.680 | 0.128 | 0.925 | 0.980 | 0.993 | 2.158 | 17.570 |

Table 2. Quantitative comparison on KITTI 2015 training set. The methods marked with ‘Mono.’ predict depths by taking single image as input, while other methods predict depths with stereo pairs. ‘S*’ denotes the method is jointly trained with a separate monocular model.

4.2. Comparative evaluation

For monocular depth estimation, we firstly evaluate TiO-Depth on the KITTI Eigen test set [12] in comparison to 4 methods trained with monocular video sequences (M) and 10 methods trained with stereo image pairs (S). The corresponding results by all the referred methods are cited from their original papers and reported in Tab. 1.

It can be seen that TiO-Depth with a post-processing as done in [60] outperforms all the comparative methods in most cases, including the methods trained with the depth pseudo labels generated by additional algorithms or networks (SGM, S.T.). Since *the same* TiO-Depth model could handle the binocular task by using the binocular path, we give its performance in binocular depth estimation (‘Bino.’) in comparison with 3 methods. As seen from Tab. 1, TiO-Depth gets the top performance among all the comparative multi-frame (2F.) and binocular methods. Several visualization results of TiO-Depth as well as two comparative meth-

ods: EPCDepth [41] and SDFA-Net [60] are given in Fig. 5. As shown in the figure, the depth maps predicted by TiO-Depth are more accurate and contain more delicate geometric details, while the performance of TiO-Depth is further improved by taking the stereo pairs as input. These results demonstrate that the TiO-Depth could predict accurate depths by taking both monocular and binocular inputs.

For binocular depth estimation, we evaluate TiO-Depth on the KITTI 2015 training set [39] in comparison to 5 self-supervised binocular depth estimation methods. It is noted that all of the comparative methods could not handle the monocular task. As seen from the corresponding results shown in Tab. 2, TiO-Depth outperforms all the methods trained with stereo pairs (S) or stereo videos (MS) in most cases, and it achieves comparable performance with StereoNet-D [7] benefited from an additional monocular depth estimation model, while the performance of TiO-Depth is boosted by itself. The monocular depth estimation results of *the same* TiO-Depth model are also given

| Method | train | test | Abs. Rel. ↓ | Sq. Rel. ↓ | RMSE ↓ | A1 ↑ |
|--------------------------|-------|------|--------------|--------------|---------------|--------------|
| PackNet [22] | D | D | 0.173 | 7.164 | 14.363 | 0.835 |
| ManyDepth (2F.) [54] | D | D | 0.146 | 3.258 | 14.098 | 0.822 |
| DepthFormer (2F.) [23] | D | D | 0.135 | 2.953 | 12.477 | 0.836 |
| <i>TiO-Depth</i> | K | D | 0.144 | 2.664 | 14.273 | 0.808 |
| MonoDepth2 [18] | C | C | 0.129 | 1.569 | 6.876 | 0.849 |
| Li <i>et al.</i> [32] | C | C | 0.119 | 1.290 | 6.980 | 0.846 |
| ManyDepth (2F.) [54] | C | C | 0.114 | 1.193 | 6.223 | 0.875 |
| SD-SSMDE [42] | C | C | 0.114 | 1.017 | 5.949 | 0.870 |
| MonoDepth2 [18] | K | C | 0.153 | 1.785 | 8.590 | 0.774 |
| SD-SSMDE [42] | K | C | 0.143 | 1.635 | 8.441 | 0.789 |
| <i>TiO-Depth</i> | K | C | 0.120 | 1.176 | 7.157 | 0.850 |
| <i>TiO-Depth (Bino.)</i> | K | C | 0.066 | 0.423 | 4.070 | 0.961 |

Table 3. Quantitative comparison on DDAD and Cityscapes. ‘C’, ‘K’, and ‘D’ denote the methods are trained or tested on the Cityscapes, KITTI and DDAD datasets respectively.

| Methods | Abs. Rel. ↓ | Sq. Rel. ↓ | A1 ↑ | EPE ↓ | D1 ↓ |
|----------------------------------|--------------|--------------|--------------|--------------|--------------|
| w. Cat module (321) | 0.069 | 0.505 | 0.947 | 2.074 | 15.952 |
| w. Attn module (321) | 0.053 | 0.439 | 0.965 | 1.377 | 7.421 |
| w. MFM (1) | 0.054 | 0.423 | 0.960 | 1.483 | 8.784 |
| w. MFM (21) | 0.052 | 0.445 | 0.965 | 1.305 | 7.077 |
| <i>TiO-Depth</i> | 0.051 | 0.429 | 0.966 | 1.281 | 6.684 |
| w/o. L_{gui} | 0.053 | 0.506 | 0.966 | 1.292 | 6.984 |
| w/o. L_{gui}, L_{cos} | 0.053 | 0.522 | 0.965 | 1.326 | 6.755 |
| w/o. $L_{gui}, L_{cos}, M_{occ}$ | 0.054 | 0.565 | 0.963 | 1.345 | 7.159 |

Table 4. Binocular depth estimation results on KITTI 2015 training set in the ablation study. The numbers in the name of methods mean the indexes of the used modules as shown in Fig. 2. All the results are evaluated after training 30 epochs.

in Tab. 2, which show that it effectively handling the monocular task at the same time, further indicating the effectiveness of TiO-Depth as a two-in-one model.

Furthermore, we train TiO-Depth on KITTI [16] and evaluate it on DDAD [22] and Cityscapes [11] for testing its cross-dataset generalization ability. The corresponding results of TiO-Depth and 6 comparative methods are reported in Tab. 3. As shown in the table, TiO-Depth not only performs best in comparison to the methods evaluated in a cross-dataset manner, but also achieves a competitive performance with the methods trained and tested on the same dataset. When the stereo pairs are available, TiO-Depth could predict more accurate binocular depths by taking the image pairs. These results demonstrate the generalization ability of TiO-Depth on the unseen dataset. Please see the supplemental material for the additional exponential results.

4.3. Ablation studies

This subsection verifies the effectiveness of each key element in TiO-Depth by conducting ablation studies on the KITTI dataset [16].

Dual-path decoder. We firstly replace the proposed Monocular Feature Matching (MFM) modules with the concatenation-based modules (Cat module) and the cross-attention-based modules without the SE layer (Attn mod-

| Steps | L_{dis} | FB. | Abs. Rel. ↓ | Sq. Rel. ↓ | RMSE ↓ | A1 ↑ |
|-------|-----------|-----|--------------|--------------|--------------|--------------|
| 1 | - | - | 0.088 | 0.556 | 4.093 | 0.904 |
| 1+2 | - | - | 0.088 | 0.557 | 4.067 | 0.906 |
| 1+2+3 | P_s^l | ✓ | 0.086 | 0.590 | 4.021 | 0.911 |
| 1+2+3 | P_h^l | ✓ | 0.085 | 0.544 | 3.919 | 0.911 |
| 1+2+3 | P_h^l | - | 0.098 | 0.695 | 4.367 | 0.892 |

Table 5. Monocular depth estimation results on the KITTI Eigen test set in the ablation study. ‘FB.’ denotes using the final branches.

ules), respectively. The corresponding results are shown in the first part of Tab. 4, which show that TiO-Depth (with MFM (321)) performs best compared to the models with other modules. Then, the impact of the number of MFMs is shown in the second part of Tab. 4. It can be seen that the binocular performances are gradually improved by using more MFMs in most cases. The monocular depth estimation results of TiO-Depth with/without the ‘final branch (FB.)’ in the SDFA modules are shown in the last two rows of Tab. 5, where the performance of TiO-Depth with the final branches is much better than that of the model without these branches. We notice that the switchable branches are important for TiO-Depth to improve the monocular results, but the SDFA block is not a necessary choice. Please see the supplemental material for more experimental results and discussions. Considering that the three MFMs only contain 1.7M parameters in total, these results indicate the effectiveness of the dual-path decoder with MFMs in the two tasks.

Multi-stage joint-training strategy. We firstly analyze the impact of each term in the stereo loss L_S in binocular depth estimation by sequentially taking out the disparity guidance loss term L_{gui} , the cost volume loss term L_{cos} and the occlusion mask M_{occ} used in L_{rec2} . The corresponding results in the third part of Tab. 4 show that the performances of the model are dropped by removing the loss terms and the mask. Then we train TiO-Depth with different numbers of step(s) and pseudo labels to validate the effectiveness of the training strategy in monocular depth estimation in Tab. 5. As shown in the table, the monocular performance could not be improved by just training TiO-Depth for learning the two tasks without distillation (*i.e.*, with ‘1+2’ steps), but it is improved in most cases by training with three steps. Compared with using the stereo probability volume P_s^l , the accuracy of the monocular results could be consistently improved by using the hybrid probability volume P_h^l in the distilled loss L_{dis} . These results demonstrate that our training strategy is helpful for TiO-Depth to learn more accurate monocular and binocular depths.

5. Conclusion

In this paper, we propose TiO-Depth, a two-in-one depth prediction model for both the monocular and binocular self-supervised depth estimation tasks, while a multi-stage joint-

training strategy is explored for training. The full TiO-Depth is used to predict depths from stereo pairs, while the partial TiO-Depth by closing the duplicate parts could predict depths from single images. The experimental results in monocular and binocular depth estimations not only prove the effectiveness of TiO-Depth but also indicate the feasibility of bridging the gap between the two tasks.

Acknowledgements. This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDA27040811), the National Natural Science Foundation of China (Grant Nos. 61991423, U1805264), the Beijing Municipal Science and Technology Project (Grant No. Z211100011021004).

References

- [1] Filippo Aleotti, Fabio Tosi, Li Zhang, Matteo Poggi, and Stefano Mattoccia. Reversing the cycle: self-supervised deep stereo through enhanced monocular distillation. In *European Conference on Computer Vision*, pages 614–632. Springer, 2020. [2](#), [3](#)
- [2] Juan Luis Gonzalez Bello and Munchurl Kim. Forget about the lidar: Self-supervised depth estimators with med probability volumes. *Advances in Neural Information Processing Systems*, 33, 2020. [4](#), [5](#)
- [3] Juan Luis Gonzalez Bello and Munchurl Kim. Self-supervised deep monocular depth estimation with ambiguity boosting. *IEEE TPAMI*, 2021. [2](#)
- [4] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo-stereo matching with slanted support windows. In *Bmvc*, volume 11, pages 1–11, 2011. [3](#)
- [5] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *CVPR*, pages 5410–5418, 2018. [1](#), [3](#)
- [6] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *ICCV*, pages 7063–7072, 2019. [2](#)
- [7] Zhi Chen, Xiaoqing Ye, Wei Yang, Zhenbo Xu, Xiao Tan, Zhikang Zou, Errui Ding, Xinming Zhang, and Liusheng Huang. Revealing the reciprocal relations between self-supervised stereo and monocular depth estimation. In *ICCV*, pages 15529–15538, 2021. [1](#), [2](#), [5](#), [6](#), [7](#)
- [8] Bin Cheng, Inderjot Singh Saggu, Raunak Shah, Gaurav Bansal, and Dinesh Bharadia. S3 net: Semantic-aware self-supervised depth estimation with monocular videos and synthetic data. In *ECCV*, pages 52–69, 2020. [2](#)
- [9] Hyesong Choi, Hunsang Lee, Sunkyung Kim, Sunok Kim, Seungryong Kim, Kwanghoon Sohn, and Dongbo Min. Adaptive confidence thresholding for monocular depth estimation. In *ICCV*, pages 12808–12818, 2021. [2](#)
- [10] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015. [4](#)
- [11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. [6](#), [8](#)
- [12] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. [1](#), [6](#), [7](#)
- [13] José M Fácil, Alejo Concha, Luis Montesano, and Javier Civera. Single-view and multi-view depth fusion. *IEEE Robotics and Automation Letters*, 2(4):1994–2001, 2017. [2](#)
- [14] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, pages 2002–2011, 2018. [1](#)
- [15] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, pages 740–756, 2016. [1](#), [2](#)
- [16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361, 2012. [6](#), [8](#)
- [17] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, pages 270–279, 2017. [1](#), [2](#), [7](#)
- [18] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, pages 3828–3838, 2019. [1](#), [2](#), [3](#), [5](#), [6](#), [8](#)
- [19] Juan Luis GonzalezBello and Munchurl Kim. Forget about the lidar: Self-supervised depth estimators with med probability volumes. *Advances in Neural Information Processing Systems*, 33:12626–12637, 2020. [2](#), [6](#)
- [20] Juan Luis GonzalezBello and Munchurl Kim. Plade-net: Towards pixel-level accuracy for self-supervised single-view depth estimation with neural positional encoding and distilled matting loss. In *CVPR*, pages 6851–6860, 2021. [2](#), [6](#)
- [21] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *IJCV*, 129(6):1789–1819, 2021. [2](#)
- [22] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *CVPR*, pages 2485–2494, 2020. [6](#), [8](#)
- [23] Vitor Guizilini, Rareş Ambruş, Dian Chen, Sergey Zakharov, and Adrien Gaidon. Multi-frame self-supervised depth with transformers. In *CVPR*, pages 160–170, 2022. [2](#), [4](#), [6](#), [8](#)
- [24] Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. Semantically-guided representation learning for self-supervised monocular depth. In *International Conference on Learning Representations (ICLR)*, 2020. [2](#)
- [25] Mu He, Le Hui, Yikai Bian, Jian Ren, Jin Xie, and Jian Yang. Ra-depth: Resolution adaptive self-supervised monocular depth estimation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, pages 565–581. Springer, 2022. [2](#)

- [26] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *CVPR*, volume 2, pages 807–814. IEEE, 2005. 2, 3, 6
- [27] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 4
- [28] Baoru Huang, Jian-Qing Zheng, Stamatia Giannarou, and Daniel S Elson. H-net: Unsupervised attention-based stereo depth estimation leveraging epipolar geometry. In *CVPR*, pages 4460–4467, 2022. 3, 6
- [29] Hyunyoung Jung, Eunhyeok Park, and Sungjoo Yoo. Fine-grained semantics-aware representation enhancement for self-supervised monocular depth estimation. In *ICCV*, pages 12642–12652, 2021. 2
- [30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
- [31] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *ECCV*, pages 582–600, 2020. 2, 6
- [32] Hanhan Li, Ariel Gordon, Hang Zhao, Vincent Casser, and Anelia Angelova. Unsupervised monocular depth learning in dynamic scenes. In *Conference on Robot Learning*, pages 1908–1917. PMLR, 2021. 8
- [33] Liang Liu, Guangyao Zhai, Wenlong Ye, and Yong Liu. Unsupervised learning of scene flow estimation fusing with local rigidity. In *IJCAI*, pages 876–882, 2019. 6, 7
- [34] Pengpeng Liu, Irwin King, Michael R Lyu, and Jia Xu. Flow2stereo: Effective self-supervised learning of optical flow and stereo matching. In *CVPR*, pages 6648–6657, 2020. 3, 7
- [35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 3, 6
- [36] Yangqi Long, Huimin Yu, and Biyang Liu. Two-stream based multi-stage hybrid decoder for self-supervised multi-frame monocular depth. *IEEE Robotics and Automation Letters*, 7(4):12291–12298, 2022. 2
- [37] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5667–5675, 2018. 5
- [38] Diogo Martins, Kevin Van Hecke, and Guido De Croon. Fusion of stereo and still monocular depth estimates in a self-supervised learning context. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 849–856. IEEE, 2018. 2
- [39] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015. 6, 7
- [40] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. 6
- [41] Rui Peng, Ronggang Wang, Yawen Lai, Luyang Tang, and Yangang Cai. Excavating the potential capacity of self-supervised monocular depth estimation. In *ICCV*, pages 15560–15569, 2021. 2, 6, 7
- [42] Andra Petrovai and Sergiu Nedevschi. Exploiting pseudo labels in a self-supervised learning framework for improved monocular depth estimation. In *CVPR*, pages 1578–1588, 2022. 2, 6, 8
- [43] Andrea Pilzer, Stephane Lathuiliere, Nicu Sebe, and Elisa Ricci. Refine and distill: Exploiting cycle-inconsistency and knowledge distillation for unsupervised monocular depth estimation. In *CVPR*, pages 9768–9777, 2019. 2
- [44] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 11(3):211–252, 2015. 6
- [45] Ashutosh Saxena, Jamie Schulte, Andrew Y Ng, et al. Depth estimation using monocular and stereo cues. In *IJCAI*, volume 7, pages 2197–2203, 2007. 2
- [46] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *ECCV*, pages 572–588, 2020. 2
- [47] Fabio Tosi, Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning monocular depth estimation infusing traditional stereo knowledge. In *CVPR*, pages 9799–9809, 2019. 2, 6
- [48] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 international conference on 3D Vision (3DV)*, pages 11–20, 2017. 7
- [49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4
- [50] Longguang Wang, Yulan Guo, Yingqian Wang, Zhengfa Liang, Zaiping Lin, Jungang Yang, and Wei An. Parallax attention for unsupervised stereo correspondence learning. *IEEE TPAMI*, 2020. 3
- [51] Yang Wang, Peng Wang, Zhenheng Yang, Chenxu Luo, Yi Yang, and Wei Xu. Unos: Unified unsupervised optical-flow and stereo-depth estimation by watching videos. In *CVPR*, pages 8071–8081, 2019. 1, 3, 6, 7
- [52] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 5
- [53] Jamie Watson, Michael Firman, Gabriel J Brostow, and Daniyar Turmukhambetov. Self-supervised monocular depth kints. In *ICCV*, pages 2162–2171, 2019. 1, 2, 6
- [54] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1164–1174, 2021. 2, 6, 8

- [55] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. Segstereo: Exploiting semantic information for disparity estimation. In *ECCV*, pages 636–651, 2018. [1](#), [2](#), [3](#)
- [56] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, pages 1983–1992, 2018. [2](#)
- [57] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *CVPR*, pages 185–194, 2019. [1](#), [3](#)
- [58] Chao Zhou, Hong Zhang, Xiaoyong Shen, and Jiaya Jia. Unsupervised learning of stereo matching. In *ICCV*, pages 1567–1575, 2017. [2](#), [3](#)
- [59] Zhengming Zhou and Qiulei Dong. Learning occlusion-aware coarse-to-fine depth map for self-supervised monocular depth estimation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6386—6395, 2022. [2](#), [5](#), [6](#)
- [60] Zhengming Zhou and Qiulei Dong. Self-distilled feature aggregation for self-supervised monocular depth estimation. In *ECCV*, pages 709–726. Springer, 2022. [2](#), [3](#), [4](#), [5](#), [6](#), [7](#)
- [61] Zhongkai Zhou, Xinnan Fan, Pengfei Shi, and Yuanxue Xin. R-msfm: Recurrent multi-scale feature modulation for monocular depth estimating. In *ICCV*, pages 12777–12786, 2021. [6](#)
- [62] Shengjie Zhu, Garrick Brazil, and Xiaoming Liu. The edge of depth: Explicit constraints between segmentation and depth. In *CVPR*, pages 13116–13125, 2020. [2](#), [5](#), [6](#)