Self-Supervised Monocular Depth Estimation by Direction-aware Cumulative Convolution Network

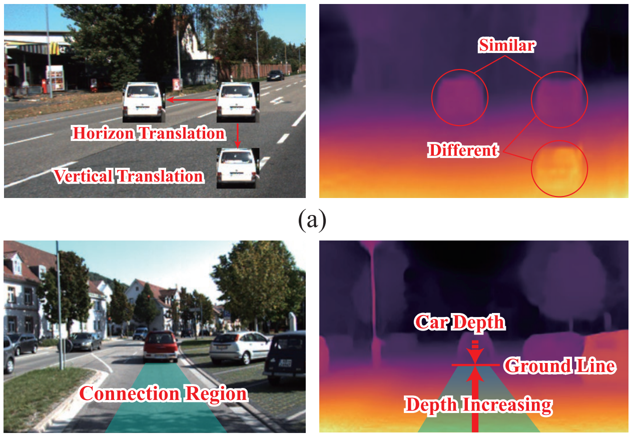
Wencheng Han[1], Junbo Yin[2], Jianbing Shen[1†]
[1] SKL-IOTSC, CIS, University of Macau, [2] Beijing Institute of Technology
{wencheng256, yinjunbocn, shenjiangbingcg}@gmail.com

Abstract

Monocular depth estimation is known as an ill-posed task in which objects in a 2D image usually do not contain sufficient information to predict their depth. Thus, it acts differently from other tasks (e.g., classification and segmentation) in many ways. In this paper, we find that self-supervised monocular depth estimation shows a direction sensitivity and environmental dependency in the feature representation. But the current backbones borrowed from other tasks pay less attention to handling different types of environmental information, limiting the overall depth accuracy. To bridge this gap, we propose a new Direction-aware Cumulative Convolution Network (DaCCN), which improves the depth feature representation in two aspects. First, we propose a direction-aware module, which can learn to adjust the feature extraction in each direction, facilitating the encoding of different types of information. Secondly, we design a new cumulative convolution to improve the efficiency for aggregating important environmental information. Experiments show that our method achieves significant improvements on three widely used benchmarks, KITTI, Cityscapes, and Make3D, setting a new state-of-the-art performance on the popular benchmarks with all three types of self-supervision. https://github.com/wencheng256/DaCCN.

1. Introduction

Monocular depth estimation is an important vision task for autonomous driving, which can generate a depth map for the image from a single camera. Unlike stereo-matching methods [38, 9, 25, 19], monocular depth estimation does not require rectified images, making it easier to be applied for self-driving cars. Because of this, monocular depth estimation methods attract much more attention from both the academic and the industrial societies, and many representative monocular depth estimation methods [23, 24, 3, 10] have been proposed during the last decade.



(a)

(b)

Figure 1. **Illustration of the direction sensitivity of self-supervised monocular depth estimation.** In (a), we translate the same car into different positions, and their depth values are shown in the right figure. In (b), we illustrate the connection region of the car and analyze the importance of this region.

The pioneering work of Eigen *et al.* [6] first developed a CNN-based network and trained the model in a fully supervised manner. To alleviate the need for the ground truth, Grag *et al.* [10] proposed a self-supervised method based on the stereo images. Zhou *et al.* [53] proposed a pose network to predict the relative position between two consecutive frames and only employ the sequence captured by a single camera in the training phase. Based on these works, a series of monocular depth estimation methods based on self-supervised learning have been proposed [33, 45, 18, 33]. In this paper, we mainly focus on the self-supervised monocular depth estimation task by fully exploring the direction sensitivity and environmental dependency information of this task.

Monocular depth estimation is an ill-posed task since the pixels of one object do not contain enough information to predict its depth. Therefore, the models highly rely on the

Corresponding author[†]: *Jianbing Shen*. This work was supported in part by the FDCT grants 0154/2022/A3 and SKL-IOTSC(UM)-2021-2023, the MYRG-CRG2022-00013-IOTSC-ICI grant and the SRG2022-00023-IOTSC grant.

Settings	Metrics		
	Abs Rel ↓	RMSE ↓	$FLOPs$
original (640 × 192)	0.115	4.863	8B
Horizon Stretch (1280 × 192)	0.118	4.875	16B
Vertical Stretch (640 × 384)	0.108	4.622	16B
Equal Stretch (1280 × 384)	0.109	4.723	32B

Table 1. **Analysis about different input ratios with monodepth2.** We adopt Abs Rel, and RMSE as our metrics. For the two metrics, lower values are better. We also provide the FLOPs of each setting.

interrelationships between the objects and environments. Previous depth estimation backbones [39, 41, 14, 40] seldom considered the depth-aware environmental encoding efficiency, which will lead to the lack of important depth clues, thereby limiting the overall performance of models.

In Fig. 1(a), the car is translated to different positions in the image, and their depth values are visualized in the right figure. From the visualization results, we find that even with the same pixels, these objects in different positions own different depth values. This demonstrates that depth prediction relies on the environment of objects. We further observed that the horizontally translated objects have little depth variance from the original object, but the depth of vertically translated objects changed a lot. Based on these observations, we infer that information from different directions plays different roles in depth estimation. The information along the view line contributes more to the depth variations, and the information from the horizontal lines keeps the depth consistency between objects. Therefore feature extraction from each direction could show different preferences. To explore their differences, we prepare a more detailed analysis in Table 1.

As mentioned in previous works [11, 47], increasing the input resolution will facilitate detailed information extraction, and a small resolution is helpful for global information encoding. Thus, we change the horizon and vertical resolutions, respectively, and train the model to compare their performances. If the feature extraction from the two directions contributes equally to the final accuracy, models with the large horizon and vertical resolution will perform similarly. As shown in Table 1, the depth estimator gets a significant performance drop when increasing the horizon resolution, indicating that the global information is preferred in this direction for better performance. While the model with a large vertical resolution obviously outperforms the one with the original resolution, which performs closely to the model with equally stretched inputs. This demonstrated that detailed information is more critical in the vertical direction for performance. Along this direction, we infer that information from the connection region is an important clue for depth estimation. As shown in Fig. 1(b), the ground line is a crucial reference for the depth estimation of the car [15], while the depth of the ground line largely relies on the region between it and the camera, which is named the connection region in this paper.

Although the depth estimation task is direction-sensitive and environmentally dependent, current backbones cannot fully use these properties. Traditional convolutional networks usually have the same receptive fields for every direction and encode the information from them similarly. This would lead to less efficiency in extracting various types of features. Moreover, convolutional operations equally aggregate information from the receptive fields into the center position. This aggregating strategy cannot efficiently utilize the critical information encoded in the connection regions.

To solve these problems, we propose a new Direction-aware Cumulative Convolution Network (DaCCN) for depth feature encoding. Our DaCCN improves the feature representation in two aspects. The first improvement is for feature extraction. DaCCN can learn to adjust the feature extraction from different directions, facilitating their information encoding. As discussed above, the encoded information from different directions in the images plays diverse roles during depth estimation. Therefore, the feature extraction from each direction should be adjusted according to the features it carries. Instead of manually adjusting the feature extraction, we design a learnable and direction-aware module to optimize it in an end-to-end manner during the offline training.

Another improvement is for feature aggregation. DaCCN can efficiently aggregate environmental information from the connection regions. The connection regions are the areas that contain all the spaces between the camera and objects and are critical for depth estimation. To efficiently aggregate information from these areas, we propose a new cumulative convolution operation, which can accumulate the environmental features from the connection regions and learn to fuse them efficiently. We integrate our DaCCN into a state-of-the-art baseline model of self-supervised depth estimation and evaluate the performance on three representative benchmarks. Experimental results show that our method achieves significant improvements with a new start-of-the-art performance.

In conclusion, the main contributions of this paper could be summarized into four folds:

- We carefully analyze the direction sensitivity and environmental dependency of self-supervised monocular depth estimation and propose the new Direction-aware Cumulative Network for better feature representation in depth estimation.

- We find that features extracted from different directions in the image play distinct roles during depth prediction and propose a learnable module to adjust the sample density and receptive fields for each direction.

- We propose a new convolutional operation for encoding the critical environmental information from the connection regions.

- Experiments on three datasets show the improvements of the proposed methods, and we set a new state-of-the-art performance on three widely used benchmarks.

2. Related Works

2.1. Supervised Monocular Depth Estimation

Depth estimation is a fundamental task in the computer vision area. It takes RGB images as input and generates depth maps as output. Each pixel in the depth map indicates the corresponding distance between the object and the camera viewpoint. Depth estimation can be functionally classified into three categories, monocular depth estimation, binocular depth estimation, and multi-view depth estimation. Among them, monocular depth estimation has drawn much attention in recent years [54, 17, 31, 48, 44, 42, 1, 42, 5, 51, 50], because of its wide application in autonomous driving.

The supervised learning approach for monocular depth estimation was first introduced, where pixel-level ground truth depth information is needed in the training phase. Eigen *et al.* [6] first proposed a deep learning model to predict the depth values under the supervision of ground truth. Their network consists of two deep network stacks, one responsible for encoding coarse-depth information and the other for fine-grained depth information. After this, different methods were proposed to improve the performance, like Li *et al.* [21] applied conditional random fields into monocular depth estimation. Some other works exploited the geometric relationship in the images. For example, Qi *et al.* [35] proposed two networks to estimate the depth and surface normal from an image. Ummenhofer *et al.* [43] developed a network to predict the depth maps according to the structure from motion (SfM) technique. Although these works have achieved promising performance, the supervised training needs a large amount of ground truth depth, which can only be gained by some special facilities, like LiDAR. The high costed data collection limits the wide application of these methods.

2.2. Self-supervised Monocular Depth Estimation

To avoid the need for labelled data in the monocular depth estimation, Garg *et al.* [8] firstly introduced a promising procedure to learn depth estimation in a self-supervised way. They employed stereo images in the training phase and formed depth optimization to the minimizing of disparity between fabricated images and real images. To release the requirement of stereo images, Zhou *et al.* [53] estimated depth map and camera pose simultaneously and only used video sequences from a single camera in the training phase. With the predicted depth and relative pose between adjacent frames, a fabricated frame can be generated, and the disparity can be calculated between it and the real frame. But the occlusion pixels between two frames and the moving objects will significantly influence the performance. Then, Godard *et al.* [11] added a minimum loss for alleviating the crucial challenges using self-supervised approaches. They found that the occlusion in the previous and subsequent frames is complimentary. Therefore the model could choose the visible frame to calculate the losses of some areas. To solve the moving object problem, they propose an efficient strategy by adding another minimum loss to ignore the loss values from these objects. After that, many works improved the performance of self-supervised monocular depth estimation [37, 56, 36, 20, 12]. Masoumian *et al.* [28] developed a multi-scale monocular depth estimation based on a graph convolutional network. Guizilini *et al.* [13] proposed a 3D packing network in this area. Watson *et al.* [46] introduced the cost volume to build a multi-frame model and achieved significant improvement. Zhou *et al.* [52] exploited the semantic information with down and up-sample procedures to improve depth estimation accuracy. However, most of these works employed a backbone network from classification-based tasks [7, 26], like U-Net [39] and HRNet [41], but few works considered the difference between depth estimation and their original tasks.

3. Method

3.1. Direction-aware Cumulative Network

Fig. 2 shows an overview framework of the proposed DaCCN. The network includes two main parts: an encoder that extracts feature maps from the input images and a decoder that produces the depth maps based on the feature maps. There are four branches in the encoder, each of which encodes features in a different resolution. To achieve the direction-aware feature extraction, we insert a learnable affinity transformation at the beginning of each branch, converting the input into the feature extraction space for direction-aware information encoding. Correspondingly, a back projection is appended at the end of each branch to keep the consistency between features and input images. The affinity transformation, the back projection, and the blocks between them together constitute the direction-aware module. Finally, the outputs of each block in the branches are concatenated and sent into the decoder for depth prediction.

There are four stages in the decoder, where each stage up-samples the current feature maps and fuses them with the corresponding feature maps from the encoder. Finally, a depth map is generated based on the fused feature map. Totally, four stages generate four depth maps with different resolutions, and four losses are calculated with the outputs. In the evaluation phase, only the depth map with the largest resolution is predicted, and the parameters for the

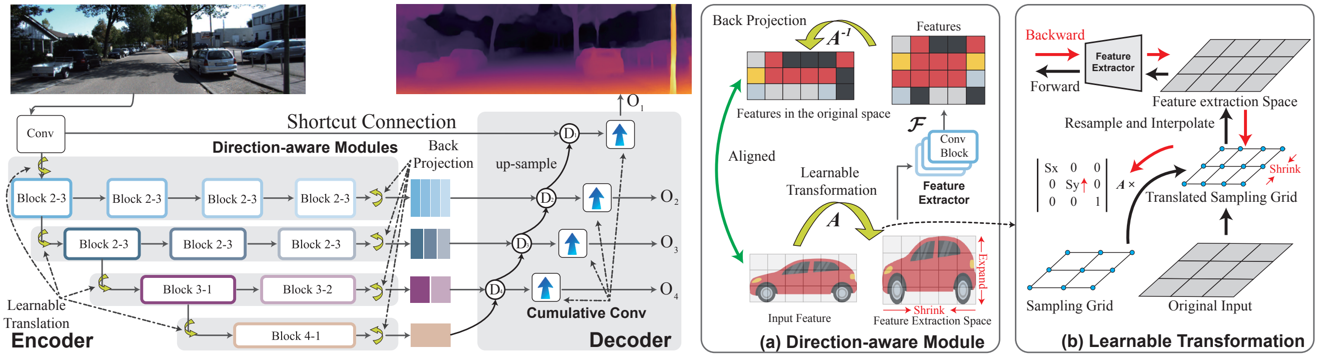


Figure 2. **The overview architecture of the proposed method.** There are mainly two modules in the proposed DaCCN architecture, a feature encoder, and a depth decoder. (a) shows the details of the direction-aware module. (b) illustrates how the learnable transformation is optimized.

other three heads are not used. We then apply a cumulative convolution on the fused feature maps for direction-aware aggregation. Features in this stage have encoded abundant semantic information and local information. Cumulative convolution will aggregate the desired environmental information from the connection regions and improve the depth estimation accuracy. Notably, we only use a single cumulative convolution at each stage because it is enough to aggregate features from the whole connection region. Direction-aware modules and cumulative convolution are two major improvements of DaCCN. In the subsequent sections, we will introduce their details and how they enhance depth prediction.

3.2. Direction-aware feature extraction

Convolutional operations in baseline networks usually similarly treat the information from any direction around the object. This is helpful for instance-aware vision tasks because the semantic information from any direction plays a similar role in these tasks. But as mentioned before, self-supervised monocular depth estimation treats the information from each direction differently. This disparity would lead to less efficiency of the model. To alleviate this problem, we propose a new direction-aware module that can learn to adjust the feature extraction from each direction.

As discussed in Table 1, the preferred information from each direction differs for the model performance. Based on this, we think the sample density and receptive fields are two direction-aware factors in feature extraction. The sampling density is defined as the number of feature vectors extracted from a unit area of the input image. Obviously, the larger the sample density, the more detailed information encoded in the feature maps. On the other side, receptive fields control the range of feature extraction, and larger receptive fields would incorporate larger ranges of pixels into each feature vector. Thus the features would pay more attention to the global information as indicated by some previous works [29, 47]. The model should use a smaller receptive field and employ more parameters for the direction that needs to be extracted more detailed information. On the contrary, large receptive fields are preferred for the global dependency direction; accordingly, the sample density should be reduced for computation efficiency. The direction-aware module is designed based on this assumption and can learn to adjust the sample density and receptive fields during the training phase.

As shown in Fig. 2, there are three parts in the direction-aware module, an affinity transformation $\boldsymbol{A}$, a convolutional feature extraction block $\mathcal{F}$, and back projection transformation $\boldsymbol{A}^{-1}$. The affinity transformation is the most important part of the module, which can transform the inputs into the feature extraction space. In this space, the sampling grid of input features is adjusted according to the information in each direction. For the direction that needs detailed information, the transformation will increase the sampling numbers and encode more details in the feature map. And for the direction focusing on a global view, the sampling number is reduced for a larger receptive field. Then, the convolutional feature extractor is employed to extract features from the resampled inputs. Finally, the back-projection transformation will transform the features back to the original space:

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\mathcal{F}(\boldsymbol{A}\boldsymbol{I}), \tag{1}$$

where $\boldsymbol{I}$ is the input of the block.

It is hard for humans to determine the suitable way to extract features from each direction. Therefore, we choose to give the learning ability to affinity transformation. To be specific, we regard the scaling ratios in the affinity matrix as two trainable parameters s_x and s_y:

$$\boldsymbol{A} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

Here, we resample and interpolate the inputs into the feature extraction space. Because the interpolation operation

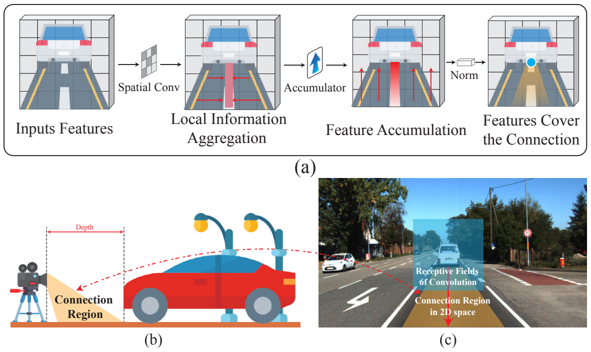

Spatial Conv
Accumulator
Norm
Inputs Features
Local Information Aggregation
Feature Accumulation
Features Cover the Connection
(a)
Depth
Connection Region
Receptive Fields of Convolution
Connection Region in 2D space
(b)
(c)

Figure 3. **Illustration of the cumulative convolution and connection region.** (a) The cumulative convolution. (b) The connection region in the 3D space. (c) Comparison between the receptive fields of convolutions and cumulative convolution.

will merge pixels in a differentiable way based on their distance to the sampling positions, the sampling points can be optimized by the gradient descent algorithms, as shown in Fig. 2(b). Then, features are extracted from the adjusted inputs. As the feature extraction space is not well aligned with the original inputs, we employ a back projection transformation with the inverse matrix $\boldsymbol{A}^{-1}$ for projecting the feature maps back to the original space.

Although some other methods can also adjust feature extraction during the training phase, such as the deformable convolution [4], they cannot achieve a similar goal as the direction-aware module. Deformable convolution learns to extract features from different positions by predicting offsets for the convolution kernels. It is designed to efficiently extract features from different-shaped objects, but the sample density cannot be adjusted during this procedure. In contrast, the direction-aware module can change the sample density in each direction and is more suitable for extracting features with different types.

3.3. Cumulative Convolution

As discussed in the introduction, monocular depth estimation is a direction-aware task where the information from the connection regions plays the most critical role. As shown in Fig. 3(b), in the 3D space, we define the space between the viewpoint and the object as the connection region. It includes the ground between the camera and the object and all the stuff on the ground. Therefore, this region contains the most crucial clues for estimating the object's depth value. Given a point $\boldsymbol{P}(X, Y, Z)$ in this region, a corresponding pixel $\boldsymbol{p}(x, y)$ in the 2D image can be gained by applying the intrinsic matrix on it:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix}, \tag{3}$$

where f_x and f_y are the pixel focal lengths and o_x, o_y are the offsets of the principal point. Because all the Z values (depth values) in the connection region are smaller than that of the objects, most projected points in the 2D image have larger y values than the objects and thus are located at the bottom areas of the objects.

We believe that better feature extraction for monocular depth estimation should fully exploit the information from connection regions. However, convolutional operation aggregates information into the center position, covering a square region around the object, making it less efficient to extract the critical information. As shown in Fig. 3(c), the blue areas are the square receptive field of a convolutional operation, and the yellow area indicates the projected connection region. To address this issue, we introduce a new cumulative convolution operation into this task. Instead of simply increasing the receptive fields of the convolution to cover the connection regions, we change the feature aggregation according to the direction where the connection regions are located.

As shown in Fig. 3(a), there are three parts of this operation. The first is a spatial convolution f that can extract spatial information from the local areas and modulate the features for the next stage. The second one is the accumulator Acc, which can cumulate the features from the bottom to the current pixels of the feature map. This operation will enlarge the receptive fields of the pixels towards the bottom line covering the whole connection region. But this operation would lead to a value imbalance between pixels because each pixel in the feature map aggregates information from a different scope. Therefore, the last part of this operation is a normalization $Norm$ operation which can normalize the aggregated features according to their position in the feature map:

$$CumulativeConv(\boldsymbol{x}) = \delta(Norm(ACC(f(\boldsymbol{x}, \eta)))), \tag{4}$$

where $\boldsymbol{x}$ is the input features, η is the weight in the spatial encoder and δ is the activation function.

In this paper, we employ a cumulative summation as our accumulator. It will cumulatively sum the features from to bottom to the current positions:

$$\boldsymbol{X}_{p,q} = \sum_{i=rows}^{p} \boldsymbol{x}'_{i,q}, \tag{5}$$

where p, q are the row and column index of the resulting feature maps $\boldsymbol{X}$ and $\boldsymbol{x}'_{i,q}$ means features in the ith raw and qth column of the input feature maps $\boldsymbol{x}'$. $rows$ is the number of rows in $\boldsymbol{x}'$. Correspondingly, we design a normalization method to match this accumulator by dividing the pixels according to their row number:

$$Norm(\boldsymbol{X}) = \frac{\boldsymbol{X}_{p,q}}{(rows - p)}. \tag{6}$$

3.4. Loss Function

Following our baselines, we employ the self-supervised method and formulate the monocular depth estimation problem as minimising the photometric reprojection error. To be specific, given two images I_t and $I_{t'}$ from different viewpoints. A pseudo target image $I_{t'\to t}$is produced by translating the image I'_t according to the predicted depth D_t, the relative position $T_{t\to t'}$ and the intrinsic K:

$$I_{t'\to t} = I_{t'} \left\langle \operatorname{proj}\left(D_t, T_{t\to t'}, K\right)\right\rangle$$

where stereo images are available, and $T_{t\to t'}$ is the relative position between two cameras; otherwise, it is the predicted position by the PoseNet introduced in [53]. Then, the disparity L_p between the pseudo image $I_{t'\to t}$ and the original target image I_t is used to measure the accuracy of the depth D_t:

$$L_p = \frac{\alpha}{2}\left(1 - \mathrm{SSIM}\left(I_{t'\to t}, I_t\right)\right) + (1-\alpha)\left\|I_{t'\to t} - I_t\right\|_1,$$

where two similarity methods are employed to calculate the difference. One is an L1 loss, and the other is the structural similarity loss (SSIM) [16]. α is a hyperparameter that controls the weight of the two similarity metrics. Besides, an edge-aware smoothness regulation is used to keep the inner-object disparity smooth:

$$L_s = \left|\partial_x d_t^*\right| e^{-\left|\partial_x I_t\right|} + \left|\partial_y d_t^*\right| e^{-\left|\partial_y I_t\right|}.$$

The final loss of our method is defined as:

$$L = L_p + \lambda L_s,$$

where λ is the weight of edge-aware smoothness regulation.

4. Experiment

We train and evaluate our models on a DGX system with an Intel E5-2698 v4CPU, 512G memory. All the training and evaluations are conducted on a single Nvidia V100 GPU. To show the improvements, we incorporate them with one newly proposed high-performance baseline DIFFNet [52], which is based on the HR-Net networks [41, 27] and it is one of the current SOTA works.

4.1. Comparison on KITTI

The KITTI dataset is known as one of the most commonly used vision datasets containing many challenges, such as optical flow [49], visual odometry [30], semantic segmentation [7]. Also, it is considered the most prevalent criterion in self-supervised monocular depth estimation. There are 56 different scenes in the dataset that are divided into 28 scenes for training and the rest for evaluation. We adopt the data split [6] as our baseline models and pre-process them as [53] for removing static frames. Finally, $39,810$ triplets are used for training and $4,424$ for validation.

SOTA comparison As shown in Table 2, we evaluate the performance of our DaCCN on Eigen split [6]. We roughly divide the results into two types of resolution, *i.e*. the low resolution and the high resolution in the table. Totally we employ 7 metrics in the comparison where $AbsRel$, $SqRel$, $RMSE$, $RMSElog$ are error-based metrics and therefore lower values are better. δ is the disparity between the predicted depth and ground truth values. $\delta < 1.25$, $\delta < 1.25^2$, $\delta < 1.25^3$ are accuracy-based metrics, and the higher values are better. According to the table, our DaCCN achieves the best performance on all three supervision types and two different input resolutions.

Obviously, the improvements on $SqRel$ and $RMSE$ are particularly prominent. The two metrics are based on square error and the large error values from the hard cases that can be magnified in these two metrics:

$$SqRel = \frac{1}{n}\sum\left(\frac{y_{pred} - y_{gt}}{y_{gt}}\right)^2$$

$$RMSE = \sqrt{\frac{1}{n}\sum\left(y_{pred} - y_{gt}\right)^2}$$

Therefore, improvements in these two metrics indicate that our methods can fix some hard cases in the original model. Compared to our baseline model DIFFNet [52], our DaCCN with monocular video training and small inputs achieves 0.003, 0.103 and 0.167 improvements in terms of $AbsRel$, $SqRel$ and $RMSE$, respectively.

With MS training, the improvement of the proposed methods is more significant. With 640×192 inputs, DaCCN achieves 0.004, 0.102 improvements in terms of AbsRel and RMSE. With 1024×320 inputs and MS training, our DaCCN also achieves the best results among all these methods and sets a new state-of-the-art performance.

Quantitive Results We also compare the qualitative performance with the baseline work DiffNet [52]. As shown in Fig. 4, for some hard cases, our DaCCN can provide the correct depth estimation result while DiffNet cannot. Also, we show a typical improvement case of the cumulation convolution module (CC). In this case, CC can easily correct some vertical errors in the prediction, as shown in the white box in (c).

4.2. Comparison on Make3D and Cityscapes

Make3D is a dataset including both monocular RGB images and its corresponding depth maps. Due to the nonexistence of stereo images and monocular sequences, this dataset cannot be used to train self-supervised monocular depth estimation models but is extensively used as a testing set to evaluate the capability of networks on a disparate

Method	Resolution	Trian	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth2 [11]	640 × 192	M	0.115	0.903	4.863	0.193	0.877	0.959	0.981
PackNet-SfM [13]	640 × 192	M	0.111	0.785	4.601	0.189	0.878	0.960	0.982
HR-Depth [27]	640 × 192	M	0.109	0.792	4.632	0.185	0.884	0.962	0.983
R-MSFM6 [55]	640 × 192	M	0.112	0.806	4.704	0.191	0.878	0.960	0.981
DIFFNet [52]	640 × 192	M	0.102	0.764	4.483	0.180	0.896	0.965	0.983
BRNet [47]	640 × 192	M	0.105	0.698	4.462	0.179	0.890	0.965	0.984
DaCCN (ours)	640 × 192	M	**0.099**	**0.661**	**4.316**	**0.173**	**0.897**	**0.967**	**0.985**
Monodepth R50 [10]	512 × 256	S	0.133	1.142	5.533	0.230	0.830	0.936	0.970
3Net (VGG) [34]	512 × 256	S	0.119	1.201	5.888	0.208	0.844	0.941	0.978
Monodepth2 [11]	640 × 192	S	0.109	0.873	4.960	0.209	0.864	0.948	0.975
BRNet [47]	640 × 192	S	0.103	0.792	4.716	0.197	0.876	0.954	0.978
DaCCN (ours)	640 × 192	S	**0.099**	**0.735**	**4.610**	**0.193**	**0.882**	**0.955**	**0.979**
Monodepth2 [11]	640 × 192	MS	0.106	0.818	4.750	0.196	0.874	0.957	0.979
HR-Depth [27]	640 × 192	MS	0.107	0.785	4.612	0.185	0.887	0.962	0.982
R-MSFM6 [55]	640 × 192	MS	0.111	0.787	4.625	0.189	0.882	0.961	0.981
DIFFNet [52]	640 × 192	MS	0.101	0.749	4.445	0.179	0.898	0.965	0.983
BRNet [47])	640 × 192	MS	0.099	0.685	4.453	0.183	0.885	0.962	0.983
DaCCN (ours)	640 × 192	MS	**0.097**	**0.647**	**4.282**	**0.172**	**0.901**	**0.967**	**0.985**
Monodepth2 [11]	1024 × 320	M	0.115	0.882	4.701	0.190	0.879	0.961	0.982
PackNet-SfM [13]	1280 × 384	M	0.107	0.802	4.538	0.186	0.889	0.962	0.981
HR-Depth [27]	1024 × 320	M	0.106	0.755	4.472	0.181	0.892	0.966	0.984
R-MSFM6 [55]	1024 × 320	M	0.108	0.748	4.470	0.185	0.889	0.963	0.982
DIFFNet [52]	1024 × 320	M	0.097	0.722	4.435	0.174	0.907	0.967	0.984
BRNet [47]	1024 × 320	M	0.103	0.684	4.385	0.175	0.889	0.965	**0.985**
DaCCN (ours)	1024 × 320	M	**0.094**	**0.624**	**4.145**	**0.169**	**0.909**	**0.970**	**0.985**
SuperDepth + pp [32]	1024 × 382	S	0.112	0.875	4.958	0.207	0.852	0.947	0.977
Monodepth2 [11]	1024 × 320	S	0.107	0.849	4.764	0.201	0.874	0.953	0.977
BRNet [47]	1024 × 320	S	0.097	0.729	4.510	0.191	0.886	**0.958**	**0.979**
DaCCN (ours)	1024 × 320	S	**0.093**	**0.699**	**4.450**	**0.190**	**0.889**	0.957	0.978
Monodepth2 [11]	1024 × 320	MS	0.106	0.806	4.630	0.193	0.876	0.958	0.980
HR-Depth [27]	1024 × 320	MS	0.101	0.716	4.395	0.179	0.899	0.966	0.983
R-MSFM6 [55]	1024 × 320	MS	0.108	0.753	4.469	0.185	0.888	0.963	0.982
BRNet [47]	1024 × 320	MS	0.097	0.677	4.378	0.179	0.888	0.965	0.984
DIFFNet [52]	1024 × 320	MS	0.094	0.678	4.250	0.172	0.911	0.968	0.984
DaCCN (ours)	1024 × 320	MS	**0.091**	**0.622**	**4.170**	**0.168**	**0.912**	**0.969**	**0.985**

Table 2. **The SOTA comparison on KITTI benchmark-Eigen Split [6].** We compare the proposed methods with the representative models on the KITTI benchmark with three self-supervision manners. **M** in the train column means training with monocular video sequences, and **S** means stereo image pairs and **MS** means training with the two types of data. For the error-based metrics , the lower value is better; and for the accuracy-based metrics , the higher value is better. The best and second best results of each set are marked in **bold** and underline.

dataset. We compare our models with other representative works on this benchmark. As shown in Table 5, our models outperform all the other methods, which demonstrates our models can be well generalized to unseen scenes. With monocular training and 640 × 192 input, our method achieves 0.290 and 6.656 in terms of $AbsRel$ and $RMSE$ with significant improvements from other SOTA models.

Citiscapes is a representative dataset in the semantic segmentation area for autonomous driving applications. Also, it includes a series of stereo video sequences that can be used to train self-supervised depth estimation models. Following the setting of [46], we train and evaluate the proposed DaCCN on the Cityscape dataset, and the results are shown in Table 4. Our DaCCN significantly outperforms many state-of-the-art models on this dataset.

Components		Abs Rel ↓	RMSE ↓	$\delta < 1.25$ ↑
DaM	CC			
		0.102	4.483	0.896
✓		0.100	4.372	0.899
	✓	0.101	4.380	0.897
✓	✓	0.099	4.316	0.897

Table 3. **Ablation study of the proposed DaCCN.** We conduct ablation studies on the two new components in our model. Our DaM means the direction-aware module and CC denotes the cumulative convolution operation.

Architecture	Abs Rel ↓	Sq Rel ↓	RMSE↓	$\delta < 1.25$ ↑
Struct2Depth [2]	0.145	1.737	7.280	0.813
Monodepth2 [11]	0.129	1.569	6.876	0.849
Videos in the Wild [12]	0.127	1.330	6.960	0.830
Li et al. [22]	0.119	**1.290**	6.980	0.846
DaCCN	**0.113**	1.380	**6.305**	**0.888**

Table 4. **Cityscape results follow the settings of [46].**

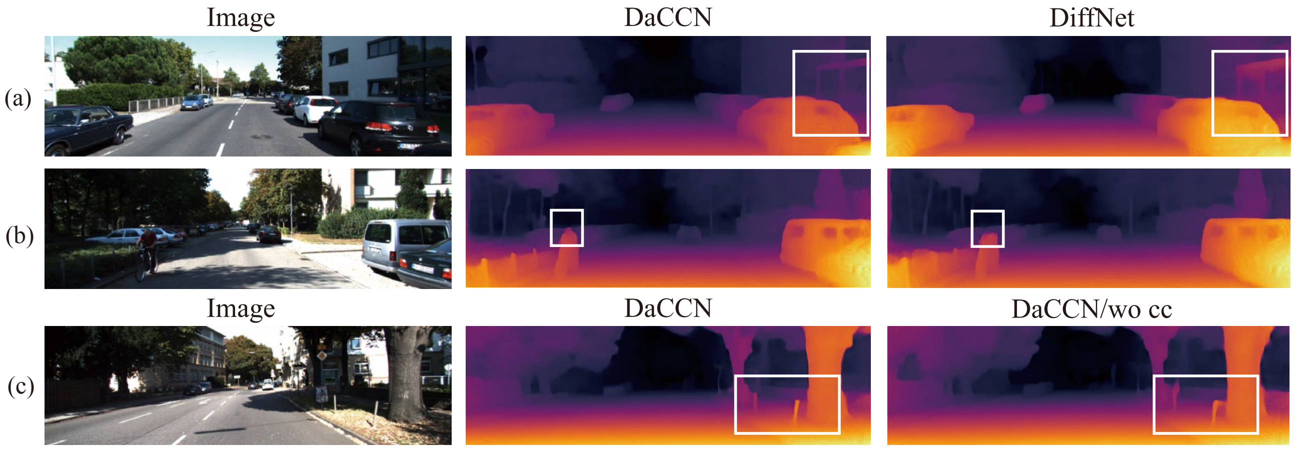


Figure 4. **Qualitative results on the KITTI Eigen split test set.** Our DaCCN can correct some errors of previous works (marked in white boxes) owning to the better feature representation.

4.3. Ablation Study

We conduct several ablation studies on the KITTI dataset. Eigen split [6] is used to validate the effectiveness of the proposed modules: direction-aware module and cumulative convolution module. In this experiment, we employ $AbsRel$, $RMSE$, and $\delta < 1.25$ as the metrics.

Direction-aware Module (DaM) Direction-aware module is responsible for adjusting feature extraction from different directions. During the training phase, s_x and s_y are optimized by the gradient descent algorithm to obtain an optimal solution. In our experiment, the optimal s_y is usually larger than s_x, because the vertical direction encodes more relative depth information, and the model needs more details to exploit it fully. On the contrary, the information extracted from the horizon direction reveals the consistency of both inner and inter objects. Thus, the model needs larger receptive fields to achieve this. As shown in Table 3, our DaM improves the performance on all the metrics.

Cumulative Convolution (CC) is responsible for enhancing environmental information aggregation. As shown in Table 3, the improvements of this module mainly lie in the $RMSE$ metric, which means this module corrects some hard cases for depth estimation. Environmental information is critical for the depth prediction of objects, while the original CNN cannot efficiently aggregate this information from the connection regions and thus fail to estimate depth values for some targets. Our cumulative convolution compensates for this limitation and improves the overall performance.

Efficiency We also compare the efficiency with more state-of-the-art approaches. In the training phase, the computation of our model will change because the sampling density is optimized by the gradient-descendent algorithm. In the evaluation phase, the sampling density has reached its optimal solution and thus the computation is fixed. As shown in Table 6, compared with other well-known methods, our model achieves a good balance between performance and efficiency.

Architecture	Abs Rel ↓	Sq Rel ↓	RMSE↓	log_{10} ↓
Monodepth	0.544	10.94	11.760	0.193
Monodepth2	0.322	3.589	7.414	0.163
BRNet	0.302	3.133	7.068	0.156
DaCCN	**0.290**	**2.873**	**6.656**	**0.149**

Table 5. **Make3D results with monocular training and** 640×192 **inputs.**

Architecture	Abs Rel	RMSE ↓	FLOPs(B)	Params(M)
Monodepth2	0.115	4.863	8	14
BRNet	0.105	4.462	31	19
PackNet-SfM	0.111	4.601	205	128
DIFFNet	0.102	4.483	2.3	12
DaCCN	0.099	4.316	4.3	13

Table 6. **Comparison results of params and computation.**

5. Conclusion

Monocular depth estimation is an ill-posed task and is very different from classification-based vision tasks in many ways. In this paper, we focus on the direction sensitivity and environmental dependency of this task and improve the efficiency of the backbone networks by exploiting a better feature representation. To achieve this, we propose a novel Direction-aware Cumulative Convolution Network (DaCCN) to strengthen the feature representation in two aspects: feature extraction and aggregation. Firstly, the direction-aware module is developed to learn to adjust the feature extraction from each direction fully facilitating the encoding of different types of features. Secondly, we propose the new cumulative convolution operation, which efficiently aggregates the information from the connection regions for improving environmental information aggregation. Experimental results show that the proposed models have achieved significant improvements on three prevalent benchmarks and set a new state-of-the-art performance.

References

[1] Ashutosh Agarwal and Chetan Arora. Depthformer: Multiscale vision transformer for monocular depth estimation with local global information fusion. *arXiv preprint arXiv:2207.04535*, 2022. 3

[2] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Unsupervised monocular depth and ego-motion learning with structure and semantics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 7

[3] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *IEEE/CVF International Conference on Computer Vision*, pages 7063–7072, 2019. 1

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *IEEE/CVF International Conference on Computer Vision*, pages 764–773, 2017. 5

[5] Tom van Dijk and Guido de Croon. How do neural networks see depth in single images? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2183–2191, 2019. 3

[6] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *IEEE/CVF International Conference on Computer Vision*, 2015. 1, 3, 6, 7, 8

[7] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017. 3, 6

[8] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, 2016. 3

[9] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *Asian conference on computer vision*, pages 25–38. Springer, 2010. 1

[10] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 1, 7

[11] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *IEEE/CVF International Conference on Computer Vision*, 2019. 2, 3, 7

[12] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *IEEE/CVF International Conference on Computer Vision*, pages 8977–8986, 2019. 3, 7

[13] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 3, 7

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 2

[15] Derek Hoiem, Alexei A Efros, and Martial Hebert. Automatic photo pop-up. In *ACM SIGGRAPH 2005 Papers*, pages 577–584. 2005. 2

[16] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010. 6

[17] Tak-Wai Hui. Rm-depth: Unsupervised learning of recurrent monocular depth in dynamic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3

[18] Varun Ravi Kumar, Senthil Yogamani, Markus Bach, Christian Witt, Stefan Milz, and Patrick Mäder. Unrectdepthnet: Self-supervised monocular depth estimation using a generic framework for handling common camera distortion models. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8177–8183. IEEE, 2020. 1

[19] Hamid Laga, Laurent Valentin Jospin, Farid Boussaid, and Mohammed Bennamoun. A survey on deep learning techniques for stereo-based depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1

[20] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 3

[21] Bo Li, Chunhua Shen, Yuchao Dai, Anton Van Den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015. 3

[22] 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. 7

[23] Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 7286–7291. IEEE, 2018. 1

[24] Shunkai Li, Fei Xue, Xin Wang, Zike Yan, and Hongbin Zha. Sequential adversarial learning for self-supervised deep visual odometry. In *IEEE/CVF International Conference on Computer Vision*, pages 2851–2860, 2019. 1

[25] Kun Liu, Changhe Zhou, Shengbin Wei, Shaoqing Wang, Xin Fan, and Jianyong Ma. Optimized stereo matching in binocular three-dimensional measurement system using structured light. *Applied optics*, 53(26):6083–6090, 2014. 1

[26] Dengsheng Lu and Qihao Weng. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28(5):823–870, 2007. 3

[27] Xiaoyang Lyu, Liang Liu, Mengmeng Wang, Xin Kong, Lina Liu, Yong Liu, Xinxin Chen, and Yi Yuan. Hr-depth: high resolution self-supervised monocular depth estimation. *CoRR abs/2012.07356*, 2020. 6, 7

[28] Armin Masoumian, David GF Marei, Saddam Abdulwahab, Julian Cristiano, Domenec Puig, and Hatem A Rashwan. Absolute distance prediction based on deep learning object detection and monocular depth estimation models. In *CCIA*, pages 325–334, 2021. 3

[29] S Mahdi H Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yagiz Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 4

[30] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. Ieee, 2004. 6

[31] Vaishakh Patil, Christos Sakaridis, Alexander Liniger, and Luc Van Gool. P3depth: Monocular depth estimation with a piecewise planarity prior. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3

[32] Sudeep Pillai, Rareş Ambruş, and Adrien Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation. In *ICRA*, 2019. 7

[33] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 1

[34] Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Learning monocular depth estimation with unsupervised trinocular assumptions. In *3DV*, 2018. 7

[35] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 3

[36] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 3

[37] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 3

[38] Paul Rogister, Ryad Benosman, Sio-Hoi Ieng, Patrick Lichtsteiner, and Tobi Delbruck. Asynchronous event-based binocular stereo matching. *IEEE Transactions on Neural Networks and Learning Systems*, 23(2):347–353, 2011. 1

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015. 2, 3

[40] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 2

[41] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*, 2019. 2, 3, 6

[42] Kunal Swami, Amrit Muduli, Uttam Gurram, and Pankaj Bajpai. Do what you can, with what you have: Scale-aware and high quality monocular depth estimation without real world labels. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3

[43] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 3

[44] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 3

[45] Jamie Watson, Michael Firman, Gabriel J Brostow, and Daniyar Turmukhambetov. Self-supervised monocular depth hints. In *IEEE/CVF International Conference on Computer Vision*, pages 2162–2171, 2019. 1

[46] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 3, 7

[47] Han Wencheng, Yin Junbo, Jin Xiaogang, Dai Xiangdong, and Shen Jianbing. Brnet: Exploring comprehensive features for monocular depth estimation. In *European Conference on Computer Vision*, 2022. 2, 4, 7

[48] Chi Xu, Baoru Huang, and Daniel S Elson. Self-supervised monocular depth estimation with 3-d displacement module for laparoscopic images. *IEEE Transactions on Medical Robotics and Bionics*, 4(2):331–334, 2022. 3

[49] Mingliang Zhai, Xuezhi Xiang, Ning Lv, and Xiangdong Kong. Optical flow and scene flow estimation: A survey. *Pattern Recognition*, 114:107861, 2021. 6

[50] Ning Zhang, Francesco Nex, George Vosselman, and Norman Kerle. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18537–18546, 2023. 3

[51] Chaoqiang Zhao, Youmin Zhang, Matteo Poggi, Fabio Tosi, Xianda Guo, Zheng Zhu, Guan Huang, Yang Tang, and Stefano Mattoccia. Monovit: Self-supervised monocular depth estimation with a vision transformer. In *2022 International Conference on 3D Vision (3DV)*, pages 668–678. IEEE, 2022. 3

[52] Hang Zhou, David Greenwood, and Sarah Taylor. Self-supervised monocular depth estimation with internal feature fusion. In *British Machine Vision Conference (BMVC)*, 2021. 3, 6, 7

[53] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 1, 3, 6

[54] Zhengming Zhou and Qiulei Dong. Self-distilled feature aggregation for self-supervised monocular depth estimation. In *European Conference on Computer Vision*, pages 709–726. Springer, 2022. 3

[55] Zhongkai Zhou, Xinnan Fan, Pengfei Shi, and Yuanxue Xin. R-msfm: Recurrent multi-scale feature modulation for monocular depth estimating. In *IEEE/CVF International Conference on Computer Vision*, 2021. 7

[56] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *European Conference on Computer Vision*, 2018. 3