Monocular depth estimation using multi-dimensional dynamic convolution

Rui Zhang[1,2,*], **Hongfei Yu**[1,3]

[1]School of Artificial Intelligence and Software, Liaoning Shihua University, Fushun, China

[2]zhangrui9895@163.com
[3]yuhfln@163.com
*corresponding author

Abstract. To address the issue of inaccurate prediction results and low accuracy at the object edges in existing self-supervised monocular depth estimation algorithms, this study proposes a self-supervised monocular depth estimation algorithm based on dynamic convolution (OE-Depth). The algorithm employs a multi-dimensional dynamic convolution feature extraction network to acquire more comprehensive feature representations, thereby enhancing the predictive capability. Additionally, the algorithm is optimized by integrating a triplet loss term and employing metric learning techniques to refine the network's performance at the object edges. Experimental evaluations conducted on the KITTI dataset validate the effectiveness of the proposed algorithm, demonstrating an optimization of the error class index by nearly 10%. Notably, the most stringent criterion $\delta < 1.25$ achieves an accuracy of 0.908 for depth prediction.

Keywords: deep learning, dynamic convolution, triplet loss, monocular depth estimation, self-supervised.

1. Introduction

Acquiring depth information for every pixel in an image is a fundamental task for applications such as autonomous driving [1] and robotics [2], underscoring its critical importance. In recent years, to enhance the performance of self-supervised monocular depth estimation networks, Godard et al. [3] proposed minimizing the reprojection loss to address occlusion problems. Additionally, researchers like Jung et al. [4], Klingner et al. [5], and Guizilini et al. [6] have integrated semantic segmentation modules to achieve clearer object edge depths. Other methods, such as PackNet [7] and CADepth-Net [8], have used 3D convolutions to compress and decompress detailed representations, incorporated attention mechanisms [9], and combined transformer structures [10] to improve monocular depth estimation algorithms.

Current self-supervised monocular depth estimation algorithms suffer from low depth accuracy at object edges due to the lack of sufficient surrounding information, making it challenging to predict depth accurately. Some depth estimation algorithms are not capable of capturing edge information effectively, leading to errors in depth estimation at object edges. To improve the overall accuracy of monocular depth estimation algorithms, this paper employs multi-dimensional dynamic convolution to extract image features and obtain more detailed object characteristics. A learnable deformation module is used to dynamically adjust the shape and size of the convolution kernel based on the input data features, thereby extracting richer image feature representations. To enhance the accuracy of self-supervised monocular depth estimation at edges, this paper introduces a triplet loss term to address the issue of objects appearing with coarser outlines in the network results. By incorporating semantic information to label object boundaries in the images, these boundary details can serve as additional constraints, guiding the depth estimation module to estimate edge depths more accurately.

2. Research Methods

The overall network architecture proposed in this paper is shown in Figure 1. The entire network architecture consists of two parts: an encoder-based depth network and a decoder-based depth network.

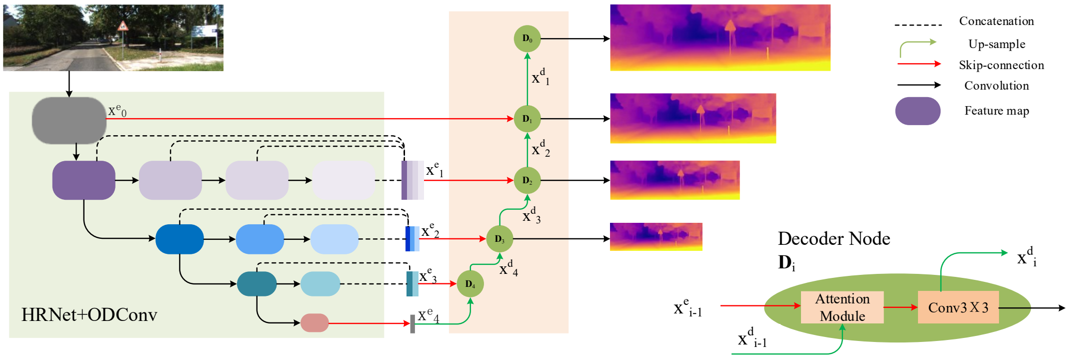

Figure 1. Overall Network Architecture

2.1. *Self-Supervised Monocular Depth Estimation*

Self-supervised monocular depth estimation involves predicting the distance from each pixel in a single color image to the camera viewpoint, resulting in a dense depth map. The network takes the original image I_t as input and the depth network generates the corresponding depth image D_t. The monocular video sequence inputs two temporally adjacent frames as the target frame I_t and the source frame I_{t+n}, where n∈{-1,+1}. The pose network predicts the relative pose change $T_{t\to t+n}$ between the two frames. Finally, using the depth map D_t and the relative pose $T_{t\to t+n}$, a reprojection operation is performed to obtain a 2D image from the new viewpoint, which serves as the supervision signal.

Most self-supervised monocular depth estimation methods construct photometric loss by minimizing the per-pixel reprojection loss:

$$L_p = \min pe\left(I_t, I'_{t+n\to t}\right) \tag{1}$$

where pe() is the photometric error composed of the L1 norm and Structural Similarity Index (SSIM).

Similarly, this paper uses an edge-aware smoothness loss to adjust the depth in low-gradient regions:

$$\mathrm{L}_s = \left|\frac{\nabla \mathrm{d}}{\partial x}\right| e^{-\left|\frac{\nabla I_t}{\partial x}\right|} + \left|\frac{\nabla d}{\partial y}\right| e^{-\left|\frac{\nabla I_t}{\partial y}\right|} \tag{2}$$

The loss function of the baseline network in this paper, shown in Equation 3, is a weighted sum of the photometric loss and the edge-aware smoothness loss, with weights μ and λ the same as in the baseline.

$$L_{baseline} = \mu L_p + \lambda L_s \tag{3}$$

2.2. Multi-Dimensional Dynamic Convolution

Conventional convolution uses static convolution kernels that are independent of the input samples, which means the feature extraction process cannot adapt to the input image. Multi-Dimensional Dynamic Convolution (ODConv) [11], however, adjusts the shape and size of the convolution kernels dynamically based on the characteristics of the input data during the convolution process. ODConv linearly weights multiple convolution kernels, with the weighting values related to the input image. This dynamic convolution depends on the input, enhancing the network's ability to extract feature representations. The structure of ODConv is shown below:

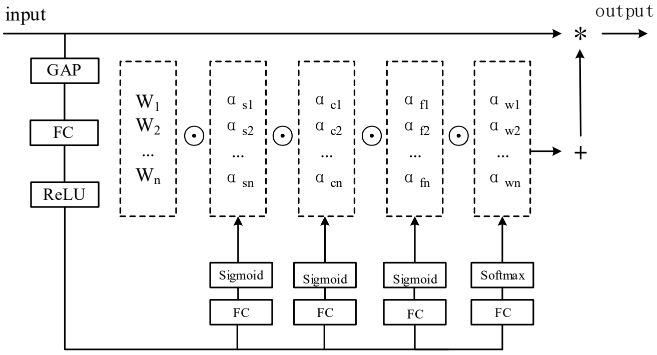

Figure 2. ODConv Structure

Specifically, ODConv first compresses the input through Global Average Pooling (GAP) into a feature vector of length c_{in}. This is followed by a Fully Connected (FC) layer and four branches. After the FC layer, a Rectified Linear Unit (ReLU) maps the compressed feature vector to a lower-dimensional space. Each branch has its own FC layer and a Softmax or Sigmoid function to generate normalized attention. The algorithm can be expressed as:

$$\text{out}put = \left(\alpha_{\omega 1} \odot \alpha_{f1} \odot \alpha_{c1} \odot \alpha_{s1} \odot W_1 + \cdots + \alpha_{\omega n} \odot \alpha_{fn} \odot \alpha_{cn} \odot \alpha_{sn} \odot W_n\right) * input (4)$$

where input and output represent the input and output features, respectively. W_n denotes the n-th convolution kernel, and $\alpha_{\omega n}$ is the attention scalar for the convolution kernel ωn. The attention scalars α_{sn}, α_{cn}, and α_{fn} are calculated along the spatial, input channel, and output channel dimensions of the convolution kernel W_n, respectively. ⊙ represents the multiplication operation along different dimensions of the kernel space. * denotes the convolution operation.

2.3. Triplet Loss

The triplet loss can be described by dividing local area samples into three parts, as shown in Figure 3. An anchor point is selected, and the depth network encourages depths within the same semantic region as the anchor pixel to be close, while regions with different semantics from the anchor pixel are pushed further away in terms of depth. The central pixel is taken as the anchor P_i, pixels with the same semantics as the anchor are considered positive pixels P_i^+, and pixels with different semantics are considered negative pixels P_i^-. The distances to positive $D^+(i)$ and negative pixels $D^-(i)$ are defined as the average Euclidean distance of L2-normalized depth features [4]:

$$D^+(i) = \frac{1}{|P_i^+|} \sum_{j \in P_i^+} \left\| \widehat{F_d}(i) - \widehat{F_d}(j) \right\|_2^2 \tag{5}$$

$$D^-(i) = \frac{1}{|P_i^-|} \sum_{j \in P_i^-} \left\| \widehat{F_d}(i) - \widehat{F_d}(j) \right\|_2^2 \tag{6}$$

where $F_d = \frac{F_d}{\|F_d\|}$.

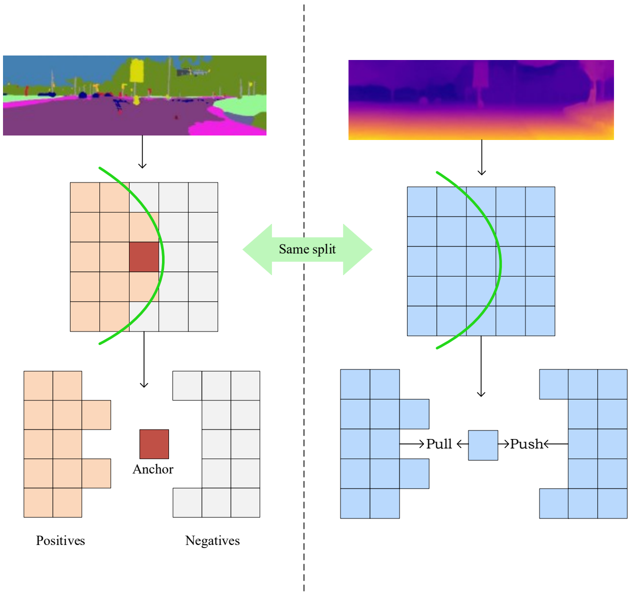

Figure 3. Semantic-Aware Triplet Loss

Actually, most of the negative pixels perform well. Some negative pixels located between the depth boundary of objects and semantic boundaries exhibit poor performance, as shown in Figure 4. Compared to the majority of well-performing negative pixels, using the average $D^-(i)$ strategy for triplet loss may significantly reduce the contribution of these poorly performing negative pixels to the network. Previous methods have used an average negative pixel distance $D^-(i)$ that is too large to optimize through triplet loss. Therefore, this paper minimizes the negative pixel distance to reduce $D^-(i)$:

$$D^-(i)' = \min_{j\in P_i^-} \left\|\widehat{F_d}(i) - \widehat{F_d}(j)\right\|_2^2 \tag{7}$$

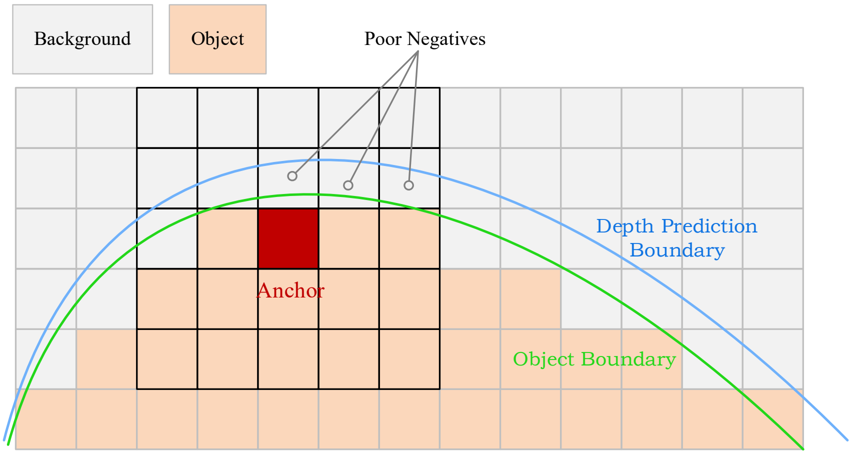

Figure 4. Example of Object Edge Coarsening

To address this, the positive pixel distance $D^+(i)$ and the minimized negative pixel distance $D^-(i)'$ can be separated from the original triplet loss. $D^-(i)'$ is compared with a new threshold m', and $D^+(i)$ is directly optimized. The triplet loss used in the experiment is as follows:

$$L_{triplet} = \frac{1}{|\Theta|} \sum_{P_i \in \Theta} \left(D^{+}(i) + \left[m^{'} - D^{-}(i)^{'} \right]_{+} \right) \tag{8}$$

where $[\,]_{+}$ denotes the hinge function, and Θ is the set of all semantic boundary pixels that meet the constraint conditions.

The final loss function used in this experiment is as follows:

$$L_{final} = \mu L_p + \lambda L_s + L_{triplet} \tag{9}$$

3. Experiments and Results

3.1. Implementation Details

The algorithm is implemented using version 1.8.1 of PyTorch as the deep learning framework. The experiments are conducted on an NVIDIA 3090 GPU with 24GB of memory for training and evaluation. The original images with a resolution of 1242*375 are first resized to 640*192. The batch size is set to 16, and the Adam optimizer with an initial learning rate of 1e-4 is used. The model is trained for a total of 20 epochs. The proposed method is initialized using the HRNet [12] pre-trained on the ImageNet dataset. The camera pose network in this paper uses ResNet18, with the input being two adjacent frames and the output being the camera pose change between the frames.

3.2. Quantitative Results

The experiments are conducted on the KITTI dataset for training and evaluation. The Eigen split is used during training, with the network being trained on 39,810 samples and validated on 4,424 samples. For evaluation, 697 samples are used to assess the performance based on the seven standard metrics proposed by Fang et al. [13]. The algorithm is compared with the baseline network under different training settings and resolutions, as shown in Tables 1 and 2. The proposed method outperforms the baseline network on multiple metrics. Figure 5 demonstrates that the proposed method achieves clearer and more accurate depth at object edges (contours), significantly improving the overall accuracy of the network's predictions.

Table 1. Comparison of Various Methods on the KITTI Dataset (Eigen Split, Low Resolution)

Method	Train	WXH	lower is better				higher is better		
			Abs Rel	Sq Rel	RMSE	RMSE log	δ1<1.25	δ2<1.25^2	δ3<1.25^3
Monodepth2[3]	M	640X192	0.115	0.903	4.863	0.193	0.877	0.959	0.981
PackNet[7]	M	640X192	0.111	0.785	4.601	0.189	0.884	0.960	0.983
HR-Depth[14]	M	640X192	0.109	0.792	4.632	0.185	0.884	0.962	0.983
CADepth[8]	M	640X192	0.105	0.769	4.535	0.181	0.892	0.964	0.983
DIFFNet[15]	M	640X192	**0.102**	0.764	4.483	0.180	**0.896**	**0.965**	0.983
OE-Depth(ours)	M	640X192	0.103	**0.706**	**4.413**	**0.179**	0.892	**0.965**	**0.984**
Monodepth2[3]	MS	640X192	0.106	0.818	4.750	0.196	0.874	0.957	0.979
HR-Depth[14]	MS	640X192	0.107	0.785	4.612	0.185	0.887	0.962	0.982
CADepth[8]	MS	640X192	0.102	0.752	4.504	0.181	0.894	0.964	**0.983**
DIFFNet[15]	MS	640X192	0.101	0.749	4.445	**0.179**	**0.898**	**0.965**	**0.983**
OE-Depth(ours)	MS	640X192	**0.099**	**0.683**	**4.389**	0.182	0.888	0.964	**0.983**

Table 2. Comparison of Various Methods on the KITTI Dataset (Eigen Split, High Resolution)

Method	Train	WXH	lower is better				higher is better		
			Abs Rel	Sq Rel	RMSE	RMSE log	δ1<1.25	δ2<1.25 $_2$	δ3<1.25 $_3$
Monodepth2[3]	M	1024X320	0.115	0.882	4.701	0.190	0.879	0.961	0.982
PackNet[7]	M	1280X384	0.107	0.802	4.538	0.186	0.889	0.962	0.983
HR-Depth[14]	M	1024X320	0.106	0.755	4.472	0.181	0.896	0.965	0.983
CADepth[8]	M	1024X320	0.102	0.734	4.407	0.178	0.898	0.966	**0.984**
Guizilini[6]	M	1280X384	0.100	0.761	4.270	0.175	0.902	0.965	0.982
DIFFNet[15]	M	1024X320	0.097	0.722	4.345	0.174	0.907	0.967	**0.984**
MonoViT[16]	M	1024X320	**0.096**	0.693	4.262	0.174	0.904	0.967	**0.984**
OE-Depth(ours)	M	1024X320	**0.096**	**0.655**	**4.192**	**0.172**	**0.908**	**0.968**	**0.984**

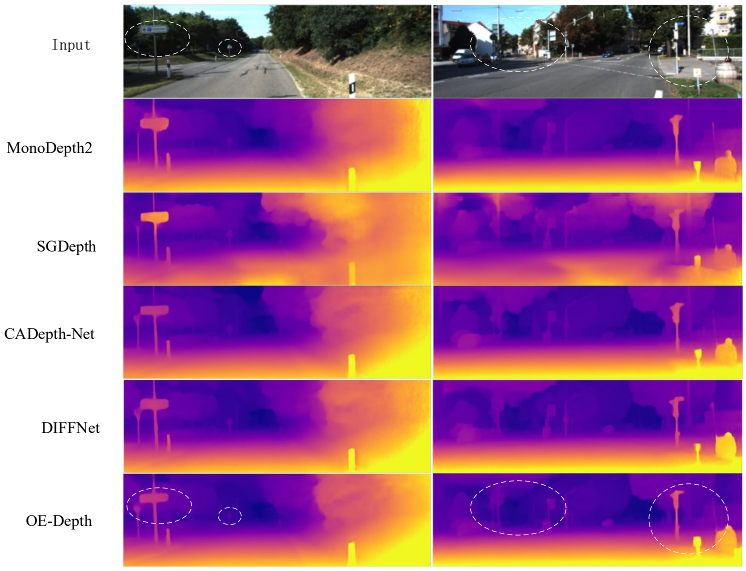

Figure 5. Comparison of Experimental Results

In Tables 1 and 2, the columns highlighted in blue represent the errors in the network's prediction results, with lower numbers indicating better performance; the columns highlighted in orange represent the accuracy of the network's prediction results, with higher numbers indicating better performance. The best results in the table are shown in bold, while the second-best results are underscored.

3.3. *Ablation Study*

In this paper, DIFFNet serves as the evaluation baseline model for the two modules mentioned above. All data in Table 3 are results obtained from monocular self-supervised training using images with a resolution of 640*192. To verify the effectiveness of each module, four sets of experiments were conducted under identical conditions. The experiments demonstrate that each introduced component has a positive impact on the baseline model, with the best results achieved when both components are added simultaneously.

Table 3. Ablation Study Comparison

ODConv	Triplet Loss	lower is better				higher is better		
		Abs Rel	Sq Rel	RMSE	RMSE log	$\delta 1<1.25$	$\delta 2<1.25^2$	$\delta 3<1.25^3$
X	X	0.109	0.838	4.648	0.186	0.887	0.963	0.982
✓	X	0.106	0.789	4.597	0.185	0.891	0.963	0.983
X	✓	0.105	0.740	4.510	0.182	0.891	0.964	0.983
✓	✓	**0.103**	**0.706**	**4.413**	**0.179**	**0.892**	**0.965**	**0.984**

4. Summary

This work primarily introduces two modules: Multi-Dimensional Dynamic Convolution and Triplet Loss. Multi-Dimensional Dynamic Convolution utilizes a novel multi-dimensional attention mechanism and parallel strategy to learn attention for convolution kernels along four dimensions in the internal kernel space, thereby enhancing the model's feature extraction capability. Triplet Loss incorporates semantic information, optimizing positive pixel distances by separating them from negative pixel distances, thereby addressing the issue of "coarse" object edges. Experimental results on public datasets demonstrate that the proposed method can generate clearer and more accurate depth at object edges. Future work will focus on addressing the problem of dynamic objects in monocular depth estimation by calculating masks for dynamic objects to improve the accuracy of depth map predictions.

References

[1] Wang, Y., Chao, W. L., Garg, D., et al. (2019). Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8445-8453).

[2] Griffin, B., Florence, V., Corso, J. (2020). Video object segmentation-based visual servo control and object depth estimation on a mobile robot. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 1647-1657).

[3] Godard, C., Mac Aodha, O., Firman, M., et al. (2019). Digging into self-supervised monocular depth estimation. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 3828-3838).

[4] Jung, H., Park, E., Yoo, S. (2021). Fine-grained semantics-aware representation enhancement for self-supervised monocular depth estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 12642-12652).

[5] Klingner, M., Termöhlen, J. A., Mikolajczyk, J., et al. (2020). Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16 (pp. 582-600). Springer International Publishing.

[6] Guizilini, V., Ambrus, R., Pillai, S., et al. (2020). 3d packing for self-supervised monocular depth estimation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2485-2494).

[7] Mallya, A., Lazebnik, S. (2018). Packnet: Adding multiple tasks to a single network by iterative pruning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 7765-7773).

[8] Yan, J., Zhao, H., Bu, P., et al. (2021). Channel-wise attention-based network for self-supervised monocular depth estimation. In 2021 International Conference on 3D Vision (3DV) (pp. 464-473). IEEE.

[9] Zhang, C., Ma, Y. X., Wan, J. W., et al. (2022). Single monocular depth estimation based on channel attention mechanism. Signal Processing, 38(11), 2332-2341. DOI:10.16798/j.issn.1003-0530.2022.11.010.

[10] Zhang, T., Zhang, X. L., Ren, Y. (2022). Monocular image depth estimation with fusion of Transformer and CNN. Journal of Harbin University of Science and Technology, 27(06), 88-94. DOI:10.15938/j.jhust.2022.06.011.

[11] Li, C., Zhou, A., Yao, A. (2022). Omni-dimensional dynamic convolution. arXiv preprint arXiv:2209.07947.

[12] Sun, K., Xiao, B., Liu, D., et al. (2019). Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 5693-5703).

[13] Fang, Z., Chen, X., Chen, Y., et al. (2020). Towards good practice for CNN-based monocular depth estimation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 1091-1100).

[14] Lyu, X., Liu, L., Wang, M., et al. (2021). HR-Depth: High Resolution Self-Supervised Monocular Depth Estimation. In National Conference on Artificial Intelligence (p. Bae J., Moon S., Im S., Eds.), Proceedings of the AAAI Conference on Artificial Intelligence, 37(1), 187-196.

[15] Zhou, H., Greenwood, D., Taylor, S. (2021). Self-supervised monocular depth estimation with internal feature fusion. arXiv preprint arXiv:2110.09482.

[16] Zhao, C., Zhang, Y., Poggi, M., et al. (2022). Monovit: Self-supervised monocular depth estimation with a vision transformer. In 2022 International Conference on 3D Vision (3DV) (pp. 668-678). IEEE.