



OPEN Lightweight monocular depth estimation using a fusion-improved transformer

Xin Sui, Song Gao✉, Aigong Xu, Cong Zhang, Changqiang Wang & Zhengxu Shi

The existing deep estimation networks often overlook the issue of computational efficiency while pursuing high accuracy. This paper proposes a lightweight self-supervised network that combines convolutional neural networks (CNN) and Transformers as the feature extraction and encoding layers for images, enabling the network to capture both local geometric and global semantic features for depth estimation. First, depth-separable convolution is used to construct a dilated convolution residual module based on a shallow network to improve the shallow CNN feature extraction receptive field. In the transformer, a multidepth separable convolution head transposed attention module is proposed to reduce the computational burden of spatial self-attention. In the feedforward network, a two-step gating mechanism is proposed to improve the nonlinear representation ability of the feedforward network. Finally, the CNN and transformer are integrated to implement a depth estimation network with a local-global context interaction function. Compared with other lightweight models, this model has fewer model parameters and higher estimation accuracy. It also has better generalizability for different outdoor datasets. Additionally, the inference speed can reach 87 FPS, achieving better real-time performance and accounting for both inference speed and estimation accuracy.

Keywords Self-supervision, Monocular depth estimation, Lightweight, CNN, Transformer

In recent years, with the rapid development of artificial intelligence technology, products such as robots, drones, and autonomous vehicles have been widely applied in human life¹. These applications require scene perception based on depth information, making accurate depth estimation crucial. Depth estimation, a core technology for autonomous systems to perceive the environment and estimate their states, provides fundamental depth information for research on visual odometry, autonomous driving, robot localization, and 3D reconstruction^{2–4}. By utilizing depth estimation for 3D reconstruction, more accurate information such as terrain height can be obtained, which is essential for areas such as environmental monitoring, urban planning, and natural disaster early warning and is crucial in various robotic systems and applications⁵. Existing depth sensors such as RGB-D cameras, LiDAR, and structured light sensors can provide accurate depth information⁶; however, these sensors have high hardware costs, large volumes, and high power consumption. Stereo cameras can obtain depth information through stereo matching methods⁷; however, such methods involve high computational complexity and demanding performance from computing units. Additionally, when temporal or spatial errors exist between two cameras in practical applications, the accumulation of errors may accelerate, further increasing algorithm requirements. In contrast, monocular depth estimation, which estimates depth maps from single RGB images, is a cost-effective and easy-to-deploy method. It does not have high hardware requirements and can obtain depth information solely from images. Therefore, methods based on single-image depth recovery have been widely studied.

Early research relied on depth cues such as image defocus, focus and defocus, and shadows to extract depth information from monocular images. However, such methods often have specific image requirements and may not be applicable to all scenes. With the development of machine learning, Saxena et al.⁸ utilized features to construct conditional random fields (CRFs) and Markov random fields (MRFs) to model depth information, considering global and long-range information, transforming the problem into a learning problem under a random field. However, traditional machine learning typically requires manual design and feature selection, making capture complex nonlinear relationships in data and achieving good generalizability is challenging in complex environments. In recent years, Eigen et al.⁹ used convolutional neural networks (CNNs) to perform monocular depth estimation tasks, estimating global coarse and local fine depth maps in two stages, realizing monocular depth estimation networks based on deep learning and achieving good results. Therefore, an increasing number of monocular depth estimation methods based on deep learning have been proposed.

School of Geomatics, Liaoning Technical University, Fuxin 123000, China. ✉email: 47221105@stu.lntu.edu.cn

In deep learning, monocular depth estimation methods can be divided into supervised and self-supervised learning methods based on whether real depth information is required during training. Supervised learning methods typically achieve high accuracy but require efficient training using large sets of images annotated with depth labels. This significantly increases the difficulty and cost of collecting large-scale, accurate, and dense ground-truth depth data. In contrast, self-supervised learning methods can avoid using large labelled datasets, effectively reducing costs and workload, thus attracting widespread attention in monocular depth estimation. Garg et al.¹⁰ regarded depth estimation as a novel view synthesis problem. They derived supervisory signals from the reprojection of images from different views based on the geometric relationships between consecutive frames to replace supervised losses based on depth labels, achieving unsupervised monocular depth estimation. Godard et al.^{11,12} proposed the Monodepth2 network. They introduced left-right disparity consistency and minimum reprojection losses to improve prediction accuracy, alleviate occlusion issues, and filter out moving objects with the same velocity as the camera using automasking. Many studies have applied transformer models¹³ in monocular depth estimation and achieved promising results. For instance, Zhao et al.¹⁴ introduced the monovision transformer as an encoder to model the global context based on Monodepth2. They progressively fused it with a convolutional decoder, enhancing the prediction accuracy by incorporating the transformer into the CNN. However, transformers are generally slower than CNNs due to their large number of parameters and quadratic complexity of multihead attention calculations, making them unsuitable for real-time tasks, especially for mobile robots and embedded devices. Diana et al.¹⁵ designed a lightweight network called FastDepth to meet real-time depth estimation requirements, which uses depth-separable convolutions throughout the network and network pruning to reduce inference time. However, such lightweight models sacrifice depth estimation accuracy to improve inference speed and cannot effectively balance speed and accuracy.

This paper adopts a combination of convolution and transformer structures as feature extractors for images and improves upon both CNNs and transformers to ensure both accuracy and improved inference speed. Shallow convolutional networks are employed, utilizing depth-separable convolutions¹⁶ to increase the receptive field of shallow networks. Stacking atrous convolution residual (ACR) modules achieves a shallow convolutional network with a larger receptive field. Additionally, an improved self-attention mechanism and feedforward network within a transformer are integrated to facilitate global context interaction. In the transformer component, a Multi-Dconv head transposed attention (MDTA) module is introduced to apply self-attention across feature dimensions, computing interchannel covariance to reduce computational complexity. The MDTA module emphasizes the spatial global context and leverages the complementary advantages of convolutional operations. It ensures global relationships are modelled between pixels and computes attention maps based on covariance. In the feedforward network section, inspired by Zamir et al.¹⁷, a two-step gated feedforward network (TSGFN) mechanism is proposed to suppress less informative features, focus on finer image features, and output high-quality feature maps.

Research methodology

Integration of Atrous convolution residual modules and an enhanced transformer for a monocular depth estimation network

The model in this paper consists of an encoder and a decoder, adopting the classic encoder-decoder structure, as illustrated in Fig. 1. In the encoder, convolution and transformer fusion are utilized to extract image features and multiscale features are aggregated in the encoding layer through four stages.

The input image is passed through the Conv-stem convolution module in the first stage. This module consists of two convolutional layers with kernel sizes of 3×3 and a stride of 2. The image undergoes two convolutions for downsampling and local feature extraction, resulting in feature maps with a size of $W/2 \times H/2 \times C$. To compensate for the loss of spatial information caused by changes in feature scale, this paper uses ResNet18 for initial feature extraction from the input image. The extracted features are then passed through Pose Net to output the camera's rotation matrix (R) and translation vector (t) for estimating the camera's pose. Finally, the extracted results are concatenated with an average pooling module, enabling the network to acquire more spatial information from the original image, providing a better understanding of the context and positional information of the target. Subsequently, downsampling is performed on the feature maps using a 3×3 convolution with a stride of 2, resulting in feature maps with a size of $H/4 \times W/4 \times C$. From stage two to stage four, ACR modules and local-global transpose self-attention modules are utilized in each stage to extract features of different scales. These features are concatenated with the output of the pooling module and fed into the next stage. Finally, feature maps with dimensions of $H/8 \times W/8 \times C$ and $H/16 \times W/16 \times C$ are output. In the decoder, only one convolutional layer is used to fuse features, further reducing the overall computational burden of the depth estimation network. Finally, the inverse depth maps at different resolutions are output by connecting them with bilinear upsampling and prediction heads.

Atrous convolution residual module

The encoding layer adopts a shallow CNN network for training to effectively reduce the model size and training parameter count. However, shallow CNNs have certain limitations in terms of the receptive field. The proposed ACR module is introduced to improve local feature extraction. This module utilizes depth separable instead of traditional convolutions to extract image features. Depth-separable convolution consists of depthwise convolution and pointwise convolution. The depthwise convolution extracts features along the channel dimension, while the pointwise convolution combines features along the spatial dimension. The feature extraction ability of the model improves by increasing the number of feature channels through linear modules, introducing nonlinear transformations, reducing computational costs, and capturing image information. This fully leverages the advantages of depth-separable convolution, as depicted in Fig. 2, which illustrates the ACR module.

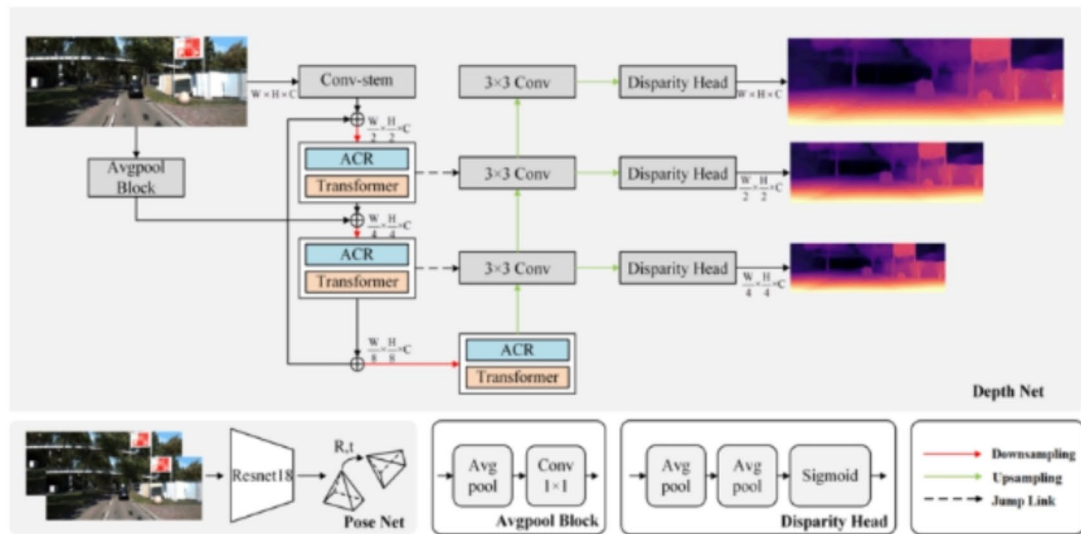


Fig. 1. Overall structure of the self-supervised monocular depth estimation network.

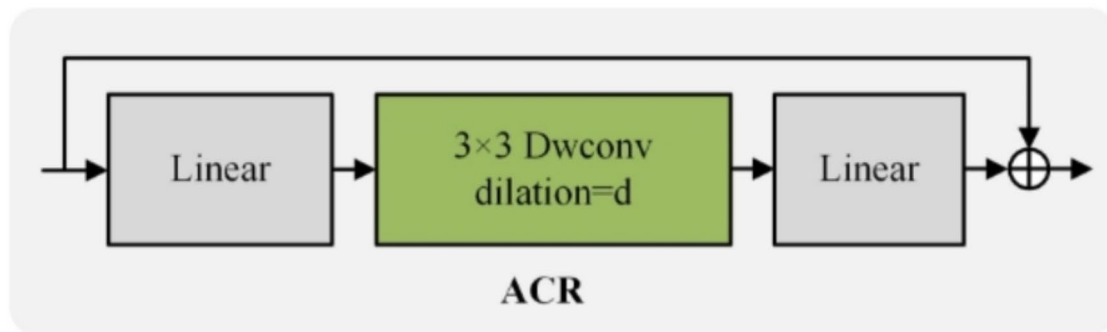


Fig. 2. The model of ACR.

Several ACR modules with different dilation rates are inserted into different stages, and the ACR modules are looped according to the stages to achieve multiscale fusion and aggregation of the local context.

The feature X with dimension $H \times W \times C$ is used as the input, and the output of the ACR module is as follows:

$$\hat{X} = X + \text{Linear}(\text{Gelu}(\text{BN}(\text{Dconv}(\text{Linear}(X)))))) \quad (1)$$

where Linear denotes a linear transformation, expanding feature channels, Dconv represents a 3×3 depthwise separable convolution with a dilation rate of d , BN denotes a batch normalization layer, Gelu denotes an activation function, and finally, the output is obtained by restoring the dimension through a fully connected layer.

Using a shallow CNN network and increasing the receptive field can better capture the global information in the image; however, it may lead to the inability to effectively capture the local fine structures in the image, which may result in the loss or neglect of detailed features. To further optimize the model performance, this paper introduces the strategy of pooling cascading¹⁸. This module is constructed by an average pooling module and 1×1 convolution, which cascades multiscale image features after each downsampling. The pooling module helps to maintain critical information while reducing dimensionality and enhances the perception of features

at different scales through multiscale fusion. By introducing the pooling cascading strategy, this paper captures local features in the image more meticulously while maintaining global information, thus further improving the model performance.

Local-global transposed transformer block

Due to the quadratic relationship between the computational complexity of self-attention and the input resolution, existing vision transformers face challenges when directly applied to visual tasks with high resolutions, such as depth estimation. This paper proposes the MDTA module to alleviate this issue, which significantly reduces the computational burden of spatial self-attention using transposed self-attention, improving the shortcomings of the original transformer architecture. The MDTA module employs a self-attention mechanism applied across channels and calculates cross-channel covariances to generate attention maps encoding the global context, thus reducing model complexity in terms of computational dimensions. As another crucial component within MDTA, a depthwise separable convolution module is introduced after the linear layer, emphasizing the local context before computing feature covariances to generate global attention maps. This helps transformer models capture relationships between input data spatial dimensions, enabling better handling of contextual information in language sequences and improving model performance. Figure 3 illustrates the specific structure of the MDTA module.

Given an input feature with dimension $H \times W \times C$, it is expanded by $N \times C$ into an image sequence, where $H \times W$ represents the image resolution, denotes the total number of pixels in the input space, and C indicates the number of image channels. Through fully connected layers and 3×3 depthwise separable convolutions, the spatial context is encoded channelwise, resulting in a query matrix $\mathbf{Q} = W_d^Q W_L^Q \mathbf{X}$, a key matrix $\mathbf{K} = W_d^K W_L^K \mathbf{X}$ and a value matrix $\mathbf{V} = W_d^V W_L^V \mathbf{X}$, each with dimensions $N \times C$, where $W_L^{(\cdot)}$ denotes the fully connected layers, and $W_d^{(\cdot)}$ represents the 3×3 depthwise separable convolutions. Therefore, the self-attention mechanism can be expressed as:

$$\hat{\mathbf{X}} = \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) + \mathbf{X} \quad (2)$$

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{V} \cdot \text{Softmax}(\mathbf{K}^T \cdot \mathbf{Q}) \quad (3)$$

where $\mathbf{X}$ and $\hat{\mathbf{X}}$ represent the input and output feature maps, respectively. Compared to the original self-attention mechanism, this paper reduces the computational complexity from $\mathcal{O}(h/N^2 + Nd)$ to $\mathcal{O}(d^2/h + Nd)$, d is the vector dimension and h is the number of attention heads.

Additionally, this paper proposes a TSGFN to achieve better contextual interactions. Similar to MDTA, the TSGFN introduces depthwise separable convolutions to encode information from spatially adjacent pixel positions. The schematic structure of the TSGFN module is illustrated in Fig. 3. In contrast to a regular multilayer perceptron (MLP) feedforward network, where the MLP updates the activation function using only the current feature and treats each feature independently, the TSGFN controls the information flow through corresponding

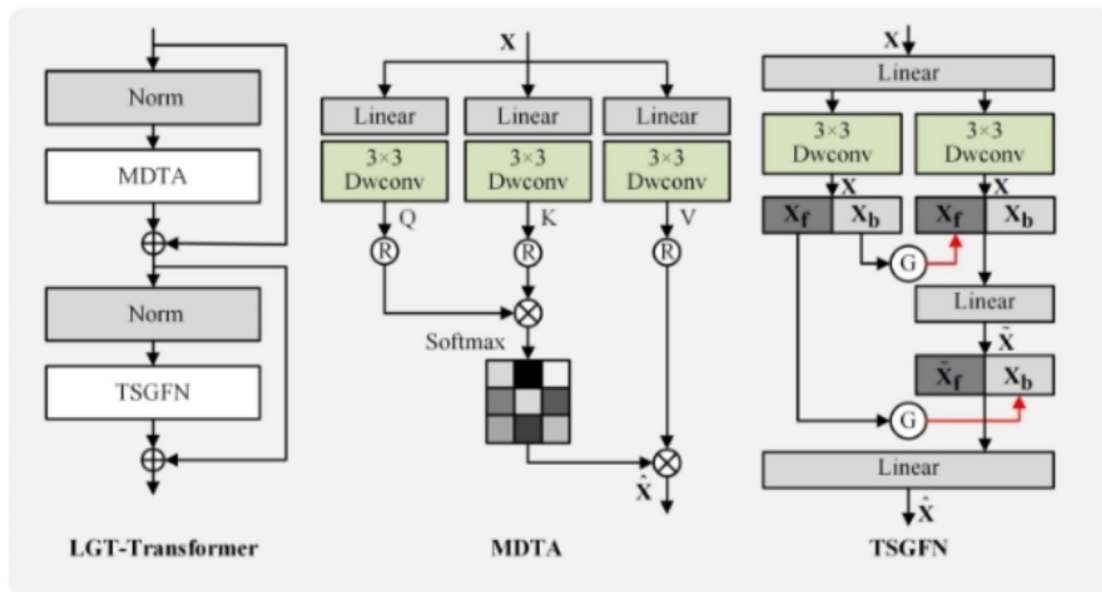


Fig. 3. Local-global transposed transformer block.

hierarchical levels in the pipeline. This allows each level to focus on fine details complementary to other levels, updating the current feature in two steps, and facilitating better contextual interactions across the entire text.

The input feature map is expanded through a fully connected layer to increase the number of feature channels. Then, image features are extracted using a 3×3 depthwise separable convolution to obtain $\mathbf{X}$. The feature map is updated in two steps. First, the features are split into two parts, $\mathbf{X}_f$ and $\mathbf{X}_b$, based on the feature channels. Part $\mathbf{X}_f$ is multiplied by $\mathbf{X}_b$ after passing through the Gaussian error linear unit (GELU) activation function, resulting in $\tilde{\mathbf{X}}$ after the first-step update. Then, the updated result $\tilde{\mathbf{X}}$ is passed through a fully connected layer. Using the same approach, the latter half $\mathbf{X}_b$ of the original input feature map is used to update the former half $\tilde{\mathbf{X}}_b$ of the current feature in the second step. Finally, the image features are restored to their original dimension and output as $\hat{\mathbf{X}}$. The TSGFN can be formulated as follows:

$$\mathbf{X} = DWConv(Linear(\mathbf{X})) \quad (4)$$

$$\tilde{\mathbf{X}} = Linear(X [Gelu(\mathbf{X}_b) \odot \mathbf{X}_f, \mathbf{X}_b]) \quad (5)$$

$$\hat{\mathbf{X}} = Linear(\mathbf{X} [\tilde{\mathbf{X}}_f, Gelu(\mathbf{X}_f) \odot \tilde{\mathbf{X}}_b]) \quad (6)$$

where $\odot$ represents elementwise matrix multiplication, and $Gelu$ denotes an activation function. Compared to the MLP, the TSGFN module utilizes feature map segmentation and elementwise multiplication operations to model contextual information through interactions between feature channels. Elementwise multiplication can selectively emphasize or suppress feature responses between different channels, enabling the capture of richer feature representations.

Loss function

This paper utilizes the difference between the source and predicted images as the signal for supervising model training. Therefore, a loss function is designed based on the difference between the two, constraining network training through optical reprojection and edge-aware smoothness loss. By utilizing the intrinsic camera function K and the predicted pose P between two adjacent views, a reconstruction target image $\hat{I}$ is obtained as a function π of the intrinsic function, pose, source image I_s and depth D_t . The loss signal $\mathcal{L}_{ss}$ is calculated as a function $\mathcal{F}$ of inputs $\hat{I}$ and I :

$$\mathcal{L}_{ss}(\hat{I}, I) = \mathcal{F}(\pi(I_s, P, D_t, K), I) \quad (7)$$

The function $\mathcal{F}$ is typically obtained as a weighted sum between the structural similarity item and the intensity difference item, calculated as the sum of the pixelwise structural similarity (SSIM) and the L1 loss between $\hat{I}$ and I :

$$\mathcal{F}(\hat{I}, I) = \frac{\alpha}{2} (1 - SSIM(\hat{I}, I)) + (1 - \alpha) \|\hat{I} - I\| \quad (8)$$

where α is typically set to 0.85. The minimum luminance loss is computed to handle out-of-view pixels in the source image and occluded objects:

$$\mathcal{L}(p) = \min_{i \in [1, -1]} \mathcal{F}(\hat{I}_i(p), I(p)) \quad (9)$$

Smoothness loss and edge-aware smoothness loss are used to improve the inverse depth map d :

$$\mathcal{L}_{smooth} = |\partial_x d_t^*| e^{-|\partial_x I_t|} + |\partial_x d_t^*| e^{-|\partial_y I_t|} \quad (10)$$

Finally, both the view reconstruction loss $\mathcal{L}_{ss}$ and the smoothness loss $\mathcal{L}_{smooth}$ are computed from the output at each scale $S \in \{1, \frac{1}{2}, \frac{1}{4}\}$ to achieve full resolution and then averaged to train the network as $\mathcal{L}_{tot}$:

$$\mathcal{L}_{tot} = \frac{1}{3} \sum_{s=1}^3 (\mu \mathcal{L}_{ss} + \lambda \mathcal{L}_{smooth}) \quad (11)$$

where $\mathcal{L}_{ss}$ ensures the similarity between the generated depth map and the original image by measuring the difference between the output result and the original features. $\mathcal{L}_{smooth}$ is used to further smooth the image, preventing the impact of excessive detail noise and irregularities, thereby improving the signal-to-noise ratio. The weights $\mu, \lambda \in [0, 1]$ adjust the balance between the reconstruction loss and the smoothing loss in different models. These two types of losses have the same effect on street scene images, so the weight factors are set to 0.5.

Experimental dataset and evaluation metrics

This section evaluates the algorithm's accuracy on the KITTI dataset¹⁹ and its generalizability on the Make 3D dataset⁸ to validate the effectiveness of the proposed method and its generalizability across different scenarios. Simultaneously, ablation experiments are conducted on various components proposed in this paper to verify the contributions of each module to the overall network.

Experimental dataset and parameter configurations

The KITTI dataset comprises 61 road scenes for autonomous driving and robotics research, captured using devices mounted on vehicles equipped with LiDAR sensors. This paper employs image segmentation as in Eigen et al.⁹ to train and evaluate the proposed method, where there are 39,180 images for training, 4424 for validation, and 697 for testing. All images used the same camera intelligence, with the camera principal point set to the image centre and the focal length set to the average of all focal lengths in KITTI. During testing, the predicted depths are constrained to the range of [0, 80] metres.

The Make 3D dataset is a monocular depth estimation dataset consisting of 534 pairs of RGB-D images, with 400 pairs used for training and 134 pairs used for testing. Towards the end of this paper, the model trained on the KITTI dataset is evaluated on 134 test images to assess the generalizability of the algorithm across various outdoor scenes.

The model in this paper is implemented on the PyTorch deep learning platform, utilizing Adam as the model optimizer, setting the number of epochs to 30, the batch size to 12, and the input image resolution to 640 × 192. The initial learning rate is set to 0.0001, with a 10x decay after the first 15 epochs, and the weight decay is set to 0.0001. Initializing the network with parameters trained on the ImageNet dataset enables rapid convergence. The experiments were conducted on an NVIDIA RTX6000 GPU with 24 GB of memory, and the training lasted for 18 h, achieving the highest accuracy on the network at the 29th epoch.

Evaluation metrics

The training results on the KITTI dataset are quantitatively analysed using the evaluation metrics proposed by Eigen et al.⁹ to validate the effectiveness of the model proposed in this paper. The primary accuracy metrics are as follows:

Absolute relative error:

$$AbsRel = \frac{1}{N} \sum_{y \in N} \frac{|y - y^*|}{y^*} \quad (12)$$

Squared relative error:

$$SqRel = \frac{1}{N} \sum_{y \in N} \frac{(y - y^*)^2}{y^*} \quad (13)$$

Root mean square error:

$$RMSE = \sqrt{\frac{1}{N} \sum_{y \in N} (y - y^*)^2} \quad (14)$$

Log root mean square error:

$$RMSE_{\log} = \sqrt{\frac{1}{N} \sum_{y \in N} (\log y - \log y^*)^2} \quad (15)$$

where y represents the depth predicted by the depth network, y^* represents the ground-truth depth, and N denotes the total number of pixels with available ground-truth depth.

Experimental results and comparative analysis

Experimental results on the KITTI dataset

The model is evaluated on the KITTI dataset and compared with mainstream algorithms to validate the effectiveness of the algorithm proposed in this paper. The comparative experimental results are shown in Table 1. The first four metrics are errors, the middle three metrics indicate the model accuracy at different thresholds, and the last metric represents the model parameter count, which is one of the indicators for evaluating how lightweight the model is.

Comparing the algorithms in Table 1, it is evident that the algorithm proposed in this paper has the least parameter count (3.0 M). Compared to Monodepth2 with a ResNet-18 backbone, the RMSE decreases by 7%, the accuracy improves by 16% on the < 1.25 metric, and the parameter count decreases by 79%. Compared to the lightweight FastDepth network, which has a faster inference speed, the proposed algorithm performs better across all the metrics. Compared to recent lightweight networks such as R-MSFM6, Lite-HR-Depth, and Lite-Mono, the proposed algorithm falls only slightly behind Lite-Mono in the Sq Rel error metric but outperforms in accuracy and other error metrics. Compared to the latest MonoFormer, the proposed algorithm performs better in all the metrics except for Abs Rel while reducing the model parameter count by 87%. In summary, the algorithm proposed in this paper can control the model parameter count within a low range while ensuring network accuracy without producing significantly greater errors.

Several lightweight algorithms are subjected to visual analysis to evaluate the advantages of the algorithm proposed in this paper. Figure 4 illustrates the visualization results on the KITTI dataset, showing the clearer boundary information extraction capability of the proposed algorithm compared to other lightweight networks. In the fourth experiment group, Monodepth2 exhibits blurred boundary extraction for road signs, failing to define

Methods	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Params
GoNet[20]	0.149	1.060	5.567	0.226	0.796	0.935	0.975	31.6 M
DDVO[21]	0.151	1.257	5.583	0.228	0.810	0.936	0.974	28.1 M
Monodepth[11]	0.148	1.344	5.927	0.247	0.803	0.922	0.964	20.2 M
Monodepth2[12]	0.115	0.903	4.863	0.193	0.877	0.959	0.981	14.3 M
FastDepth[15]	0.150	0.890	5.321	0.207	0.808	0.945	0.981	3.96 M
R-MSFM6[22]	0.112	0.806	4.704	0.191	0.878	0.960	0.981	3.5 M
Lite-HR-Depth[23]	0.116	0.845	4.841	0.190	0.866	0.957	0.982	3.1 M
MonoFormer[24]	0.104	0.846	4.580	0.183	0.891	0.962	0.982	23.9 M+
Lite-Mono[25]	0.107	0.765	4.561	0.183	0.886	0.963	0.983	3.1 M
Ours	0.105	0.810	4.506	0.183	0.891	0.963	0.983	3.0 M

Table 1. Depth accuracy and error of different methods on the KITTI dataset. Significant values are in bold.

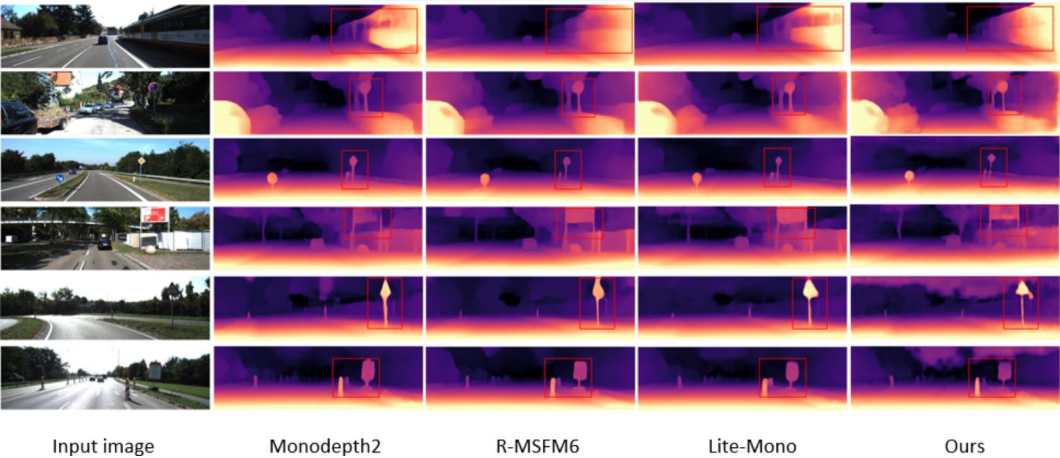


Fig. 4. Visualization results on the KITTI dataset.

their boundaries clearly, whereas our algorithm is able to extract the boundaries of road signs more accurately, showing superior boundary clarity. In the fifth experiment group, other algorithms failed to identify distant signs, while the proposed algorithm identified both signs relatively well, demonstrating better performance in recognizing distant objects with higher accuracy. Moreover, in the sixth experiment group, other algorithms confused the boundary information of the ground landmarks, while the proposed algorithm provided more accurate boundary information for each separately. This is attributed to the ability of the proposed algorithm to capture both local geometry and global semantic information in the environment through the local-global transposed transformer module, controlling the flow of effective information. This experiment demonstrates that the proposed algorithm can provide more accurate depth information for small but important objects in autonomous driving scenarios, such as signs, traffic lights, and barriers, thus exhibiting superior monocular depth estimation performance with significant implications for autonomous driving and related areas.

Generalization validation results on the make 3D dataset

In this paper, generalization experiments are conducted on the Make 3D dataset to evaluate the ability of the model to generalize to different real-world scenarios. Initially trained on the KITTI dataset using input images with a resolution of, the model is directly evaluated on the Make 3D dataset without any modifications following the evaluation method provided in Monodepth2. Figure 5 presents the visualization results of various algorithms on the Make 3D dataset. Despite no further training on this dataset, the proposed algorithm maintains relatively accurate depth extraction precision, with good detail features observed at the boundaries of some objects. Table 2 compares the generalization results between the proposed algorithm and other depth estimation algorithms. The experimental results indicate that compared to Monodepth2, the proposed algorithm achieves a decrease of 0.5 in RMSE and a decrease of 0.295 compared to R-MSFM6. Additionally, the proposed algorithm exhibits the lowest errors across all other error metrics. A quantitative and qualitative analysis of experimental results shows that the proposed algorithm can perform well on different outdoor datasets, demonstrating its strong generalizability.

Model complexity evaluation

In this paper, experiments were conducted on a Raspberry Pi 3 Model B, an edge device Syncbot SBC-T800, and an Nvidia Titan X. The test images used were from the KITTI dataset with a resolution of 640×192. The

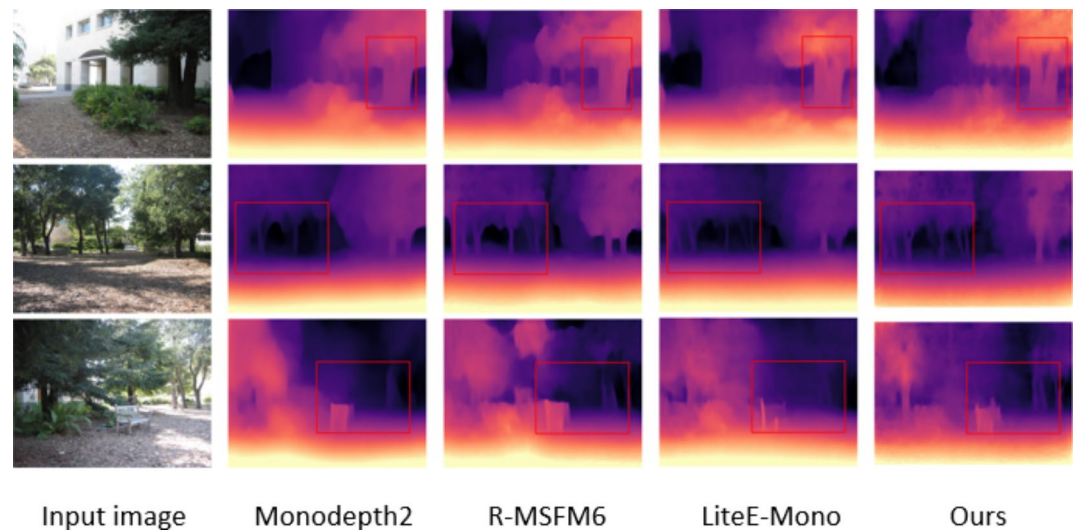


Fig. 5. Visualization results on the make 3D dataset.

Methods	Abs Rel	Sq Rel	RMSE	RMSE log
Monodepth	0.544	10.94	11.760	0.193
Monodepth2	0.322	3.589	7.417	0.163
R-MSFM6	0.334	3.285	7.212	0.169
Lite-Mono	0.305	3.060	6.981	0.158
Ours	0.295	3.042	6.917	0.150

Table 2. Evaluation results on the make 3D dataset. Significant values are in bold.

model speed and complexity were evaluated based on parameters, frames per second (FPS), and floating point operations (FLOPs). Figure 6 illustrates the inference speed of different algorithms on three different devices. The algorithms positioned towards the top-right corner exhibit higher efficiency and accuracy, highlighting the advantages of the proposed algorithm. None of the algorithms achieved real-time depth estimation due to the lower configuration and absence of GPU deployment on the Raspberry Pi 3 Model B. However, when deployed on the edge device SBC-T800 without a GPU, the proposed algorithm achieved real-time performance. Furthermore, it maintained high inference speed on more advanced GPUs while demonstrating good performance in terms of estimation accuracy and error metrics.

Table 3 presents the quantitative analysis results of the complexity and inference speed of the proposed algorithm compared to those of other algorithms. The proposed algorithm significantly outperforms FastDepth, which is based on a lightweight CNN architecture and exhibits higher real-time performance, in depth estimation accuracy and RMSE metrics. While FastDepth achieves higher real-time performance through pruning, the proposed algorithm maintains superior accuracy. Monodepth2 and the proposed algorithm share similar real-time performance; however, Monodepth2 has twice the number of FLOPs and 5.1 times the number of parameters of the proposed algorithm, indicating higher model complexity. Despite its smaller parameter count, R-MSFM6 has the slowest inference speed in all experiments, lacking real-time performance, especially on edge devices. Compared to Lite-Mono, which shares a similar CNN-Transformer fusion architecture, the proposed algorithm achieves fewer parameters through depth-separable convolutions and exhibits higher accuracy and stronger inference capabilities by improving the transformer component.

In summary, the proposed algorithm achieves high accuracy while maintaining lower model complexity and faster inference speed. It is deployable on edge devices and meets real-time requirements.

Ablation experiments

Ablation experiments were conducted on each module proposed in this paper to understand better the effect of the proposed modules on network performance. All experiments were conducted on the KITTI dataset, with the image resolution set to 640×192. The experimental results are shown in Table 4.

Depth-separable convolution: In this paper, depth-separable convolutions are employed in the encoding layers to extract features. In the ablation experiment, the dilation rate was set to 1, and the subsequent pointwise convolution was removed to convert it into a regular convolution. The visualized result for Model A is shown in Fig. 7. The network parameter count decreased by 0.165 M, and the inference speed of the model decreased by 3 FPS. This ablation experiment resulted in the largest decrease in accuracy. The reason lies in the significance of the ACR module in expanding the receptive field of shallow networks, which is crucial for improving local

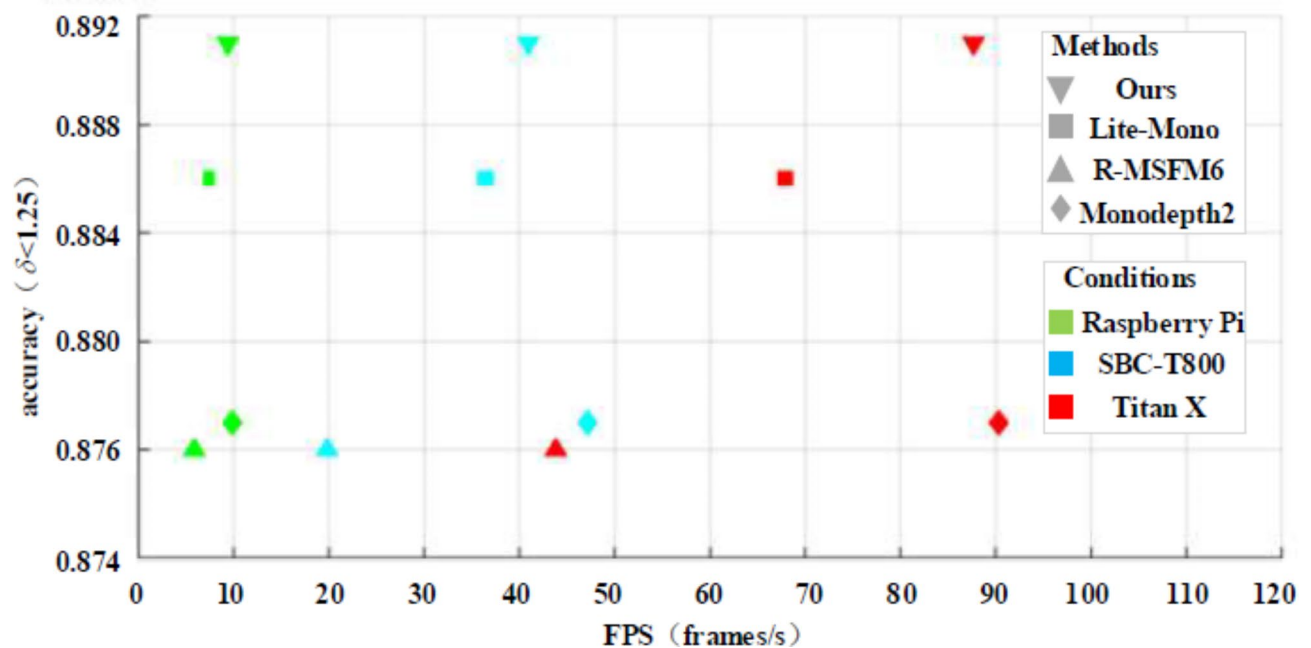


Fig. 6. The running speed of algorithms on different devices.

Methods	Params (M)	FLOPs (G)	RMSE	$\delta < 1.25$	FPS		
					Raspberry Pi	SBC-T800	Titan X FPS
FastDepth	3.96	3.6	5.321	0.808	11.39	57.68	116.20
Monodepth2	14.3	8.3	5.927	0.877	9.96	47.16	90.37
R-MSFM6	3.8	31.2	4.704	0.876	5.87	19.87	43.84
Lite-Mono	3.1	5.1	4.561	0.886	7.66	35.45	67.92
Ours	3.0	4.6	4.506	0.891	8.79	40.95	87.71

Table 3. Model complexity and speed evaluation of different methods. Significant values are in bold.

Model	Dconv	LGTF	TSGFN	Abs Rel	Sq Rel	RMSE	$\delta < 1.25$	Params (M)	FPS
Ours	✓	✓	✓	0.105	0.810	4.506	0.891	3.041	87.71
A	×	✓	✓	0.112	0.873	4.704	0.879	2.876	84.59
B	✓	×	✓	0.114	0.853	4.856	0.882	2.835	97.36
C	✓	✓	×	0.112	0.842	4.783	0.883	2.990	78.25

Table 4. Ablation experiment results. Significant values are in bold.

feature extraction. Image features cannot be effectively extracted by relying only on shallow CNNs, leading to a decrease in model accuracy.

Transformer Module: In the ablation experiment results, when the transformer module was removed, the model's parameter count decreased by 0.206 M, and the inference speed increased by 5 FPS. However, model accuracy significantly decreased. Additionally, the model's ability to extract object boundaries also decreased, as shown in Model B in Fig. 7. The LGTF module provides global context information interaction for the entire network, improving the inherent locality of feature extraction in CNN networks. When removed, the model relies solely on shallow networks for global information extraction, which has limitations in capturing global information, leading to a decrease in model accuracy.

TSGFN Module: To further verify the role of the two-step gated feedforward network module in the transformer, an ablation experiment was conducted by replacing this module with a MLP feedforward neural network. After the replacement, the parameter count and inference speed remained largely unchanged; however, there was a slight decrease in model accuracy. This module in the transformer controls the feature responses between different channels, selectively emphasizing or suppressing image features. When this part of the

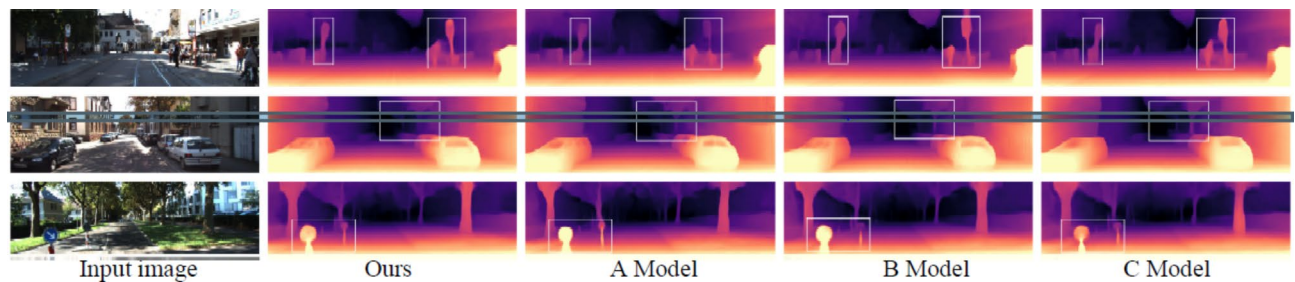


Fig. 7. Visualization results of the ablation experiment.

module is replaced, the flow of information in the network is not affected by the gating mechanism, leading to uncontrolled noise information in the network, resulting in increased model error.

Conclusion

The inference speed and estimation accuracy of monocular depth estimation networks are crucial metrics for evaluating algorithms. However, there must be a balance between inference speed and estimation accuracy. A lightweight self-supervised monocular depth estimation method is proposed by integrating an improved transformer to address the issues of large model parameters and low computational efficiency in existing monocular depth estimation networks. This paper validates the effectiveness of the proposed method on two outdoor datasets, namely, the KITTI and Make 3D datasets, and conducts model complexity and real-time verification on three types of devices. The experimental results demonstrate that the proposed method has a model parameter count of only 3.0 M. On the KITTI dataset, it achieves an accuracy of 89.1% on the < 1.25 metric and performs optimally on the Make 3D dataset, indicating good generalizability. Visualization results from both datasets illustrate the ability of the network to effectively extract boundary information of objects such as signs and obstacles in the scene. With an inference speed of 87 FPS on the Nvidia Titan X, real-world experiments on edge devices confirm that the proposed algorithm can be deployed on edge devices, meeting real-time requirements. Compared to other lightweight networks, this proposed network achieves a better balance between accuracy and network inference speed. Through experiments, it was demonstrated that the proposed algorithm can maintain high accuracy while improving computational efficiency, realizing a lightweight self-supervised monocular depth estimation approach.

Data availability

The datasets analysed during the current study are available in the KITTI dataset and Make 3D dataset, [https://www.cvlibs.net/datasets/kitti] [http://make3d.cs.cornell.edu/data.html].

Received: 15 July 2024; Accepted: 10 September 2024

Published online: 28 September 2024

References

- Hou, H., Lan, C. & Xu, Q. UAV absolute positioning method based on global and local deep learning feature retrieval from satellite image. *J. Geo-Inf. Sci.* **25**(5), 1064–1074. <https://doi.org/10.12082/dqxxkx.2023.220827> (2023).
- Jin, S., Li, X., Yang, F. & Zhang, W. 3D object detection in road scenes by pseudo-LiDAR point cloud augmentation. *J. Image Graph* **28**(11), 3520–3535. <https://doi.org/10.11834/jig> (2023).
- Wang, S., Fang, L., Chen, C. & Huang, M. A single view type 3D reconstructive method for architecture based on structured scene. *J. Geo-Inf. Sci.* **18**(8), 1022–1029. <https://doi.org/10.3724/SP.J.1047.2016.01022> (2016).
- Hu, X., Zhou, Y., Lan, C., Huang, G. & Zhao, L. Virtual real registration assisted by structural semantic constraint for digital city scene. *J. Geo-Inf. Sci.* **25**(5), 883–895. <https://doi.org/10.12082/dqxxkx.2023.220544> (2023).
- Zhang, Y., Wu, Y. & Chen, H. Research progress on simultaneous visual localization and mapping based on deep learning. *J. Instrum.* **44**(7), 214–241. <https://doi.org/10.19650/j.cnki.cjsi.J2311081> (2023).
- Liu, Y., Liu, H., Li, Y., Zhao, S. & Yang, Y. Building BIM modeling based on multi-source laser point cloud fusion. *J. Geo-Inf. Sci.* **23**(5), 763–772. <https://doi.org/10.12082/dqxxkx.2021.200378> (2021).
- Zhu, Q. & Wang, H. Stereo matching algorithm for occlusion recovery using image segmentation. *J. Huazhong Univ. Sci. Technol. (Nat. Sci. Ed.)* **38**(1), 81–84. <https://doi.org/10.13245/j.hust.2010.01.024> (2010).
- Saxena, A., Sun, M. & Ng, A. Make3D: Learning 3D scene structure from a single still image. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(5), 824–840. <https://doi.org/10.1109/TPAMI.2008.132> (2009).
- Eigen, D., Puhrsch, C. & Fergus, R. Depth map prediction from a single image using a multi-scale deep network, presented at the 28th NIPS, Montreal, Canada, 6–11 (2014).
- Garg, R., Bg, V., Carneiro, G. & Reid, I. Unsupervised CNN for single view depth estimation: Geometry to the rescue, presented at the 14th Computer Vision–ECCV, Amsterdam, Netherlands, 11–14 (2016).
- Godard, C., Mac Aodha, O. & Brostow, G. Unsupervised monocular depth estimation with left-right consistency, presented at the 15th CVPR, Hawaii, United States, 21–26 (2017).
- Godard, C., Mac Aodha, O., Firman, M. & Brostow, G. Digging into self-supervised monocular depth estimation, presented at the 18th ICCV, Seoul, South Korea, 27–2 (2019).
- Vaswani, A. et al. Attention is all you need, presented at the 31th NIPS, Long Beach, California, United States, 4–9 (2017).
- Zhao, C. et al. Monovit: Self-supervised monocular depth estimation with a vision transformer, presented at the 10th 3DV, Prague, Czech Republic, 12–16 (2022).

15. Wofk, D., Ma, F., Yang, T., Karaman, S. & Sze, V. Fastdepth: Fast monocular depth estimation on embedded systems, presented at the 35th ICRA, Montpellier, France, 20–24 (2019).
16. Cheng, R., Yang, Y., Li, L., Wang, Y. & Wang, J. Classification of hyperspectral images using lightweight residual networks based on depth wise separable convolution. *Acta Opt. Sin.* **43**(12), 311–320. <https://doi.org/10.3788/AOS221848> (2023).
17. Zamir, S. et al. Restormer: Efficient transformer for high-resolution image restoration, presented at the 35th CVPR, Hawaii, United States, 26–1 (2022).
18. Wang, C. & Chen, Y. Self-supervised monocular depth estimation based on full-scale feature fusion. *J. Comput. Aided Des. Comput. Graph* **35**(5), 667–675. <https://doi.org/10.3724/SP.J.1089.2023.19418> (2023).
19. Geiger, A., Lenz, P. & Stiller, C. Vision meets robotics: The KITTI Dataset. *IJRR* **32**(11), 1231–1237. <https://doi.org/10.1177/027836491349> (2013).
20. Yin, Z. & Shi, J. Geonet Unsupervised learning of dense depth, optical flow and camera pose, presented at the 31th CVPR, Salt Lake City, Utah, United States, 18–22 (2018).
21. Wang, C., Buenaposada, J., Zhu, R. & Lucey, S. Learning depth from monocular videos using direct methods, presented at the 31th CVPR, Salt Lake City, Utah, United States, 18–22 (2018).
22. Zhou, Z., Fan, X., Shi, P. & Xin, Y. R-MSFM: Recurrent multi-scale feature modulation for monocular depth estimating, presented at the 19th ICCV, Montreal, Canada, 11–17 (2021).
23. Lyu, X., Liu, L., Wang, M., Kong, X. & Yuan, Y. HR-Depth: High resolution self-supervised monocular depth estimation, presented at the 35th AAAI, Online 2–9 (2021).
24. Bae, J., Moon, S. & Im, S. Deep digging into the generalization of self-supervised monocular depth estimation, presented at the 37th AAAI, Honolulu, Hawaii, United States, 7–14 (2023).
25. Zhang, N., Nex, F., Vosselman, G. & Kerle, N. Lite-mono: A lightweight CNN and transformer architecture for self-supervised monocular depth estimation, presented at the 76th CVPR, Vancouver, Canada, 18–22 (2023).

Author contributions

Conceptualization, S.G. and A.X.; methodology, S.G. and X.S.; software, S.G.; validation, S.G., C.Z. and Z.S.; formal analysis, S.G. and C.W.; investigation, S.G.; data curation, S.G., C.Z. and Z.S.; writing-original draft preparation, S.G.; writing-review and editing, S.G., X.S., A.X., C.Z., Z.X. and C.W.; funding acquisition, A.X. All authors have read and agreed to the published version of the manuscript.

Funding

This research was funded in part by the National Natural Science Foundation of China (No. 42074012). This research was funded in part by the Liaoning Revitalization Talents Program (Nos. XLYC2002101 and XLYC2008034).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to S.G.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2024