Multi-Scale Depth Guidance Transformer for Monocular Depth Estimation

Canbin Li[1] **and Yiguang Liu**[1](✉)

Abstract Recently, works based on transformer networks show great progress in monocular depth estimation task, but they often overlook the invaluable invariances and priors in the scene space, thus leading to the loss of details in depth estimation. To tackle this problem, a transformer architecture is proposed with a depth guidance decoder, which utilizes multi-scale depth guidance layers in the decoding phase. Compared to existing methods, our proposed depth guidance layers introduce the regularity of the scene into the network and significantly improve the estimation accuracy. A large number of experiments and ablation study show that our proposed method achieves the state-of-the-art results on challenging benchmarks and can converge faster than other architectures.

Keywords Machine Learning, Transformer, Depth Estimation, Monocular Depth Estimation

1 Introduction

Over the last decade, neural networks have led to significant advancements in 3D computer vision field, such as multi-view stereo [1], novel view synthesis [2], visual simultaneous localization and mapping [3], etc. Among several 3D vision tasks, one of the challenging tasks to solve is the monocular depth estimation problem. This is a problem of estimating a high quality dense depth map from a single RGB input image. It is a classical problem in computer vision that is essential for numerous computer vision applications [4][5][6][7]. At the same time, it is an ill-posed problem. Given an image, there are infinite possible world scenes may have produced it. Of course, most of these are physically impossible for real-world spaces, and thus the depth may still be predicted with respectable accuracy. However, a good solution to monocular depth estimation is highly desirable in robotics, self-driving vehicles or computer vision fields performed in 3D pace[9].

1 Sichuan University, No.24 South Section 1, Yihuan Road, Chengdu, 610065, China. E-mail: C. Li, licanbin1@stu.scu.edu.cn; Y. Liu, liuyg@scu.edu.cn(✉)

(a) Input RGB (b) Adabins[8] (c) Ours

Fig. 1 Illustration of our work: **Left**: input RGB images. **Middle**: depth predicted by Adabins[8]. **Right**: depth predicted by our proposed mulit-scale depth guidance transformer. Note that the predicted depth in our work can present more details.

In recent years, significant advancements have been made in machine learning based depth estimation methods. Eigen *et al.*[10] introduced a multi-scale deep network that employs a global network for coarse depth prediction and a local network for refining depth prediction.

Motivated by the work of Eigen *et al.*[10], convolution networks have been widely utilized for depth estimation. For instance, CLIFFNet[11] employs a multi-scale fusion convolutional framework to generate high-quality depth estimation results. More recently, Transformer networks have garnered considerable attention in the computer vision field. Building on the success of recent applications of Transformers to solve computer vision problems, DPT[12] propose to replace convolution operations with Transformer layers, leading to further improvements in network performance.

While the methods mentioned above have significantly enhanced depth prediction accuracy, they often suffer from slow convergence due to the treatment of monocular depth estimation as a regression task. Another line of research introduces the discretization of continuous depth into multiple intervals, framing deep network learning as a per-pixel classification problem. For instance, DORN[13] introduces an efficient depth estimation loss for ordinal classification and incorporates the ASPP[14] module to extract multi-level information. Building on this work, Diaz[15] softens the classification target during training, leading to performance

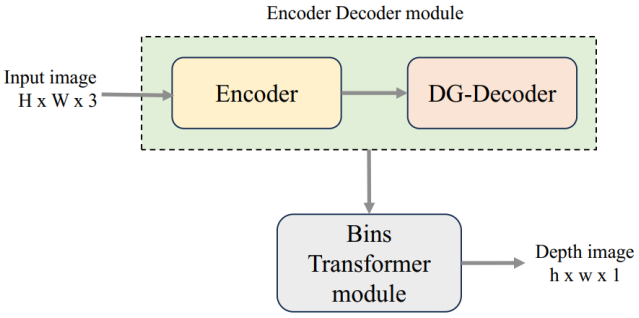


Fig. 2 Overview of our proposed network architecture. Our architecture consists of three major components: an EfficientNet B5[16] encoder block, our proposed depth guidance decoder block(DG-Decoder), a transformer block based on vision transformer.The input to our network is an RGB image of size $H \times W \times 3$, and the output is a $h \times w \times 1$ depth image.

improvement. Additionally, some approaches [17] reframe the problem as classification regression to alleviate the visual artifacts associated with the discretization of depth values, especially at sharp depth discontinuities. In pursuit of further improving model performance, Adabins [8] introduces an adaptive bin strategy, a critical component for accurate depth estimation. However, methods relying on bin strategies often overlook the invaluable real-world geometric cues, such as surface normal constraints or planar relationships, which is essential in monocular depth estimation problems.

2 RELATED WORK

In monocular depth estimation, supervised methods take a single RGB image and use depth data measured by range sensors such as RGB-D cameras or multi-channel laser scanners as the ground truth for training. Eigen et al.[10] introduced a convolutional architecture that initially learns coarse global depth predictions in one part of the network and progressively refines them using another part of the network. In contrast to earlier single-image depth estimation approaches, their network learns representations directly from raw pixels, eliminating the need for hand-crafted features. Building on the success of this approach, many CNN-based methods have treated monocular depth estimation as a regression task, aiming to predict dense depth maps from a single RGB image [10][18][19]. In our work, we utilize multi-scale deep guidance layers to learn dense features at different scales through supervised learning.

Recently, transformer networks are gaining much attention as a viable building block outside of their traditional use in NLP tasks and are incorporated into computer vision tasks[20][21][22]. We propose to leverage two Transformer encoders as a building block for recovering the decoder feature. It leverages the idea of transforming the depth prediction problem into a bins classification problem.

Encoder-decoder networks have played a pivotal role in addressing various vision-related challenges, including image segmentation [1], optical flow estimation[23], and image restoration [24]. In our work, we employ an encoder-decoder architecture to extract depth information using our depth guidance layers and estimate depth through a transformer network.

In this paper, to introduce geometry guidance to neural network, we propose a network architecture that utilizes novel multi-scale depth guidance layers in the decoding phase. We let the multi-scale depth guidance layers to learn 6-dimensional plane coefficients and use them together to reconstruct depth decoder feature in the decoder for the transformer to estimate final depth. As a consequence of multi-scale layers combination, individual spatial cells in each resolution are distinctively trained while the training progress. And they can concatenate together to compensate each other by different scale feature. Experiments on challenging datasets demonstrate that our proposed method achieves the state-of-the-art results.

3 PROPOSED METHOD

Fig. 2 shows an overview of our proposed depth estimating architecture. Our architecture consists of three major components: 1) an encoder block built on a pretrained EfficientNet B5 [16] encoder. 2) our proposed decoder block consists of multi-scale depth guidance layers. 3) a transformer architecture use two transformer encoders. The main contribution of our work is the multi-scale depth guidance layers in the decoder, which introduce the invaluable geometry invariances and priors to our neural network architecture.

3.1 Motivation

Monocular depth estimation involves the task of learning a dense mapping, denoted as $f_\theta : I(u, v) \longrightarrow D(u, v)$, where I represents the input image with dimensions $H \times W$, D corresponds to the depth map of the same resolution, and (u, v) represent pixel coordinates within the image space. The parameter set θ defines the mapping function f. In supervised learning, each input image I in the training set is associated with a ground-truth depth map D^*. During the training process, the parameters θ are optimized to minimize the loss between the predicted depth and the ground-truth depth across the entire training dataset Γ. This optimization process can be formally expressed as follows:

$$\min_{\theta} \sum_{(I,D^*)\in\Gamma} \mathcal{L}(f_\theta(I), D^*) \tag{1}$$

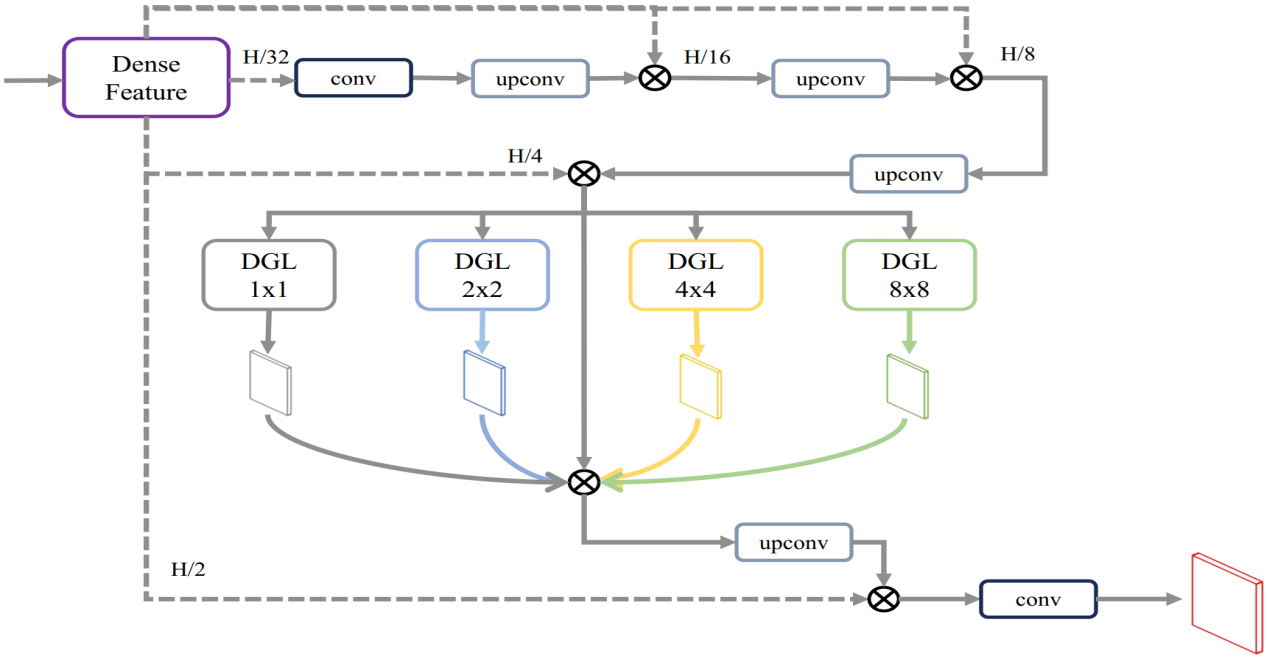


Fig. 3 Overview of the proposed depth guidance decoder. The decoder is composed of dense feature extractor (extract the encoder features), depth guidance layers and their dense connection for final depth decoder features estimation. Note that the outputs from multiple scale depth guidance layers have the same resolution H/4 but use different scale depth guidance. We also use skip connections from dense feature extractor to link with internal outputs in the decoding phase with corresponding spatial resolutions.

where $\mathcal{L}$ is a loss function that penalizes deviations between the prediction and the ground truth. Additionally, with a depth map D and knowledge of the camera intrinsics, we can perform the process of backprojection for each pixel into 3D space. Utilizing the pinhole camera model and provided parameters such as the focal lengths (f_x, f_y) and the principal point (u_0, v_0), we can map each pixel $p = (u, v)^T$ to a corresponding 3D point $P = (X, Y, Z)^T$ using the following transformation:

$$X = \frac{Z(u - u_0)}{f_x}, Y = \frac{Z(v - v_0)}{f_y}, Z = D(u, v) \quad (2)$$

Recent research has focused extensively on designing powerful neural networks, often overlooking the valuable geometric guidance inherent in spatial scenes. In our work, we incorporate this geometry guidance into our network as follows.Suppose we have a backprojected 3D point P that corresponds to a planar component of the 3D scene. The equation of the associated plane in point-normal form is expressed as $\mathrm{n} \cdot \mathrm{P} + d = 0$, where $\mathrm{n} = (a, b, c)^T$ represents the normal vector to the plane, and $-d$ denotes the plane's distance from the origin. Substituting P from Eq. (2) into the point-normal equation yields:

$$\frac{1}{\mathrm{Z}} = \frac{-a}{f_x d}u + \frac{-b}{f_y d}v + \frac{1}{d}\left(\frac{a}{f_x}u_0 + \frac{b}{f_y}v_0 - c\right) \quad (3)$$

This equation reveals that, for image regions depicting planar 3D surfaces, the inverse depth is an affine function of pixel position. The coefficients in this function encode both the camera intrinsic parameters and the properties of the 3D plane. We reformulate Eq. (3) as:

$$\mathrm{Z} = \left(\frac{-a}{f_x d}u + \frac{-b}{f_y d}v + \frac{1}{d}\left(\frac{a}{f_x}u_0 + \frac{b}{f_y}v_0 - c\right)\right)^{-1} \quad (4)$$

Based on this assumption, we employ a convolutional neural network to learn the parameters in Eq. (4), which enhances the effectiveness of our neural network.

3.2 Multi-Scale Depth Guidance Decoder

The central concept of our work lies in efficiently defining the relationship between internal features and decoder output. Diverging from conventional methods that rely on simple nearest-neighbor upsampling layers and skip connections during the decoding stage to restore the original resolution, we introduce novel multi-scale depth guidance layers. These layers leverage geometric prior assumptions to extract features effectively, enabling us to achieve the desired resolution and obtain the decoder features. Our decoder architecture with multi-scale depth-guided layers can be seen in Fig.3. Given a feature map having spatial resolution H/4, our proposed layers estimate for each spatial cell the coefficients that fit a locally defined $k \times k (k \in \{1, 2, 4, 8\})$ patch on the resolution H/4, then they are concatenated together and upsampled for

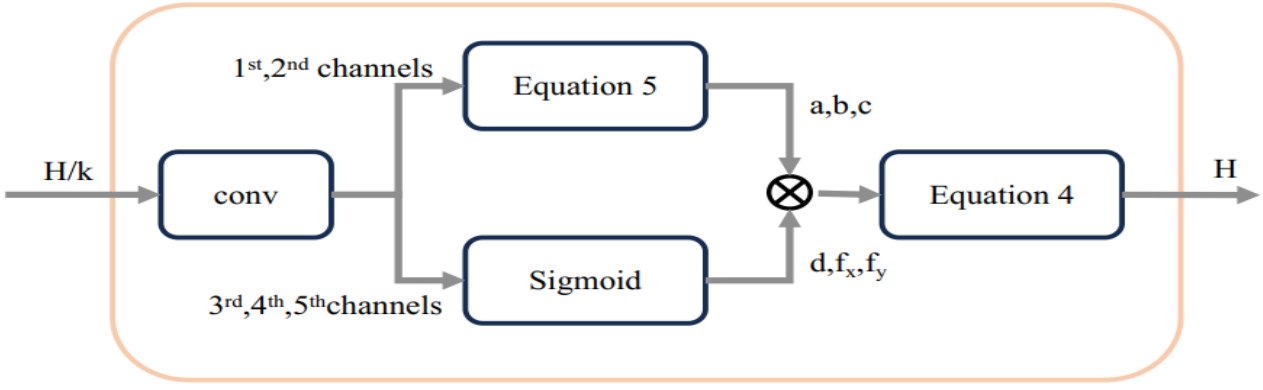


Fig. 4 The depth guidance layer. We use a stack of 1 × 1 convolutions to get the 6D coefficients estimations. (i.e., H/k×H/k×6). Then the channels are split to pass through two different activation mechanisms to ensure coefficients' constraint. Finally, they are fed into the depth guidance module to compute depth estimation features.

the estimation of the final decoder features.

With the geometry guidance as Eq. (4), we propose our depth guidance layer as Fig.4. We regard the first two channels of the encoded feature map as the angles and convert them to unit normal vectors by following equations, since a unit normal vector has two degrees of freedom (i.e., polar and azimuthal angles θ, ϕ).

$$a = \sin(\theta)\cos(\phi), b = \sin(\theta)\sin(\phi), c = \cos(\theta) \quad (5)$$

Finally, they are concatenated again and used for estimation of Z using Eq. (4).

We designed a depth guidance layer with multiple scale sizes ($k \times k(k \in \{1, 2, 4, 8\})$). Our goal is to enable the learning of multi-scale features across different regions. By utilizing features from the same spatial location at different scales to produce decoder features, we aim to capture global information at coarser scales and local details at finer scales. Additionally, these different scale features can interact with each other to extract depth-related features.

During training with depth guidance layers, details for regions would be learned at fine scales(1, 2) while major structures at coarse scales(4, 8).

3.3 Bins Transformer architecture

We design our bins transformer architecture based on the idea of transforming the regression problem into a classification problem as Fig. 5 shows. Following Adabins[8], we use vision transformer ViT[22] to extract bins width vector information and range attention feature map. However, different from Adabins[8], we use two different vision transformer architecture to estimate bins width and range attention feature maps, which can avoid the two results blemishing each other. The adaptive bins transformer encoder learns an adaptive bins vector for each decoder feature map adaptively. The attention maps encoder learns different attention maps for each bins vector. Then we transform the bins vector into bin centers and the attention maps into probabilities scores. The depth is estimated using bin centers and probabilites scores as Eq. (7) shown.

With the bins vector, we calculate depth-bin-centers as follows:

$$c(b_i) = d_{min} + (d_{max} - d_{min})(\frac{b_i}{2} + \sum_{j=1}^{i-1} b_j) \quad (6)$$

Range attention maps are passed through a 1 × 1 convolutional layer to obtain n channels which can be interpreted as probabilities scores $p_k, k = 1, ..., N$ at each pixel.Finally, the final depth value $\tilde{d}$ is calculated from the linear combination of probabilities scores p_k at that pixel and the depth bin centers $c(k)$ as follows:

$$\tilde{d} = \sum_{k=1}^{N} c(b_k)p_k \quad (7)$$

3.4 Training Loss

Following Adabins[8], we define two loss function for bins and pixels loss then add together to get the final loss.First we use a scaled version of the Scale-Invariant loss introduced by Eigen *et al.*[10] as follows:

$$\mathcal{L}_{pixel} = \alpha\sqrt{\frac{1}{T}\sum_i g_i^2 - \frac{\lambda}{T^2}(\sum_i g_i)^2} \quad (8)$$

where $g_i = \log\tilde{d}_i - \log d_i$ and the ground truth depth d_i and T denotes the number of pixels.This loss function calculate the loss of the estimated depth and the ground truth, which is the main loss function. We experimented with different combinations of hyperparameters α and λ, including $\alpha = 1, 10, 20, 50, 100$ and $\lambda = 0.8, 0.85, 0.9, 1$. Finally we set

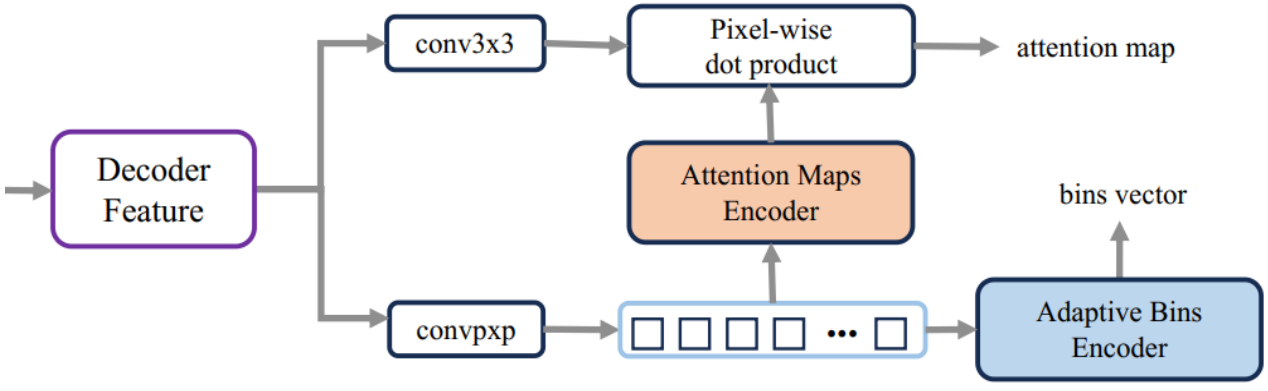


Fig. 5 An overview of the Adaptive bins transformer block. The input to the block is the decoder feature maps generated by depth guidance decoder. The block includes two transformer encoders.An adaptive bins transformer encoder that is applied on patch embeddings of the decoder feature for the purpose of learning to estimate bin widths. An attention maps Transformer encoder that is applied on patch embeddings of the input for the purpose of learning a set of convolutional kernels needed to compute attention maps.

$\alpha = 10$ and $\lambda = 0.85$, which achieved the best result.

Chamfer Loss[25] is important in 3D object reconstruction using point clouds. It defines the distance between two set S_1, S_2 as:

$$d(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2 \quad (9)$$

And we use the bi-directional Chamfer Loss to encourage the distributuion of bin centers to follow the distribution of depth values in the groud truth as:

$$\mathcal{L}_{bins} = chamfer(X, c(\mathbf{b})) + chamfer(c(\mathbf{b}), X) \quad (10)$$

Finally, we define the final loss as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{pixel} + \beta \mathcal{L}_{bins} \quad (11)$$

We experimented with different hyperparameters β, including $\beta = 0, 0.1, 0.2, 0.5$. Finally we set $\beta = 0.1$, which achieved the best result.

4 EXPERIMENTS

We conducted experiments on the standard depth estimation from a single image datasets. In the following, we first briefly describe the dataset and the evaluation metrics, and then present comparisons to the state-of-the-art in supervised monocular depth estimation.

4.1 Datasets

NYU Depth v2 is a dataset that provides images and depth maps for different indoor scenes captured at a size of 640 × 480 [37]. The dataset contains 120K training samples and 654 testing samples. We train our network on a 20K subset. The depth maps have an upper bound of 10 meters. We evaluate our network on the pre-defined center cropping by Eigen *et al.*[10]. At test time, we compute the final output by taking the average of an image's prediction and the prediction of its mirror image which is commonly used in previous work.

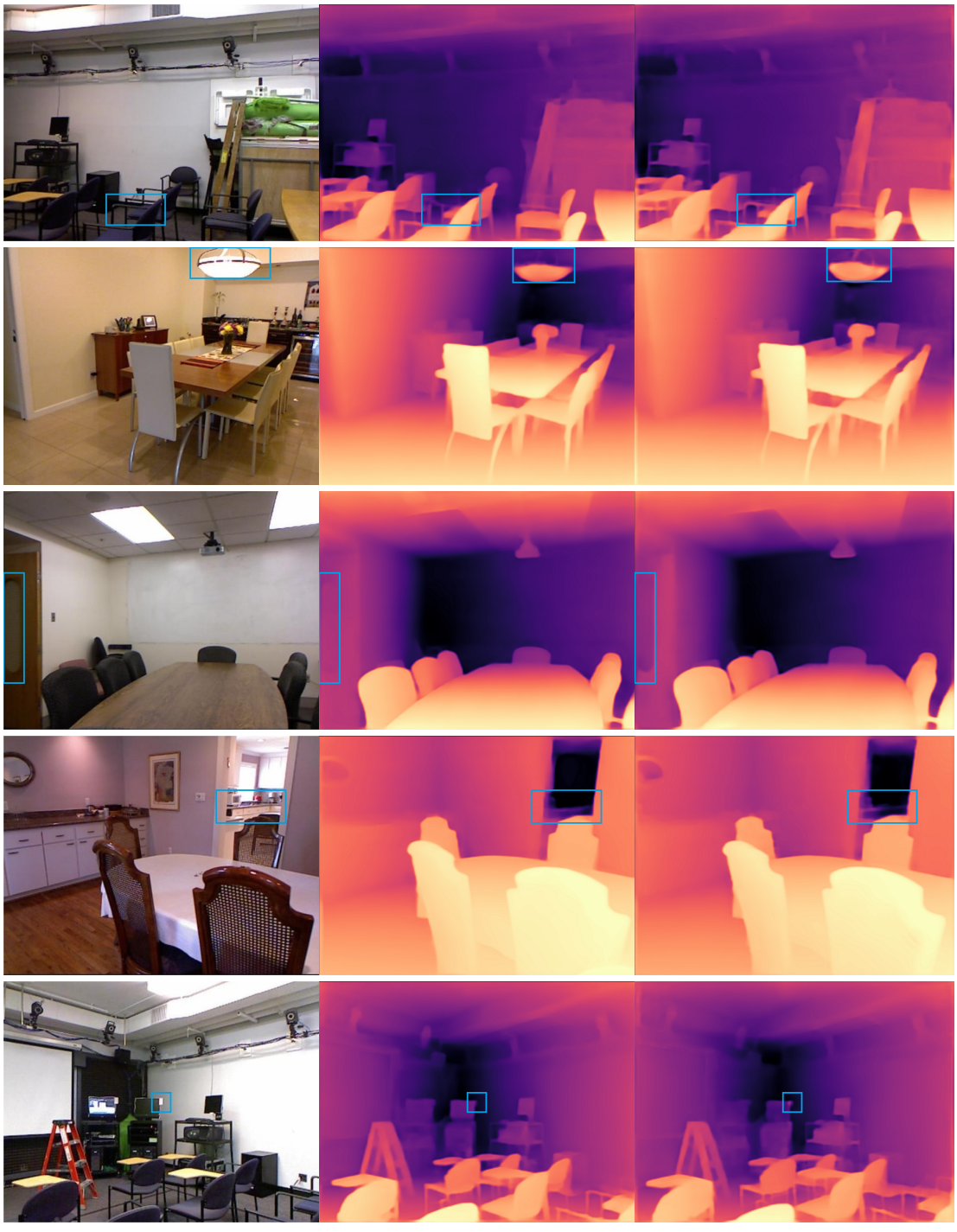
KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute) is one of the most popular datasets for use in mobile robotics and autonomous driving[38].It consists of hours of traffic scenarios recorded with a variety of sensor modalities, including high-resolution RGB, grayscale stereo cameras, and a 3D laser scanner. The RGB images have a resolution of around 1241 × 376. We train our network on a subset of around 23K images. The depth maps have an upper bound of 80 meters. The final output is computed by taking the average of an image's prediction and the prediction of its mirror image.

4.2 Evaluation metrics

We use the standard six metrics to compare our method against state of-the-art. These error metrics are defined as: average relative error (REL): $\frac{1}{n}\sum_n^p \frac{|y_p - \hat{y}_p|}{y}$; root mean squared error (RMS): $\sqrt{\frac{1}{n}\sum_n^p (y_p - \hat{y}_p)^2}$; average ($log_{10}$) error: $\frac{1}{n}\sum_n^p |log_{10}(y_p) - log_{10}(\hat{y}_p)|$; threshold accuracy (δ_i) : % of y_p s.t. $max(\frac{y_p}{\hat{y}_p}, \frac{\hat{y}_p}{y_p}) = \delta < thr$ for $thr = 1.25, 1.25^2, 1.25^3$; where y_p is a pixel in depth image y, $\hat{y}_p$ is a pixel in the predicted depth image $\hat{y}$, and n is the total number of pixels for each depth image. And for KITTI dataset, we use the two standard metrics: Squared Relative Difference (Sq. Rel): $\frac{1}{n}\sum_p^n \frac{\|y_p - \hat{y}_p\|^2}{y}$; and RMSE log: $\sqrt{\frac{1}{n}\sum_p^n \|\log y_p - \log \hat{y}_p\|^2}$.

4.3 Comparison to the state-of-the-art

We train and evaluate our models on the NYU Depth V2 dataset and the KITTI dataset. As shown in Table 1 and



(a) Input RGB (b) Adabins[8] (c) Ours

Fig. 6 Illustration of our work: **Left**: input RGB images. **Middle**: depth predicted by Adabins[8]. **Right**: depth predicted by our proposed mulit-scale depth guidance transformer. Note that the predicted depth in our work can present more details.

Table 1 QUANTITATIVE RESULTS ON NYU DEPTH V2.**THE BEST** AND THE SECOND-BEST ARE HIGHLIGHTED.

Method	$\delta_1\uparrow$	$\delta_2\uparrow$	$\delta_3\uparrow$	REL↓	RMS↓	$log_{10}\downarrow$
Eigen *et al.*[10]	0.769	0.950	0.988	0.158	0.641	-
Lee *et al.*[26]	0.815	0.963	0.991	0.139	0.572	-
Fu *et al.*[13]	0.828	0.965	0.992	0.115	0.509	0.051
Qi *et al.*[27]	0.834	0.960	0.990	0.128	0.569	0.057
Adabins[8]	0.874	0.953	0.967	0.130	0.418	0.141
Focal-WNet[28]	0.875	0.976	0.989	0.116	0.398	0.048
VNL[29]	0.875	0.980	0.995	0.111	0.416	0.048
Ours	**0.896**	**0.985**	**0.997**	**0.106**	**0.370**	**0.045**

Table 2 QUANTITATIVE RESULTS ON KITTI.**THE BEST** AND THE SECOND-BEST ARE HIGHLIGHTED.

Method	$\delta_1\uparrow$	$\delta_2\uparrow$	$\delta_3\uparrow$	REL↓	Sq Rel↓	RMS↓	$log_{10}\downarrow$
Liu *et al.*[30]	0.680	0.898	0.967	0.201	1.584	6.471	0.273
Eigen *et al.*[10]	0.702	0.898	0.967	0.203	1.548	6.307	0.282
Godard *et al.*[31]	0.861	0.949	0.976	0.114	0.898	4.935	0.206
Kuznietsov *et al.*[32]	0.862	0.960	0.986	0.113	0.741	4.621	0.189
SC-Depth[33]	0.873	0.960	0.982	0.114	-	4.706	0.191
DNet[34]	0.877	0.960	0.981	0.113	-	4.812	0.191
Gan *et al.*[35]	0.890	0.964	0.985	0.098	0.666	3.933	0.173
Focal-WNet[28]	0.926	0.986	0.997	0.082	-	3.076	0.120
Fu *et al.*[13]	0.932	0.984	0.994	0.072	0.307	2.727	0.120
Yin *et al.*[36]	0.938	0.990	0.998	0.072	-	3.258	0.117
Lee *et al.*[26]	0.956	0.993	0.998	0.059	0.245	2.756	0.096
Ours	**0.963**	**0.995**	**0.999**	**0.058**	**0.195**	**2.365**	**0.025**

Table 2, our models achieve state-of-the-art performance in all metrics compared to previous monocular depth estimation methods. Our models also outperform the previous bins transformer based method [8] by a large margin. As Fig. 6 shown, the predicted depth in our work can present more details than others. This shows that our depth guidance based transformer model is better at to capturing local context information, which mainly benefits from our multiple scale design. The multi-scale depth guidance layers can concatenate different scale context information and compensate each other. Then the extracted global and local features are used for final depth estimation together.

When training our network on the dataset, we observe that our network converge faster than Adabins[8]. As Fig. 7 shown, our network takes just 30k iterations(10 epoches) to achieve the state-of-the-art results while Adabins[8] takes 75k iterations(25 epoches). We think it benefits from the depth guidance layers which helps the network learn depth feature faster.

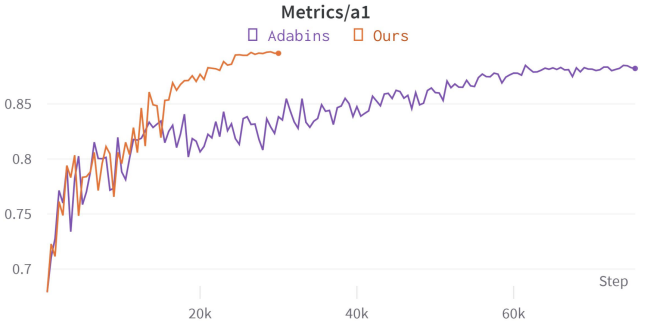


Fig. 7 An overview of the a1 metric in training progress of Adabins[8] and ours network. Adabins uses 25 epoches for training, ours just uses 10 epoches and converges in less time.

4.4 Ablation study

For our ablation study, we evaluate the influence of the following design choices on our results:

Depth guidance decoder: We first evaluate the importance of our depth guidance decoder module. We remove the multi-scale depth guidance decoder(MDG-Decoder) from the architecture and use a normal decoder(N-Decoder) to predict the feature map. The normal decoder use the same architecture as depth guidance decoder only without the multi-scale depth guidance layer. Table 3 shows that we achieve greater performance gain by employing the proposed depth guidance decoder (3rd row, 4th row) compared to the normal decoder architecture (1st row, 2nd row). This result indicates that some of the predictions are improved and our network can improve

Table 3 ABLATION STUDY OF THE PROPOSED METHOD ON NYU DEPTH V2. **THE BEST** AND THE SECOND-BEST ARE HIGHLIGHTED.

Variant	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	$REL \downarrow$	$RMS \downarrow$	$\log_{10} \downarrow$
N-Decoder + S-ViT	0.882	0.981	0.995	0.113	0.379	0.047
N-Decoder + D-ViT	0.887	0.982	0.996	0.109	0.377	0.046
MDG-Decoder + S-ViT	0.890	0.984	**0.997**	0.110	0.374	0.046
MDG-Decoder + D-ViT	**0.896**	**0.985**	**0.997**	**0.106**	**0.370**	**0.045**

the prediction in more details situation.

Transformer encoder effect: We then evaluate the effect of our double vision transformer encoder(D-ViT) design. We remove the adaptive bins encoder and the attention maps encoder from the network and use just a single vision transformer encoder(S-ViT) to predict the bins vector and range attention maps. Table 3 shows that the architecture with double transformer encoder (2nd row, 4th row) performs better than other variants(1st row, 3th row). This result indicates that using different transformer encoder for bins vector and attention maps can avoid the two prediction from blemishing each other, which improves the accuracy of our net work.

5 CONCLUSIONS

In this work, we have presented a depth guidance layers based monocular depth estimation network and achieved state-of-the-art results. Benefiting from recent advances in deep learning, we design a network architecture that uses novel depth guidance layers and transformer encoder, giving an explicit relation from encoder feature maps to the geometry guidance decoder feature maps for better training of the network. By deploying the proposed layers on multiple scales in the decoding phase, we gained an improved experimental results on challenging benchmark. In future work, we would like to investigate how to introduce different scenes geometry guidance to improve the accuracy and generalization ability of neural network in depth estimation field.

Acknowledgements

This work is supported by NSFC under grants 61860206007 and U19A2071, Sichuan Science and Technology Program under grant 2023YFG0334, as well as the funding from Sichuan University under grant 2020SCUNG205.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation, 2015.

[2] Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 2021, 65(1): 99–106, doi:10.1145/3503250.

[3] Teed Z, Deng J. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras, 2021, publisher Copyright: © 2021 Neural information processing systems foundation. All rights reserved.; 35th Conference on Neural Information Processing Systems, NeurIPS 2021 ; Conference date: 06-12-2021 Through 14-12-2021.

[4] Lee W, Park N, Woo W. Depth-assisted real-time 3D object detection for augmented reality. In *ICAT*, volume 11, 2011, 126–132.

[5] Moreno-Noguer F, Belhumeur PN, Nayar SK. Active refocusing of images and videos. *ACM Transactions On Graphics (TOG)*, 2007, 26(3): 67–es.

[6] Hazirbas C, Ma L, Domokos C, Cremers D. FuseNet: Incorporating Depth into Semantic Segmentation via Fusion-Based CNN Architecture. In SH Lai, V Lepetit, K Nishino, Y Sato, editors, *Computer Vision – ACCV 2016*, volume 10111, Springer International Publishing, 213–228, doi:10.1007/978-3-319-54181-5_14, series Title: Lecture Notes in Computer Science.

[7] Du R, Turner E, Dzitsiuk M, Prasso L, Duarte I, Dourgarian J, Afonso J, Pascoal J, Gladstone J, Cruces N, Izadi S, Kowdle A, Tsotsos K, Kim D. DepthLab: Real-time 3D Interaction with Depth Maps for Mobile Augmented Reality. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, ACM, 829–843, doi:10.1145/3379337.3415881.

[8] Farooq Bhat S, Alhashim I, Wonka P. AdaBins: Depth Estimation Using Adaptive Bins. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 4008–4017, doi:10.1109/CVPR46437.2021.00400.

[9] Hu X, Yang K, Fei L, Wang K. Acnet: Attention based network to exploit complementary features for rgbd semantic segmentation, 2019.

[10] Eigen D, Puhrsch C, Fergus R. Depth map prediction from a single image using a multi-scale deep network. *Advances in*

neural information processing systems, 2014, 27.

[11] Wang L, Zhang J, Wang Y, Lu H, Ruan X. CLIFFNet for Monocular Depth Estimation with Hierarchical Embedding Loss. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V*, 2020.

[12] Ranftl R, Bochkovskiy A, Koltun V. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, 12179–12188.

[13] Fu H, Gong M, Wang C, Batmanghelich K, Tao D. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, 2002–2011.

[14] Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille A. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016, PP, doi:10.1109/TPAMI.2017.2699184.

[15] Diaz R, Marathe A. Soft Labels for Ordinal Regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[16] Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks, 2019.

[17] Johnston A, Carneiro G. Self-Supervised Monocular Trained Depth Estimation Using Self-Attention and Discrete Disparity Volume. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[18] Laina I, Rupprecht C, Belagiannis V, Tombari F, Navab N. Deeper depth prediction with fully convolutional residual networks, 2016.

[19] Xu D, Ricci E, Ouyang W, Wang X, Sebe N. Multi-scale Continuous CRFs as Sequential Deep Networks for Monocular Depth Estimation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 161–169, doi:10.1109/CVPR.2017.25.

[20] Parmar N, Vaswani A, Uszkoreit J, Kaiser L, Shazeer N, Ku A, Tran D. Image Transformer. In J Dy, A Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80, 2018, 4055–4064.

[21] Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-End Object Detection with Transformers. In A Vedaldi, H Bischof, T Brox, JM Frahm, editors, *Computer Vision – ECCV 2020*, volume 12346, Springer International Publishing, 213–229, doi:10.1007/978-3-030-58452-8_13, series Title: Lecture Notes in Computer Science.

[22] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, et al.. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[23] Dosovitskiy A, Fischer P, Ilg E, Hausser P, Hazirbas C, Golkov V, Smagt PVD, Cremers D, Brox T. FlowNet: Learning Optical Flow with Convolutional Networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2758–2766, doi:10.1109/ICCV.2015.316.

[24] Wang X, Girshick R, Gupta A, He K. Non-local Neural Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, 7794–7803, doi:10.1109/CVPR.2018.00813.

[25] Fan H, Su H, Guibas L. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2463–2471, doi:10.1109/CVPR.2017.264.

[26] Lee J, Heo M, Kim K, Kim CS. Single-Image Depth Estimation Based on Fourier Domain Analysis. In *Proceedings - 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018*, 2018.

[27] Qi X, Liao R, Liu Z, Urtasun R, Jia J. GeoNet: Geometric Neural Network for Joint Depth and Surface Normal Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[28] Manimaran G, Swaminathan J. Focal-WNet: An Architecture Unifying Convolution and Attention for Depth Estimation. In *2022 IEEE 7th International conference for Convergence in Technology (I2CT)*, IEEE, 1–7, doi:10.1109/I2CT54291.2022.9824488.

[29] Yin W, Liu Y, Shen C, Yan Y. Enforcing Geometric Constraints of Virtual Normal for Depth Prediction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, 5683–5692, doi:10.1109/ICCV.2019.00578.

[30] Liu F, Shen C, Lin G, Reid I. Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, 38, doi:10.1109/TPAMI.2015.2505283.

[31] Godard C, Mac Aodha O, Brostow G. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[32] Kuznietsov Y, Stückler J, Leibe B. Semi-Supervised Deep Learning for Monocular Depth Map Prediction. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017: 2215–2223.

[33] Bian JW, Zhan H, Wang N, Li Z, Zhang L, Shen C, Cheng MM, Reid I. Unsupervised Scale-Consistent Depth Learning from Video. *International Journal of Computer Vision*, 2021, 129, doi:10.1007/s11263-021-01484-6.

[34] Xue F, Zhuo G, Huang Z, Fu W, Wu Z, Ang MH. Toward Hierarchical Self-Supervised Monocular Absolute Depth Estimation for Autonomous Driving Applications. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, 2330–2337, doi:10.1109/IROS45743.2020.9340802.

[35] Gan Y, Xu X, Sun W, Lin L. Monocular Depth Estimation with Affinity, Vertical Pooling, and Label Enhancement. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part III*, 2018.

[36] Yin W, Liu Y, Shen C, Yan Y. Enforcing Geometric Constraints of Virtual Normal for Depth Prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[37] Silberman N, Hoiem D, Kohli P, Fergus R. Indoor Segmentation and Support Inference from RGBD Images. In A Fitzgibbon, S Lazebnik, P Perona, Y Sato, C Schmid, editors, *Computer Vision – ECCV 2012*, Springer Berlin Heidelberg, 746–760.

[38] Geiger A, Lenz P, Stiller C, Urtasun R. Vision meets robotics: the KITTI dataset. *The International Journal of Robotics Research*, 2013, 32: 1231–1237, doi:10.1177/0278364913491297.

Author biography

Canbin Li Canbin Li received his B.S.degree in computer science and technology from Sichuan University in 2021. He is currently working toward his M.S. degree at the Sichuan University. His research interests include 3D computer vision and deep learning.

Yiguang Liu Yiguang Liu is a professor at the College of Computer Science, Sichuan University. He received his Ph.D. degree from the Sichuan University in 2004. His current research interests are Information Detection and Intelligent Perception, including detection imaging and the corresponding analysis and processing of incomplete and uncertain information.