arXiv:2203.04538v1 [cs.CV] 9 Mar 2022

Monocular Depth Distribution Alignment with Low Computation

Fei Sheng^{1,†}, Feng Xue^{1,†}, Yicong Chang¹, Wenteng Liang¹ and Anlong Ming^{1,*}

Abstract—The performance of monocular depth estimation generally depends on the amount of parameters and computational cost. It leads to a large accuracy contrast between light-weight networks and heavy-weight networks, which limits their application in the real world. In this paper, we model the majority of accuracy contrast between them as the difference of depth distribution, which we call 'Distribution drift'. To this end, a distribution alignment network (DANet) is proposed. We firstly design a pyramid scene transformer (PST) module to capture inter-region interaction in multiple scales. By perceiving the difference of depth features between every two regions, DANet tends to predict a reasonable scene structure, which fits the shape of distribution to ground truth. Then, we propose a local-global optimization (LGO) scheme to realize the supervision of global range of scene depth. Thanks to the alignment of depth distribution shape and scene depth range, DANet sharply alleviates the distribution drift, and achieves a comparable performance with prior heavy-weight methods, but uses only 1% floating-point operations per second (FLOPs) of them. The experiments on two datasets, namely the widely used NYUDv2 dataset and the more challenging iBims-1 dataset. demonstrate the effectiveness of our method. The source code is available at https://github.com/YiLiM1/DANet.

I. INTRODUCTION

Monocular depth estimation (MDE) aims to infer the 3D space for a given 2D image, which has been widely applied in many computer vision and robotics tasks, e.g., visual SLAM [1], [2], [3], monocular 3D object detection [4], [5], obstacle avoidance [6], [7], [8], and augmented reality [9]. They raise high demand for both the accuracy and speed of MED.

With the development of deep learning, many impressive works [10], [11], [12], [13] have emerged, most of which focus on improving accuracy of predicted depth. However, when we attempt to reduce the parameter and computation of these models their accuracy drops sharply. The degradation is mainly caused by the inadequate feature representation of pixel-wise continuous depth value. This phenomenon also appears in some recent algorithms [14], [15] that realize realtime MDE. To our knowledge, it is difficult for the prior methods to run in low latency while achieving a similar performance as the networks focusing on accuracy.

The motivation of this paper is the observed major degradation of the light-weight MDE models compared to heavyweight MDE models. We found that for the light-weight MDE networks, there are usually whole pieces of pixel in



Predicted Depth Distribution Predicted Depth

Fig. 1. Illustration of distribution drift phenomenon. The depth distribution is represented by the histogram of depth values, green for correct depth and red for predicted depth. The error map describes the pixel-wise error of depth, with red indicating too far and blue indicating too close.

the prediction that are smaller or larger than the correct depth monolithically, which is the main indicator of accuracy degradation. As shown in the second error map of [16] in Fig. 1, almost all pixels on the wall are predicted farther, which can be observed more intuitively in the depth distribution. Depth distribution shows the proportion of pixels with different depth values. The light-weight MDE models tend to get a completely different depth distribution from the ground truth, which is reflected in two differences, i.e., the shape of depth distribution and the full depth range. We call this issue 'Distribution Drift'. As shown in Fig. 1, [16] using a light-weight backbone obtains a different shape of depth distribution and depth range from ground truth.

In this paper, we propose a distribution alignment network to alleviate the distribution drift, making our method to achieve the performance comparable to the state-of-the-art methods, while with low latency. Firstly, to address the shape deviation of depth distribution, we propose a pyramid scene transformer (PST). Since the light-weight models are limited in network depth, they only extract depth cues in short range. However, minimal depth changes in a short range can hardly be perceived, which causes the wrong predicted depth of the whole slice. In the proposed PST, we capture the long-range interaction between every two regions in multiple scales, which constrains the depth relationship

[†]Equal contribution.

^{*}Corresponding author.

¹Beijing University of Posts and Telecommunications, Beijing, China, {shengfei,xuefeng,yicongchang,liangwenteng,mal} @bupt.edu.cn

This work was supported by the national key R & D program intergovernmental international science and technology innovation cooperation project 2021YFE0101600, and Excellent Ph.D. Students Foundation CX2020114.



Fig. 2. Cases of accuracy degradation of the prior state-of-the-art methods ([16], [17], [18] from left to right). For each method, the first row shows the prediction, error map and depth distribution of the models using the heavy-weight backbone, and the second row for the light-weight backbone.

between different regions. Thus PST is beneficial to realize a reliable scene structure. Then, to align the depth range, a local-global optimization (LGO) scheme is proposed to optimize the local depth value and the global depth range simultaneously. By using maximum and minimum depth as supervision, the value range of the scene depth is estimated to be aligned with the ground truth. Experiments prove that we indeed align the distribution of the scene depth, which helps our method to achieve comparable performance with state-of-the-art methods on NYUDv2 and iBims-1 datasets.

The main contributions of this work lie in:

- The distribution drift is studied to reveal the major degradation of light-weight models, which inspires us to propose a distribution alignment network (DANet). The DANet exceeds all prior light-weight works, and achieves a comparable accuracy with heavy-weight models but uses only 1% FLOPs of them.
- A pyramid scene transformer (PST) module is proposed to gain long-range interaction between multi-scale regions, helping DANet to alleviate the shape deviation of predicted depth distribution.
- A local-global optimization (LGO) scheme is proposed to jointly supervise the network with local depth value and global depth statistics.

II. RELATED WORK

A. Monocular Depth Estimation

Several early monocular depth estimators utilize the handcrafted features to estimate depth [19], [20] but suffer from insufficient expression ability. Recently, many CNN-based methods achieve great performance gain. Eigen *et al.* [21], [22] propose a coarse-to-fine CNN to estimate depth. Laina *et al.* [10] propose the up-projection for MDE to achieve higher accuracy. Xue *et al.* [23] improve the boundary accuracy of the predicted depth by a boundary fusion module. Yin *et al.* [17] enforce geometric constraints of virtual normal for depth prediction. Different from these works, Fu *et al.* [11] define MDE as a classification task. They divide the depth range into a set of bins with a predetermined width. Bhat *et al.* [24] compute bins adaptively for each image.

However, these prior methods focus heavily on achieving high accuracy with the cost of complexity and runtime, because they construct a large number of convolutions to obtain sufficient feature representation. Fig. 2 shows three models to verify the issue we find. Once the network capacity drops, they suffer a sharp degradation in accuracy which limits their applications. In this paper, we solve this problem by aligning the depth distribution, thus our method achieves a better trade-off between accuracy and computation.

B. Real-time Monocular Depth Estimation Methods

To reduce latency in inference, several methods are proposed for MDE in recent years. Wofk *et al.* [15] design an extremely light-weight network. It adopts the MobileNet [25] and depth-wise separable convolution [26] to build the whole network, followed by network pruning to further reduce computation. Nekrasov *et al.* [14] boost depth estimation by learning semantic segmentation and distilling structured knowledge from large model to light-weight model. However, due to the limited capacity of the model, distribution drift still appears in these light-weight networks.

C. Context Learning

Context plays an important role in computer vision tasks [27], [28], [29], [30], [31]. Zhao *et al.* [27] propose the pyramid pooling to aggregate global context information. Lu *et al.* [32] propose the multi-rate context learner to capture image context by dilated convolution. Vaswani *et al.* [33] design the transformer to obtain global context by self-attention, which is used in multiple vision tasks [34], [35], [36]. This paper proposes a pyramid scene transformer to capture context interaction between multi-scale regions.

III. PROBLEM FORMULATION

In contrast to other methods [16], [18], following [24], the depth range ([0, 10] in NYUDv2 dataset) of the whole scene is divided into $N_{\mathbf{b}}$ bins, and the goal of depth estimation is formulated as follow: for an input image $I \in \mathbb{R}^{H \times W \times 3}$, two tensors are jointly predicted, namely, the center values of depth bins $\mathbf{c} \in \mathbb{R}^{1 \times 1 \times N_{\mathbf{b}}}$, and the bin-probability maps $\mathbf{P} \in \mathbb{R}^{H \times W \times N_{\mathbf{b}}}$ indicating the probability of each pixel falling into the corresponding depth bin. In the final predicted depth map, denoted as \mathcal{Y} , each pixel can be formulated as the linear combination of bin probabilities and the bin centers.

$$y_i = \sum_{n=1}^{N_{\mathbf{b}}} \mathbf{P}_i(n) \mathbf{c}(n) \tag{1}$$

where y_i denotes the *i*-th pixel in the prediction \mathcal{Y} .

IV. METHODOLOGY

The first subsection outlines the whole architecture of DANet. The second subsection illustrates the pyramid scene transformer (PST), and the following subsection presents



Fig. 3. Network architecture of DANet which consists of an encoder-decoder network, pyramid scene transformer. The pyramid scene transformer is between the encoder and the decoder, which predicts the center of depth bin to combine with the output of decoder.

the local-global optimization (LGO) scheme for aligning the depth range of the scene to the correct range.

A. Network Structure

Fig. 3 illustrates the architecture of DANet that consists of an encoder, a pyramid scene transformer, and a decoder. Given an image I, a light-weight backbone EfficientNet B0 [37] is used to extract features. Assuming that the *i*th level feature map of the backbone is denoted as $\mathbf{x}_i \in$ $\mathbb{R}^{\frac{H}{2^i} \times \frac{W}{2^i} \times C_i}, (i \in \{1, 2, 3, 4, 5\}), \text{ where } C_i \text{ is the channel}$ number. The feature compression block (FCB), composed of a 3×3 convolution and a 1×1 convolution, is used to reduce the channel number of feature \mathbf{x}_i $(i \in \{1, 2, 3, 4\})$ to 16. The four FCBs provide multi-level scene detail information with low time cost. At the end of the encoder, the PST is employed to capture the interaction between multi-scale regions from \mathbf{x}_5 , meanwhile predict the center values of the depth bin, i.e., c, in the scene (see Section IV-B). In the decoder, four up-scaling stages are employed to gradually enlarge the resolution from $\frac{H}{32} \times \frac{W}{32}$ to $\frac{H}{2} \times \frac{W}{2}$. Each stage upsamples the last stage output, and sums it with the same-size feature given by FCB. Then, a residual structure containing three 3×3 convolutions is used to fuse these features. The channel numbers of features in the decoder are set to 16 to meet requirements of low latency and light weight. At the end of the decoder, we use a 1×1 convolution to learn $N_{\rm b}$ dimensional bin-probability map P from the finest resolution feature. Referring to Eq. 1, the final prediction \mathcal{Y} is obtained by linear combination of **P** and **c**.

B. Pyramid Scene Transformer

Context interaction models the inter-region relationship of depth features, which helps to correctly estimate depth difference between regions. In the global view of the scene, it plays a significant role in suppressing the shape deviation of the depth distribution. To this end, we design the PST to capture context interaction, which consists of three independent parallel paths, as shown in Fig. 4. These paths divide the scene into various-size patches, respectively, to cover various-size scene components. And the relationship between every two patches is captured by a transformer structure [34].



Fig. 4. The detailed structure of pyramid scene transformer.

Specifically, an adaptive embedding convolution (AEC) is firstly designed to gain the multi-scale context embeddings adaptively. Given the input feature resolution $H_i \times W_i$ and the expected output resolution $H_o \times W_o$, AEC is defined as:

- Stride in X direction: $s_{\mathbf{x}} = \lfloor W_{\mathbf{i}} / W_{\mathbf{o}} \rfloor$
- Stride in Y direction: $s_{\mathbf{y}} = \lfloor H_{\mathbf{i}}/H_{\mathbf{o}} \rfloor$
- Kernel size: $(H_{\mathbf{i}} s_{\mathbf{y}}(H_{\mathbf{o}} 1)) \times (W_{\mathbf{i}} s_{\mathbf{x}}(W_{\mathbf{o}} 1))$

By using AEC, these paths re-scale \mathbf{x}_5 into tensors with three sizes: $\mathbf{x}_{\mathbf{p}}^j \in \mathbb{R}^{\frac{H}{2^{j+4}} \times \frac{W}{2^{j+4}} \times C_e}$, where $j \in \{1, 2, 3\}$ is the path number. Each pixel in $\mathbf{x}_{\mathbf{p}}^j$ represents the C_e -dimensional context embedding of a patch in the scene. Secondly, in a path, all embeddings are fed into a transformer encoder after adding a 1-D learned positional encoding [34]. The transformer encoder is utilized to perceive the interaction between every two embeddings, and output a sequence of embeddings with the same size as the input embeddings. Note that the first path is different from the two others. It appends an additional C_e -dimensional embedding $\mathbf{b_i}$ together with context embeddings, and outputs a special embedding $\mathbf{b}_{\mathbf{o}}$ which has the same size as $\mathbf{b}_{\mathbf{i}}$. Thirdly, in each path, the output embeddings are reshaped to build a tensor $\mathbf{x}_{\mathbf{o}}^{i}, i \in$ $\{1,2,3\}$ which has the same size as $\mathbf{x}_{\mathbf{p},i}^{i} \in \{1,2,3\}$. Then, all output tensors $\mathbf{x}_{\mathbf{o}}^{i}$ are upsampled to $\frac{H}{32} \times \frac{W}{32} \times C_{e}$, so that they can be concatenated. The concatenated feature is then compressed to 16 channels through two 3×3 convolution and a 1×1 convolution, and fed into the decoder. Meanwhile,



Fig. 5. The distribution of a depth map and value of λ_i^D with different v.

in the first path, the output special embedding $\mathbf{b_o}$ is fed into a multi-layer perceptron to obtain a $N_{\mathbf{b}}$ -dimensional vector. Subsequently, in the same way as [24], the vector \mathbf{b}' is normalized to obtain the depth-range widths vector $\mathbf{b}: b_i = \frac{b'_i + \tau}{\sum_{j=1}^{N_{\mathbf{b}}} (b'_j + \tau)}, i \in \{1, 2, ..., N_{\mathbf{b}}\}$. And the center of bin \mathbf{c} is obtained as follows: $c_i = d_{min} + (d_{max} - d_{min})(b_i/2 + \sum_{j=1}^{i-1} b_j)$, where d_{min}, d_{max} are the minimum and maximum depth values, and c_i is the *i*-th value in \mathbf{c} .

Since the transformer extracts the context interaction of each two patches in a scene, each output embedding encodes the depth interaction from one patch to all other patches. And different paths correspond to the depth correlation of patch in various scales. Moreover, unlike [24], PST is between the encoder and decoder to minimize the amount of computation.

C. Local-Global Optimization for Depth Range Learning

To align the global depth range, we propose a localglobal optimization (LGO) scheme, which trains DANet by two stages. In the local stage, we perform two local errors referred from [24] as supervision. In the global stage, we propose min-max loss and range-based pixel weight to learn the global depth range and optimize the whole depth.

1) Loss of local stage: The local stage aims to optimize the pixel-wise depth. To this end, a scaled version of the Scale-Invariant (SSI) loss [38] is used to minimize the pixelwise error between the predicted depth and correct depth:

$$L_{pixel} = \sqrt{\frac{1}{N} \sum_{i} h_i^2 - \frac{u}{N^2} \left(\sum_{i} h_i\right)^2} \tag{2}$$

where $h_i = (\lambda_i + 1) \cdot (\log y_i - \log g_i)$, and N is the pixel number of an image. y_i, g_i are the predicted and correct depth respectively. λ_i is a weight parameter of pixel *i*. In the local stage, λ_i is set to 0. Furthermore, following [24], the bidirectional Chamfer Loss [24] is employed as a regularizer to optimize the bin centers **c** to be close to the ground truth.

$$L_{bin} = \sum_{x \in \mathcal{X}} \min_{c_i \in \mathbf{c}} \|x - c_i\|^2 + \sum_{c_i \in \mathbf{c}} \min_{x \in \mathcal{X}} \|x - c_i\|^2 \quad (3)$$

where \mathcal{X} is the set of all depth values in the ground truth.

2) Loss of global stage: The global stage aims to learn the depth range. In this stage, we supervise the first and last bin center in c by a new designed min-max loss:

$$L_{minmax} = \|c_1 - min(g)\|_1 + \|c_{N_{\mathbf{b}}} - max(g)\|_1 \quad (4)$$

where c_i is *i*-th value of **c**. *min* and *max* are the operation of taking minimum and maximum value, respectively. The min-max loss affects all pixels during back-propagation by supervising the bins, so that it squeezes all predicted depth values into range $[g_{min}, g_{max}]$.

However, since the amount of pixels with the largest and smallest depth is small in scenes, the network might be insensitive to these pixels. Thus, we additionally assign a depth-related weight to each pixel. In the global stage, the parameter λ_i^D in Eq. 2 is taken as the depth-related weight of a pixel, which is proportional to the difference from the pixel's depth to the median depth value in ground truth.

$$\lambda_i^D = \begin{cases} \frac{v(med(g)-g_i)}{med(g)-min(g)} & if \ g_i \le med(g) \\ \frac{v(g_i-med(g))}{max(g)-med(g)} & otherwise \end{cases}$$
(5)

where *i* is pixel index, and *v* is a coefficient. *med* denotes the operation of taking medium value. As shown in Fig. 5, if the correct depth g_i is close to max(g) or min(g), the λ_i^D is close to *v*, which means that the network pays more attention to this pixel *i*. If the correct depth g_i is close to med(g), the λ_i^D tends to be 0. In this way, DANet pays more attention to pixels with small and large depth, and predicts a more reasonable depth range.

3) Training scheme: Combined with the min-max loss and λ_i^D , the total loss is formulated as:

$$L_{total} = \alpha L_{pixel} + \beta L_{bin} + \gamma L_{minmax} \tag{6}$$

where α, β, γ are hyper-parameters. In the first stage, we set $\alpha = 10, \beta = 0.1, \gamma = 0, u = 0.85$ and $\lambda_i = 0$. The first stage optimizes the pixel-wise depth preliminarily. In the second stage, we set $\alpha = 10, \beta = 0.1, \gamma = 0.1, u = 0.85, v = 1$ and $\lambda_i = \lambda_i^D$. The second stage further optimizes the depth range based on the learned weight of the first stage.

V. EXPERIMENTS

In this section, we evaluate the proposed method on several datasets, and compare to the prior methods. Moreover, we give more discussions for the network design.

A. Dataset and Implementation Details

Datasets: NYUDv2 [41] and iBims-1 dataset [42] are used to conduct experiments. NYUDv2 [41] is an indoor dataset that collects 464 scenes with 120K pairs of RGB and depth maps. Following [16], [18], we train DANet on 50k images sampled from raw training data and adopt the same data augmentation strategy as [18]. The test set includes 654 images with filled-in depth values. iBims-1 dataset [42] contains 100 pairs of high-quality depth map and high-resolution image. Since the dataset lacks training set, we evaluate the generalization on it by using the model trained on NYUDv2 dataset.

Implementation Details: DANet is constructed on the Pytorch framework using a single NVIDIA 3090 GPU. Our backbone, namely, EfficentNet b0, is pre-trained on ILSVRC [43]. Other parameters are randomly initialized. The Adam optimizer is adopted with parameters (β_1 , β_2) = (0.9, 0.999). The weight decay is 10^{-4} . We train our model for 20 epochs with batch size of 24, 10 epochs for the local stage and 10 epochs for the global stage. The initial learning rate is set to 0.0002 and reduced by 10 % for every 5 epochs. **Metrics:** Following [24], we evaluate our method based on following metrics: mean absolute relative error (REL), root mean squared error (RMS), mean \log_{10} Error (log10),

TABLE I

COMPARISONS ON NYUDV2 DATASET. GROUP ^① CONTAINS NON-LIGHTWEIGHT METHODS. GROUP ^② CONTAINS LIGHT-WEIGHT METHODS. GROUP ^③ CONTAINS THE RE-IMPLEMENTED MODELS USING A SAME BACKBONE WITH OUR METHOD.

Groups	Methods	Backbone	Resolution	FLOPs	Params	REL \downarrow	RMS \downarrow	$\log 10\downarrow$	$\delta_1\uparrow$	$\delta_2\uparrow$	$\delta_3\uparrow$
	Eigen et al.[22]	VGG16	240×320	31G	240M	0.215	0.772	0.095	0.611	0.887	0.971
	Eigen et al. [21]	VGG16	228×304	23G	-	0.158	0.565	-	0.769	0.950	0.988
	Laina et al. [10]	ResNet50	240×320	17G	63M	0.127	0.573	0.055	0.811	0.953	0.988
	Fu et al. [11]	ResNet101	240×320	102G	85M	0.118	0.498	0.052	0.828	0.965	0.992
	Lee et al. [39]	DenseNet161	224×224	96G	268M	0.126	0.470	0.054	0.837	0.971	0.994
Ū	Hu et al. [16]	ResNet50	228×304	107G	67M	0.130	0.505	0.057	0.831	0.965	0.991
	Chen et al. [18]	SENet154	228×304	150G	258M	0.111	0.420	0.048	0.878	0.976	0.993
	Yin et al. [17]	ResNet101	384×384	184G	90M	0.105	0.406	0.046	0.881	0.976	0.993
	Lee et al. [38]	ResNet101	416×544	132G	66M	0.113	0.407	0.049	0.871	0.977	0.995
	Bhat <i>et al.</i> [24]	EfficientNet b5	426×560	186G	77M	0.103	0.364	0.044	0.902	0.983	0.997
2	Wofk et al. [15]	MobileNet	224×224	0.75G	3.9M	0.162	0.591	-	0.778	0.942	0.987
	Nekrasov et al. [14]	MobileNet v2	480×640	6.49G	2.99M	0.149	0.565	-	0.790	0.955	0.990
	Yin et al. [17]	MobileNet v2	338×338	15.6G	2.7M	0.135	-	0.060	0.813	0.958	0.991
	Hu et al. [40]	MobileNet v2	228×304	-	1.7M	0.138	0.499	0.059	0.818	0.960	0.990
3	Hu et al. [16] †	EfficientNet b0	228×304	14G	5.3M	0.142	0.505	0.059	0.814	0.961	0.989
	Chen et al. [18] †	EfficientNet b0	228×304	8.22G	12M	0.135	0.514	-	0.828	0.963	0.990
	Yin et al. [17] †	EfficientNet b0	384×384	18G	4.6M	0.145	0.567	0.067	0.771	0.947	0.988
	Ours	EfficientNet b0	228×304	1.5G	8.2M	0.135	0.488	0.057	0.831	0.966	0.991

TABLE II

COMPARISONS ON IBIMS-1 DATASET. THE 1-ST GROUP IS NON-LIGHT WEIGHT METHODS. THE 2-ND GROUP IS LIGHT-WEIGHT METHODS.

Methods	REL↓	RMS↓	log10↓	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$
Eigen et al. [22]	0.32	1.55	0.17	0.36	0.65	0.84
Eigen et al. [21]	0.25	1.26	0.13	0.47	0.78	0.93
Laina et al. [10]	0.23	1.20	0.12	0.50	0.78	0.91
Hu et al. [16]	0.24	1.20	0.12	0.48	0.81	0.92
Chen et al. [18]	0.25	1.07	0.10	0.56	0.86	0.94
Fu et al. [11]	0.23	1.13	0.12	0.55	0.81	0.92
Lee et al. [39]	0.23	1.09	0.11	0.53	0.83	0.95
Yin et al. [17]	0.24	1.06	0.11	0.54	0.84	0.94
Bhat <i>et al.</i> [24]	0.21	0.91	0.10	0.55	0.86	0.95
Wofk et al. [15]	0.38	1.76	0.21	0.30	0.56	0.74
Nekrasov et al. [14]	0.52	1.57	0.16	0.33	0.66	0.87
Ours	0.26	1.11	0.11	0.55	0.86	0.94

and the accurate under threshold ($\delta_k < 1.25^k$, k = 1, 2, 3). Referring to [24], in order to make a fair comparison, we re-evaluated some methods [22], [21], [10], [11], [16], [18], [17], in which the performance will be slightly different.

B. Comparison with the prior methods

Quantitative Evaluation: Table I shows the comparison between our method and the prior methods on NYUDv2 dataset. The backbones of three non-real time networks [16], [17], [18] are replaced for comparison (Group ③). DANet achieves a comparable RMS and accuracy of several non-real time networks [39], [11], [16], but only expending $1.4\% \sim$ 1.56% FLOPs of them. It also outperforms all light-weight networks [15], [14], [40] by a large margin. Furthermore, compared to the state-of-the-art methods with EfficientNet b0, DANet gains the best performance on all metrics, which expresses the effectiveness of distribution alignment in lightweight network. Although DANet uses more parameters than light-weight models, it is much slighter than heavy-weight models, enough to run well on the embedded platforms.

Table II shows the cross-dataset evaluation on iBims-1 dataset by using the model trained on NYUDv2 dataset without fine-tuning. Note that we do not re-normalize the depth range of the results to iBims-1. Although iBims-1

TABLE III QUANTITATIVE RESULTS OF OUR PROPOSED MODULE.

Models	RMS↓	$\delta_1 \uparrow$	FLOPs	Params
Baseline	0.510	0.810	1.0G	3.7M
+ PST	0.498	0.820	1.5G	8.2M
+ PST + min-max loss	0.498	0.825	1.5G	8.2M
+ PST + depth-related weight	0.496	0.822	1.5G	8.2M
+ LGO	0.496	0.822	1.0G	3.7M
+ PST + LGO	0.488	0.831	1.5G	8.2M

TABLE IV

QUANTITATIVE RESULTS OF CONTEXT LEARNING MODULE.

Models	RMS↓	$\delta_1\uparrow$
Ours with PPM [27]	0.497	0.823
Ours with ASPP [44]	0.494	0.825
Ours with mini ViT [24]	0.496	0.822
Ours with PST	0.488	0.831

dataset has a totally different data distribution from NYUDv2 dataset, DANet achieves the fifth best RMS and tied for 2rd best accuracy of δ_1 with state-of-the-art methods [24], [11]. Furthermore, DANet exceeds the prior real-time works [14], [15] by a large margin on all metrics. The reason is that DANet gains an outstanding performance on scenes with a similar depth range to NYUD v2, which proves the generalization of our method with distribution alignment.

Qualitative Evaluation: Fig. 6 shows the qualitative results on NYUDv2 dataset (first two rows) and iBims-1 dataset (last two rows). In the first scene, several methods predict a wrong depth of the wall behind the sofa, thus suffering from the wrong depth range. DANet gets a depth distribution almost coinciding with ground truth. In the second scene, the farthest region is occluded by the cabinet. The lightweight models obtain the wrong farthest region, causing a large distribution drift. Our method correctly estimates the depth together with other state-of-the-art methods. The third and fourth rows show two scenes that have never been seen. Many methods suffer from the deviation of depth range, especially the light-weight models [14], [15]. Our method still estimates the depth distribution almost perfectly,



Fig. 6. Visualizations on the NYUDv2 (first two rows) and iBims-1 datasets (last two rows). $1^{st} - 2^{nd}$ columns are input and ground truth, and $3^{th} - 9^{th}$ columns for [39], [16], [18], [24], [15], [14] and DANet. The depth distribution is under the depth maps with green for correct depth and red for prediction.



Fig. 7. Qualitative results of each contributions.

and predicts a reasonable depth image. These visualizations further prove the effectiveness of proposed paradigm.

C. Detailed Discussions

Ablation studies: We verify our PST and LGO in Table III. They are added one by one to test the effectiveness of each proposal. Note that the baseline is an encoder-decoder network without PST and LGO. Compared with baseline, PST and LGO achieve 1% and 1.2% gain in δ_1 . Moreover, Baseline+PST+LGO achieves the best performance in all metrics of evaluation. We further validate the effectiveness of min-max loss L_{minmax} and depth-related weight λ_i^D in LGO. Compared to Baseline+PST, the performance of the model is improved after using L_{minmax} and λ_i^D respectively.

Fig. 7 illustrates the visualized results of these variants. The model using LGO squeezes the depth range into a narrower space, but fails to optimize the distribution shape. The model using PST obtains a similar distribution with ground truth, but suffers from the wrong depth range. The model using all of them aligns the depth distribution well.

Effectiveness of multi-scale interaction. To evaluate the multi-scale interaction, PST is replaced by other contextual learning modules, i.e., Pyramid Pooling Module (PPM) [27], ASPP [44], and mini ViT [24], respectively. As shown



Fig. 8. The performance of δ_1 with various coefficient v.

in Table IV, DANet with PST outperforms others over all metrics, because the interaction of multi-scale regions directly models the relationship between every two regions. **Coefficient of depth-related weight**. To explore the best coefficient v of depth-related weight. To explore the best $\{0, 1, 2, 3, 4, 5\}$. Fig. 8 demonstrates the comparisons. It can be seen that δ_1 rises at the beginning and continues to decrease as v increases, which reveals that excessive attention to the far and near areas leads to performance saturates. Therefore, the coefficient is set to 1 in this paper.

VI. CONCLUSIONS

In this work, our DANet is designed to solve the distribution drift problem in light-weight MDE network. To obtain an aligned depth distribution shape, the PST is introduced, which captures the interaction between multi-scale regions. In addition, a local-global optimization is proposed to guide the network to obtain a reliable depth range. Experimental results on NYUDv2 and iBims-1 datasets prove that DANet achieves comparable performance with state-of-the-art methods with only 1% FLOPs of them. In the future, we will further achieve real-time running time on the embedding platform, so that it can be used to improve depth-dependent tasks [45], [46] and mobile robot applications [47].

REFERENCES

- Y. Li, Y. Ushiku, and T. Harada, "Pose graph optimization for unsupervised monocular visual odometry," in *International Conference* on Robotics and Automation (ICRA), 2019.
- [2] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Realtime dense monocular slam with learned depth prediction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [3] L. Tiwari, P. Ji, Q. H. Tran, B. Zhuang, and M. Chandraker, "Pseudo rgb-d for self-improving monocular slam and depth prediction," in *European Conference on Computer Vision (ECCV)*, 2020.
- [4] X. Weng and K. Kitani, "Monocular 3d object detection with pseudolidar point cloud," in *IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2019.
- [5] Y. Cai, B. Li, Z. Jiao, L. H, X. Zeng, and X. Wang, "Monocular 3d object detection with decoupled structured polygon estimation and height-guided depth estimation," in AAAI Conference on Artificial Intelligence (AAAI), 2020.
- [6] M. Mancini, G. Costante, P. Valigi, and T. A. Ciarfuglia, "J-mod²: Joint monocular obstacle detection and depth estimation," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 3, pp. 1490–1497, 2018.
- [7] L. Xie, S. Wang, A. Markham, and N. Trigoni, "Towards monocular vision based obstacle avoidance through deep reinforcement learning," in *Robotics: Science and System workshop (RSS Workshop)*, 2017.
- [8] F. Xue, A. Ming, and Y. Zhou, "Tiny obstacle discovery by occlusionaware multilayer regression," *IEEE Transactions on Image Processing* (*TIP*), vol. 29, pp. 9373–9386, 2020.
- [9] X. Luo, J.-B. Huang, R. Szeliski, K. Matzen, and J. Kopf, "Consistent video depth estimation," ACM Transactions on Graphics (TOG), vol. 39, no. 4, 2020.
- [10] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *International Conference on 3D Vision (3DV)*, 2016.
- [11] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [12] J. Jiao, Y. Cao, Y. Song, and R. Lau, "Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss," in *The European Conference on Computer Vision (ECCV)*, 2018.
- [13] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zeroshot cross-dataset transfer," *IEEE Transactions on Pattern Analysis* and Machine Intelligence (TPAMI), 2020.
- [14] V. Nekrasov, T. Dharmasiri, A. Spek, T. Drummond, C. Shen, and I. Reid, "Real-time joint semantic segmentation and depth estimation using asymmetric annotations," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.
- [15] D. Wofk, F. Ma, T.-J. Yang, S. Karaman, and V. Sze, "Fastdepth: Fast monocular depth estimation on embedded systems," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.
- [16] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, "Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019.
- [17] W. Yin, Y. Liu, C. Shen, and Y. Yan, "Enforcing geometric constraints of virtual normal for depth prediction," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [18] X. Chen, X. Chen, and Z.-J. Zha, "Structure-aware residual pyramid network for monocular depth estimation," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.
- [19] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 31, no. 5, pp. 824– 840, 2009.
- [20] M. Liu, M. Salzmann, and X. He, "Discrete-continuous depth estimation from a single image," in *IEEE Conference on Computer Vision* and Pattern Recognition (CVPR), 2014.
- [21] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [22] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Neural Information Processing Systems (NIPS)*, 2014.

- [23] F. Xue, J. Cao, Y. Zhou, F. Sheng, Y. Wang, and A. Ming, "Boundaryinduced and scene-aggregated network for monocular depth prediction," *Pattern Recognition (PR)*, vol. 115, p. 107901, 2021.
- [24] S. F. Bhat, I. Alhashim, and P. Wonka, "Adabins: Depth estimation using adaptive bins," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [25] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *ArXiv*, vol. abs/1704.04861, 2017.
- [26] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [27] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [28] Y. Gao, X. Li, J. Zhang, Y. Zhou, D. Jin, J. Wang, S. Zhu, and X. Bai, "Video text tracking with a spatio-temporal complementary model," *IEEE Transactions on Image Processing (TIP)*, vol. 30, pp. 9321– 9331, 2021.
- [29] Y. Zhou, X. Bai, W. Liu, and L. Latecki, "Fusion with diffusion for robust visual tracking," in Advances in Neural Information Processing Systems (NIPS), 2012.
- [30] Y. Zhou, X. Bai, W. Liu, and L. J. Latecki., "Similarity fusion for visual tracking," *International Journal of Computer Vision (IJCV)*, vol. 118, no. 3, pp. 337–363, 2016.
- [31] M. Zhou, J. Ma, A. Ming, and Y. Zhou, "Objectness-aware tracking via double-layer model," in *IEEE International Conference on Image Processing (ICIP)*, 2018.
- [32] R. Lu, F. Xue, M. Zhou, A. Ming, and Y. Zhou, "Occlusion-shared and feature-separated network for occlusion relationship reasoning," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Neural Information Processing Systems (NIPS)*, 2017.
- [34] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ArXiv*, vol. abs/2010.11929, 2021.
- [35] D.-J. Chen, H.-Y. Hsieh, and T.-L. Liu, "Adaptive image transformer for one-shot object detection," in *IEEE Conference on Computer Vision* and Pattern Recognition (CVPR), 2021.
- [36] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [37] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning (ICML)*, 2019.
- [38] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," *ArXiv*, vol. abs/1907.10326, 2020.
- [39] J.-H. Lee and C.-S. Kim, "Monocular depth estimation using relative depth maps," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [40] J. Hu, C. Fan, H. Jiang, X. Guo, X. Lu, and T. L. Lam, "Boosting light-weight depth estimation via knowledge distillation," *ArXiv*, vol. abs/2105.06143v1, 2021.
- [41] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *The European Conference on Computer Vision (ECCV)*, 2012.
- [42] T. Koch, L. Liebel, F. Fraundorfer, and M. Körner, "Evaluation of cnn-based single-image depth estimation methods," in *European Conference on Computer Vision Workshop (ECCV Workshop)*, 2019.
- [43] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and F.-F. Li, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, p. 211–252, 2015.
- [44] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 40, no. 4, pp. 834–848, 2018.
- [45] Y. Zhou, Y. Yang, Y. Meng, X. Bai, W. Liu, and L. J. Latecki., "Online multiple person detection and tracking from mobile robot

in cluttered indoor environments with depth camera," *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, vol. 28, no. 1, pp. 1455 001.1–1455 001.28, 2014.
[46] Z. Liu, X. Zhao, T. Huang, R. Hu, Y. Zhou, and X. Bai, "Tanet: Robust

- [46] Z. Liu, X. Zhao, T. Huang, R. Hu, Y. Zhou, and X. Bai, "Tanet: Robust 3d object detection from point clouds with triple attention," in AAAI Conference on Artificial Intelligence(AAAI), 2020.
- [47] F. Xue, A. Ming, M. Zhou, and Y. Zhou, "A novel multi-layer framework for tiny obstacle discovery," in *International Conference* on Robotics and Automation (ICRA), 2019.