Abstract—Monocular depth estimation (MDE) is a fundamental task in computer vision and has drawn increasing attention. Recently, some methods reformulate it as a classification-regression task to boost the model performance, where continuous depth is estimated via a linear combination of predicted probability distributions and discrete bins. In this paper, we present a novel framework called BinsFormer, tailored for the classification-regression-based depth estimation. It mainly focuses on two crucial components in the specific task: 1) proper generation of adaptive bins; and 2) sufficient interaction between probability distribution and bins predictions. To specify, we employ a Transformer decoder to generate bins, novelly viewing it as a direct set-to-set prediction problem. We further integrate a multi-scale decoder structure to achieve a comprehensive understanding of spatial geometry information and estimate depth maps in a coarse-to-fine manner. Moreover, an extra scene understanding query is proposed to improve the estimation accuracy, which turns out that models can implicitly learn useful information from the auxiliary environment classification task. Extensive experiments on the KITTI, NYU, and SUN RGB-D datasets demonstrate that BinsFormer surpasses state-of-the-art MDE methods with prominent margins. Code and pretrained models are made publicly available at https://github.com/zhyever/Monocular-Depth-Estimation-Toolbox/tree/main/configs/binformer.

Index Terms—Monocular depth estimation, adaptive bins, multi-scale refinement, auxiliary task, transformer.

I. INTRODUCTION

MONOCULAR depth estimation (MDE) is a fundamental yet challenging task in computer vision, which requires the algorithm to predict each pixel’s depth within a single input RGB image [1]. A widespread of various depth-dependent tasks related to autonomous driving [2], [3], [4], [5], [6], virtual reality [7], [8], and scene understanding [9], [10], [11], [12] provide strong demand for effective MDE methods that can accurately reconstruct the 3D world. As a result, there has been an increasing interest in MDE in recent years.

Existing works for MDE can be divided into three categories based on the supervision manner: supervised, self-supervised, and semi-supervised. Supervised MDE methods [13], [14], [15], [16], [17] apply RGB and ground-truth pairs during the training stage. To reduce the cost of labeling, self-supervised methods [18], [19], [20] view depth estimation as a view synthesis task and explore supervision in video frames. While no ground-truth is required during training, the model performance is far less than the supervised methods. To combine the advantages of both training strategies, semi-supervised based MDE methods [21], [22] propose to simultaneously utilize supervised loss and self-supervised loss terms during the training. While achieving performance gains, the bottleneck still lies in the supervised baseline methods. In this paper, we mainly focus on supervised MDE.

So far, there have been numerous mainstream supervised methods formulating MDE as per-pixel regression, such as DAV [23], DPT [24] and TransDepth [25] (Fig. 1a), where a regression loss is applied to each pixel prediction. Per-pixel regression methods can neatly predict pixel-wise depth, thus becoming a universal paradigm. In spite of their significant achievements, such methods still face problems of slow convergence and unsatisfied results [26].

Another line of research [26], [27] proposes to discretize continuous depth into several intervals and cast the depth network learning as a per-pixel classification problem (Fig. 1b). While this strategy significantly improves the model performance, it is worth noting that the discretization of depth values will result in poor visual quality with apparent sharp discontinuities.

To solve the issue, some methods [15], [19] reformulate depth estimation as a per-pixel classification-regression task (Fig. 1c), learning probabilistic representations on each pixel and predicting the final depth value as a linear combination with bin centers. The bin centers are pre-defined in Uniform/Log-uniform space (UD/SID) or trained ones (for each dataset). They combine the best of both tasks and achieve an improving performance. On top of that, Adabins [15] observes the extreme variation of depth distribution among changing scenes and further proposes the adaptive bins generation module to predict bins centers adaptively.

While Adabins [15] boosts depth estimation performance to a remarkable extent, several dilemmas still exist. 1) It directly applies bins prediction depending on the highest resolution feature map (the output of the last layer of the Decoder), leading to the difficulty of squaring up the global information...
In this paper, we propose a conceptually simple yet effective approach called BinsFormer, tailored for classification-regression-based MDE. It Novelly views adaptive bins generation as a direct set prediction problem. We employ a separate Transformer decoder to compute a set of pairs, each consisting of a bins length and a bins embedding vector. The bins embedding vector is used to get the probabilistic representations via a dot product with the per-pixel representations obtained from an underlying fully-convolutional decoder. Finally, BinsFormer predicts depth values as a linear combination of bins centers and probabilistic representations. Such disentangled decoder avoids bins embeddings and fine-grained per-pixel representations blemishing each other and combines the best of global and pixel-wise information. We further integrate a multi-scale decoder structure to comprehensively understand spatial geometry information and estimate depth maps in a coarse-to-fine manner. It enables sufficient interactions and aggregations of features via successive alternating cross-attention and self-attention mechanisms. To further improve the estimation accuracy, we equip an extra scene understanding query to the Transformer decoder, which aims to predict the classification of the input environment. It can benefit models to generate appropriate bins via auxiliary and implicit supervision.

We evaluate BinsFormer on three depth estimation datasets with various settings: NYU [31] (indoor), KITTI [32] (outdoor), and SUN-RGBD [33] (indoor, directly fine-tuning). Numerous experiments demostrate BinsFormer achieves the new state-of-the-art on all these datasets with Swin Transformer [34] backbone, outperforming other methods with large margins. Exhaustive ablation studies further validate the effectiveness of each proposed component.

In summary, the main contributions of this paper are as follows:

- We design a new framework for depth estimation, where a separate Transformer decoder is adopted to predict discrete bins in a set-to-set manner. It explicitly models the interaction between per-pixel representations and bins.
- We propose a multi-scale refinement strategy and equip the Transformer decoder with an additional scene classification query. These methods further improve the model performance.
- Experimental results show that our method achieves superior performance over previous state-of-the-art methods over three different datasets.

II. RELATED WORK

A. Depth Estimation

MDE plays a critical role in three-dimensional reconstruction and perception. There has been tremendous witnessed progress achieved by learning-based depth estimation methods in recent years. Eigen et al. [13] groundbreaking proposes a multi-scale deep network, consisting of a global network and a local network to predict the coarse depth and refine predictions, respectively. Motivated by [13], convolutional architectures have been intensively studied for depth estimation. For instance, CLIFFNet [35] applies a multi-scale fusion convolutional framework to generate high-quality depth.

![Diagram of BinsFormer](image)
prediction. Recently, Transformer networks are gaining greater interest in the computer vision community [29], [30], [34], [36]. Following the success of recent trends that apply the Transformer to solve computer vision tasks, TransDepth [25] and DPT [24] propose to replace convolution operations with Transformer layers, which further boosts model performance. Though the above methods have significantly improved depth prediction accuracy, they suffer from relatively slow convergence and sub-optimal solutions since they regard MDE as a regression task [26]. Another line of research [26], [27] proposes to discretize continuous depth into several intervals and cast the depth network learning as a per-pixel classification problem. DORN [26] also designs an effective ordinal classification depth estimation loss and develops an ASPP [37] module to extract multi-level information. Based on [26], [27] softens the classification target during training and achieves improving performance. Moreover, some methods [15], [19] reformulate the problem as classification-regression to alleviate poor visual quality with apparent sharp depth discontinuities caused by discretization of depth values. Johnston et al. [19] introduces the stereo DDV for MDE. To further improve the model performance, Adabins [15] proposes an adaptive bins strategy, which is crucial for accurate depth estimation.

In this paper, we further investigate the adaptive bins strategy and propose BinsFormer, tailored for classification-regression-based MDE. We novelly treat the adaptive bins generation as a set prediction process [29] and develop Transformer layers to resort to the problem, which is intuitively different from previous work [24], [25] that only utilize Transformer to strengthen the encoder capability. Furthermore, we melt an effective multi-scale refinement strategy into the Transformer-based decoder in a neat fashion. An extra scene understanding task further improves the model performance. Indeed, various strategies have been explored to benefit MDE, such as self-supervised learning [18], multi-task training [38], specific supervision losses [35], sparse ordinal [39] or relative depth estimation [40]. Our method focuses on the most basic framework design, which can be plugged into any other method as a strong baseline.

B. Transformer

Transformer was first proposed for natural language process (NLP) tasks [41] and has drawn increasing interest in the computer vision community recently [29], [30], [34], [36]. Tons of vision methods successfully adopt Transformer as a stronger alternative to the convolutional network, leading to new trends and bringing much vitality to computer vision [42]. Among them, one kind of method rethink the formulation of specific tasks and install Transformer seamlessly. The most representative work is DETR [29], in which the authors treat object detection as a set-to-set prediction task. They propose the object query to encode instance-level information and use Transformer to achieve interaction with queries and image features. Following it, various types of queries are proposed for different tasks, such as pose query [43] and track query [44], [45]. In this work, we are motivated by the mask queries [46], [47] in semantic segmentation and propose a bins query for depth estimation. But different from the semantic queries encoding per-class semantic information, bins queries try to represent the space in a discrete range manner, which is also in line with the motivation of the plane sweep strategy in depth estimation. However, instead of constructing cost volume for stereo matching, we discrete the continuous range to facilitate model training following [15].

III. Methods

In this section, we first present the overview of BinsFormer. Then, we introduce our instantiation of the adaptive bins generation strategy with the help of Transformer decoder [29]. Finally, we present the auxiliary scene understanding task and the multi-scale prediction refinement strategies, which can be neatly melted into the framework and improve the depth estimation performance.

A. Framework Overview

BinsFormer mainly consists of three essential components (see Fig. 2): the pixel-level module, the Transformer module, and the depth estimation module. Moreover, we propose the auxiliary scene classification and the multi-scale prediction refinement strategies to further boost model performance.

Given an input RGB image, the pixel-level module first extracts image features and decodes them into multi-scale immediate features $F$ and the per-pixel representations $f_p$. Benefiting from the encoder-decoder framework with skip connections, BinsFormer can fully extract local information for fine-grained depth estimation. Then, queries in the Transformer module interact with $F$ with the help of attention mechanisms. Independent MLPs are applied to project the queries into bins predictions $b$ and bins embeddings $f_b$, respectively. By novelly viewing the bins generation as a set-to-set prediction problem and applying Transformer, BinsFormer can also explore the global information and predict appropriate bins for integral depth estimation. Finally, the depth estimation module aggregates the best of the abovementioned modules and predicts final depth. It first calculates the probability distributions $P$ and then combines them with bins centers $c(b)$ via linear combinations.

On top of that, we equip an extra scene understanding query to the Transformer decoder, which aims to predict the classification of the input environment. Similarly, an MLP projects the query to the classification result. The extra task can benefit models to generate appropriate bins via auxiliary and implicit supervision. Moreover, we further integrate a multi-scale decoder structure to comprehensively understand spatial geometry information and estimate depth maps in a coarse-to-fine manner. The Transformer queries progressively interact with multi-scale features $F$, enabling sufficient aggregations of features via successive attention modules.

B. BinsFormer

1) Per-Pixel Module: Per-pixel module takes an image $I$ as input. A backbone is applied to extract a set of feature maps. Then, a commonly used decoder gradually upsamples
the features to generate the per-pixel representation \( f_p \in \mathbb{R}^{H \times W \times C_f} \), where \( C \) is the representation dimension, \( H \times W \) is the resolution. The decoder produces \( S \) scale immediate feature maps \( F = (f_i)^{S}_{i=1} \) as well, where \( f_i \) indicates the feature map at scale \( i^{th} \). This process can be formulated as:

\[
F, f_p = M_{\text{pixel}}(I),
\]

where \( M_{\text{pixel}} \) denotes the blue pixel-wise feature extraction module.

Note that any depth estimation model fits the pixel-wise feature extraction module design including Transformer-based models [15], [24], [25]. However, unlike previous methods [24], [25] that only adopt Transformer to replace convolutional operations in models, BinsFormer seamlessly converts Transformer to solve the bins generation and leaves the per-pixel backbone untouched.

2) Transformer Module: Transformer module applies the standard Transformer decoder [30], transforming \( N \) embeddings using multi-head self- and cross-attention mechanisms. Following [29], BinsFormer decodes the \( N \) bins in parallel at each decoder layer, which serve as bins queries to interact with image features \( F \) and are transformed into an output embedding \( e_L \) by the decoder. Queries are zero initialized before being fed into the Transformer decoder and are associated with learnable positional embeddings. Without loss of generality, we denote the input query embeddings of the Transformer decoder as:

\[
e_0 = [e^1, e^2, \ldots, e^N],
\]

where each embedding \( e \) is randomly initialized following previous methods [29], [46], [47] and is mapped into a latent \( C_1 \)-dimensional bins embedding space using a trainable linear projection layer. There are \( L \) Transformer layers which consist of multi-headed cross-attention (MCA), multi-headed self-attention (MSA), and feed-forward network (FFN) blocks. At each layer \( \ell \), the input of the cross-attention block is a triplet of \( Q \) (query), \( K \) (key), and \( V \) (value), similar with [41], computed from \( f_1 \) and \( e_{\ell-1} \in \mathbb{R}^{N \times C_1} \) as:

\[
Q = e_{\ell-1} \times W_Q, K = f_1 \times W_K, V = f_1 \times W_V, \quad (3)
\]

where \( W_Q, W_K, W_V \in \mathbb{R}^{C_1 \times d} \) are the learnable parameters and \( d \) is the dimension of \( Q, K, V \). The cross-attention is calculated as:

\[
a = \left( \frac{Q \times K^T}{\sqrt{d}} \cdot V \right), \quad (4)
\]

where \( a \) is short for attention result and \( d \) is the dimension of the attention block. We also adopt the multi-head strategy in the attention result will be calculated \( m \) times by independent weight matrices. It can be formulated as:

\[
\text{MCA}(e_{\ell-1}) = e_{\ell-1} + \cat(a_1, a_2, \ldots, a_m) \times W_h, \quad (5)
\]

where \( W_h \in \mathbb{R}^{md \times C_1} \) and \( \cat \) is short for concatenation. After the cross-attention, we follow a self-attention to enable interaction among queries. The process is similar except the \( Q, K, V \) are generated by the query embeddings. The output of MSA is then transformed by the FFN block with residual skip as the layer output. Hence, we formulate a Transformer layer as:

\[
e_L = \text{FFN}(\text{MSA}(\text{MCA}(e_{\ell-1}, f_1))). \quad (6)
\]

Then, we apply a linear perceptron with softmax on top of the output embeddings \( e_L \) to yield \( N \) bins length \( b = [b_i]_i^{N} \). Moreover, we utilize a 3-layer perceptron with ReLU activation function on \( e_L \) to predict \( N \) bins embeddings \( f_b \in \mathbb{R}^{C \times N} \) to calculate the similarity with per-pixel representations \( f_p \) in the depth prediction module. As a result, the Transformer module \( M_{\text{trans}} \) can be formulated as:

\[
b, f_b = M_{\text{trans}}(e_0, f_1). \quad (7)
\]

We present the Transformer layer in Fig. 3 right for a clear understanding of the architecture.

---

**Fig. 2.** BinsFormer overview: We use a backbone and a pixel decoder to extract and upsample image features. A transformer decoder attends to multi-scale image features \( F \) and generate \( 1 + N \) output embedding. The first one predicts environment classification and the other \( N \) ones independently predict \( N \) bins lengths \( b \) and \( N \) bins embeddings \( f_b \), respectively. Then the model predicts the probability distribution map \( P \) via a dot product between pixel representations \( f_p \) and bins embeddings \( f_b \). Note, the dimensions for the dot \( \otimes \) are shown in gray. The final depth estimation is calculated by a linear combination between the probability distribution map \( P \) and post-processed \( N \) bins centers \( c(b) \).
3) Depth Prediction Module: Depth prediction module aggregates outputs from the pixel-wise feature extraction module and the Transformer module to predict depth. Given the predicted bins lengths 2 from the Transformer module, it first converts them to bins centers via a simple post-process following [15]:

\[
c(b_i) = d_{min} + (d_{max} - d_{min}) \left( \frac{b_j + i - 1}{2} + \sum_{j=1}^{i-1} b_j \right),
\]

where \(c(b_i)\) is center depth of the \(i^{th}\) bins, \(d_{max}\) and \(d_{min}\) are the max and the min valid depth values of the dataset, respectively.

Meanwhile, we obtain a similarity map via a dot product between the pixel-wise representations \(f_p\) from the pixel-wise feature extraction module and the bins embeddings \(f_b\) from the Transformer module. We then convert it to a probability distribution map \(P \in \mathbb{R}^{H \times W \times N}\) by a Softmax function. Finally, at each pixel, the final depth value \(\hat{d}\) is calculated from a linear combination of the probability distribution at that pixel and the depth-bin-centers \(c(b)\) as follows:

\[
\hat{d} = \sum_{i=1}^{N} c(b_i)p_i,
\]

Hence, the depth module \(M_{depth}\) can be written as

\[
\hat{D} = M_{depth}(f_p, f_b, b),
\]

where \(\hat{D}\) is the predicted depth map.

Compared to Adabins [15], we disentangle the bins generation and avoid bins embeddings and fine-grained per-pixel representations blushing each other. This enables us to predict more accurate depth without large-area failures, as can be seen in Fig. 6.

After predicting final depth maps, we apply a scaled version of the Scale-Invariant loss (SI) introduced by Eigen et al. [13]:

\[
L_{reg} = \alpha \frac{1}{T} \sum_i g_i^2 - \frac{\lambda}{T^2} \left( \sum_i g_i \right)^2,
\]

where \(g_i = \log \hat{d}_i - \log d_i\) with the ground truth depth \(d_i\) and predicted depth \(\hat{d}_i\). \(T\) denotes the number of pixels having valid ground truth values. Following [15], we use \(\lambda = 0.85\) and \(\alpha = 10\) for all our experiments.

4) Auxiliary Scene Classification: Auxiliary scene classification is an auxiliary subtask to provide implicit guidance to the bins generation. Beyond the bins embeddings, we equip the Transformer decoder with an extra scene query \(e_a\), which is used to classify the scene environment. As a result, the query embedding set in Eq. 2 is extended as:

\[
e = [e^0, e^1, e^2, \ldots, e^N].
\]

The auxiliary query can learn the global semantic information and transfer such knowledge through successive alternating self-attention to the bins queries. Similar to the bins embeddings, we adopt a 3-layer perceptron with ReLU activation function on the output embedding to yield the final classification \(f_a\). During training, a simple CrossEntropy classification loss \(L_{cls}\) is applied:

\[
L_{cls} = \text{CrossEntropy}(f_a, l_a),
\]

where \(l_a\) is the one-hot classification ground-truth.

Unlike Adabins [15], which applies chamfer loss to constrain the distribution of bins, our method avoids introducing such futile inductive bias. It means the supervision from the auxiliary classification subtask is implicit. The bins queries can adaptively absorb the global semantic information by self-attention with the scene query. This strategy only leads to a negligible overhead during the training process, compared with the computational pixel-wise chamfer loss.

5) Multi-Scale Prediction Refinement: Multi-scale prediction refinement is applied to obtain a global understanding of the image structure information and exploit a coarse-to-fine depth refinement. It is intuitively reasonable that the model can effectively square up the global information in low-resolution feature maps \(f_1\) where the structure information is well reserved while high-frequency details are discarded. However, since we combine the bins and the per-pixel representations to predict final depth, learning the fine-grained details in high-resolution feature maps is also crucial for high-quality depth estimation. Hence, we propose the multi-scale prediction refinement strategy, which can be seamlessly adopted with the help of the Transformer module.

As shown in Fig 3, we feed one resolution of the multi-scale feature \(F\) to one Transformer decoder layer at a time. Each scale of the Transformer decoder contains \(L\) Transformer layers that consists of a cross-attention module, a self-attention module and a FFN. During the forward propagation, queries
at scale $s$ first query the input image feature $f_s$ via a cross-attention module, and then aggregate information among themselves through a self-attention module. We also apply the depth estimation module at each Transformer layer to provide auxiliary supervision. Therefore, the total loss can be formulated as:

$$\mathcal{L}_{total} = \sum_{s=1}^{S} \left( w_s \sum_{l=1}^{L} \left( \mathcal{L}_{reg}^{s,l} + \mu \mathcal{L}_{cls}^{s,l} \right) \right),$$

where $\mu = 10^{-3}$ is a hyperparameter to balance the auxiliary classification loss and the depth estimation loss. $w$ is a scale weight, which is set to $\{0, 1, 2, \ldots, S\}$. By simply reweight the multi-scale punishment to estimation results, we seamlessly integrate the multi-scale refinement strategy to the Transformer module.

IV. EXPERIMENTS

In this section, we evaluate the performance of BinsFormer by comparing it against several baselines, starting by introducing datasets, evaluation metrics, and implementation details. Then, we present the comparison to the SoTA methods (containing a cross-dataset generalization evaluation), ablation studies, and uncertainty predictions.

A. Datasets and Evaluation Metrics

1) Datasets: We assess the proposed method using KITTI [32], NYU-Depth-v2 [31], and SUN RGB-D [33] datasets. KITTI is a dataset that provides stereo images and corresponding 3D laser scans of outdoor scenes captured by equipment mounted on a moving vehicle [32]. Following the standard Eigen training/testing split [13], we use around 26K images from the left view for training and 697 frames for testing. When evaluation, we use the crop as defined by Garg et al. [58] and upsample the prediction to the ground truth resolution. For the online KITTI depth prediction, we use the official benchmark split [59], which contains around 72K truth resolution. For the online KITTI depth prediction, we use the official 25 classes divided by folder names for the auxiliary scene understanding task. For KITTI, since the outdoor dataset is tough to classify, we omit the scene classification loss and only use ground truth depth to provide supervision.

2) Backbone: BinsFormer is compatible with any backbone architecture. In our work we use the standard convolution-based ResNet [62] backbones (ResNet-18 and ResNet-50, respectively) and recently proposed Transformer-based Swin-Transformer [34] backbones.

3) Pixel Decoder: As for the pixel decoder in Fig. 2, any depth estimation decoder can be adopted (e.g., [14], [26], [63]). There are numerous depth estimation methods use modules like ASPP [37] or CBAM [64] to capture long-range correspondings. Since our Transformer module attends to all image representations, collecting both the global and local information to generate bins, we can omit the computational context aggregation in per-pixel module. Therefore, following [47], a light-weight pixel decoder is applied based on the popular FPN network [65].

4) Transformer Decoder: We stack $L = 3$ Transformer layers for each scale of prediction refinement (i.e., 9 layers total) and 64 queries by default. The auxiliary loss is added to every intermediate Transformer decoder layer. Following [46], we adopt a simple deformable encoder [66] to enhance the multi-scale image features. In our experiments, we observe that BinsFormer is competitive for depth estimation with a single decoder layer as well.

C. Comparison With the SoTA

This section compares the proposed approach with the current SoTA MDE methods.

1) KITTI: We evaluate on the Eigen split [13] and Tab. I reports the results. BinsFormer achieves very competitive results, including NeWCRFs [53] and VA-DepthNet [54], which are the SoTA Swin backbones for MDE. Qualitative comparisons can be seen in the Fig. 4. We then evaluate the proposed method on the online KITTI depth prediction benchmark server\footnote{https://www.cvlibs.net/datasets/kitti/eval_depth.php?benchmark=depth_prediction} and report the results in Tab. II. While
TABLE I
COMPARISON OF PERFORMANCES ON THE KITTI DATASET. THE REPORTED NUMBERS ARE FROM THE CORRESPONDING ORIGINAL PAPERS. MEASUREMENTS ARE MADE FOR THE DEPTH RANGE FROM 0m TO 80m. BEST / SECOND BEST RESULTS ARE MARKED BOLD / UNDERLINED. E-B5 ARE SHORT FOR EFFICIENTNET-B5 [48], TAND † REPRESENT THE MODELS ARE PRE-TRAINED BY IMAGENET-22K AND AUXILIARY DEPTH ESTIMATION DATASET, RESPECTIVELY

| Method        | Ref     | Backbone | δ1† | δ2† | δ5† | RBL | Sq-rel | RMS | RMS log |
|---------------|---------|----------|------|------|------|-----|-------|-----|---------|
| Godard et al. [49] | CVPR 2017 | ResNet-50 | 0.861 | 0.849 | 0.976 | 0.114 | 0.898 | 4.935 | 0.206 |
| Johnston et al. [19] | CVPR 2020 | ResNet-101 | 0.889 | 0.962 | 0.982 | 0.106 | 0.861 | 4.699 | 0.185 |
| Gan et al. [50] | ECCV 2018 | ResNet-101 | 0.890 | 0.964 | 0.985 | 0.098 | 0.666 | 3.933 | 0.173 |
| DORN et al. [26] | CVPR 2018 | ResNet-101 | 0.932 | 0.984 | 0.994 | 0.072 | 0.307 | 2.727 | 0.120 |
| Yin et al. [51] | ICCV 2019 | ResNext-101 | 0.938 | 0.990 | 0.998 | 0.072 | - | 3.258 | 0.117 |
| PGA-Net [52] | TPAMI 2020 | ResNet-50 | 0.952 | 0.992 | 0.992 | 0.063 | 0.267 | 2.634 | 0.101 |
| BTS [14] | Arxiv 2019 | DenseNet-161 | 0.956 | 0.993 | 0.998 | 0.059 | 0.245 | 2.756 | 0.096 |
| TransDepth [25] | ICCV 2021 | ResNet50-ViT-B† | 0.956 | 0.994 | 0.999 | 0.064 | 0.232 | 2.755 | 0.098 |
| DPT [24] | ICCV 2021 | ResNet50-ViT-B† | 0.959 | 0.995 | 0.999 | 0.062 | 0.189 | 2.573 | 0.092 |
| AdaBins [15] | CVPR 2021 | EfficientNet-B5 | 0.964 | 0.995 | 0.999 | 0.058 | 0.190 | 2.360 | 0.088 |
| NeWCRFs [53] | CVPR 2022 | Swin-Large | 0.974 | 0.997 | 0.999 | 0.052 | 0.155 | 2.129 | 0.079 |
| VA-DepNet [54] | ICLR 2023 | Swin-Large | 0.977 | 0.997 | 0.999 | 0.050 | 0.148 | 2.093 | 0.076 |

Table II
COMPARISON OF PERFORMANCES ON THE KITTI DEPTH ESTIMATION BENCHMARK TEST SET. REPORTED NUMBERS ARE FROM THE OFFICIAL BENCHMARK WEBSITE

| Method         | SILog | SqRel | absErrRel | SqRMSE |
|----------------|-------|-------|-----------|-------|
| DORN [26]      | 11.77 | 2.23  | 8.78      | 12.98 |
| BTS [14]       | 11.67 | 2.21  | 9.04      | 12.33 |
| BA.Net [55]    | 11.55 | 2.31  | 9.34      | 12.17 |
| PWA [56]       | 11.45 | 2.30  | 9.05      | 12.32 |
| VGP-D [57]     | 10.80 | 2.19  | 8.94      | 11.77 |
| BinsFormer     | 10.14 | 1.69  | 8.23      | 10.90 |

a saturation phenomenon persists in SILog, BinsFormer still achieves 6.1% improvement on this metric.

2) NYU-Depth-v2: Tab. III lists the performance comparison results on the NYU-Depth-v2 dataset. While the performance of the SoTA models tends to approach saturation, NeWCRFs [53], VA-DepNet [54], and the proposed BinsFormer outperform all the rest competitors with prominent margins in all metrics. It indicates the effectiveness of our proposed methods. Qualitative comparisons can be seen in Fig. 6.

3) SUN RGB-D: Following Adabins [15], we conduct a cross-dataset evaluation by training our models on the NYU-Depth-v2 dataset and evaluating them on the test set of the SUN RGB-D dataset without any fine-tuning. As shown in Tab. IV, significant improvements in all the metrics indicate an outstanding generalization performance of BinsFormer. Qualitative results are shown in Fig. 7.

D. Ablation Studies
In this section, we adopt Swin-T as the default backbone and conduct experiments on the NYU dataset to demonstrate the effectiveness of each component in BinsFormer. We also provide various visualizations and in-depth discussions on experimental results.

1) BinsFormer: We first evaluate each component of BinsFormer as shown in Tab. V. We start with the per-pixel...
regression baseline. The Reg. Baseline uses the pixel-level module of BinsFormer and directly outputs per-pixel depth predictions (i.e., w/o Transformer and depth estimation module). For a fair comparison, we design the Reg. Baseline+, which adds the transformer module and query embedding MLP to the Reg. Baseline. The prediction is still from the per-pixel module (i.e., w/o depth estimation module). The transformer module is totally in line with the description in Sec. III-B.2, where only $f_1$ from per-pixel module will interact with the queries via the cross-attention. Then, for the BinsFormer (the 6th line in Tab. V), we add the depth estimation module and introduce the elaborate adaptive bins strategy for the classification-regression depth estimation. Here, in terms of classification-regression methods, we compare our bins generation method with pre-defined fixed UI/SID and Adabins [15]. Results presented in Tab. V demonstrate the effectiveness of

![Fig. 5. Qualitative results on KITTI online benchmark.](image_url)

**TABLE III**

| Method        | Encoder       | $\delta_1$ | $\delta_2$ | $\delta_3$ | REL | RMS | log10 |
|--------------|---------------|------------|------------|------------|-----|-----|-------|
| DORN [26]    | ResNet-101    | 0.828      | 0.965      | 0.992      | 0.115 | 0.509 | 0.051 |
| Yin et al. [51] | ResNeXt-101 | 0.875      | 0.976      | 0.994      | 0.108 | 0.416 | 0.048 |
| BTS [14]     | DenseNet-161  | 0.885      | 0.978      | 0.994      | 0.110 | 0.392 | 0.047 |
| DAV [23]     | DRN-D-22      | 0.882      | 0.980      | 0.996      | 0.108 | 0.412 |       |
| TransDepth [25] | Res-50+ViT-B+ | 0.900      | 0.983      | 0.996      | 0.106 | 0.365 | 0.045 |
| DPT [24]     | Res-50+ViT-B+ | 0.904      | 0.988      | 0.998      | 0.110 | 0.357 | 0.045 |
| AdaBins [15] | EfficientNet-B5 | 0.903  | 0.984      | 0.997      | 0.103 | 0.364 | 0.044 |
| NeWCRFs [53] | Swin-Large    | 0.922      | 0.992      | -          | 0.095 | 0.334 | 0.041 |
| VA-DepthNet [54] | Swin-Large | 0.937      | 0.992      | 0.998      | 0.086 | 0.304 | -     |

**TABLE IV**

| Method        | Encoder       | $\delta_1$ | $\delta_2$ | $\delta_3$ | REL | RMS | log10 |
|--------------|---------------|------------|------------|------------|-----|-----|-------|
| Chen et al. [67] | SENet [69]    | 0.757      | 0.943      | 0.984      | 0.166 | 0.494 | 0.071 |
| Yin et al. [51] | ResNeXt-101 | 0.696      | 0.912      | 0.973      | 0.183 | 0.541 | 0.082 |
| BTS [14]     | DenseNet-161  | 0.740      | 0.933      | 0.980      | 0.172 | 0.515 | 0.075 |
| Adabins [15] | EfficientNet-B5 | 0.771  | 0.944      | 0.983      | 0.159 | 0.476 | 0.068 |
| BinsFormer   | Swin-Tiny     | 0.760      | 0.945      | 0.985      | 0.162 | 0.478 | 0.069 |
| BinsFormer   | Swin-Large+   | 0.805      | 0.963      | 0.990      | 0.143 | 0.421 | 0.061 |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Fig. 6. Qualitative comparison on the NYU-Depth-v2 dataset.

BinsFormer. Moreover, the proposed auxiliary scene understanding query and the multi-scale architecture further boost the model performance with engaging margins.

As presented in the introduction and related work, we design the bins queries to correspondingly predict the length of bins. Since each query associates one bins length, it is interesting.
TABLE V
ABLATION STUDY RESULTS ON THE NYU DATASET. WE COMPARE BINSFORMER WITH REGRESSION BASED METHODS AND CLASSIFICATION-REGRESSION BASED METHODS WITH DIFFERENT BINS GENERALIZATION STRATEGIES. FURTHERMORE, WE INVESTIGATE THE EFFECTIVENESS OF EACH COMPONENT IN BINSFORMER

| Method          |Cls.-Reg.| Ada. Bins | Bins Query | Aux. Info. |Multi-S. | δ1↑ | REL↓ | RMS↓ |
|-----------------|---------|-----------|------------|------------|---------|-----|------|------|
| Reg. Baseline   |         |           |            |            |         | 0.852 | 0.130 | 0.422 |
| Reg. Baseline+  |         | ✓         |            | ✓          |         | 0.870 | 0.123 | 0.403 |
| Fix UD          | ✓       |           |            |            |         | 0.851 | 0.130 | 0.424 |
| Fix SID         | ✓       |           |            |            | ✓       | 0.825 | 0.145 | 0.453 |
| Adabins [15]    | ✓       | ✓         | ✓          |            | ✓       | 0.850 | 0.136 | 0.434 |
| BinsFormer      | ✓       | ✓         | ✓          | ✓          | ✓       | 0.878 | 0.116 | 0.397 |
|                 |         |           |            |            |         | 0.882 | 0.115 | 0.388 |
|                 |         |           |            |            |         | 0.890 | 0.113 | 0.379 |

Fig. 8. Visualization of different areas that queries are responsible for sensing, uncertainty maps, and probability distribution of selected points in images. We randomly select three points in the RGB images and plot their distributions in the Prob. plots, where Bins centers are presented by the small ticks upon the x-axis. Figure Query N visualizes the similarity of Nth query and the per-pixel representation $f_p$.

Fig. 9. Visualization of various distribution of predicted bins centers.

to investigate whether each query is responsible to perceive a certain depth range of input images and adapt to various scenes. Hence, we visualize the bins predictions of Fig. 9 and predicted probability distributions $P$ in Fig. 8. They indicate BinsFormer can adaptively estimate suitable bins for various dynamic scenes (e.g., for images containing large areas of distant pixels, predicted bins rather approaches the max depth). We highlight that we uniformly select four channels (8, 24, 50, 64) of $P$ and visualize the weight maps in Fig. 8. Since the $P$ is the softmaxed probability of the similarity between per-pixel representations and bins embeddings, the weight maps essentially indicate the sensing areas of corresponding bins. In line with our claim, ranked in a depth-increasing order, queries can correctly understand the scene structure and roughly respond their spatial locations via attention mechanisms.

Moreover, since we predict probability distribution maps for input images, it is possible to compute the measurement uncertainty [69] for each ray by measuring the Maximum Likelihood Estimates (MLE) following [70]. Fig. 8 shows a trend where uncertainty increases with distance, which has also been observed in unsupervised models that are capable of estimating uncertainty [19]. Areas of fringes show very high uncertainty, likely attributed to the drastic variation of depth values and the lack of depth cues in these regions.

2) Number of Bins: To study the influence of the number of bins, we train our network for various values of $N$ bins and measure the model performance. Results are presented in Tab. VI. The enhancing performance gained by increasing the number of queries diminishes above $N = 64$. Hence,

TABLE VI
ABLATION STUDY ON THE NYU DATASET: EFFECT OF NUMBER OF BINS (N) ON PERFORMANCE. SIMILAR TO ADABINS [15], WE OBSERVE THAT PERFORMANCE STARTS TO SATURATE AS N INCREASES ABOVE 64

| # of queries | δ1↑ | δ2↑ | δ3↑ | REL↓ | RMS↓ | log10b |
|--------------|-----|-----|-----|------|------|--------|
|              |     |     |     |      |      |        |
| 8            | 0.889 | 0.981 | 0.995 | 0.115 | 0.390 | 0.048 |
| 16           | 0.887 | 0.981 | 0.995 | 0.113 | 0.384 | 0.048 |
| 32           | 0.889 | 0.982 | 0.995 | 0.114 | 0.380 | 0.047 |
| 64           | 0.890 | 0.983 | 0.996 | 0.113 | 0.379 | 0.047 |
| 128          | 0.889 | 0.982 | 0.994 | 0.112 | 0.380 | 0.046 |
| 256          | 0.886 | 0.980 | 0.995 | 0.115 | 0.383 | 0.048 |
| 512          | 0.883 | 0.977 | 0.992 | 0.119 | 0.393 | 0.050 |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
we use \( N = 64 \) for our final model. On top of that, we exemplify different areas that queries are responsible for sensing in Fig. 8. Since queries and predicted bins are one-to-one correspondences, there is a strong correlation between depth and interest area of queries.

3) Multi-Scale Strategy: We then study the effectiveness of the proposed multi-scale strategy by changing the input features \( \mathbf{F} \) of the Transformer module. Results are presented in Tab. VII. Interestingly, the performance does not always improve by adding more scale information. In detail, the performance increases significantly with the scale of the feature increasing until the number reaches three. We also provide immediate scale predictions in Fig. 10, which demonstrates predictions get sharper and more accurate with the help of the proposed multi-scale refinement strategy. Also interestingly, from the visualization images, we observe that the results of scale 1 and scale 2 predictions tend to be more distant and closer, respectively. It indicates that the resolution of derived feature maps potentially influences the prediction.

4) Auxiliary Scene Classification: Results in Tab. V demonstrate that the auxiliary scene understanding task can improve the model performance by introducing implicit supervision. Furthermore, we visualize attention maps of the scene classification query \( e^\text{a} \) with the multi-scale features \( \mathbf{F} \) to investigate the principle of the auxiliary scene classification. As shown in Fig. 10, the classification query considers different areas of features at different scales and fully aggregates spatial information to provide hints for bins embeddings via self-attention in the Transformer module.

![Fig. 10. Visualization of depth estimation results and attention maps of the scene understanding query at the progressive multi-scale prediction refinement process.](image)

**TABLE VII**

Ablation Study on the NYU Dataset: Performance of BinsFormer for Different Scales Refinement

| \( f^a \) | # layers | \( \Delta_1^\text{a} \) | \( \Delta_2^\text{a} \) | \( \Delta_3^\text{a} \) | REL\(_\text{a}^\dagger\) | RMS\(_\text{a}^\dagger\) |
|---|---|---|---|---|---|---|
| \( f_1 \) | 3 | 0.882 | 0.980 | 0.995 | 0.115 | 0.388 |
| \( f_2, f_4 \) | 6 | 0.888 | 0.982 | 0.995 | 0.113 | 0.380 |
| \( f_1, f_2, f_3, f_4 \) | 9 | 0.890 | 0.983 | 0.996 | 0.113 | 0.379 |
| \( f_1, f_2, f_3, f_4 \) | 12 | 0.881 | 0.979 | 0.984 | 0.118 | 0.392 |

**TABLE VIII**

Fair Comparisons of the Effectiveness and Efficiency With Previous SoTA Methods by Aligning Encoders on NYU Dataset

| Method | Encoder | \( \Delta_1^\dagger \) | REL\(_{\text{FPS}}^\dagger \) | FPS\(_\text{FPS}^\dagger \) |
|---|---|---|---|---|
| Adabins [15] | EfficientNet-B5 | 0.903 | 0.103 | 11.48 |
| Ours | EfficientNet-B5 | 0.899 | 0.097 | 7.53 |
| Adabins [15] | ResNet-50 | 0.850 | 0.136 | 13.79 |
| Ours | ResNet-50 | 0.890 | 0.113 | 10.52 |
| DepthFormer [71] | Swin-T | 0.891 | 0.120 | 13.36 |
| BinsFormer | Swin-T | 0.890 | 0.113 | 14.06 |
| Adabins [15] | Swin-L | 0.917 | 0.102 | 7.63 |
| Ours | Swin-L | 0.925 | 0.094 | 5.28 |

E. Fair Comparison

We observe that previous methods adopt various backbones, leading to an unfair comparison on both fidelity and inference speed. In this section, we align the backbone with other methods and measure the inference speed on the same device to achieve fair comparisons. As shown in Tab. VIII, BinsFormer runs a little slower than the SoTA method with convolution-based architecture (Adabins) but faster than the one with Transformer designs (DepthFormer [71]). More importantly, BinsFormer achieves superior qualitative results with large improvement margins compared with previous methods.

V. Conclusion

In this paper, we propose a novel classification-regression based monocular depth estimation (MDE) method, called BinsFormer, that incorporates a Transformer module to prediction bins in a set-to-set manner, a per-pixel module to estimate high-resolution pixel-wise representations, and a depth estimation module to aggregate information to predict final depth maps. BinsFormer can adaptively generate bins and per-pixel probability distribution for accurate depth estimation. We propose an auxiliary scene understanding task and a multi-scale prediction refinement strategy that can be seamlessly integrated into the Transformer module. These two methods further boost model performance and only introduce negligible overhead. The performance of BinsFormer achieves...
new state-of-the-art on two popular benchmark datasets. Moreover, the generalization ability of the method was further demonstrated in cross dataset experiments.

REFERENCES

[1] C. Zhao, Q. Sun, C. Zhang, Y. Tang, and F. Qian, “Monocular depth estimation based on deep learning: An overview,” Sci. China Technol. Sci., vol. 63, no. 9, pp. 1612–1627, Sep. 2020.

[2] T. Wang, J. Pang, and D. Lin, “Monocular 3D object detection with depth from motion,” 2022, arXiv:2207.12988.

[3] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, “Categorical depth distribution network for monocular 3D object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognition (CVPR), Jun. 2021, pp. 8555–8566.

[4] Y. Wang and K. Kitani, “Monocular 3D object detection with pseudo-LiDAR point cloud,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW), Oct. 2019, pp. 857–866.

[5] Z. Li et al., “Unsupervised domain adaptation for monocular 3D object detection via self-training,” 2022, arXiv:2204.11590.

[6] Y. Wang, V. C. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, “DETR3D: 3D object detection from multi-view images via 3D-to-2D queries,” in Proc. ICLR, 2022, pp. 180–191.

[7] C. Armbruster, M. Wolter, T. Kahlen, W. Spjikkers, and B. Finn, “Depth perception in virtual reality: Distance estimations in peri- and extrapersonal space,” CyberPsychology Behav., vol. 11, no. 1, pp. 9–15, Feb. 2008.

[8] N. Gerig, J. Mayo, K. Baur, F. Wittmann, R. Riener, and P. Wolf, “Missing depth cues in virtual reality limit performance and quality of three dimensional reaching movements,” PLoS ONE, vol. 13, no. 1, Jan. 2018, Art. no.e189275.

[9] S. Zhu, G. Brazil, and X. Liu, “The edge of depth: Explicit constraints between segmentation and depth,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognition (CVPR), Jun. 2020, pp. 13113–13122.

[10] W. Wang and U. Neumann, “Depth-aware CNN for RGB-D segmentation,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 135–150.

[11] W. Zhang, L. Su, A. Geiger, T. Meinhardt, A. Kirillov, and C. Damokos, and D. Cremers, “FuseNet: Incorporating depth into semantic segmentation via fusion-based CNN architecture,” in Proc. ACCV, 2016, pp. 213–228.

[12] T.-H. Yu, H. Jain, M. Bucher, M. Cord, and P. P. Pérez, “DADA: Depth-aware domain adaptation in semantic segmentation,” in Proc. ICCV, 2019, pp. 7364–7373.

[13] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in Proc. NeurIPS, vol. 25, no. 9, pp. 3691–3702, Aug. 2018.

[14] C. Godard, O. M. Aodha, M. Firman, and G. Brostow, “Digging into self-supervised monocular depth estimation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 3828–3838.

[15] A. Johnston and G. Carneiro, “Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognition (CVPR), Jun. 2021, pp. 4009–4018.

[16] X. Xu, Z. Chen, and F. Yin, “Multi-scale spatial attention-guided monocular depth estimation with semantic enhancement,” IEEE Trans. Image Process., vol. 30, pp. 8811–8822, 2021.

[17] Z. Zhang, C. Xu, J. Yang, J. Gao, and Z. Cui, “Progressive hard-mining network for monocular depth estimation,” IEEE Trans. Image Process., vol. 25, no. 8, pp. 3691–3702, Aug. 2018.

[18] C. Godard, V. Kalogerakis, and G. Brostow, “Dense prediction,” in Proc. Int. Conf. Pattern Recognit. (ICPR), Sep. 2021, pp. 1334–1337.

[19] A. Johnston and G. Carneiro, “Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognition (CVPR), Jun. 2020, pp. 4756–4765.

[20] X. Song et al., “MLDA-Net: Multi-level dual attention-based network for self-supervised monocular depth estimation,” IEEE Trans. Image Process., vol. 30, pp. 4691–4705, 2021.

[21] Y. Kuznetsov, J. Buckler, and B. Leibe, “Semi-supervised deep learning for monocular depth map prediction,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognition, Jun. 2017, pp. 6647–6655.

[22] V. Guizilini, J. Li, R. Ambrus, S. Pillai, and A. Gaidon, “Robust semi-supervised monocular depth estimation with reprojected distances,” in Proc. CoRL, 2020, pp. 503–512.
M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 6105–6114.

C. Godard, O. M. Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 270–279.

Y. Gan, X. Xu, W. Sun, and L. Lin, “Monocular depth estimation with affinity, vertical pooling, and label enhancement,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Oct. 2018, pp. 224–239.

W. Yin, Y. Liu, C. Shen, and Y. Yan, “Enforcing geometric constraints of virtual normal for depth prediction,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., Oct. 2019, pp. 5684–5693.

D. Xu, X. Alameda-Pineda, W. Ouyang, E. Ricci, X. Wang, and N. Sebe, “Probabilistic graph attention network with conditional kernels for pixel-wise prediction,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 5, pp. 2673–2688, May 2022.

W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan, “Neural window fully-connected CRFs for monocular depth estimation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), New Orleans, LA, USA, Jun. 2022, pp. 3916–3925.

R. Garg, B. G. V. Kumar, G. Carneiro, and I. Reid, “Unsupervised CNN for single view depth estimation: Geometric to the rescue,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 740–756.

J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, “Sparsity invariant CNNs,” in Proc. Int. Conf. 3D Vis. (3DV), Oct. 2017, pp. 11–20.

A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” in Proc. 33rd Conf. Neural Inf. Process. Syst., 2019, pp. 8026–8037.

D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

I. Alhashim and P. Wonka, “High quality monocular depth estimation via transfer learning,” 2018, arXiv:1812.11941.

S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: Convolutional block attention module,” in Proc. Eur. Conf. Comput. Vis., Sep. 2018, pp. 3–19.

T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 2117–2125.

X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable DETR: Deformable transformers for end-to-end object detection,” in Proc. ICLR, 2021.

X. Chen, X. Chen, and Z.-J. Zha, “Structure-aware residual pyramid network for monocular depth estimation,” 2019, arXiv:1907.06023.

J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proc. CVPR. Jun. 2018, pp. 7132–7141.

A. Kendall and Y. Gal, “What uncertainties do we need in Bayesian deep learning for computer vision?” in Proc. NeurIPS, 2017, pp. 5574–5584.

C. Liu, J. Gu, K. Kim, S. G. Narasimhan, and J. Kautz, “Neural RGB-D sensing: Depth and uncertainty from a video camera,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 10986–10995.

Z. Li, Z. Chen, X. Liu, and J. Jiang, “DepthFormer: Exploiting long-range correlation and local information for accurate monocular depth estimation,” 2022, arXiv:2203.14211.

Zhenyu Li received the B.S. and M.S. degrees in computer science from Harbin Institute of Technology, China, in 2021 and 2023, respectively. He is currently pursuing the Ph.D. degree with the King Abdullah University of Science and Technology (KAUST), Saudi Arabia. His research interests include depth estimation and 3D object detection.

Xuyang Wang received the B.S. degree in electrical engineering and automation from Harbin Institute of Technology, China, in 2020, and the M.S. degree from Australian National University, Canberra, ACT, Australia, in 2023, where she is currently pursuing the Ph.D. degree. Her research interests include 3D vision.

Xianming Liu (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from Harbin Institute of Technology (HIT), Harbin, China, in 2006, 2008, and 2012, respectively. In 2011, he spent half a year at the Department of Electrical and Computer Engineering, McMaster University, Canada, as a Visiting Student, where he was a Postdoctoral Fellow from 2012 to 2013. He was a Project Researcher with the National Institute of Informatics (NII), Tokyo, Japan, from 2014 to 2017. He is currently a Professor with the School of Computer Science and Technology, HIT. He was a recipient of the IEEE ICME 2016 Best Student Paper Award.

Junjunjiang (Senior Member, IEEE) received the B.S. degree in mathematics from Huazhao University, Quanzhou, China, in 2009, and the Ph.D. degree in computer science from Wuhan University, Wuhan, China, in 2014. From 2015 to 2018, he was an Associate Professor with the School of Computer Science, China University of Geosciences, Wuhan. From 2016 to 2018, he was a Project Researcher with the National Institute of Informatics (NII), Tokyo, Japan. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China. His research interests include image processing and computer vision.