GLPanoDepth: Global-to-Local Panoramic Depth Estimation
Jayyang Bai, Haoyu Qin, Shuichang Lai, Jie Guo, and Yanwen Guo

Abstract—Depth estimation is a fundamental task in many vision applications. With the popularity of omnidirectional cameras, it becomes a new trend to tackle this problem in the spherical space. In this paper, we propose a learning-based method for predicting dense depth values of a scene from a monocular omnidirectional image. An omnidirectional image has a full field-of-view, providing much more complete descriptions of the scene than perspective images. However, fully-convolutional networks that most current solutions rely on fail to capture rich global contexts from the panorama. To address this issue and also the distortion of equirectangular projection in the panorama, we propose Cubemap Vision Transformers (CViT), a new transformer-based architecture that can model long-range dependencies and extract distortion-free global features from the panorama. We show that cubemap vision transformers have a global receptive field at every stage and can provide globally coherent predictions for spherical signals. As a general architecture, it removes any restriction that has been imposed on the panorama in many other monocular panoramic depth estimation methods. To preserve important local features, we further design a convolution-based branch in our pipeline (dubbed GLPanoDepth) and fuse global features from cubemap vision transformers at multiple scales. This global-to-local strategy allows us to fully exploit useful global and local features in the panorama, achieving state-of-the-art performance in panoramic depth estimation.

Index Terms—Depth estimation, panorama, transformer, cubemap, fusion.

I. INTRODUCTION

THE boom of numerous 3D vision applications (e.g., autonomous driving [3], [26], [52], 3D scene reconstruction [38], [69] and human motion analysis [6]) has witnessed the effectiveness of adopting depth as an important modality besides RGB color. To obtain the depth information, one could resort to some depth sensors such time-of-flight (TOF) cameras or could infer it from RGB images. Compared with capturing depth from specialized depth sensors, it is more attractive to infer depth from color images since they are still the only sources in many scenarios. Much effort has been devoted to solve this ill-posed problem for single perspective image (image captured by conventional cameras with a limited field-of-view). Among them, deep learning-based methods have achieved very promising results [24], [27], [34].

With the emergence of consumer-level omnidirectional (360°) cameras (e.g., Ricoh Theta, Samsung Gear360 and Insta360 ONE), 360° panoramas are currently experiencing a surge in popularity, making depth estimation from these panoramas a hot topic both in academia and in industry [13], [16], [21], [22], [37], [50], [55], [56], [62], [67], [68]. However, directly applying existing deep neural networks trained on perspective images to omnidirectional images has been shown to achieve sub-optimal performance due to the space-varying distortion of equirectangular projection [51], [68]. Consequently, some methods try to reduce the distortion by specially-designed convolution operations [51] or by combining features from distortion-free cubemaps [21], [55]. Moreover, compared with perspective images, 360° panoramas provide a rich source of information about the entire scene around the viewer. Many neural networks fail to capture the rich features [64] that are beneficial for high-quality depth estimation. A recent neural network named SliceNet [37] tries to exploit the full-geometric context involved in a single panorama. Slicing is performed over multiple levels to preserve global information in the panorama. However, it assumes that the equirectangular image should be aligned to the gravity vector, which does not always hold in practice.

In this paper, we tend to address the above two issues (i.e., space-varying distortion and rich global information) simultaneously for panoramic depth estimation, avoiding any strong assumptions on the scenes. Since fully-convolutional networks have a limited receptive field which grows linearly with the depth of the network, they will fail to capture long-range relations and global features that are rich in panoramas [64]. Considering this, we resort to the vision transformer (ViT) [12], a new backbone architecture that is particularly effective at modeling long-range dependencies over all pixels in the input image. Generally, ViT has a global receptive field at every stage. However, directly applying ViT for panoramic depth estimation does not result in higher accuracy than equal-sized convolutional neural networks (CNN) since ViT lacks 2D local structures, i.e., spatial correlations between neighboring pixels. Moreover, ViT also fails to handle distortion in omnidirectional images. Considering these two issues, we design a two-stream network dubbed GLPanoDepth which incorporates two branches to extract global and local features, respectively. First, we adopt cubemap projection and construct a new Cubemap Vision Transformers (CViT) to...
Fig. 1. We propose a global-to-local approach (i.e., GLPanoDepth) to recover a dense depth map from a single panorama. Benefited from the rich global information that have been extracted, the proposed method is able to preserve both global structures and local details as shown in 2D depth maps and 3D point clouds, outperforming previous methods (e.g., BiFuse [55] and SliceNet [37]).

To summarize, the contribution of this paper is threefold:

- We propose a two-stream network named GLPanoDepth to recover dense depth values from a single omnidirectional image, without imposing any restriction on the input panorama.
- We introduce Cubemap Vision Transformers, a general architecture designed for panoramas that can capture long-range relations and distortion-free global features from wide-field spherical signals.
- We employ a Gated Fusion Module to better fuse global and local features from input panoramas.

We have tested our method on multiple datasets. Comparison with the state-of-the-art approaches and extensive ablation studies validate the effectiveness of the proposed method.

II. RELATED WORK

The goal in monocular depth estimation is to predict the depth value of each pixel, given only a single RGB image as input. There has been much work which contributes to this field. In this section, we will review previous methods that closely related to our work.

A. Depth Estimation for Perspective Images

One of the first learning-based methods in monocular depth estimation for perspective images is proposed by Saxena et al. [42], using a patch-based model and Markov random fields. After that, many approaches have been presented using hand crafted representations [17], [40], [43]. With the rapid development of deep learning, the performance of depth estimation from a single image has been significant improvement. Eigen et al. [15] first applied convolutional neural networks to monocular depth estimation. Following this work, monocular depth estimation based on depth learning has gradually become a research hotspot. Eigen and Fergus [14] extended their work [15] with a deeper and more discriminative model, using VGG features. Lately, Laina et al. [28] proposed FCRN which uses ResNet [20] as the encoder and utilizes an up-projection module for upsampling, along with the reverse Huber loss [29]. Lee et al. [30] also designed a ResNet-based depth estimation network, and resorted to Fourier analysis to well solve the single-image depth estimation problem. As a useful tool, conditional random fields (CRF) have shown great power in optimizing depth estimation results [4], [11], [32], [57]. To ease the requirement of ground-truth depth images, unsupervised training methods are developed [18], [60], [63], [65]. However, unsupervised training methods often achieve sup-optimal results compared with supervised training methods. Similar to ours, Ranftl et al. [39] also tried to recover dense depth values from a single image by vision transformers. However, they only focused on distortion-free perspective images. For a comprehensive review on depth estimation for perspective images, please refer to some recent surveys [24], [27], [34].

B. Depth Estimation for Omnidirectional Images

Perspective images contain limited geometric contexts because of the restricted field-of-view [64]. With the popularity of omnidirectional cameras, researches on depth estimation for omnidirectional images have emerged [56], [64], [70]. It has been shown that directly applying existing deep learning methods designed for perspective images to the omnidirectional domain achieves sub-optimal results. ACDNet [66] introduces adaptively combined dilated convolutions as a replacement for regular convolutions to effectively extract features from panoramic images. Nonetheless, the inherent distortion in panoramas still hinders the accuracy of depth estimation. To deal with the distortion of equirectangular projection in omnidirectional images, some methods utilize cubemaps as an additional cue. Cheng et al. [8] first suggested to convert an equirectangular image into a cubemap to address the issue of space-varying distortion, while keeping the connection...
between each adjacent face of the cubemap. Wang et al. [54] also adopted the cubemap representation in a self-supervised learning manner. To properly combine features from different representations of spherical signals, specially designed fusion strategies are proposed [21], [55]. Another way to handle distortion is by specially-designed spherical convolutions [9], [10], [48], [49]. Several methods use them to make the networks explicitly aware of the distortion. By incorporating spherical convolutions [48], Zioulis et al. [68] utilized omnidirectional content in the form of equirectangular images to perform depth estimation. Most recently, Pintore et al. [37] achieved remarkable results using a slice-based representation of omnidirectional images. They introduced a LSTM multi-layer module [45] to effectively recover long and short term spatial relationships between slices. Sun et al. [50] proposed a Latent Horizontal Feature representation (LHFFeat) and HoHoNet to recover per-pixel depth from an indoor 360° panorama. However, both SliceNet [37] and and HoHoNet [50] rely heavily on the assumption that the gravity direction is aligned with the image’s y-axis. In comparison, our method is free from this restriction and can handle panoramas with arbitrary gravity directions.

C. Vision Transformers

Transformers [53] have recently achieved tremendous success in a wide range of tasks in natural language processing and have recently outperformed their convolutional counterparts in many vision applications, including image classification [2], [7], semantic segmentation [58], 3D object detection [35] and action anticipation [41]. The vision transformer (ViT) [12] is the first to validate the feasibility of pure transformer architectures for vision tasks. To lower the computational cost and better model local context in ViT, some recent methods have introduced some design changes. For instance, Transformer-in-transformer (TNT) [19] utilizes an outer transformer to process the patch embeddings and an inner transformer to model the relation between pixel embeddings, thus allowing to model both patch-level and pixel-level representation. Swin transformer [33] achieves high efficiency by limiting self-attention computation to non-overlapping local windows with a hierarchical transformer architecture that is computed with shifted windows. A recent work [39] also tries to recover depth values from a single image by leveraging ViT. However, it only handles perspective images. 360Joint [61] employs transformers for panoramic depth estimation and trains the model using a combination of supervised and self-supervised learning. It introduces a non-local fusion block to retain global information from the vision transformer. Nonetheless, it does not consider the distortion in panoramas when applying transformers, which limits its performance. HiMODE [23] proposes a hybrid architecture, which comprises a CNN-based feature extractor and a transformer to effectively reduce the artifacts and recover the surface depth data. Nonetheless, both 360Joint and HiMODE neglect the distortion inherent in equirectangular projection and directly input features or patches from equirectangular images into transformers, potentially hindering their performance. OmniFusion [59] addresses this concern by projecting inputs onto multiple patches at multiple viewpoints. These distortion-free patches are processed with an encoder-decoder network to produce patch-wise depth maps, which are then merged into final equirectangular depth maps. However, it fails to completely remove the artifacts caused by patch-wise discrepancy, especially in a cross-domain setup. In contrast, we fuse local features with global features with a gated fusion module, enabling the direct generation of equirectangular images.

III. Method

A. Motivation and Overview

Our goal is to predict a dense depth map from an omnidirectional image. With the advent of practical deep learning, many methods apply convolutional neural networks to extract and decode the features from the input image. Unfortunately, these convolutional neural networks are not well suited for omnidirectional or spherical images. Convolutions are linear operations that have a limited receptive field. They can hardly acquire sufficiently broad context as well as significant high-level representations from omnidirectional images. The limited receptive field and the limited expressiveness of an individual convolution operation necessitate sequential stacking many convolutional layers into very deep architectures. In contrast, vision transformers [12] outperform CNNs in long-range feature learning. However, they have difficulty in learning local structures especially for panoramas with heavy space-varying distortions.

In view of these challenges, we design a two-stream network, named GLPanoDepth, for panoramic depth estimation. An overview of the complete architecture is shown in Fig. 2. The key idea of our method is to extract global and local features with two separate branches, which leads to fine-grained and globally coherent prediction. To extract global features, we employ cubemap vision transformers (CViT), a new transformer-based architecture which learns long-range contextual relationships and distortion-free global features through multi-head self-attention (MHSA) [53] and multi-layer perceptrons (MLP). As transformers are set-to-set functions, they do not perform well in retaining the information of the spatial positions and local features. To address this issue, we leverage a convolution-based subnetwork to learn highly correlated local structures from panoramas. Finally, a gated fusion module is designed to progressively combine features from two branches into the final dense prediction.

Our network has at least two novel designs. First, the combination of cubemap projection and ViT addresses the distortion issue and preserves global structures simultaneously, showing better performance than spherical CNN or classic ViT. Second, our Gated Fusion Module takes full use of respective advantages of transformer and convolution, progressively fusing global and local features into final dense prediction.

B. Network Architecture

As shown in Fig. 2, our network can be logically separated to an encoder part and a decoder part. The encoder consists of a CViT branch and a CNN branch, both of which provide
features at various resolutions. Given an input panorama \( P \) with a \( H \times W \) resolution, the CViT branch is designed to capture global features \( F_G \) while the CNN branch is expected to learn local features \( F_L \). The decoder then progressively combine these features into a full resolution prediction using the gated fusion module.

**CViT Branch** We leverage vision transformers [12] as the backbone of CViT branch. Vision transformers are believed to capture global representations among the compressed patch embeddings with dynamic attention, global context fusion and better generation. Due to the distortion in omnidirectional images, our CViT branch takes the cubemap projection as the input. Specifically, we first reproject the input panorama \( P \) into a corresponding cubemap \( C \in \mathbb{R}^{6 \times \frac{H}{2} \times \frac{W}{2} \times 3} \), where 6 means six faces corresponding to the planes on the back, down, front, left, right and up. Note that the image of each face is distortion-free while all the faces form a complete 360° field-of-view. Then, we split the images from the six faces into a sequence of flattened patches of size \( p \times p \) pixels. All these non-overlapping square patches are flattened into vectors and individually embedded using a linear projection, resulting in \( N_p = (6 \times \frac{H}{2} \times \frac{W}{2}) / p^2 \) tokens in total.

Since we have lost the information of the spatial positions of individual tokens during flatting, we add a learnable position embedding to the image embeddings. After that, input tokens are transformed into multi-scale representations \( T_k \) using \( K \) transformer blocks. \( T_k \) is the output of \( k \)-th transformer block. Fundamentally, each transformer block contains two basic operations, as shown in the left part of Fig. 3. First, a multi-head self-attention operation is adopted to model inter-element relations. This attention-based operation is the key to capture long-range contextual relationships in the panorama. Second, an MLP is used to model relations within each element. These two operations are often intertwined with layer normalization and residual connections. Since each transformer block is performed on tokens from all six faces, it naturally addresses the discontinuity issue of cubemap projection.

To reassemble tokens from different transform blocks, we map the set of tokens into image-like feature representations. The details of reassembling is provided in the right part of Fig. 3. Formally, we apply a linear projection and spatial reshape operation to adjust \( N_p \) \( C \)-channel tokens \( T_k \in \mathbb{R}^{N_p \times C} \) into a feature map \( F_L = \mathbb{R}^{H \times W \times C} \). Since the spatial reshape operation has limited information for scales, we add a recover layer to scale the representation. This layer is implemented using \( 1 \times 1 \) convolutions and pixelshuffle [44]. Through this layer, feature maps from earlier blocks have higher resolutions and have a global receptive field. Currently, we use 12 transformer blocks, i.e., \( K = 12 \), in the CViT branch. We only extract global feature maps from the fourth, seventh, tenth and twelfth transformer blocks, i.e.,

\[
F_G = \{ F_{T_4}, F_{T_7}, F_{T_{10}}, F_{T_{12}} \}. \tag{1}
\]

These feature maps will be fed into the decoder.

**CNN Branch** The CNN branch is devoted to capture local relationships using local receptive fields and shared weights in CNN. It takes the panorama \( P \) with equirectangular projection as the input and outputs local feature maps \( F_L \) with the same size as those from the CViT branch. Currently, we do not choose any spherical variant of convolution to handle distortion in the panorama because our ultimate goal is to
produce an equirectangular depth map. The CViT branch is responsible for correcting distorted regions.

In the CNN branch, we have designed four residual blocks [20] and downsampling layers to extract local features from detail-rich shallow layers and context-critical deep layers. In the residual block, two $3 \times 3$ convolutional layers are used to address limited-receptive-field learning for specific local details, each followed by a rectified linear unit (ReLU). We apply a $3 \times 3$ convolutional layer with stride 2 for downsampling. Our CNN branch extracts a hierarchical ensemble of local features with multiple limited reception fields from different layers. These features, denoted as $\mathcal{F}_L$, have the same resolution as $\mathcal{F}_G$ from the CViT branch.

**Gated Fusion Module** To combine information from two branches, we propose a gated fusion module to adaptively adjust the weights assigned to global and local features. As illustrated in Fig. 4, we first fuse feature maps ($\mathcal{F}_G^l$ and $\mathcal{F}_L^l$) from $l$-th layer of the two branches via an element-wise summation, and then pass the result to two convolutional layers and a sigmoid function to generate a gate map $G^l$:

$$G^l = \text{Sigmoid}(\text{Conv}(\text{Conv}(\mathcal{F}_G^l + \mathcal{F}_L^l))) \quad (2)$$

$G^l$ ranges from 0 to 1, serving as a gate that select different features from the two branches. We use $G^l$ and $1 - G^l$ as soft attention matrices for $\mathcal{F}_G^l$ and $\mathcal{F}_L^l$ respectively. Then, we obtain a fused feature map $\mathcal{F}_{fuse}^l$ as

$$\mathcal{F}_{fuse}^l = \mathcal{F}_G^l \otimes G^l + \mathcal{F}_L^l \otimes (1 - G^l) \quad (3)$$

where $\otimes$ denotes element-wise product.

Our gated fusion module has two fundamental differences against bi-projection fusion module proposed by Wang et al. [55]. First, we fuse different feature maps into a single decoder—instead of two—to produce an omnidirectional image directly. Second, the feature maps generated by the CViT branch do not require a cubemap to equirectangular transformation and suffer less from inconsistency in cubemap boundaries thanks to the transformer blocks.

**Discussion** To show the necessity and benefit of fusing features from both CViT branch and CNN branch, we visualize the attention weights of extracted feature maps from pure CNN, fused feature maps from ViT+CNN, and fused feature maps from CViT+CNN (our complete model) in Fig. 5, respectively. Since CNN has a limited receptive field, the extracted features from pure CNN only focus on local correlations in the panorama. Therefore, important global structures are overwhelmed by excessive local details, as shown in the second column of Fig. 5. Although ViT can capture long-range relations, it fails to preserve clear edges in the panorama due to the distortion raised by equirectangular projection, as shown in the third column of Fig. 5. Thanks to the cubemap projection, CViT can extract distortion-free global features and learn global contextual representations much better. As seen in the fourth column of Fig. 5, the fused feature map from both CViT branch and CNN branch preserves clear global structures and sufficient local details.

**C. Loss Function and Training Details**

Following most recent works on panoramic depth estimation, we adopt the BerHu loss [28] as the objective function in training:

$$L(y, \hat{y}) = \begin{cases} |y - \hat{y}|, & |y - \hat{y}| \leq T \\ (|y - \hat{y}|^2 + T^2) / (2T), & |y - \hat{y}| > T \end{cases} \quad (4)$$

where $y$ is the ground truth depth value and $\hat{y}$ is the prediction. The threshold $T$ is set to 0.2 in our experiments.

We implement our GLPanoDepth using the PyTorch framework [36]. Adam [25] optimizer is used with the default learning rate of 0.0001. Our model is trained on two NVIDIA RTX 3090 GPUs with a batch size of 8. We jointly train two branches for 80 epochs. Our network contains 120 million trainable parameters. It costs 77 ms and 2.87 GB GPU memory to infer a depth panorama with a resolution of $1024 \times 512$.

**IV. EXPERIMENTS**

In this section, we present extensive experimental results to validate the proposed method. We report both quantitative and qualitative results of three large-scale datasets and compare our method with state of the arts. Ablation studies are carried out to validate our design choices.

**A. Datasets and Evaluation Metrics**

We conduct our experiments mainly on three benchmark datasets: 360D [68], Matterport3D [5] and Stanford2D3D [1]. Matterport3D and Stanford2D3D are real-world datasets collected by Matterport’s Pro 3D Camera. Matterport3D contains 10,800 panoramas and their corresponding depth maps. Stanford2D3D contains 1,413 panoramas collected from six large
scale indoor areas of three kinds of buildings in the real world. As ToF sensors usually cause noise or missing value in certain areas, the ground-truth depth maps in Matterport3D and Stanford2D3D are incomplete and even inaccurate in some areas. Following many recent approaches [37], [55], we resize the resolution of images and depth maps in Matterport3D and Stanford2D3D into 512 × 1024. We use official splits which take some rooms for training and the others for testing for both datasets. 360D comprises multi-modal stereo renders of scenes from realistic and synthetic large-scale 3D datasets including Matterport3D [5], Stanford2D3D [1] and SunCG [47]. In total, 360D contains 35,977 panoramas, where 33,879 of them are used for training, 800 used for validation and the rest is for testing. We use the split from Zioulis et al. [68] and the resolution is resized to 512 × 256.

To evaluate the performance, we use standard metrics (listed in Table I) including mean absolute error (MAE), root mean square error (RMSE) and the root mean square error in log space (RMSElog). In addition, we calculate the percentages of pixels where the ratio between the estimated depth and ground truth depth is smaller than the thresholds \( \delta \) to evaluate the accuracy.

### B. Comparisons With Previous Methods

For fair and reproducible experiments, we quantitatively and qualitatively compare our method with three baselines in this field: FCRN [28], OmniDepth [68] and BiFuse [55]. We also make comparisons with four state-of-the-art methods: SliceNet [37], HoHoNet [50], ACDNet [66] and OmniFusion [59], both of which also try to handle rich global contexts in omnidirectional images.

1) Quantitative Comparisons: We first show quantitative comparisons on three benchmark datasets in Table I. Generally, our method improves state-of-the-art performance of dense panoramic depth estimation for many numerical metrics on these three datasets. In particular, our method outperforms three baselines (FCRN, OmniDepth and BiFuse) in terms of accuracy for all metrics. Note that these three baselines do not take rich global features in the panoramas into consideration.
since they are all based on some kind of local convolution operations. Among the three datasets, four state-of-the-art methods showcase their efficacy on individual datasets. In contrast, our approach demonstrates robust competitiveness across all three datasets simultaneously, with the majority of evaluation metrics ranking within the top three. Moreover, our method exhibits outstanding generalization capabilities, particularly on the 360D dataset. Both the Matterport3D and Stanford2D3D datasets contain incomplete or inaccurate areas in their ground-truth depth maps and a limited number of training samples, which potentially leads to overfitting. In contrast, the 360D dataset includes a significantly larger number of training samples, covering realistic scenes from both the Matterport3D and Stanford2D3D datasets, as well as synthetic scenes. Consequently, the 360D dataset provides a diverse range of scene types, effectively reducing the risk of overfitting. On this dataset, our method outperforms others on most metrics, except RMSE and RMSElog, where it ranks second. Besides these aforementioned metrics on depth maps, we incorporate an additional metric on transformed point clouds to evaluate the performance of depth estimation results. Given the predicted depth maps, we assign the pixel-wise estimated depth channels to their corresponding pixels in RGB panoramas thus obtaining their coordinates. Consequently, we transform depth maps into point clouds. To evaluate the accuracy of the predicted point clouds, we employ the Chamfer Distance metric, which calculates the squared distance to its nearest neighbor in the other set for each point and sums these distances up. The results in Table II further validate that our method achieves state-of-the-art performance in comparison with other methods. Both SliceNet and HoHoNet have noticed the importance of the large field-of-view provided by 360° images and they exploit gravity-aligned features to capture long- and short-term relationships in the panoramas. Although these two methods have advanced the performance of panoramic depth estimation on the benchmark datasets, they rely heavily on the assumption that the gravity direction should be aligned with the panorama’s y-axis. Hence, they are vulnerable to gravity misalignment. To show this, we conduct an experiment in which all test panoramas in the Stanford2D3D dataset have been modified by randomly rotating the up vector of the camera. We introduce ±2° and ±5° maximum inclination error (MIE) to the test panoramas. The performance of different methods is reported in Table III. As seen, the accuracy of estimated depth maps decreases apparently for both SliceNet and HoHoNet when MIE increases, indicating that both methods achieve low performance if the assumption of gravity alignment fails. In comparison, our method still achieves high accuracy even on non-gravity-aligned panoramas, since we do not impose any restriction on the panorama. Our architecture is robust to gravity misalignment. Note that the captured panoramas could be non-gravity-aligned in practice.

Besides the robustness to gravity misalignment, our method also shows better generalizability than previous methods. To verify this, we conduct cross-dataset validation and make comparisons with SliceNet, HoHoNet, ACDNet, and OmniFusion. We report the quantitative results in Table IV. Here, each model is trained and evaluated on a different dataset. Note that since 360D contains many cases from both Matterport3D and Stanford2D3D, the reverse validation is inappropriate. The results clearly show that our method performs much better than SliceNet in all cases. It also outperforms HoHoNet and ACDNet on most numerical metrics. The performance of HoHoNet drops significantly when evaluating on 360D using models trained with either Matterport3D or Stanford2D3D. This is probably because both Matterport3D and Stanford2D3D contain real-world examples while 360D also has synthetic data. There is a domain gap between these two types of data that cannot be handled by HoHoNet. When inferring on a different dataset, OmniFusion does not provide the learnable fusion layer for the evaluated dataset, leading to significant artifacts and decreased scores. In contrast, our trainable gated fusion module is capable of operating effectively across different datasets. In summary, our two-stream network has better generalizability than previous convolution-based methods and ACDNet.

2) Qualitative Comparisons: In Fig. 6, 7 and 8, we show visual comparisons against BiFuse, SliceNet, HoHoNet, ACDNet, and OmniFusion. Overall, with fine-grained and globally coherent predictions, our GLPanoDepth performs favorably against state-of-the-art panoramic depth estimation methods. Although cubemaps have already been used in some previous methods, e.g., BiFuse [55], their usages are fundamentally differently. Previous methods rely on cubemaps to handle the space-varying distortion of equirectangular projection. In contrast, we leverage cubemaps to extract global features by our specially-designed CViT. We observe that BiFuse tends to produce smooth predictions on test sets, resulting in overly blur depth map. Based on the assumption that equirectangular projection is aligned to the gravity vector, SliceNet recovers panoramic depth maps through a convolutional long short-term memory (LSTM) network to retain the global information. As shown in the results, SliceNet enhances edges with the help of global information. Instead of using LSTM, our GLPanoDepth leverages CViT branch to extract global features. MHSAs in CViT is an inherently global operation, as every embedding token can attend to and thus influence other tokens globally. ACDNet [66] extends the receptive field and gets diverse attention areas by combining the convolution kernels with different dilations in the equirectangular projection. However, its direct operation on distorted equirectangular projections limits its performance. Similar to our CViT branch, Omnifusion employs a transformer to extract

| Methods   | Matterport3D | Stanford2D3D | 360D   |
|-----------|--------------|--------------|--------|
| HoHoNet [50] | 0.732        | 1.344        | -      |
| BiFuse [55]   | 0.785        | 0.112        | 0.047  |
| SliceNet [37] | 0.736        | 0.117        | 0.039  |
| ACDNet [66]   | 1.961        | 0.122        | -      |
| OmniFusion [59] | -            | 0.117        | 0.146  |
| Ours        | 0.766        | 0.111        | 0.035  |

TABLE II QUANTITATIVE COMPARISONS USING CHAMFER DISTANCE FOR THE POINT CLOUD. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD, AND THE SECOND ARE UNDERLINED.
TABLE III
QUANTITATIVE PERFORMANCE ON NON-GRAVITY-ALIGNED PANORAMAS. MIE MEANS MAXIMUM INCLINATION ERROR. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

| MIE   | Method      | MAE ↓ | RMSE ↓ | RMSElog ↓ | $\delta \leq 1.25 \uparrow$ | $\delta \leq 1.25^2 \uparrow$ | $\delta \leq 1.25^3 \uparrow$ |
|-------|-------------|-------|--------|------------|-----------------------------|-------------------------------|-------------------------------|
| $\pm 2^\circ$ | SliceNet [37] | 0.3021 | 0.4539 | 0.1201 | 0.7501 | 0.9203 | 0.9601 |
|       | HoHoNet [50] | 0.2239 | 0.4011 | 0.1022 | 0.8889 | 0.9591 | 0.9802 |
|       | Ours        | 0.2021 | 0.3707 | 0.0712 | 0.8939 | 0.9711 | 0.9869 |
| $\pm 5^\circ$ | SliceNet [37] | 0.3639 | 0.4901 | 0.1322 | 0.7199 | 0.9156 | 0.9548 |
|       | HoHoNet [50] | 0.2659 | 0.4198 | 0.1131 | 0.8429 | 0.9359 | 0.9724 |
|       | Ours        | 0.2407 | 0.4012 | 0.0791 | 0.8441 | 0.9647 | 0.9839 |

TABLE IV
QUANTITATIVE PERFORMANCE FOR CROSS-DATASET VALIDATION. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD, AND THE SECOND ARE UNDERLINED.

| Dataset                          | Method      | MAE ↓ | RMSE ↓ | RMSElog ↓ | $\delta \leq 1.25 \uparrow$ | $\delta \leq 1.25^2 \uparrow$ | $\delta \leq 1.25^3 \uparrow$ |
|----------------------------------|-------------|-------|--------|------------|-----------------------------|-------------------------------|-------------------------------|
| Trained on Stanford2D3D,         | SliceNet [37] | 0.5426 | 1.0435 | 0.1994 | 0.4254 | 0.6443 | 0.7718 |
| evaluated on Matterport3D        | HoHoNet [50] | 0.6016 | 0.8987 | 0.1434 | 0.5925 | 0.8481 | 0.9289 |
|                                 | OmniFusion [59] | 0.6843 | 0.9970 | 0.1773 | 0.5066 | 0.7737 | 0.8986 |
|                                 | ACDNet [66] | 0.5686 | 0.8541 | 0.1477 | 0.6049 | 0.8415 | 0.9268 |
|                                 | Ours        | 0.4182 | 0.6405 | 0.1036 | 0.7584 | 0.9301 | 0.9709 |
| Trained on Matterport3D,         | SliceNet [37] | 0.9910 | 2.3545 | 0.3101 | 0.6627 | 0.7815 | 0.8252 |
| evaluated on Stanford2D3D        | HoHoNet [50] | 0.4569 | 1.0411 | 0.1475 | 0.7730 | 0.9174 | 0.9569 |
|                                 | ACDNet [66] | 0.4855 | 0.9994 | 0.1751 | 0.7311 | 0.8552 | 0.9121 |
|                                 | Ours        | 0.3872 | 0.6547 | 0.1345 | 0.7651 | 0.8766 | 0.9277 |
| Trained on Matterport3D,         | SliceNet [37] | 0.7181 | 1.0478 | 0.2122 | 0.3651 | 0.6946 | 0.8434 |
| evaluated on 360D                | HoHoNet [50] | 1.0728 | 1.4471 | 0.3239 | 0.3672 | 0.5330 | 0.6674 |
|                                 | ACDNet [66] | 0.4678 | 0.7466 | 0.1067 | 0.6211 | 0.8533 | 0.9409 |
|                                 | Ours        | 0.4201 | 0.7027 | 0.1304 | 0.7461 | 0.8780 | 0.9377 |
| Trained on Stanford2D3D,         | SliceNet [37] | 0.6125 | 0.9797 | 0.1779 | 0.5351 | 0.7912 | 0.8903 |
| evaluated on 360D                | HoHoNet [50] | 2.9877 | 4.0339 | 1.3742 | 0.0634 | 0.1325 | 0.2032 |
|                                 | OmniFusion [59] | 0.5909 | 0.8747 | 0.1715 | 0.5154 | 0.7873 | 0.9083 |
|                                 | ACDNet [66] | 0.4396 | 0.6712 | 0.1228 | 0.6525 | 0.8907 | 0.9567 |
|                                 | Ours        | 0.3956 | 0.5872 | 0.1050 | 0.6773 | 0.9479 | 0.9784 |

Fig. 6. Qualitative results on 360D. Invalid pixels are indicated with the black area in the ground truth.
the balustrade, and the bookshelf. This validates that our method is aware of the global structures of the scene through CViT and can leverage global information to improve depth estimation.

Furthermore, our two-branch network can better learn global coherence in areas such as large homogeneous regions and relative depth arrangement across the image. Noted that our model can split the objects with a close depth plane, such as jars on the shelf. Our prediction produces cleaner and finer-grained delineations of object boundaries and in some cases less cluttered.

As the predictions from different methods look similar in terms of the panoramic depth maps, we convert the predictions to point clouds to better compare the details of the predictions. It is accomplished by assigning the pixel-wise estimated depth channels to pixels in RGB panoramas and calculating their coordinates. These point clouds are visualized in Fig. 9. With the help of the global features from our CViT branch, GLPanoDepth has a global understanding of the spherical images, producing coherent values for the geometric planes, e.g., the vertical wall or the cabinet.

In addition, we visually compare our method with SliceNet and HoHoNet on non-gravity-aligned panoramas in Fig. 10. We find that SliceNet produces blurrier depth boundary when its assumption of gravity-aligned input does not hold. Since HoHoNet focuses on learning the most prominent signals of a column, it would generate some artifacts on non-gravity-aligned panoramas. By contrast, our method is robust to handle
both of global structures and local details indicated by the red boxes in Fig. 10.

C. Model Complexity and Inference Performance

In Table V, we report the model complexity and the average frame per second during the inference for HoHoNet [50], BiFuse [55], SliceNet [37] and our model. The complexity metrics include the number of neural model parameters and the Giga Floating-point Operations Per Second (Gflops) with varying input resolution. These experiments are conducted on an NVIDIA RTX 3090 GPU. Table V shows that HoHoNet possesses the advantage of a small model and fast inference. This is accomplished through an efficient height compression model that operates on compact 1D latent horizontal features. To better exploit long-term and short-term relationships among gravity-aligned slices, SliceNet builds a multi-resolution spatial representation and sequentially processes it with a convolutional long short-term memory (LSTM) network. However, LSTM is harder to parallelize, leading to a significant decrease in frames per second. Similar to our model, BiFuse incorporates two branches and fusion schemes. However, BiFuse utilizes Bi-Projection modules on both the encoding and decoding stages and leverages a spherical convolutional layer. These choices contribute to a high level of complexity and significant time consumption. As a result,
TABLE V

MODEL COMPLEXITY COMPARISONS AND INFERENCE PERFORMANCE WHEN THE RESOLUTION OF THE INPUT VARIES

| Inputs   | Metrics | HoHoNet [50] | SliceNet [37] | BiFuse [55] | Ours  |
|----------|---------|--------------|----------------|-------------|-------|
| 256 × 512 | GFlops  | 21.28 G      | 32.39 G        | 193.81 G    | 186.18 G |
|          | Parameters | 45.25 M    | 75.32 M       | 253.08 M    | 119.63 M |
|          | Fps     | 69.69        | 15.33          | 1.89        | 20.18  |
| 512 × 1024 | GFlops  | 82.90 G      | 101.77 G       | 775.24 G    | 755.41 G |
|          | Parameters | 45.27 M    | 79.51 M       | 253.08 M    | 131.44 M |
|          | Fps     | 51.18        | 7.77           | 0.94        | 9.56   |

Fig. 11. Visually comparing our complete model with four variants. Ours(CViT) removes the CNN branch and Ours(CNN) removes the CViT branch. Ours(CViT-C) modifies Ours(CViT) to take cubemap projection as both the input and output. Ours(ViT+CNN) replaces CViT with ViT. The major differences are highlighted in red boxes.

Fig. 12. Validation of the effectiveness of the gated fusion module, as compared with simple concatenation.

BiFuse nearly doubles the number of parameters compared to our model. In contrast, Our model strikes a balance between model complexity and inference performance, with a reasonable number of parameters. Moreover, we achieve shorter inference times compared to SliceNet and BiFuse.

D. Ablation Study

We conduct various ablation experiments to investigate the effectiveness of our design choices.

1) Effectiveness of Our Two-Branch Architecture: To validate the effectiveness of our two-branch architecture, we design two variants of our method: Ours(CViT) and Ours(CNN). Ours(CViT) removes the CNN branch from GLPanoDepth, while Ours(CNN) removes the CViT branch. Taking cubemap projection as input, Ours(CViT) model extracted features \( F_{G} \) from consecutive transformer blocks. These features are then combined using a RefineNet-based feature fusion block [31] and progressively upsampled by a factor of two to generate a fine-grained equirectangular prediction. The quantitative results of model variants are reported in Table I and we qualitatively compare them in Fig. 11. As a single CNN branch has limited receptive fields, it cannot fully exploit global information, resulting in the limited performance of Ours(CNN) model. On the other hand, without the CNN branch, Ours(CViT) also achieves sub-optimal results since a single CViT branch fails to preserve important and strong local features. Consequently, many small structures are missing in the depth map predicted by Ours(CViT) as illustrated in the top-left image of Fig. 11. Considering that the cubic and equirectangular mapping might not be easy to learn in Ours(CViT), we further introduce a model variant named Ours(CViT-C). Ours(CViT-C) extracts features from cubemap projection and decodes the features to output cubemap projection with convolution layers. However, the numerical metrics in Table I indicate that a single CViT branch or CViT-C branch is hard to converge to the appropriate point with only extracted global features. In contrast, combining features from the CViT branch and CNN branch offers highly synergistic improvements. The fusion of global and local features in our two-branch architecture leads to enhanced performance in depth estimation, as demonstrated by the quantitative results and qualitative comparisons.

To study the advantage of CViT, we replace it with ViT and name this variant Ours(ViT+CNN). Since ViT directly reshapes omnidirectional images into patches, it is susceptible to distortion. The quantitative results in Table I show that using cubemaps reinforces the effect of transformers to extract global contents from spherical signals.

2) Effectiveness of Our Gated Fusion Module: To explore the effect of the gated fusion module, we remove it from
our GLPanoDepth and use simple concatenation to combine features from different branches. Quantitative results in Table I clearly show that using concatenation as a fusion module makes our model converge to an inferior solution. As shown in Fig. 12, our gated fusion module yields sharper edges on the depth map, as well as clearer foreground-background separation, as compared with concatenation. In addition, we observe that the model using our gated fusion module converges faster than that using concatenation, which further verifies the efficacy of our gated fusion module.

3) Vanilla Convolution vs. Spherical Convolution in the CNN Branch: It appears that adopting spherical convolution in the CNN branch is a natural choice to handle spherical signals. However, sub-optimal performance is achieved when we replace vanilla convolution in the CNN branch with spherical convolution, as shown in Table VI. We also qualitatively compare our model with the model that utilizes spherical convolution instead of vanilla convolution in Fig. 13. The results demonstrate that the model using spherical convolution produces noisy depth values along the boundary of the wall while the spherical edges produced by our model are clear and close to the ground truth. We believe this is because the CViT branch has already addressed the issue of space distortion while spherical convolution will introduce additional complexities and challenges in training. Therefore, we currently use vanilla convolution in our CNN branch to achieve more favorable results.

4) Comparison with Transformer-based Networks: To further demonstrate the effectiveness of our model, which combines transformer and convolution, we conduct a comparison with transformer-based methods including DPT [39] and Swin Transformer [33]. In our comparison, we fine-tune the official DPT using equirectangular images from the Stanford2D3D dataset. The official DPT is trained on MIX 6 [39] and the NYUv2 dataset [46], both of which only provide pairs of aligned indoor perspective images and depth images. With the same training equirectangular images, Swin Transformer is trained from scratch. The quantitative results are reported in Table VII and we qualitatively compare these models in Fig. 14. DPT leverages vision transformers as the backbone and employs a convolutional decoder to progressively assemble and combine tokens for generating full-resolution predictions. However, all features in DPT are extracted by transformers, which inherently struggle to capture local structures, especially in panoramas with significant space-varying distortions. As a result, DPT achieves sub-optimal quantitative performance. To extract multi-scale features, Swin Transformer constructs hierarchical feature maps by starting from small-sized patches and gradually merging neighboring patches in deeper transformer layers. These hierarchical feature maps are built through a shifted window-based self-attention module. However, the spatially varying spherical distortions present in equirectangular images pose challenges for computing self-attention across shifting windows. Consequently, Swin Transformer produces over-smooth depth maps in Fig. 14. In contrast, GLPanoDepth outperforms DPT and Swin Transformer across all quantitative metrics and produces superior fine-grained depth maps.

E. Transfer Learning Capability

We evaluate the transfer learning capability of our model by applying our model to the panoramic semantic segmentation task. Similar to depth estimation, the semantic segmentation task is also a pixel2pixel task. We keep the structure of our network unchanged and train it directly on the Stanford2D3D dataset, which consists of 1000 pairs of panoramas and their ground truth for the panoramic semantic segmentation task. For comparison, we also train HoHoNet on the same dataset. The performance of HoHoNet and our model is quantitatively evaluated using metrics including Intersection over Union (IoU) and accuracy, and the results are presented in Table VIII. The results demonstrate that our model surpasses HoHoNet in terms of IoU, highlighting the advantage of our transfer learning capability.

F. Limitations

We have shown how the proposed CViT architecture is beneficial for extracting rich global information from omnidirectional images. However, like other transformers [12], CViT
requires a large-scale dataset for training. In other words, if the dataset is small in size, the accuracy of prediction will decrease. As we have mentioned previously, our method with a single CViT branch fails to converge on the Stanford2D3D dataset, since this dataset only has 1,413 panoramas in total, most of which contain large areas of missing depth value. Two failure cases from the Stanford2D3D dataset are shown in Fig. 15. To maximize the effectiveness of CViT and to improve the generalization capability of the trained model, a large-scale dataset containing a large amount of diverse training data is needed.

V. CONCLUSION

To conclude, we have proposed GLPanoDepth, a new method for end-to-end panoramic depth estimation. The key idea of GLPanoDepth is to extract global and local features respectively by two separate branches: one is based on cube-map vision transformers and the other is based on traditional convolutional layers. Different features are progressive combined by a gated fusion module. We validate the benefits of our proposed method on multiple datasets. Qualitative and quantitative comparisons against previous methods show superiority of our method. Ablation studies further exemplify that the specially-designed cube-map vision transformers and gated fusion module are able to better capture rich and distortion-free global features from spherical signals compared with other methods.

REFERENCES

[1] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, “Joint 2D-3D-semantic data for indoor scene understanding,” 2017, arXiv:1702.01105.
[2] S. Bhajanpal, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner, and A. Veit, “Understanding robustness of transformers for image classification,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 10231–10241.
[3] G. Bresson, Z. Alsabay, L. Yu, and S. Glasner, “Simultaneous localization and mapping: A survey of current trends in autonomous driving,” IEEE Trans. Intell. Vehicles, vol. 2, no. 3, pp. 194–220, Sep. 2017.
[4] Y. Cao, Z. Wu, and C. Shen, “Estimating depth from monocular images as classification using deep fully convolutional residual networks,” in Proc. IEEE Trans. Circuits Syst. Video Technol. (TCSVT), Aug. 2018, pp. 3174–3182.
[5] A. Chang et al., “Matterport3D: Learning from RGB-D data in indoor environments,” in Proc. Int. Conf. 3D Vis. (3DV), Oct. 2017, pp. 667–676.
[6] L. Chen, H. Wei, and J. Ferryman, “A survey of human motion analysis using depth imagery,” Pattern Recognit. Lett., vol. 34, no. 15, pp. 1995–2006, Nov. 2013.
[7] M. Chen et al., “Generative pretraining from pixels,” in Proc. Int. Conf. Mach. Learn., Jul. 2020, pp. 1691–1703.
[8] H.-T. Cheng, C.-H. Chao, J.-D. Dong, H.-K. Wen, T.-L. Liu, and M. Sun, “C rude padding for weakly-supervised saliency prediction in 360° videos,” in Proc. IEEE/CVF Comput. Vis. Pattern Recognit., Jun. 2018, pp. 1420–1429.
[9] T. S. Cohen, M. Geiger, J. Kohler, and M. Welling, "Spherical CNNs," in Proc. Int. Conf. Learn. Represent, 2018, pp. 1–15.
[10] B. Coors, A. P. Condurache, and A. Geiger, “SphereNet: Learning spherical representations for detection and classification in omnidirectional images,” in Computer Vision—ECCV, Ferrari, M. Hebert, C. Sinmichesescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 525–541.
[11] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci, “Structured attention guided convolutional neural field for monocular depth estimation,” in Proc. IEEE/CVF Comput. Vis. Pattern Recognit., Jun. 2018, pp. 3917–3925.
[12] A. Dosovitskiy et al., “An image is worth 16×16 words: Transformers for image recognition at scale,” in Proc. ICLR, 2021, pp. 1–22.
[13] M. Eder, P. Moulon, and L. Guan, “Fano popups: Indoor 3D reconstruction with a plane-aware network,” in Proc. Int. Conf. 3D Vis. (3DV), Sep. 2019, pp. 76–84.
[14] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 2650–2658.
[15] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in Proc. 27th Int. Conf. Neural Inf. Process. Syst. (NIPS), vol. 2, 2014, pp. 2366–2374.
[16] B. Y. Feng, W. Yao, Z. Liu, and A. Varshney, “Deep depth estimation on 360° images with a double quaternion loss,” in Proc. Int. Conf. 3D Vis. (3DV), Nov. 2020, pp. 524–533.
[17] R. Furukawa, R. Sagawa, and H. Kawasaki, “Depth estimation using structured light flow—Analysis of projected pattern flow on an object’s surface,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 4650–4658.
[18] 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.
[19] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, “Transformer in transformer,” in Proc. Adv. Neural Inf. Process. Syst., vol. 34, 2021, pp. 15908–15919.
[20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.
[21] H. Jiang, Z. Sheng, S. Zhu, Z. Dong, and R. Huang, “UniFuse: Unidirectional fusion for 360° panorama depth estimation,” IEEE Robot. Autom. Lett., vol. 6, no. 2, pp. 1519–1526, Apr. 2021.
[22] L. Jin et al., “Geometric structure based and regularized depth estimation from 360° indoor imagery,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 886–895.
[23] M. S. Junayed, A. Sadeghzadeh, M. B. Islam, L.-K. Wong, and C. Sminchisescu, and Y. Weiss, “Understanding robustness of transformers for image classification,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 10231–10241.
[24] G. Bresson, Z. Alsabay, L. Yu, and S. Glasner, “Simultaneous localization and mapping: A survey of current trends in autonomous driving,” IEEE Trans. Intell. Vehicles, vol. 2, no. 3, pp. 194–220, Sep. 2017.
[25] Y. Cao, Z. Wu, and C. Shen, “Estimating depth from monocular images as classification using deep fully convolutional residual networks,” in Proc. IEEE Trans. Circuits Syst. Video Technol. (TCSVT), Aug. 2018, pp. 3174–3182.
[26] A. Chang et al., “Matterport3D: Learning from RGB-D data in indoor environments,” in Proc. Int. Conf. 3D Vis. (3DV), Oct. 2017, pp. 667–676.
[27] L. Chen, H. Wei, and J. Ferryman, “A survey of human motion analysis using depth imagery,” Pattern Recognit. Lett., vol. 34, no. 15, pp. 1995–2006, Nov. 2013.
[28] I. Laina, V. Belagiannis, F. Tombari, and N. Navab, “Deeper depth prediction with fully convolutional residual networks,” in Proc. 4th Int. Conf. 3D Vis. (3DV), Oct. 2016, pp. 239–248.

[29] S. Lambert-Lacroix and L. Zwald, “The adaptive BerHu penalty in robust regression,” J. Nonparametric Statist., vol. 28, no. 3, pp. 487–514, Jul. 2016.

[30] J.-H. Lee, M. Heo, K.-R. Kim, and C.-S. Kim, “Single-image depth estimation based on Fourier domain analysis,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 330–339.

[31] G. Lin, A. Milan, C. Shen, and I. Reid, “ReineNet: Multi-path refinement networks for high-resolution semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 5168–5177.

[32] F. Liu, C. Shen, and G. Lin, “Deep convolutional neural fields for depth estimation from a single image,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 5162–5170.

[33] Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 9902–10002.

[34] Y. Meng, M. Fang, C. Fan, and H. Yu, “Deep learning for monocular depth estimation: A review,” Neurocomputing, vol. 438, pp. 14–33, May 2021.

[35] I. Misra, R. Girdhar, and A. Joulin, “An end-to-end transformer model for 3D object detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 2906–2917.

[36] A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds, Curran Associates, 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288ee7022ef97012774740-Paper.pdf

[37] G. Pintore, M. Agus, E. Almansa, J. Schneider, and E. Gobbetti, “SliceNet: Deep dense depth estimation from a single indoor panorama using a slice-based representation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 11531–11540.

[38] G. Pintore, C. Mura, F. Ganovelli, L. Fuentes-Perez, R. Pajarola, and E. Gobbetti, “State-of-the-art in automatic 3D reconstruction of structured indoor environments,” Comput. Graph. Forum, vol. 39, no. 2, pp. 667–699, May 2020.

[39] R. Ranftl, A. Bochkovskiy, and V. Koltun, “Vision transformers for dense prediction,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 12179–12188.

[40] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun, “Dense monocular depth estimation in complex dynamic scenes,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 4058–4066.

[41] D. Roy and B. Fernando, “Action anticipation using pairwise human-object interactions and transformers,” IEEE Trans. Image Process., vol. 30, pp. 8116–8129, 2021.

[42] A. Saxena, S. Chung, and A. Ng, “Learning depth from single monocular images,” in Proc. Adv. Neural Inf. Process. Syst., vol. 18, Y. Weiss, B. Schölkopf, and J. Platt, Eds, Cambridge, MA, USA: MIT Press, 2006, pp. 1–8.

[43] A. Saxena, M. Sun, and A. Y. Ng, “Make3D: Learning 3D scene structure from a single still image,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 3, pp. 824–840, May 2009.

[44] W. Shi et al., “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1874–1883.

[45] S. Xingjian, C. Zhouroug, W. Hao, Y. Dit-Yan, W. Wai-Kin, and W. Wang-Chun, “Convolutional lstm network: A machine learning approach for precipitation nowcasting,” in Proc. 27th Int. Conf. Neural Inf. Process. Syst., vol. 1, Cambridge, MA, USA: MIT Press, 2015, pp. 802–810.

[46] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from RGBD images,” in Proc. 12th Eur. Conf. Comput. Vis., Berlin, Germany: Springer, 2012, pp. 746–760.

[47] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, “Semantic scene completion from a single depth image,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 190–198.

[48] Y.-C. Su and K. Grauman, “Learning spherical convolution for fast features from 360° imagery,” in Proc. Adv. Neural Inf. Process. Syst., vol. 30, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds, Curran Associates, 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/0c74b718409a4022a2c45a5ca3e19-Paper.pdf

[49] Y.-C. Su and K. Grauman, “Kernel transformer networks for compact spherical convolution,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 9434–9443.

[50] C. Sun, M. Sun, and H.-T. Chen, “HoHoNet: 360 indoor holistic understanding with latent horizontal features,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 2573–2582.

[51] K. Tateno, N. Navab, and F. Tombari, “Distortion-aware convolutional filters for spherical depth prediction,” in Proc. Computer Vision—ECCV, F. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds, Cham, Switzerland: Springer, 2018, pp. 732–750.

[52] K. Tateno, F. Tombari, I. Laina, and N. Navab, “CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 6565–6574.

[53] A. Paszke et al., “PyTorch: A new tensor library for efficient computation,” Comm. Comput. Inform. Sci., vol. 31, no. 5, pp. 824–840, May 2009.