PanoFormer: Panorama Transformer for Indoor 360° Depth Estimation

Zhijie Shen[1,2], Chunyu Lin[1,2], Kang Liao[1,2,3], Lang Nie[1,2], Zishuo Zheng[1,2], and Yao Zhao[1,2]

[1] Institute of Information Science, Beijing Jiaotong University, China
[2] Beijing Key Laboratory of Advanced Information Science and Network Technology
[3] Max Planck Institute for Informatics, Germany
Corresponding Author: Chunyu Lin
{zhjshen, cylin, kang_liao, nielang, zszheng, yzhao}@bjtu.edu.cn
Code: https://github.com/zhijieshen-bjtu/PanoFormer

Abstract. Existing panoramic depth estimation methods based on convolutional neural networks (CNNs) focus on removing panoramic distortions, failing to perceive panoramic structures efficiently due to the fixed receptive field in CNNs. This paper proposes the panorama transformer (named ***PanoFormer***) to estimate the depth in panorama images, with tangent patches from spherical domain, learnable token flows, and panorama specific metrics. In particular, we divide patches on the spherical tangent domain into tokens to reduce the negative effect of panoramic distortions. Since the geometric structures are essential for depth estimation, a self-attention module is redesigned with an additional learnable token flow. In addition, considering the characteristic of the spherical domain, we present two panorama-specific metrics to comprehensively evaluate the panoramic depth estimation models' performance. Extensive experiments demonstrate that our approach significantly outperforms the state-of-the-art (SOTA) methods. Furthermore, the proposed method can be effectively extended to solve semantic panorama segmentation, a similar pixel2pixel task.

1 Introduction

Depth information is important for computer systems to understand the real 3D world. Monocular depth estimation has attracted researchers' [6,7,8,15,40,39] attention with its convenience and low cost, especially for panoramic depth estimation [41,43,34], where the depth of the whole scene can be obtained from a single 360° image.

Since estimating depth from a single image is an ill-posed and inherently ambiguous problem, current solutions almost use powerful CNNs to extract explicitly or implicitly prior geometric to realize it [3,5]. However, when applied to panoramic tasks, these SOTA depth estimation solutions for perspective imagery [21] show a dramatic degradation because the 360° field-of-view (FoV)

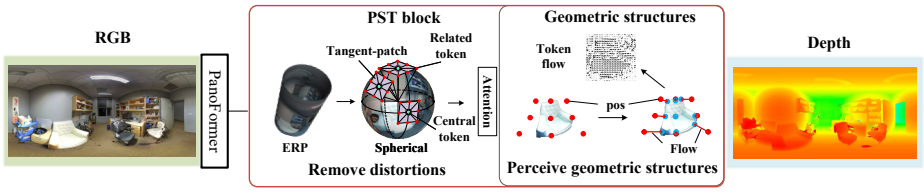

Fig. 1. We present PanoFomer to establish panoramic perception capability. The tangent-patch is proposed to remove panoramic distortions, and the token flows force the token positions to fit the structure of the sofa better. More details refer to Sec. 3

from panorama brings geometric distortions that challenge the structure perception. Specifically, distortions in panoramas (usually represented in equirectangular projection—ERP) increase from the center to both sides along the latitude direction, severely deforming objects' shapes. Due to the fixed receptive field, CNNs are inferior for dealing with distortions and perceiving geometric structures in panoramas [5]. To deal with the distortions in panoramas, some researchers [17,28,29,34] adopt the projection-fusion strategy. But this strategy needs to cover the domain gap between different projections, and the extra cross-projection fusion module increases computational burdens. Other researchers [9,10,13,30,44,30,37,38,16,19,22] employ various distortion-aware convolution filters to make CNN-based depth estimation solutions adapt to 360° images. However, the fixed sampling positions still limit their performance. Pintore *et al*. [26] focuses on the full geometric context of an indoor scene, proposing SliceNet but losing detailed information when reconstructing the depth map. We note that all the existing methods cannot perceive the distorted geometric structures with the fixed receptive field.

To address the above limitations, we propose the first panorama Transformer (PanoFormer) to enable the network's panoramic perception capability by removing distortions and perceiving geometric structures simultaneously (shown in Fig. 1). To make the Transformer suitable for panoramic dense prediction tasks (e.g., depth estimation and semantic segmentation), we redesign its structure. First, we propose a dense patches dividing method and handcrafted tokens to catch detailed features. Then, we design a relative position embedding method to reduce the negative effect of distortions, which utilizes a central token to locate the eight most relevant tokens to form a tangent patch (it differs from directly dividing patches on the ERP domain in traditional vision Transformers). To achieve this goal, we propose an efficient spherical token locating model (STLM) to guide the 'non-distortion' token sampling process on the ERP domain directly by building the Transformations among the three domains (shown in Fig. 3). Subsequently, we design a Panoramic Structure-guided Transformer (PST) block to replace the traditional block in a hierarchical architecture. Specifically, we redesign the self-attention module with additional learnable weight to push token flow, so as to flexibly capture various objects' structures. This module encourages the PanoFormer to further perceive geometric structures effec-

tively. In this way, we establish our network's perception capability to achieve panoramic depth estimation. Moreover, the proposed PST block can be applied to other learning frameworks as well.

Furthermore, current evaluation metrics for depth estimation are suitable for perspective imagery. However, these metrics did not consider distortions and the seamless boundary property in panoramas. To comprehensively evaluate the depth estimation for panoramic images, we design a Pole Root Mean Square Error (P-RMSE) and Left-Right Consistency Error (LRCE) to measure the accuracy on polar regions and depth consistency around the boundaries, respectively.

Extensive experiments demonstrate that our solution significantly outperforms SOTA algorithms in panoramic depth estimation. Besides, our solution achieves the best performance when applied to semantic segmentation, which is also a pixel2pixel panoramic task. The contributions of this paper are summarized as follows:

- We present ***PanoFormer***, the first panorama Transformer, to establish the panoramic perception capability by reducing distortions and perceiving geometric structures for the panoramic depth estimation task.
- We propose a PST block that divides patches on the spherical tangent domain and reshapes the self-attention module with the learnable token flow. Moreover, the proposed block can be applied in other learning frameworks.
- Considering the difference between Panorama and normal images, we design two new panorama-specific metrics to evaluate the panoramic depth estimation.
- Experiments demonstrate that our method significantly outperforms the current state-of-the-art approaches on all metrics. The excellent panorama semantic segmentation results also prove the extension ability of our model.

2 Related Work

2.1 Panoramic Depth Estimation

There are two main fusion methods to reduce distortions while estimating depth on ERP maps. One is the equirectangular-cube fusion method represented by Bifuse [34], and the other is the dual-cube fusion approach described by Shen [28]. Specifically, Bifuse [34] propose a two-branch method of fusing equirectangular projection and cube projection, which improves the tolerance of the model to distortions. Moreover, UniFuse [17] also uses a dual projection fusion scheme only at the encoding stage to reduce computation cost. Noting that the single-cube projection method produces significant discontinuities at the cube boundary, Shen *et al* [28] proposed a dual-cube approach based on a 45° rotation to reduce distortions. This class of methods can attenuate the negative effect of distortions, but they need to repeatedly change the projection for fusion, increasing the model's complexity.

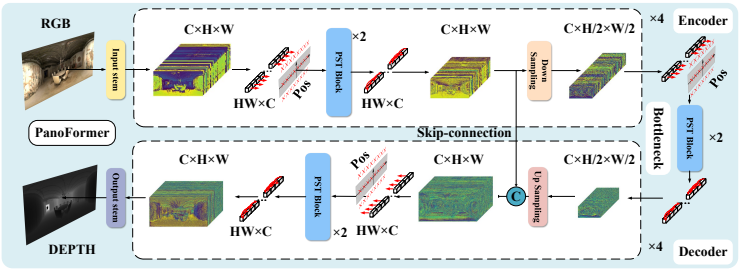

Fig. 2. Our PanoFormer takes a monocular RGB panoramic image as the input and outputs the high-quality depth map

To apply depth estimation models of normal images to panoramas, Tateno *et al*. [33] obtained exciting results by designing distortion-aware convolution filters to expand the perceptual field. Zioulis *et al*. [45] demonstrated that monocular depth estimation models trained on conventional 2D images produce low-quality results, highlighting the necessity of learning directly on the 360° domain. Jin *et al*. [18] demonstrated the effectiveness of geometric prior for panoramic depth estimation. Chen *et al*. [5] used strip pooling and deformable convolution to design a new encoding structure for accommodating different degrees of distortions. Moreover, Pintore *et al*. [26] proposed SliceNet, a network similar to HorizonNet [31], which uses a bidirectional Long Short-Term Memory (LSTM) to model long-range dependencies. However, the slicing method ignores the latitudinal distortion property and thus cannot accurately predict the depth near the poles. Besides, [2,11] proved that on large-scale datasets, Transformer-based depth estimation for normal images are superior to CNN.

2.2 Vision Transformer

Unlike CNN-based networks, the Transformer has the nature to model long-range dependencies by global self-attention [27]. Inspired by ViT [11], researchers have designed many efficient networks that have the advantages of both CNNs and Transformers. To enhance local features extraction, convolutional layers are added into muti-head self-attention (CvT [36]) and feed-forward network (FFN) (CeiT [42], LocalViT [23]) is replaced by locally-enhanced feed-forward network (LeFF) (Uformer [35]). Besides, CvT [36] demonstrates that the padding operation in CNNs implicitly encodes position, and CeiT [42] proposes the image-to-tokens embedding method. Inspired by SwinT [24], Uformer [35] proposes a shifted windows-based multi-head attention mechanism to improve the efficiency of the model. But all these solutions are developed based on normal FoV images, which cannot be applied to panoramic images directly. Based on these previous works, we further explore suitable Transformer structure for panoramic images and adapt it to the dense prediction task.

3 PanoFomer

3.1 Architecture Overview

Our primary motivation is to make the Transformer suitable for pixel-level omni-directional vision tasks by redesigning the standard components in conventional Transformers. Specifically, we propose a pixel-level patch division strategy, a relative position embedding method, and a panoramic self-attention mechanism. The proposed pixel-level patch division strategy is to enhance local features and improve the ability of Transformers to capture detailed features. For position embedding, we renounce the conventional absolute position embedding method and get the position of other related tokens on the same patch by the central token (described in 3.3). This method not only eliminates distortions, but also provides position embedding. Furthermore, we establish a learnable flow in the panorama self-attention module to perceive panoramic structures that are essential for depth estimation.

As shown in Fig. 2, the PanoFomer is a hierarchical structure with five major parts: input stem, output stem, encoder, decoder and bottleneck. For the input stem, a 3×3 convolution layer is adopted with size $H \times W$ to form the features with dimension C. Then the features are fed into the encoder. There are four hierarchical stages in encoder and decoder, and each of them contains a position embedding, two PST blocks (sharing the same settings), and a convolution layer. Specifically, a 4×4 convolution layer is adopted for increasing dimension and down-sampling in the encoder, while a 2×2 transposed convolution layer is used in the decoder for decreasing dimension and up-sampling. Finally, the output features from the decoder share the same resolution and dimension as the features from the input stem. Furthermore, the output stem, implemented by a 3×3 convolution, is employed to recover the depth map from features. More specifically, the number of heads is sequentially set as [Encoder:1, 2, 4, 8; Bottleneck: 16; Decoder: 16, 8, 4, 2]. As for all padding operations in convolution layers, we utilize circular padding for both horizontal sides of the features.

3.2 Transformer-customized Spherical Token

In vision Transformers, the input image is first divided into patches of the same size. For example, ViT [11] divides the input image into patches with size of 16×16 to reduce the computational burden. Then, these patches are embedded as tokens in a learning-based way via a linear layer. However, this strategy loses much detailed information, which is a fatal drawback for dense prediction tasks, such as depth estimation. To overcome this issue, we propose a pixel-level patches dividing method.

First, the input features are divided into pixel-level patches, which means each sampling position in the features corresponds to a patch centered on it. Such a dense division strategy allows the network to learn more detailed features, which is beneficial for dense prediction tasks. Furthermore, we make each patch consist of 9 features at different positions (one central position and eight

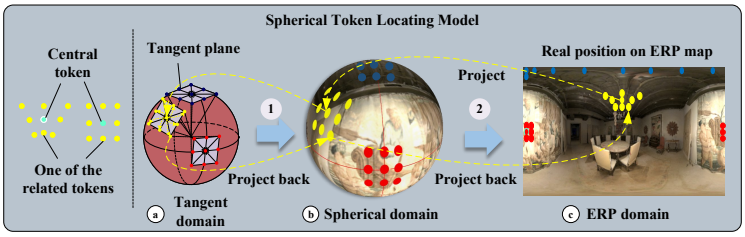

Fig. 3. Spherical Token Locating Model (STLM): locate related tokens on ERP domain. 1: tangential domain of unit sphere to spherical domain; 2: spherical domain to ERP domain

surrounding positions, illustrated in Fig. 3 left) to balance the computational burden. Unlike standard Transformers that embed patches as tokens by a linear layer, our tokens are handcrafted. We define the features at the central position as the central token and those from the other 8 surrounding positions as the related tokens. The central token can determine the position of related tokens by looking up the eight most relevant tokens among the features. To remove distortion and embed position information for the handcrafted tokens, we propose a distortion-based relative position embedding method in Sec. 3.3.

3.3 Relative Position Embedding

Inspired by the cube projection, we note that the spherical tangent projection can effectively remove the distortion (see Supplementary Materials for proof). Therefore, we propose STLM to initialize the position of related tokens. Unlike the conventional Transformers (e.g., ViT [11]), which directly adds absolute position encoding to the features, we "embed" the position information via the central token. Firstly, the central token is projected from the ERP domain to the spherical domain; then, we use the central token to look up the position of eight nearest neighbors on the tangent plane; finally, these positions are all projected back to the ERP domain (the three steps are represented by yellow arrows in Fig. 3). We call patches formed in this way as tangent patches. To facilitate locating the related tokens in the ERP domain, we further establish the relationship among the three domains (illustrated in Fig. 3).

Tangent domain to spherical domain: Let the unit sphere be S^2, and $S(0,0) = (\theta_0, \phi_0) \in S^2$ is the spherical coordinate origin. $\forall S(x,y) = (\theta, \phi) \in S^2$, we can obtain other 8 points (related tokens) around it (current token) on the spherical domain.

$$\begin{aligned} S(\pm 1, 0) &= (\theta \pm \Delta\theta, \phi) \\ S(0, \pm 1) &= (\theta, \phi \pm \Delta\phi) \\ S(\pm 1, \pm 1) &= (\theta \pm \Delta\theta, \phi \pm \Delta\phi) \end{aligned} \tag{1}$$

where (θ, ϕ) denotes the unit spherical coordinates, and $\theta \in (-\pi, \pi)$, $\phi \in (-\frac{\pi}{2}, \frac{\pi}{2})$; $\Delta\theta$,$\Delta\phi$ is the sampling step size.

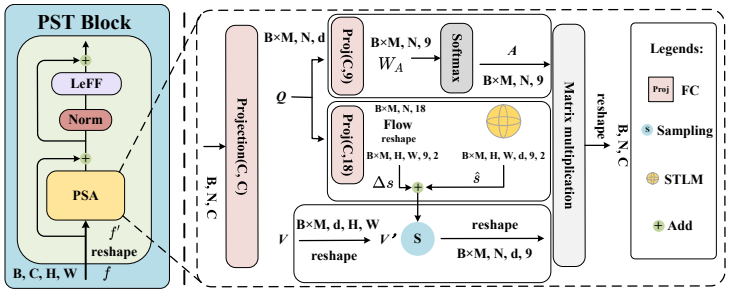

Fig. 4. The proposed PST Block can remove the negative effect of distortions and perceive geometric structures

By the geocentric projection [25], we can calculate the local coordinates $(T(x, y))$of the sampling point in tangent domain [9] (the current token in tangent domain is represented as $T(0, 0) = T(\theta, \phi) = (0, 0)$):

$$\begin{aligned} T(\theta \pm \Delta\theta, \phi) &= (\pm \tan \Delta\phi, 0) \\ T(\theta, \phi \pm \Delta\phi) &= (0, \phi \pm \tan \Delta\theta) \\ T(\theta \pm \Delta\theta, \phi \pm \Delta\phi) &= (\pm \tan \Delta\phi, \pm \sec \Delta\phi \tan \Delta\theta) \end{aligned} \tag{2}$$

By applying the inverse projection described in [9], we can get the position of all tokens of a tangent patch in the spherical domain.

Spherical domain to ERP domain: Furthermore, by utilizing the projection equation [28], we can get the position of each tangent patch in the ERP domain. This whole process is named Spherical Token Locating Model (STLM).

3.4 Panorama Self-Attention with Token Flow

Based on the traditional vision Transformer block, we replace the original attention mechanism with panorama self-attention. To further enhance local features interaction, we replace FFN with LeFF [42] for our pixel-level depth estimation task. Specifically, as illustrated in Fig. 4, when the features $f \in \mathbb{R}^{C\times H\times W}$ with a height of H and a width of W are fed into PST block, they are flattened and reshaped as $f' \in \mathbb{R}^{N\times C}$, where $N = H \times W$. Then a fully connected layer is applied to obtain query $Q \in \mathbb{R}^{N\times d}$ and value $V \in \mathbb{R}^{N\times d}$, where $d = C/M$, and M is the head number. The Q and V will pass through three parallel branches for computing attention score ($A \in \mathbb{R}^{N\times 9}$), token flows ($\Delta s \in \mathbb{R}^{N\times 18}$), and re-sampling features. In the top branch, a full connection layer is adopted to get attention weights $W_A \in \mathbb{R}^{N\times 9}$ from Q, and then softmax is employed to calculate the attention score A. In the middle branch, another fully connection layer is used to learn a token flow Δs and it is further reshaped to $\Delta s' \in \mathbb{R}^{d\times H\times W\times 9\times 2}$, sharing the same dimension with $\hat{s}$ (the initialed position from the STLM). Moreover, $\Delta s'$ and $\hat{s}$ are added together to calculate the final token positions. In the bottom branch, the value V is reshaped to $V' \in \mathbb{R}^{C\times H\times W}$ and are sampled to form

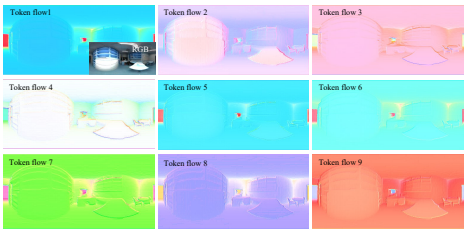

Fig. 5. Visualization of the token flows from the first PST block, which suggest the panoramic structures

the divided patches (described in 3.2) by looking up the related tokens in the final token positions. Afterward, the PSA can be represented as follows:

$$\begin{aligned} \mathrm{PSA}(f, \hat{\boldsymbol{s}}) = \textstyle\sum_{m=1}^{M} W_m * \\ \left[\textstyle\sum_{q=1}^{H\times W} \sum_{k=1}^{9} A_{mqk} \cdot W'_m f\left(\hat{s}_{mqk} + \Delta s_{mqk}\right)\right], \end{aligned} \tag{3}$$

where $\hat{s} = \boldsymbol{STLM}(f)$, and $\boldsymbol{STLM}(\cdot)$ denotes the spherical token locating model; m indexes the head of self-attention, M is the whole heads, q index the current point (token), k indexes the tokens in a tangent patch, $\Delta \boldsymbol{s}_{mqk}$ is the learned flow of each token, A_{mqk} represents the attention weight of each token, and W_m and W'_m are normal learnable weights of each head.

From the above process, we can see that the final positions of the tokens are determined by two steps: position initialization from STLM and additional learnable flow. Actually, the initialized position realizes the division of the tangent patch (described in 3.3) and removes the panoramic distortion. Furthermore, the learnable flow exhibits a panoramic geometry by adjusting the spatial distribution of tokens. To verify the effectiveness of the token flow, we visualize all tokens from the first PST block in Fig. 5. It can be observed that this additional flow provides the network with clear scene geometric information, which helps the network to estimate the panorama depth with the structure as a clue.

3.5 Objective Function

For better supervision, we combine reverse Huber [14] (or Berhu [21]) loss and gradient loss [28] to design our objective function as commonly used in previous works [26,28]. In our objective function, the Berhu loss β_δ can be written as:

$$\beta_\delta(g,p) = \begin{cases} |g-p| & \text{for } |g-p| \le \delta \\ \frac{|(g-p)^2|+\delta^2}{2\delta} & \text{otherwise} \end{cases} \tag{4}$$

where g, p denote the ground truth and predicted values, respectively.

Similar to SliceNet [26], we apply gradient loss to Berhu loss. To obtain depth edges, we use two convolution kernels to obtain gradients in horizontal and vertical directions, respectively. They are represented as K_h and K_v, where

K_h = [-1 0 1, -2 0 2, -1 0 1], and $K_v = (K_h)^T$. Denote the gradient function as G, the horizontal gradient I_h and vertical gradient I_v of the input image I can be expressed as $I_h = G(K_h, I)$ and $I_v = G(K_v, I)$, respectively. In this paper, δ = 0.2 and the final objective function can be written as

$$\ell_{final} = \beta_{0.2}(g, p) + \beta_{0.2}(G(K_h, g), G(K_h, p)) + \beta_{0.2}(G(K_v, g), G(K_v, p)), \quad (5)$$

4 Panorama-specific Metrics

Rethinking the spherical domain, we note that two significant properties cannot be neglected: the spherical domain is continuous and seamless everywhere; the distortion in the spherical domain is equal everywhere. For the first issue, we propose LRCE to measure the depth consistency of left-right boundaries. For the second issue, since distortions on ERP maps vary in longitude, RMSE cannot visually reflect the model's ability to adapt to distortions. Therefore, we provide P-RMSE to focus on the regions with massive distortions to verify the model's panoramic perception capability.

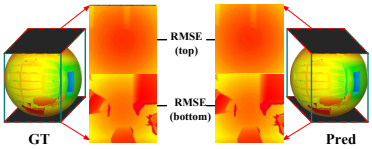

Fig. 6. P–RMSE: calculate the RMSE of the polar regions

Pole Root Mean Square Error. Cube projection is a special spherical tangent projection format that projects the sphere onto the cube's six faces. The top and bottom faces correspond to the polar regions of the spherical domain, so we select the two parts to design P-RMSE (illustrated in Fig. 6). Define the function of converting ERP to Cube as $E2C(\cdot)$, the converted polar regions of the ERP image E can be expressed as $Select(E2C(E), T, B)$, where T, B represent the top and bottom parts, respectively. The error C_e between the ground truth GT and the predicted depth map P at the polar regions can be expressed as

$$C_e = Select(E2C(GT), T, B) - Select(E2C(P), T, B) \quad (6)$$

The final P-RMSE can be written as

$$\text{P-RMSE} = \sqrt{\frac{1}{N_{C_e}} \sum_{i=1}^{N_{C_e}} |C_e^i|} \quad (7)$$

where N_{C_e} is the number of values in C_e.

Left-Right Consistency Error. We can evaluate the depth consistency of the left-right boundaries by calculating the horizontal gradient between the both sides of the depth map. Define that the horizontal gradient G_E^H of the image E can be written as $G_E^H = E_{first}^{col} - E_{last}^{col}$, where E_{first}^{col}/ E_{last}^{col} represent the values in the first/last columns of the image E. But consider an extreme case where if the edge of an object in the scene happens to be on the edge of the depth map, then there is ambiguity in reflecting continuity only by G_E^H. We cannot tell whether this discontinuity is real or caused by the model. Therefore, we add ground truth to our design. The horizontal gradient of ground truth and the predicted depth map are denoted as G_{GT}^H and G_P^H (where $G_{GT}^H = GT_{first}^{col} - GT_{last}^{col}$, $G_P^H = P_{first}^{col} - P_{last}^{col}$), respectively. The final expression can be as follows:

$$\text{LRCE} = \frac{1}{N_{error}} \sum_{i=1}^{N_{error}} |error_i| \tag{8}$$

where $error = G_{GT}^H - G_P^H$ and N_{error} is the number of values in $error$.

5 Experiments

In the experimental part, we compare the state-of-the-art approaches on four popular datasets and validate the effectiveness of our model.

5.1 Datasets and Implementations

Four datasets are used for our experimental validation, they are Stanford2D3D [1], Matterport3D [4], PanoSUNCG [29] and 3D60 [45].

Stanford2D3D and Matterport3D are two real-world datasets. They were rendered from a common viewpoint. Previous work used a dataset that was rendered only on the equator and its surroundings, ignoring the area near the poles, which undermined the integrity of the panorama. We strictly follow the previous works and employ the official datasets (Notice that the Stanford2D3D and Matterport3D that are contained in 3D60 have a problem that the light in the scenarios will leak the depth information). PanoSUNCG is a virtual panoramic dataset. And 3D60 is an updated version of 360D (360D is no longer available now). It consists of data from the above three datasets. There is a gap between the distributions of these three datasets, which makes the dataset more responsive to the model's generalizability. Note that we divide the dataset as the previous work and eliminate the samples that failed to render [5,34].

In the implementation, we conduct our experiments on two GTX 3090 GPUs, and the batch size is set to 4. We choose Adam [20] as the optimizer and keep the default settings. The initialized learning rate is 1×10^{-4}. The number of parameters of our model is 20.37 M.

Table 1. Quantitative comparisons on Matterport3D, Stanford2D3D, PanoSUNCG and 3D60 Datasets.

Dataset	Method	Classic metrics					
		$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	**RMSE**$\downarrow$	MRE$\downarrow$	MAE$\downarrow$
		Higher the better			Lower the better		
Matterport3D	FCRN [21]	0.7703	0.9174	0.9617	0.6704	0.2409	0.4008
	OmniDepth [45]	0.6830	0.8794	0.9429	0.7643	0.2901	0.4838
	Bifuse [34]	0.8452	0.9319	0.9632	0.6295	0.2408	0.3470
	UniFuse [17]	0.8897	0.9623	0.9831	0.4941	–	0.2814
	SliceNet [26]	0.8716	0.9483	0.9716	–	0.1764	0.3296
	Ours	**0.9184**	**0.9804**	**0.9916**	**0.3635**	**0.0571**	**0.1013**
Stanford2D3D	FCRN [21]	0.7230	0.9207	0.9731	0.5774	0.1837	0.3428
	OmniDepth [45]	0.6877	0.8891	0.9578	0.6152	0.1996	0.3743
	Bifuse [34]	0.8660	0.9580	0.9860	0.4142	0.1209	0.2343
	UniFuse [17]	0.8711	0.9664	0.9882	0.3691	–	0.2082
	SliceNet [26]	0.9031	0.9723	0.9894	–	0.0744	0.1048
	Ours	**0.9394**	**0.9838**	**0.9941**	**0.3083**	**0.0405**	**0.0619**

Dataset	Method	Classic metrics				New metrics	
		$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	**RMSE**$\downarrow$	P-RMSE$\downarrow$	LRCE$\downarrow$
PanoSUNCG	FCRN [21]	0.9532	0.9905	0.9966	0.2833	0.1094	0.1119
	OmniDepth [45]	0.9092	0.9702	0.9851	0.3171	0.0929	0.0913
	Bifuse [34]	0.9590	0.9838	0.9907	0.2596	0.0967	0.0735
	UniFuse [17]	0.9655	0.9846	0.9912	0.2802	0.0826	0.0884
	Ours	**0.9780**	**0.9961**	**0.9987**	**0.1503**	**0.0537**	**0.0442**
3D60	FCRN [21]	0.9532	0.9905	0.9966	0.2833	0.1681	0.2100
	OmniDepth [45]	0.9092	0.9702	0.9851	0.3171	0.1373	0.1941
	Bifuse [34]	0.9699	0.9927	0.9969	0.2440	0.1229	0.1357
	UniFuse [17]	0.9835	0.9965	0.9987	0.1968	0.0829	0.1021
	DAMO [5]	0.9865	0.9966	0.9987	0.1769	–	–
	SliceNet [26]	0.9788	0.9952	0.9969	–	0.1746	0.1600
	Ours	**0.9876**	**0.9975**	**0.9991**	**0.1492**	**0.0501**	**0.0898**

5.2 Comparison Results

We selected the metrics used in previous work and the two proposed metrics for the quantitative comparison, including RMSE, $\delta(1.25,\ 1.25^2,\ 1.25^3)$ and panorama-specific metrics, LRCE and P-RMSE (We cannot calculate the proposed new metrics due to limitation of the two real-world datasets). RMSE reflects the overall variability. δ exhibits the difference between ground truth and the predicted depth.

Quantitative Analysis. Table 1 shows the quantitative comparison results with the current SOTA monocular panoramic depth estimation solutions on the four popular datasets. As shown in the table, our model achieves the first place in all metrics. In particular, the RMSE metric of our model achieves a 16% improvement on Stanford2D3D, 26% on Matterport3D. Even on the virtual dataset

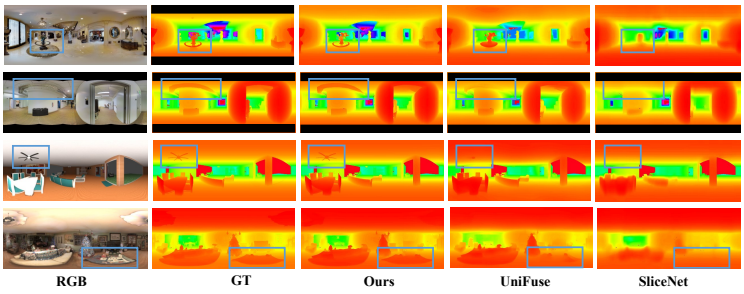

Fig. 7. Qualitative results on Matterport3D, Stanford2D3D, PanoSUNCG, and 3D60. More results can be found in Supplementary Materials

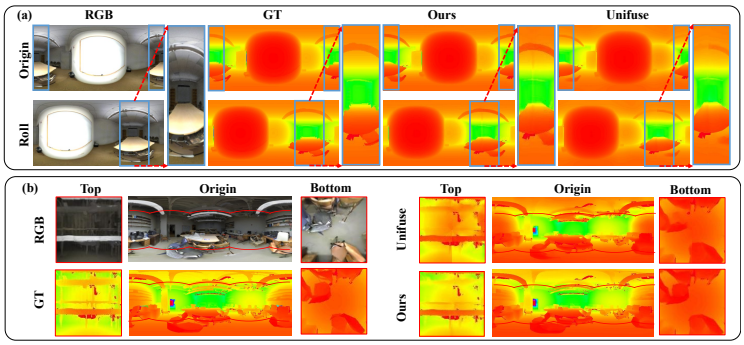

Fig. 8. Visualization of the new metrics' comparison between our method and Unifuse [17]. (a) We stitch the ERP results to observe the depth consistency. (b) We project the areas with massive distortions to cube face to compare the models' performance

PanoSUNCG, there is a 42% improvement on RMSE. But there is just a 16% improvement on 3D60 dataset with RMSE. The improvement is not particularly significant compared to the other three datasets because 3D60 dataset is more extensive, the difference between the models is not obvious. The improvement on δ performance further demonstrates that our model can obtain more accurate prediction results. On the new metric P-RMSE, we achieved an average gain of about 40% on the other two virtual datasets. It indicates that our model is more resilient to the distortion in panoramas. In addition, on LRCE, our model outperforms 40% on PanoSUNCG and 12% on 3D60, showing that our model can better constrain the depth consistency of the left-right boundaries in panoramas, because our network fully considers the seamless property of the sphere.

Qualitative Analysis. Fig. 7 shows the qualitative comparison with the current SOTA approaches. From the figures, we can observe that SliceNet is relatively accurate in predicting regions without distortion. However, the model performance degrades dramatically in regions with distortions or large object deformations.

Although SliceNet can efficiently focus on the global panoramic structures, the depth reconstruction process cannot accurately recover the details, which affects the model's performance. UniFuse can deal with deformation effectively, but it still suffers from incorrect estimating and tends to lose detailed information. From Fig. 8, we can observe that our results are very competitive at boundary and pole areas.

Table 2. Ablation study. We trained on Stanford2D3D for 70 epochs. *a* is the baseline structure developed with CNNs

Index	Transformer	STLM	Token Flow	RMSE	P-RMSE	LRCE
a	✗	✗	✗	0.6704	0.2258	0.2733
b	✓	✗	✗	0.4349	0.2068	0.2155
c	✓	✓	✗	0.3739	0.1825	0.1916
d	✓	✓	✓	**0.3366**	**0.1793**	**0.1784**

5.3 Ablation study

With the same conditions, we validated the key components of our model by ablation study on Stanford2D3D (real-world dataset, small-scale, challenging). As illustrated in Table 2, *a* presents the baseline structure that we use convolutional layers to replace PST blocks; Our network with the traditional attention mechanism is expressed with *b*; *c* indicates our attention module without token flow; Our entire network is shown as *d*.

Transformer vs. CNN. From Table 2, we can observe that the Transformer gains 35% improvements over CNNs in terms of RMSE. Furthermore, qualitative results in *b* are more precise than the CNNs. Essentially, CNNs are a special kind of self-attention. Since the convolutional kernel is fixed, it requires various components or structures or even deeper networks to help the model learn the data patterns. On the other hand, the attention in Transformer is more flexible, and it is relatively easier to learn the patterns.

Effectiveness of Tangent-patches for Transformer. To illustrate the effectiveness of the tangent-patch dividing method, we compared an alternative attention structure that currently performs SOTA in vision Transformers. From Table 2, our network with tangent-patches (*c*) outperforms the attention mechanism (*b*) with 21% on RMSE, 10% on P-RMSE and 12% on LRCE. It proves that tangent-patch can help networks deal with panoramic distortions.

Effectiveness of Token Flow. Since the geometric structures are essential for depth estimation, we add the additional token flows to perceive geometric structures. The results in Table 2 show that our model with the token flow can make P-RMSE more competitive. In Fig. 9, we can observe that the token flow allows the model to estimate the depth details more accurately.

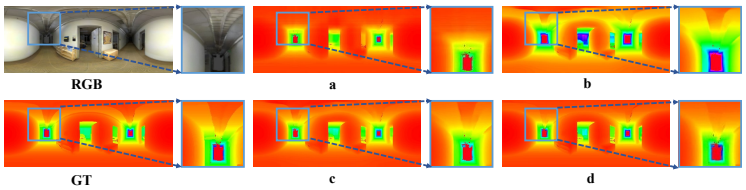

Fig. 9. Qualitative comparison of ablation study. a, b, c, d are the same as Table 2

5.4 Extensibility

We also validate the extensibility of our model by the panoramic segmentation that is also a pixel2pixel task. We did not change any structure of our network and strictly followed the experimental protocol in [32]. As listed in Table 3, the experimental results show that our model outperforms the current SOTA approaches. Due to page limitations, more qualitative comparisons and the results with a high resolution can be found in the supplementary material.

Table 3. Quantitative comparison for semantic segmentation on Stanford2D3D. Results are averaged over the official 3 folds [32]

Dataset	Method	mIoU↑	mAcc↑
Stanford2D3D	TangentImg [12]	41.8	54.9
	HoHoNet [32]	43.3	53.9
	Ours	**48.9**	**64.5**

6 Conclusion

In this paper, we propose the first panorama Transformer (PanoFormer) for indoor panoramic depth estimation. Unlike current approaches, we remove the negative effect of distortions and further model geometric structures by using learnable token flow to establish the network's panoramic perceptions. Concretely, we design a PST block, which can be effectively extended to other learning frameworks. To comprehensively measure the performance of the panoramic depth estimation models, we propose two panorama-specific metrics based on the priors of equirectangular images. Experiments demonstrate that our algorithm significantly outperforms current SOTA methods on depth estimation and other pixel2pixel panoramic tasks, such as semantic segmentation.

Acknowledgement. *This work was supported by the National Key R&D Program of China (No.2021ZD0112100), and the National Natural Science Foundation of China (Nos. 62172032, U1936212, 62120106009).*

References

1. Armeni, I., Sax, S., Zamir, A.R., Savarese, S.: Joint 2d-3d-semantic data for indoor scene understanding. arXiv preprint arXiv:1702.01105 (2017)
2. Bhat, S.F., Alhashim, I., Wonka, P.: Adabins: Depth estimation using adaptive bins. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4009–4018 (2021)
3. Bhoi, A.: Monocular depth estimation: A survey. arXiv preprint arXiv:1901.09402 (2019)
4. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niebner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. In: 2017 International Conference on 3D Vision (3DV). pp. 667–676. IEEE Computer Society (2017)
5. Chen, H.X., Li, K., Fu, Z., Liu, M., Chen, Z., Guo, Y.: Distortion-aware monocular depth estimation for omnidirectional images. IEEE Signal Processing Letters **28**, 334–338 (2021)
6. Cheng, H.T., Chao, C.H., Dong, J.D., Wen, H.K., Liu, T.L., Sun, M.: Cube padding for weakly-supervised saliency prediction in 360 videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1420–1429 (2018)
7. Cheng, X., Wang, P., Zhou, Y., Guan, C., Yang, R.: Omnidirectional depth extension networks. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 589–595. IEEE (2020)
8. Cohen, T.S., Geiger, M., Köhler, J., Welling, M.: Spherical cnns. In: International Conference on Learning Representations (2018)
9. Coors, B., Condurache, A.P., Geiger, A.: Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In: Proceedings of the European conference on computer vision (ECCV). pp. 518–533 (2018)
10. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
12. Eder, M., Shvets, M., Lim, J., Frahm, J.M.: Tangent images for mitigating spherical distortion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12426–12434 (2020)
13. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. Advances in neural information processing systems **27** (2014)
14. Esmaeili, A., Marvasti, F.: A novel approach to quantized matrix completion using huber loss measure. IEEE Signal Processing Letters **26**(2), 337–341 (2019)
15. Esteves, C., Allen-Blanchette, C., Makadia, A., Daniilidis, K.: Learning so (3) equivariant representations with spherical cnns. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 52–68 (2018)
16. Jiang, C., Huang, J., Kashinath, K., Marcus, P., Niessner, M., et al.: Spherical cnns on unstructured grids. arXiv preprint arXiv:1901.02039 (2019)
17. Jiang, H., Sheng, Z., Zhu, S., Dong, Z., Huang, R.: Unifuse: Unidirectional fusion for 360 panorama depth estimation. IEEE Robotics and Automation Letters **6**(2), 1519–1526 (2021)

18. Jin, L., Xu, Y., Zheng, J., Zhang, J., Tang, R., Xu, S., Yu, J., Gao, S.: Geometric structure based and regularized depth estimation from 360 indoor imagery. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 889–898 (2020)
19. Khasanova, R., Frossard, P.: Geometry aware convolutional filters for omnidirectional images representation. In: International Conference on Machine Learning. pp. 3351–3359. PMLR (2019)
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
21. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: 2016 Fourth international conference on 3D vision (3DV). pp. 239–248. IEEE (2016)
22. Lee, Y., Jeong, J., Yun, J., Cho, W., Yoon, K.J.: Spherephd: Applying cnns on a spherical polyhedron representation of 360deg images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9181–9189 (2019)
23. Li, Y., Zhang, K., Cao, J., Timofte, R., Van Gool, L.: Localvit: Bringing locality to vision transformers. arXiv preprint arXiv:2104.05707 (2021)
24. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)
25. Pearson, I.F.: Map Projections: Theory and Applications (1990)
26. Pintore, G., Agus, M., Almansa, E., Schneider, J., Gobbetti, E.: Slicenet: deep dense depth estimation from a single indoor panorama using a slice-based representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11536–11545 (2021)
27. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12179–12188 (2021)
28. Shen, Z., Lin, C., Nie, L., Liao, K., Zhao, Y.: Distortion-tolerant monocular depth estimation on omnidirectional images using dual-cubemap. In: 2021 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6. IEEE (2021)
29. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic scene completion from a single depth image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1746–1754 (2017)
30. Su, Y.C., Grauman, K.: Kernel transformer networks for compact spherical convolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9442–9451 (2019)
31. Sun, C., Hsiao, C.W., Sun, M., Chen, H.T.: Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1047–1056 (2019)
32. Sun, C., Sun, M., Chen, H.T.: Hohonet: 360 indoor holistic understanding with latent horizontal features. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2573–2582 (2021)
33. Tateno, K., Navab, N., Tombari, F.: Distortion-aware convolutional filters for dense prediction in panoramic images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 707–722 (2018)

34. Wang, F.E., Yeh, Y.H., Sun, M., Chiu, W.C., Tsai, Y.H.: Bifuse: Monocular 360 depth estimation via bi-projection fusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 462–471 (2020)
35. Wang, Z., Cun, X., Bao, J., Liu, J.: Uformer: A general u-shaped transformer for image restoration. arXiv preprint arXiv:2106.03106 (2021)
36. Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L.: CVT: Introducing convolutions to vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22–31 (2021)
37. Xiong, B., Grauman, K.: Snap angle prediction for 360 panoramas. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–18 (2018)
38. Xu, Y., Zhang, Z., Gao, S.: Spherical dnns and their applications in 360° images and videos. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
39. Yan, Z., Li, X., Wang, K., Zhang, Z., Li, J., Yang, J.: Multi-modal masked pre-training for monocular panoramic depth completion. arXiv preprint arXiv:2203.09855 (2022)
40. Yan, Z., Wang, K., Li, X., Zhang, Z., Xu, B., Li, J., Yang, J.: Rignet: Repetitive image guided network for depth completion. arXiv preprint arXiv:2107.13802 (2021)
41. Yu-Chuan, S., Kristen, G.: Flat2sphere: Learning spherical convolution for fast features from 360 imagery. In: Proceedings of International Conference on Neural Information Processing Systems (NIPS) (2017)
42. Yuan, K., Guo, S., Liu, Z., Zhou, A., Yu, F., Wu, W.: Incorporating convolution designs into visual transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 579–588 (2021)
43. Yun, I., Lee, H.J., Rhee, C.E.: Improving 360 monocular depth estimation via non-local dense prediction transformer and joint supervised and self-supervised learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 3224–3233 (2022)
44. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable transformers for end-to-end object detection. In: International Conference on Learning Representations (2020)
45. Zioulis, N., Karakottas, A., Zarpalas, D., Daras, P.: Omnidepth: Dense depth estimation for indoors spherical panoramas. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 448–465 (2018)