High-Resolution Depth Estimation for 360° Panoramas through Perspective and Panoramic Depth Images Registration

Chi-Han Peng
National Yang Ming Chiao Tung University
pengchihan@nycu.edu.tw

Jiayao Zhang
ByteDance
jiayao.zhang@bytedance.com

Abstract

We propose a novel approach to compute high-resolution (2048x1024 and higher) depths for panoramas that is significantly faster and qualitatively and qualitatively more accurate than the current state-of-the-art method [23]. As traditional neural network-based methods have limitations in the output image sizes (up to 1024x512) due to GPU memory constraints, both [23] and our method rely on stitching multiple perspective disparity or depth images to come out a unified panoramic depth map. However, to achieve globally consistent stitching, [23] relied on solving extensive disparity map alignment and Poisson-based blending problems, leading to high computation time. Instead, we propose to use an existing panoramic depth map (computed in real-time by any panorama-based method) as the common target for the individual perspective depth maps to register to. This key idea made producing globally consistent stitching results from a straightforward task. Our experiments show that our method generates qualitatively better results than existing panorama-based methods, and further outperforms them quantitatively on datasets unseen by these methods.

1. Introduction

Panoramas with depth information are very useful for 3D computer vision tasks such as novel view synthesis [12, 4, 40], 3D scene understanding (*e.g*. room layout estimation [30]), omnidirectional SLAM [36], and virtual reality (VR) applications [24]. Traditional monocular depth estimation methods built for perspective images cannot directly work on panoramas, as a panorama cannot be converted to a perspective image since the latter can not have field-of-view (FOV) angles exceeding 180 degrees. Therefore, many depth estimation methods that are specially built for panoramas have been proposed [47, 33, 31, 19, 11, 14, 13].

Most of these methods are based on deep neural networks trained on panorama datasets with ground-truth depths such as Matterport3D [6] and Stanford2D3D [2]. The ground-truth depths in such datasets have been calibrated such that they are *globally consistent* (i.e., having the same scale and shift) across all the viewing directions from the camera position. As a result, these methods estimate consistent depth maps for the whole panoramas.

However, we observe two main shortcomings of these methods. First, due to the limitation of GPU memory during the training of the CNN architecture being used, they can only output depth images with resolutions as high as 1024x512. This is far below the native resolutions of RGB panoramas (most modern commodity 360° cameras can shoot panoramas with resolutions exceeding 4096x2048) and is insufficient for VR applications. For example, a 90°-by-90° perspective view rendered out of a 1024x512 panorama would have a resolution of only 256x256. Second, we observe that depth maps estimated by panorama-based methods often lack the same level of detail as the results generated by perspective methods. This may be because they were trained on panoramic RGB-D datasets, which are much smaller and less diverse than the traditional perspective RGB-D datasets that perspective methods were trained on. See Figure 1 for examples.

To address the shortcomings, the authors of [23] proposed a *stitching*-based approach as follows. First, the panorama is partitioned into several perspective views. They used a 20-view partition mimicking the 20 triangle faces of an icosahedron (*i.e*. "tangent images" in [8]). Each view is fed to a modern perspective depth estimation method to get a perspective disparity map (they used MiDaS [22]). Each perspective map is projected back to a common equirectangular domain. A fully panoramic depth map can then be computed by stitching together the projected perspective disparity maps and then converted to depth values. The main challenge of such stitching-based methods is that the perspective maps tend to have different scales and shifts of values because the perspective depth estimation method was run on different fields of view of the panorama. To fix this inconsistency problem, they relied on extensive disparity map alignment and Poisson blending strategies, both become very expensive for high-resolution

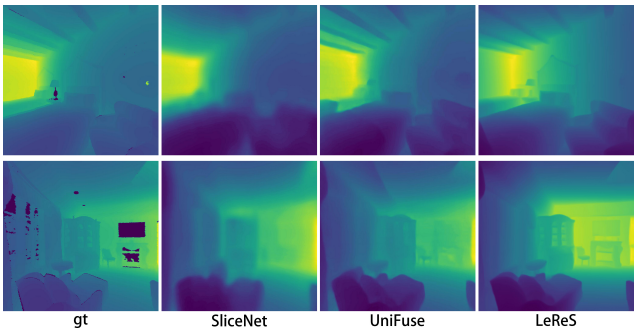

Figure 1. Comparisons of depth estimations by panorama-based methods (SliceNet [19] and UniFuse [11]), and perspective method (LeRes [43]) of the same region in a panorama. We find that LeReS produced qualitatively better results with better levels of details than any panorama-based methods by a large margin.

images.

Instead, we propose a simple solution to the global consistency problem: *leveraging a panoramic depth map as the common target for the perspective depth maps to register to*. Note that such "reference" panoramic depth maps can be generated in real-time by an existing panorama-based method, which is known to produce globally consistent values. Our stitching pipeline is as follows.

We begin with partitioning the panorama into several rectangular regions. For each region, we generate a perspective depth map using LeReS [43]. Next, we generate a full panoramic depth map using a recent panorama-based method such as SliceNet [19] or Unifuse [11]. For each perspective map, we register a low degree (*e.g.* quadratic or cubic) transform function by fitting a set of sampled pixels in the perspective map to the corresponding pixels in the panoramic map in a least-squared sense. Being an optimization problem with only a few variables, the computational cost of this step is very low.

The previous step minimized the differences of scales and shifts between the perspective maps but does not completely erase the visible seams in between. Therefore, we further blend the registered perspective maps by a Poisson-based approach similar to [18]. We opt for computing the final panoramic depth map as the one that fits the Laplacians of the perspective depth maps in a least-squared sense. As the optimization problem is ill-posed (being shift invariant), we regularize it by the $L2$ distances to the reference panoramic depth map with a small weight. Note that our blending approach is similar to the disparity map blending step in [23] with the main difference that our approach no longer needs the spatial-varying weights (*e.g.* "radial" or "frustum"-shaped) in [23] to enforce smoother transitions between the perspective maps.

Through experiments, we show that our method produced qualitatively and qualitatively better results than [23] at a significantly faster speed. Our contributions are summarized as follows:

- We propose a simple, effective, and cost-efficient solution to the global consistency problem of stitching-based methods for high-resolution panoramic depth map generations, namely the registration-based approach.
- Benefited from the effectiveness of the aforementioned step, we find that a straightforward Poisson blending-based approach is sufficient to blend the registered perspective depth maps and erases the visible seams in between, without the needs for complex spatial-varying weights as was done in [23].

2. Related Work

2.1. Panorama-based 3D Modeling and Datasets

Leveraging the increasing popularity of 360° cameras, 3D modeling of indoor scenes based on panoramic image inputs have become a popular research field in recent years. Key tasks include depth estimation [47, 33, 19, 11], room layout estimation [48, 42, 29, 20, 44, 34], object detection and segmentation [37, 31], and more generally 3D reconstruction tasks such as registration of multiple panoramas [41, 35].

Panoramic image datasets with depth information are summarized as follows. Matterport3D [6] and Standford2D3D [3] provide real-world panoramas of diverse indoor scenes with ground-truth depths. Note that although the depths of one panorama are captured through multiple 3D scans (the 3D scanners have limited field-of-views and relied on motors to take multiple scans in a controlled manner), they have been calibrated so that the depth values are consistent across all viewing directions - we don't see any seams between the different views in the depth maps. SunCG [27], Structure3D [45], and Replica360 [28] are synthetic datasets of indoor scenes. They provide photo-realistically rendered indoor RGB-D images and annotations of 3D structures including room layouts. 3D60 [46] is a collective dataset consisting of real and synthetic sources.

2.2. Monocular perspective depth estimation

Monocular (*i.e.* needed just a single image as input instead of stereo ones) perspective depth estimation is a very active research topic in which modern methods can now reliably predict generally accurate depths from arbitrary images in the wild (*i.e.* in unseen datasets). We list methods proposed in recent years with competitive performances: Xian et al. 2018 [38], MegaDepth [15], MiDaS v2 [21] and v3 [22], SGR [39], Huynh et al. 2021 [9], AdaBins [5], and LeReS [43]. Note that some of the methods predict disparity values (e.g., MiDaS), which are the inverse of depths. Authors of [17] proposed a derivative method that calls

an external perspective depth estimation method repeatedly on carefully chosen subsets of the input image, and blend the partial depth estimations to form the final output. The downside is that it is much slower than standard methods.

2.3. Panoramic Depth Estimation

Tateno et al. [32] proposed a deformable convolutional filter that maps perspective views to the equirectangular domain in a distortion-aware manner and used it to perform panoramic depth predictions via numerous, densely sampled perspective depth-prediction calls that could be trained on both panoramic and perspective datasets for perspective depth predictions. In [7] an elaborate discussion of such filters are discussed. OmniDepth [47] was an early method to estimate depths specifically for panoramas in an end-to-end manner. They reported that applying monocular perspective methods directly on equirectangular images led to inferior results. To solve the problem, they trained a customized encoder-decoder network (called "UResNet" in their paper) directly on panoramas with ground-truth depths from synthetic sources. Later, BiFuse [33] found that injecting perspective views of the panoramas (in cubemap formats) into the training process increased the performance considerably. Inspired by the gravity-aligned nature of indoor scenes, SliceNet [19] proposed a novel network architecture that encodes the input panorama into feature vectors that each correspond to a vertical slice of the panorama. HoHoNet [31] leveraged a similar idea in their method that predicts room layouts, depths, and semantic labels simultaneously. Unifuse [11] iterates on the "two-branch" (one feeds the equirectangular projection and another feeds the cubemap projection of the input panorama) network design of BiFuse and proposed a simpler version. PanoDepth [14] uses stereo matching ideas to improve panoramic depth estimations. OmniFusion [13] is a panoramic method via aligning and blending tangent depth maps using transformers. A trend of network design in recent methods is to leverage perspective views sampled on panoramic images while using various strategies to eliminate the distortions and discontinuity artifacts caused in the process [26, 13, 25]. Finally, 360MonoDepth [23] proposed a novel derivative approach to generate high resolution panoramic depth maps by stitching perspective depth maps generated by external methods.

3. Method

We use notations as follows. We assume a 3D world space with right-hand rule in which the +z axis is the "straight-up" direction and the +x axis is chosen arbitrarily. A *viewing direction* is equivalent to a point on the unit sphere centered at the origin, which can be denoted by a spherical coordinate, (ϕ, θ), in which $0^\circ \leq \phi < 360^\circ$ is the *azimuth angle* from the +x axis on the x-y plane in counterclockwise order and $0^\circ \leq \theta \leq 180^\circ$ is the *zenith angle* to the +z axis. An *equirectangular projection* maps a viewing direction to a 2D point trivially as (ϕ, θ). The *equirectangular domain* denotes the rectangle $(0^\circ, 0^\circ)$, $(360^\circ, 0^\circ)$, $(360^\circ, 180^\circ)$, $(0^\circ, 180^\circ)$ in the 2D plane. A *perspective view* is a perspective projection defined by a viewing direction and field-of-view angles in the horizontal and vertical directions ($FOVx$ and $FOVy$ angles in short) in which the camera is at the origin and +z axis is the "up" direction. An *perspective-to-equirectangular (P2E) projection* projects a perspective view to a region in the equirectangular domain. Note that the projected region would not be a rectangle anymore (see Figure 3). A summary of our method is given in Figure 2 and its description. We explain the key steps in the following sub-sections.

3.1. Equirectangular-to-Perspective Partitions

Our method produces depths for a rectangular subset of the equirectangular domain that spans azimuth from 0° to 360° and zenith from 25° to 155°, where all panorama datasets have depth values. We denote this region as the "target domain". We partition the target domain into rectangular sub-regions along horizontal and vertical lines, and then use a perspective depth estimation method such as LeReS [43] to estimate the depth values for each sub-region. We denote a rectangular region that covers azimuth from ϕ_0° to ϕ_1° and zenith from θ_0° to θ_1° as a "partition" $P_{\phi_0,\phi_1,\theta_0,\theta_1}$. Recall that we assume all perspective views have the camera position at origin and the "up" direction at +z axis. Even so, there are still infinitely many ways to construct a perspective view that sufficiently covers a partition. Therefore, we describe an algorithm to find one such perspective view that would bound the region tightly as follows.

First we note that the shape of a perspective view's P2E-projected domain is invariant to rotations along the z-axis, so we limit our analysis to the case of azimuth equals zero. Also, there are reflective symmetries vertically w.r.t. the equator so we need only to analysis the "tilted-up" cases (zenith not greater than 90°) and the corresponding tilted-down cases can be derived trivially. Now, as shown in Figure 3, we observe all possible cases of a perspective view's P2E-projected domains and rectangles centered at the viewing direction (point M in the equirectangular domain) bounded within. These rectangles are the rectangular partitions that a perspective view may cover. We find that in the equirectangular domain, the upper edges of the P2E-projected domain are always closer or equal to the viewing direction (M) than the lower edges along the y-axis. Therefore, we choose the rectangle with left and right edges horizontally aligned to the bottom two corners (same azimuths) of the P2E-projected domain that goes as up as possible until touching the upper edge.

The viewing direction M's 3D coordinate is $(cos(\Theta), 0, sin(\Theta))$, Θ is the vertical tilted-up angle

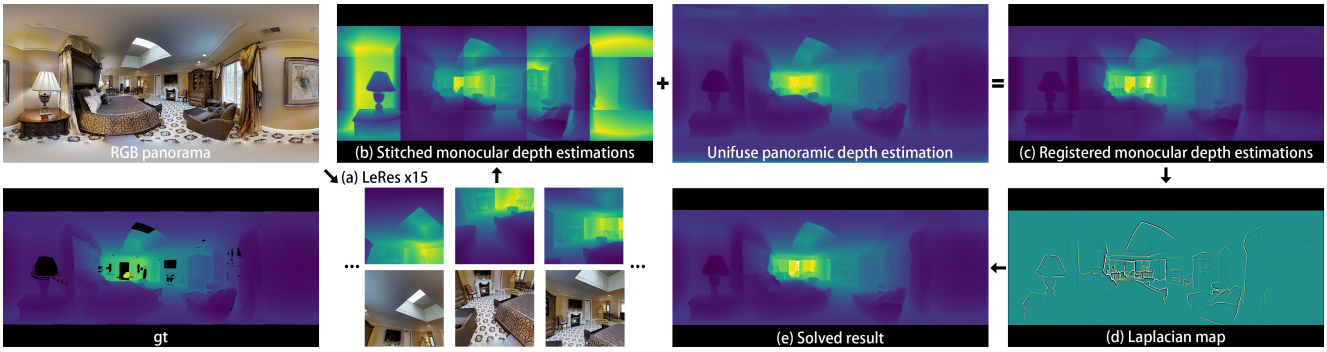

Figure 2. Overview of our method. (a) We first partition the panorama into several perspective views and feed each perspective image to a monocular depth estimation method such as LeReS [43]. Note that most panoramic RGB-D datasets don't have depths in the top and bottom so our method skips those regions. (b) The predicted perspective depth maps are projected to the equirectangular domain and are stitched together to form a panoramic depth map. (c) For each projected perspective depth map, we solve a low-degree function in a least-squared sense that transforms the pixels in the perspective map to the corresponding pixels in a shared "reference" panoramic depth map, which is generated in real-time by a panorama-based method such as SliceNet [19] or UniFuse [11]. (d) We generate a panoramic Laplacian map of the registered depth maps. (e) Finally, we optimize for a new panoramic depth map that fits the Laplacians of the registered depth maps with a small regularization term using L2 distances to the reference panoramic depth map.

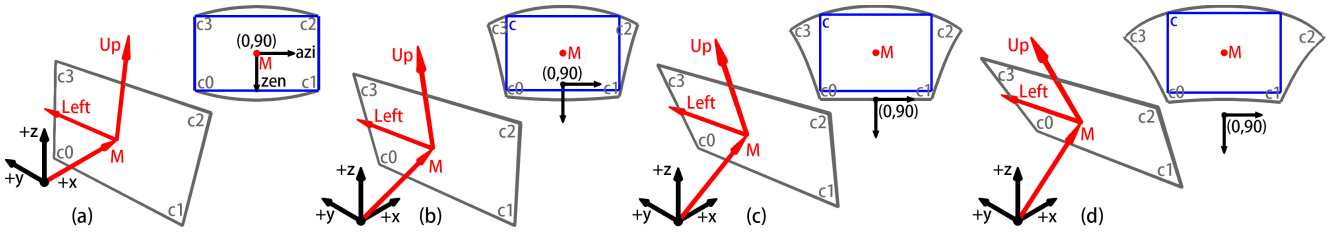

Figure 3. (a) to (d) show four cases of a perspective view with $FOVx = 80°$ and $FOVy = 60°$ in 3D space where in (a) the looking direction, M, 's spherical coordinate is $(0, 90°)$ (on the equator), in (b) it is $(0, 70°)$ (slightly tilted up), in (c) it is $(0, 60°)$ (tilted up and the bottom edge of the image plane lies on the x-y plane), and in (d) it is $(0, 50°)$ (further tilted up). $c0$ to $c3$ denote the four corners of the image plane. $Left$ and Up denote the two axes of the image plane in 3D. In each figure's upper-right corner, we show the P2E-projected region of the perspective view in the equirectangular domain. We show the partition that each view is designated to cover in blue.

from the equator. The two axis of the image plane in 3D are $Left = (0, 1, 0)$ and $Up = (-sin(\Theta), 0, cos(\Theta))$, respectively. To derive the lower-left corner ($c0$)'s azimuth, we first find $c0$'s 3D position as:

$$M + tan(FOVx/2) * Left - tan(FOVy/2) * Up, \quad (1)$$

$FOVx$ and $FOVy$ are the field-of-view angles of the perspective view. $c0$'s azimuth can then be computed and denoted as $c0_\phi$. Note that the 3D positions of the other three corners (denoted as $c1$, $c2$, and $c3$, in counterclockwise order) can be derived similarly. We can now uniquely locate the point c on the upper edge with azimuth equals to $c0_\phi$.

Finally, given a rectangular partition that the perspective view is designated to cover, we derive the corresponding c_ϕ and c_θ. c's 3D position is now fixed. We then solve $FOVy$ such that the upper edge would intersect c. Next, we solve $FOVx$ using the formula for $c0$'s azimuth. ■

For each perspective view of the panorama, the corresponding monocular depth map is computed by a perspective method such as LeReS. By default, we partition the target domains into 3 rows and 5 columns (15 total) of rectangular sub-regions. The rows are divided along zeniths 25°, 60°, 120°, and 155°. The columns are divided along azimuths 0°, 72°, 144°, 216°, and 288°. Comparisons of different ways to partition the target domain are in Section 4.3.

3.2. Perspective-to-Equirectangular Registration

As shown in Figure 2 (b), each partition of the target domain is filled with depth values from the corresponding P2E-projected perspective depth map, using looking directions and $FOVx$ and $FOVy$ chosen according to the algorithm described in Section 3.1. However, each filled partition tends to have different scales and shifts of values. With the common scale and shift being unknown, as has been shown in [23], aligning them using optimization could be an expensive process, especially for high-resolution images.

Instead, we propose the idea of using a "reference" panoramic depth map, in which the depth values are known to be consistent across the whole target domain, as the common target for the perspective depth maps to register to. The reference panoramic depth map can be generated by one of the existing panorama-based methods such as SliceNet, or

UniFuse. For each partition $P_{\phi_0,\phi_1,\theta_0,\theta_1}$, the registration is summarized as solving a linear least-squares optimization problem as follows:

$$\underset{a,b,c,d}{\text{argmin}} \quad \sum_{i=0}^{N-1} (a(x_i)^3 + b(x_i)^2 + cx_i + x_i - X_i)^2 \qquad (2)$$

where x_i and X_i denote the depth values of the i-th sampled pixel in the partition and the corresponding pixel in the reference panoramic depth map, respectively. N denotes the number of sampled pixels in the partition. We sample pixels at every 1 degree horizontally and vertically. Here, we show an algorithm design with a cubic function. We also experimented with other choices (*e.g*. quadratic and linear) and report the finding in Section 4.3. We use Google Ceres Solver [1] to solve the optimization problem. We then use the solved registration function to transform every pixel values in a partition (*i.e*. not just the sampled pixels).

3.3. Laplacian-based Poisson Blending

The aforementioned registration process aligned the scales and shifts of the depth values in every partition (*e.g*. comparing Figure 2 (b) and (c)) but does not completely remove the visible seams in between. To solve this issue, we blend the partitions using a Poisson-based approach. Similar to the traditional gradient-based Poisson blending algorithms (which was used in [23]), we opt for optimizing for a panoramic depth map that fits the Laplacians of the registered perspective maps directly.

We use the standard 3x3 discrete Laplacian operator (shown in Equation 4). For each partition, we also sample "padded" depth values outside its region in the equirectangular domain by slightly expanding the equirectangular regions of each partition and updating the FOV angles respectively. By default, we expand the region of each partition 5 pixels horizontally and 2 pixels vertically (for both 2K and 4K cases). This means that there would be small overlaps of Laplacian values between the partitions and we take the average of Laplacians at the overlapping pixels. This padding step serves the purpose of slightly smoothing out the Laplacians between adjacent partitions. The Laplacian-based blending is expressed as an optimization problem as follows:

$$\begin{aligned} \underset{x_{i,j},\, 0\le i<W,\, 0\le j<H}{\text{argmin}} \quad & \sum_{j=1}^{H-2} \sum_{i=1}^{W-2} ((l_{i,j} - L_{i,j})^2 \\ & + \gamma\,(x_{i,j} - X_{i,j})^2) \\ \text{subject to} \quad & x_{i,j} \ge 0, \forall i,j \end{aligned} \qquad (3)$$

where $x_{i,j}$ is the depth value at pixel coordinate (i,j) in the panoramic depth map to be solved. W and H are the width and height of the panoramic depth map. $l_{i,j}$ is the Laplacian value at pixel (i,j) computed by the standard 3x3 discrete Laplacian operator:

$$\begin{aligned} l_{i,j} = 4x_{i,j} - x_{i-1,j} - x_{i+1,j} - x_{i,j-1} - x_{i,j+1}, \\ \forall\, 1 \le i \le W-2, 1 \le j \le H-2. \end{aligned} \qquad (4)$$

and $L_{i,j}$ are the "target" Laplacian values computed by the same formula in the registered depth values at each partitions (one example is shown in Figure 2 (d)). $X_{i,j}$ is the depth value at pixel coordinate (i,j) in the reference panoramic map. In short, the objective function has two terms - a Laplacian term and a regularization term using L2 distances to the reference panoramic depth map. We set γ, the weight for the regularization term, to 1e-4.

The optimization problem is solved using the standard Jacobi iterative method with the depth values of the reference panoramic depth map as the initial guess. To speed up solving, we solve the problem in a multi-scale manner. That is, we first solve the problem in a reduced-resolution version of the image buffer, pass the solved values to a finer-resolution buffer, solve again, and so on until the problem is solved in the original resolution. For the 2048x1024 case, we solve in 3 levels (*i.e*. 512x256, 1024x512, and 2048x1024). The iterations for the finest level are chosen to be 50 as we observed that the problem usually converged (with residuals lower than 0.1% of initial values) during iteration 40 to 50. We then set iterations for the subsequent levels to be 100 and 200. For the 4096x2048 case, we solve in 4 levels (50, 100, 150, and 200 iterations).

4. Results and Analysis

We tested our method and [23] on a computer with Intel i7-10700 CPU, 32GB ram, and NVidia RTX 2070 GPU. We compare the timing statistics of our method versus [23] in Table 4. In summary, our method is 3.05 times faster (2048x1024 outputs), and 1.94 times faster (4096x2048 outputs), than [23]. We draw each perspective view in a 1024x989 resolution, which is enough to form 4K outputs when stitched. Same as in prior methods (including OmniDepth, BiFuse, SliceNet, and UniFuse), when comparing a computed depth map, ω, to a ground-truth depth map, Ω, a "median-scaling" step is first taken by multiplying every depth value in ω by the median value of Ω divided by the median value of ω. Note that for the Matterport dataset, depth values are reported in meters. We choose HoHoNet [31], SliceNet [19], and UniFuse [11], as the methods to generate the panoramic depth maps for our method.

4.1. Quantitative Comparisons

In Table 4.1, we quantitatively compare our method versus recent panorama-based methods and [23] on the Matterport dataset testing split in 2K resolution outputs and the synthetic Replica360 dataset in 2K and 4K resolutions

Res	Pers.D	Pano.D	Reg.	Blend	Total	[23]
2K	4.27	0.122	0.22	10.89	15.50	47.3
4K	4.27	0.122	0.23	34.27	38.89	75.3

Table 1. Timing comparisons. We list the times (in seconds) for our perspective-view rendering and depth estimation (LeReS 15 times), panoramic depth estimation (UniFuse), registration step (cubic), and blending step. We solved for 100 panoramas and calculated the average. For [23], to reproduce similar results as shown in their paper, we set the options to: Poisson blending, 3 levels of disparity map alignments, and MiDaS V2. Note that [23] also has a one-time "pre-processing" step which took about 40 seconds.

provided by [23]. Same as in [23], we up-sample 1K results generated by previous methods to 2K by bilinear interpolation. Our method produced quantitatively better results than [23] on all three datasets by large margins. Our method, when paired with SliceNet or HoHoNet, scored nearly as good as the top performers (SliceNet and HoHoNet) on the Matterport 2K dataset. But when tested on the Replica360 2K and 4K datasets, in which the previous panorama-based methods were not trained on, our method outperforms all existing panorama-based methods. Also note that on Replica360 2K and 4K datasets, our method improved the quantitative score of the panorama-based method that is being used in every cases. We find [23] to perform competitively on the Replica360 2K and 4K datasets (through accuracy metrics), but not on the Matterport 2K dataset.

4.2. Qualitative Comparisons

We show qualitative comparisons in Figure 5 on the Matterport 2K and Replica360 2K and 4K datasets. We found UniFuse produced qualitatively best results among panorama-based methods. For stitching-based methods, we found [23] generates clearer images than UniFuse, while our method generated slightly clearer results than [23] with finer details in general. We found that in [23]'s results, the estimated depths can deviate from the ground truth when observed at larger scales. For example, observe the inconsistent depths of the two white walls (at roughly the same distances to the camera) of the Replica360 2K example in Figure 5. More examples are shown in the Supplementary Materials.

4.3. Analysis

See Table 4.1 and Figure 4 for supporting quantitative statistics and qualitative examples.

Ablation studies: Only performing the registration step led to blocks with visible seams (*e.g.* Figure 2 (c)). Similarly, only performing the blending step led to worse quantitative and qualitative results. Skipping the padding in the blending step may led to glitches at the partition boundaries.

Method	RMSE	MAE	AbsRel	RMSE_{log}
Reg. only	+3.97%	+8.49%	+14.62%	+8.23%
Blend only	+8.32%	+12.23%	+26.69%	+56.13%
No padding	+0.02%	+0.00%	+0.01%	+0.13%
Avg. s/s	+6.79%	+11.26%	+24.18%	+47.20%
Smoothing	+6.36%	+9.92%	+17.49%	+28.01%
4-fold	+0.36%	+0.70%	+1.51%	+1.47%
3-fold	+0.05%	+0.41%	+0.08%	+0.35%
MiDaS	+0.26%	+1.06%	+1.75%	+2.89%
Linear	-0.52%	-0.77%	-1.53%	-1.60%
Quadratic	-0.51%	-0.75%	-0.88%	-0.77%

Table 2. Quantitative statistics of ablation studies and alternative design choices. We show relative ratios w.r.t. the case of using UniFuse as the panorama-based method on Matterport 2K dataset. "Reg. only" means only performing the registration step. "Blend only" means only performing the blending step. "No padding" means skipping the padding in the blending step. "Avg. s/s" means simply registering each partition to the average scale and shift. "Smoothing" means replacing the blending step by smoothing depth values at partition boundaries. "4-fold" means taking a partition of 3 rows by 4 columns (horizontally divided along azimuths 0°, 90°, 180°, and 270°), and "3-fold" means taking a partition of 3 rows by 3 columns (along azimuths 0°, 120°, and 240°). "MiDaS" means using MiDaS for perspective depth estimations. "Linear" and "Quadratic" means using alternative degree registration functions.

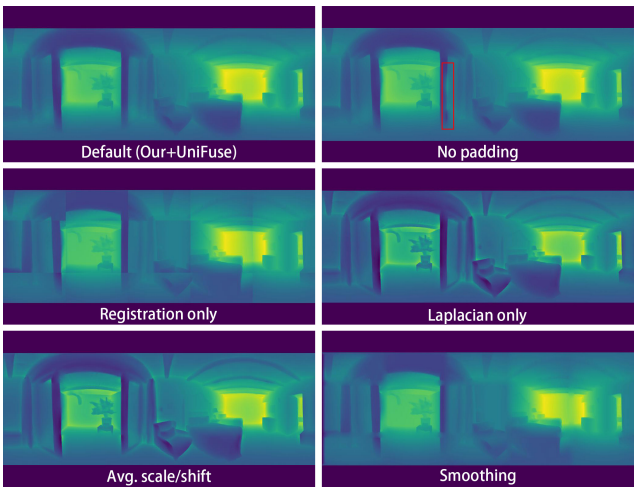

Figure 4. . Qualitative comparisons of ablation studies and alternative design choices. Only performing the blending step led to over-saturated results. Similarly, replacing the registration step with a trivial averaging of scales and shifts led to similar results. Skipping the padding may lead to glitches at partition boundaries. The "smoothing" result shows that a simple color smoothing cannot effectively remove the inconsistency between partitions.

Next, we tried swapping key designs of our method with trivial approaches. First, instead of using a common reference map, we tried to register the individual perspective depth maps by transforming them to the average of scale

DS	Method	Error metric ↓ RMSE	MAE	AbsRel	$RMSE_{log}$	Accuracy metric ↑ δ_1	δ_2	δ_3
Matterport 2K	Bifuse [33]	0.6350	0.3675	0.1367	0.0901	82.77%	94.46%	97.53%
	HoHoNet [31]	0.4707	0.2620	0.0967	0.0629	90.50%	97.27%	97.09%
	SliceNet [19]	0.4463	0.2153	0.0665	0.0513	95.17%	98.07%	99.54%
	UniFuse [11]	0.6040	0.3309	0.1110	0.0728	87.79%	95.70%	98.38%
	360MonoDepth [23]	0.7729	0.5106	0.2653	0.1253	60.38%	85.55%	94.70%
	Our (HoHoNet)	0.4791	0.2655	0.1004	0.0662	90.23%	97.09%	98.93%
	Our (SliceNet)	0.4949	0.2569	0.0883	0.0648	91.51%	97.21%	99.10%
	Our (UniFuse)	0.6107	0.3333	0.1152	0.0766	87.08%	95.36%	98.20%
Replica360 2K	Bifuse [33]	0.0555	0.0416	0.2150	0.1121	71.56%	91.39%	96.32.%
	HoHoNet [31]	0.0300	0.0193	0.1116	0.0671	90.31%	95.90%	98.11%
	SliceNet [19]	0.0403	0.0279	0.1590	0.0896	85.15%	93.88%	96.44%
	UniFuse [11]	0.0362	0.0248	0.1336	0.0774	86.87%	95.94%	97.72%
	360MonoDepth [23]	0.0706	0.0456	0.1813	0.0865	78.48%	93.56%	98.34%
	Our (HoHoNet)	0.0272	0.0182	0.1074	0.0643	90.98%	96.07%	98.28%
	Our (SliceNet)	0.0380	0.0272	0.1553	0.0862	85.35%	94.31%	96.77%
	Our (UniFuse)	0.0354	0.0247	0.1334	0.0762	87.00%	96.08%	97.80%
Replica360 4K	Bifuse [33]	0.0642	0.0485	0.2446	0.1266	63.27%	89.13%	95.65%
	HoHoNet [31]	0.0357	0.0249	0.1359	0.0744	85.17%	94.63%	96.61%
	SliceNet [19]	0.0473	0.0341	0.1891	0.0994	78.31%	93.17%	96.77%
	UniFuse [11]	0.0394	0.0289	0.1480	0.0818	82.20%	96.26%	98.54%
	360MonoDepth [23]	0.0611	0.0400	0.1667	0.0815	80.04%	95.25%	98.47%
	Our (HoHoNet)	0.0332	0.0239	0.1309	0.0709	86.07%	94.98%	96.76%
	Our (SliceNet)	0.0444	0.0335	0.1831	0.0975	76.80%	93.27%	97.34%
	Our (UniFuse)	0.0380	0.0281	0.1447	0.0795	82.69%	96.66%	98.65%

Table 3. Quantitative comparisons of our method (using HoHoNet, SliceNet, or UniFuse to generate the reference panoramic depth maps) versus previous panorama-based methods and the stitching-based method proposed in [23]. RMSE, MAE, AbsRel, and $RMSE_{log}$ measure the root mean squared error, mean absolute error, mean relative error, and RMSE in log-10 space (same as in UniFuse and BiFuse), of depth values. δ_1, δ_2, and δ_3 measure the ratios of pixels with mutual relative errors below 1.25, 1.25^2, and 1.25^3, respectively. Highlighting: best, second-best, third-best.

and shift. Second, we tried blending the registered perspective depth maps by simply smoothing the depth values at partition boundaries. In both cases, the results become worse quantitatively and qualitatively.

Alternative degree registration functions: We tried linear and quadratic registration functions instead of cubic functions. We found that using simpler functions actually led to slightly better quantitative scores. We still opt for cubic functions because we think the additional degrees of freedom may be beneficial for coping with unseen cases.

Switching to MiDaS: To verify that the differences of performance between [23] and our method are not simply due to the different perspective methods that were used, we tried using MiDaS instead of LeReS to do the perspective depth estimations. We find the results to be slightly worse, but do not make up all the differences between the two methods.

Different partitions: We tried other ways to partition the target domain into perspective views. We find that sparser partitions (larger FOVs) led to slightly worse performance. This may be because perspective methods run on larger FOVs output less detailed estimations in general (see discussions in [17]). A benefit of using fewer partitions is shorter computation times of the perspective depth maps generation step (which is not the performance bottleneck).

Limitations: Our results may be negatively affected by major errors of the panoramic depth maps being used. For examples, for unseen cases such as outdoor scenes, some of the existing panorama-based methods may produce very low quality results and negatively affect our results (two such examples are OmniDepth [47] and BiFuse run on outdoor scenes, as shown in [23]'s website). Same as in [23], another limitation is the accuracy of the estimated perspective depth maps being used. In either cases, our method is poised to benefit from advances in both panoramic and perspective depth estimation methods.

5. Conclusion

We show that a bottleneck of stitching-based panoramic depth estimation methods, *i.e.* the global consistency problem, can be solved satisfactorily and very efficiently by

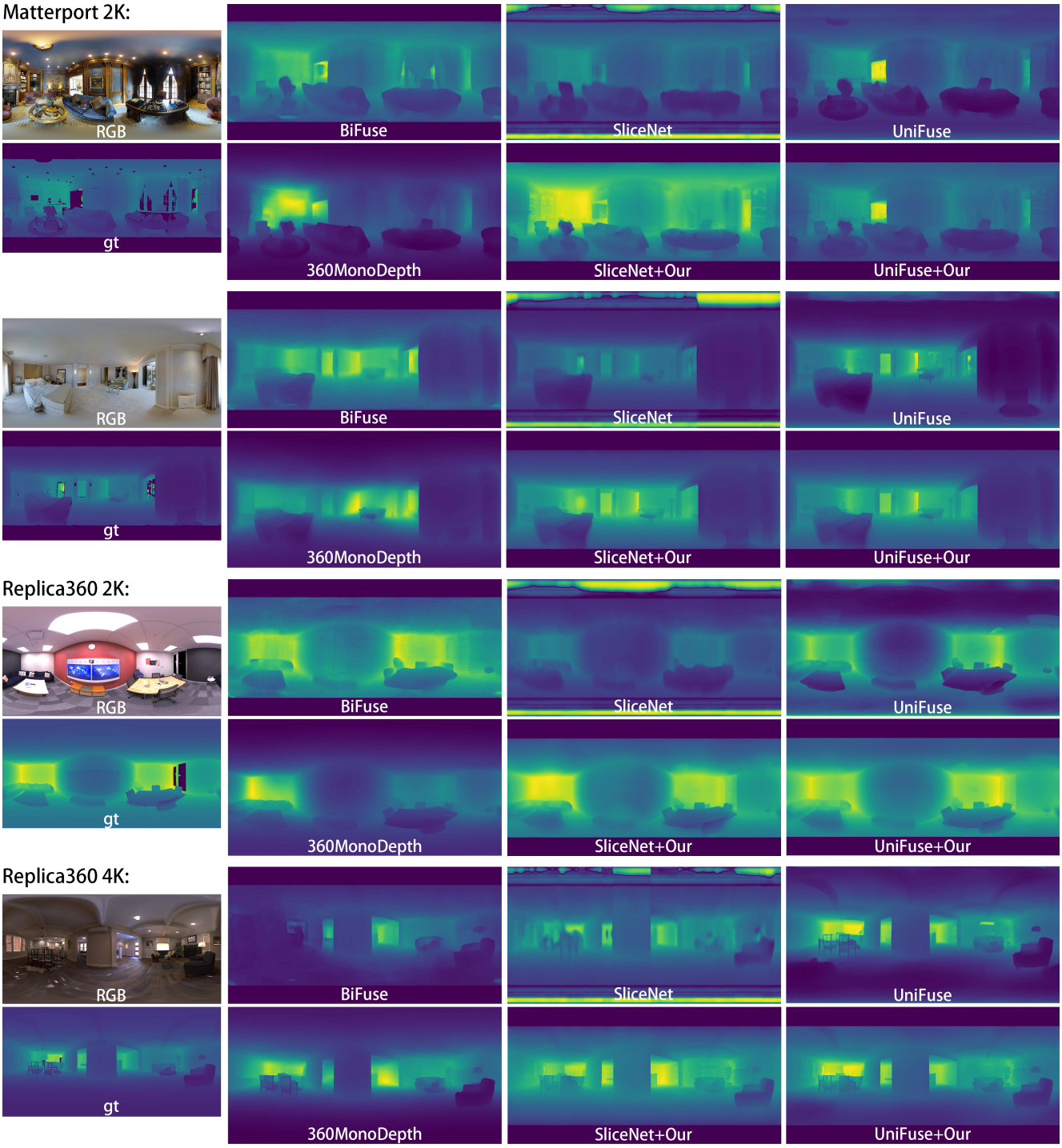

Figure 5. . Qualitative comparisons of results generated by our method (using SliceNet or UniFuse to generate the reference panoramic depth maps), previous panorama-based methods, and the stitching-based method in [23] (360MonoDepth).

our registration-based approach. Accordingly, we propose a streamlined stitching pipeline that outperforms current state-of-the-arts method [23] quantitatively and qualitatively and is much faster. For future work, our main goal is speed improvement, either by developing GPU-based Laplacian solvers inspired by [10] to solve the blending step or alternatively, using style transfer-based approaches.

References

[1] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver, 3 2022.

[2] I. Armeni, A. Sax, A. R. Zamir, and S. Savarese. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *ArXiv e-prints*, Feb. 2017.

[3] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017.

[4] Tobias Bertel, Mingze Yuan, Reuben Lindroos, and Christian Richardt. OmniPhotos: Casual 360° VR photography. *ACM Transactions on Graphics*, 39(6):266:1–12, Dec. 2020.

[5] S. Farooq Bhat, I. Alhashim, and P. Wonka. Adabins: Depth estimation using adaptive bins. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4008–4017, Los Alamitos, CA, USA, jun 2021. IEEE Computer Society.

[6] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

[7] Marc Eder, True Price, Thanh Vu, Akash Bapat, and Jan-Michael Frahm. Mapped convolutions. 2019.

[8] Marc Eder, Mykhailo Shvets, John Lim, and Jan-Michael Frahm. Tangent images for mitigating spherical distortion. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[9] Lam Huynh, Phong Nguyen-Ha, Jiří Matas Esa Rahtu, and Janne Heikkilä. Guiding monocular depth estimation using depth-attention volume. 2021.

[10] Stefan Jeschke, David Cline, and Peter Wonka. A gpu laplacian solver for diffusion curves and poisson image editing. *ACM Trans. Graph.*, 28(5):1–8, dec 2009.

[11] Hualie Jiang, Zhe Sheng, Siyu Zhu, Zilong Dong, and Rui Huang. Unifuse: Unidirectional fusion for 360° panorama depth estimation. *IEEE Robotics and Automation Letters*, 2021.

[12] Johannes Kopf, Kevin Matzen, Suhib Alsisan, Ocean Quigley, Francis Ge, Yangming Chong, Josh Patterson, Jan-Michael Frahm, Shu Wu, Matthew Yu, Peizhao Zhang, Zijian He, Peter Vajda, Ayush Saraf, and Michael F. Cohen. One shot 3d photography. *Transactions on Graphics (Proceedings of SIGGRAPH)*, 39(4), 2020.

[13] Yuyan Li, Yuliang Guo, Zhixin Yan, Xinyu Huang, Duan Ye, and Liu Ren. Omnifusion: 360 monocular depth estimation via geometry-aware fusion. In *2022 Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, USA, June 2022.

[14] Yuyan Li, Zhixin Yan, Ye Duan, and Liu Ren. Panodepth: A two-stage approach for monocular omnidirectional depth estimation. In *2021 International Conference on 3D Vision (3DV)*, pages 648–658. IEEE, 2021.

[15] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[16] Yiqun Mei, Yuchen Fan, and Yuqian Zhou. Image super-resolution with non-local sparse attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3517–3526, June 2021.

[17] S. Mahdi H. Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yağız Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. 2021.

[18] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, jul 2003.

[19] Giovanni Pintore, Marco Agus, Eva Almansa, Jens Schneider, and Enrico Gobbetti. Slicenet: deep dense depth estimation from a single indoor panorama using a slice-based representation: Supplementary material. 2021.

[20] Giovanni Pintore, Marco Agus, and Enrico Gobbetti. AtlantaNet: Inferring the 3D indoor layout from a single 360 image beyond the Manhattan world assumption. In *Proc. ECCV*, August 2020.

[21] Renx00E9; Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12159–12168, 2021.

[22] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis amp; Machine Intelligence*, 44(03):1623–1637, mar 2022.

[23] Manuel Rey-Area, Mingze Yuan, and Christian Richardt. 360MonoDepth: High-resolution 360deg $monocular depth estimation. In CVPR$, 2022.

[24] Ana Serrano, Incheol Kim, Zhili Chen, Stephen DiVerdi, Diego Gutierrez, Aaron Hertzmann, and Belen Masia. Motion parallax for 360° rgbd video. *IEEE Transactions on Visualization and Computer Graphics*, 2019.

[25] Zhijie Shen, Chunyu Lin, Kang Liao, Lang Nie, Zishuo Zheng, and Yao Zhao. Panoformer: Panorama transformer for indoor 360 depth estimation. *arXiv e-prints*, pages arXiv–2203, 2022.

[26] Z. Shen, C. Lin, L. Nie, K. Liao, and Y. Zhao. Distortion-tolerant monocular depth estimation on omnidirectional images using dual-cubemap. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, Los Alamitos, CA, USA, jul 2021. IEEE Computer Society.

[27] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[28] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[29] Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1047–1056, 2019.

[30] Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[31] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Hohonet: 360 indoor holistic understanding with latent horizontal features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2573–2582, 2021.

[32] Keisuke Tateno, Nassir Navab, and Federico Tombari. Distortion-aware convolutional filters for dense prediction in panoramic images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[33] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Bifuse: Monocular 360 depth estimation via bi-projection fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 462–471, 2020.

[34] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Led2-net: Monocular 360deg layout estimation via differentiable depth rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12956–12965, 2021.

[35] Haiyan Wang, Will Hutchcroft, Yuguang Li, Zhiqiang Wan, Ivaylo Boyadzhiev, Yingli Tian, and Sing Bing Kang. Psmnet: Position-aware stereo merging network for room layout estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[36] Changhee Won, Hochang Seok, Zhaopeng Cui, Marc Pollefeys, and Jongwoo Lim. Omnislam: Omnidirectional localization and dense mapping for wide-baseline multicamera systems. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 559–566, 2020.

[37] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9068–9079, 2018.

[38] Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, Yang Xiao, Ruibo Li, and Zhenbo Luo. Monocular relative depth perception with web stereo data supervision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[39] Ke Xian, Jianming Zhang, Oliver Wang, Long Mai, Zhe Lin, and Zhiguo Cao. Structure-guided ranking loss for single image depth prediction. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[40] Jiale Xu, Jia Zheng, Yanyu Xu, Rui Tang, and Shenghua Gao. Layout-guided novel view synthesis from a single indoor panorama. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16438–16447, 2021.

[41] Sheng Yang, Beichen Li, Yan-Pei Cao, Hongbo Fu, Yu-Kun Lai, Leif Kobbelt, and Shi-Min Hu. Noise-resilient reconstruction of panoramas and 3d scenes using robot-mounted unsynchronized commodity rgb-d cameras. *ACM Transactions on Graphics (TOG)*, 39(5):1–15, 2020.

[42] Shang-Ta Yang, Fu-En Wang, Chi-Han Peng, Peter Wonka, Min Sun, and Hung-Kuo Chu. Dula-net: A dual-projection network for estimating room layouts from a single rgb panorama. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3363–3372, 2019.

[43] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (CVPR)*, 2021.

[44] Wei Zeng, Sezer Karaoglu, and Theo Gevers. Joint 3d layout and depth prediction from a single indoor panorama image. In *European Conference on Computer Vision*, pages 666–682. Springer, 2020.

[45] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Proceedings of The European Conference on Computer Vision (ECCV)*, 2020.

[46] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, Federico Alvarez, and Petros Daras. Spherical view synthesis for self-supervised 360 depth estimation. In *2019 International Conference on 3D Vision (3DV)*, pages 690–699. IEEE, 2019.

[47] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. Omnidepth: Dense depth estimation for indoors spherical panoramas. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 448–465, 2018.

[48] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2051–2059, 2018.