# GPR-Net: Multi-view Layout Estimation via a Geometry-aware Panorama Registration Network

Jheng-Wei Su[1] Chi-Han Peng[2] Peter Wonka[3] Hung-Kuo Chu[1]

[1]National Tsing Hua University [2]National Yang Ming Chiao Tung University
[3]KASUT

## Abstract

*We present a room layout estimation framework that jointly learns wide baseline panorama registration and layout estimation given a pair of 360° panoramas. To effectively tackle the wide baseline registration problem, we introduce a novel end-to-end supervised Geometry-aware Panorama Registration Network or GPR-Net that exploits the layout geometry and computes fine-grained correspondences on the layout boundary, instead of the global pixel-space. GPR-Net consists of two main parts. The geometry transformer learns a set of 1D horizon features sampled on the panorama. These 1D feature maps encode geometric cues describing the ceiling-wall and floor-wall layout boundaries, and the correspondence and co-visibility between layout boundaries. These learned geometric cues are further used for direct regression of relative pose (translation and rotation) with a pose transformer. The final layout is then obtained by registering the two layouts using the estimated pose and taking the union of the two individual layouts derived from the estimated layout boundary maps. Experimental results indicate that our method achieves state-of-the-art performance in both panorama registration and layout estimation on a large-scale indoor panorama dataset ZInD [3]. Our code is available online*[1].

## 1. Introduction

In this paper, we tackle the problem of room layout estimation from multiple 360° panoramas. Many approaches that estimate room layouts from a single panorama have been proposed [16, 23, 29, 32]. However, these methods did not take advantage of "multi-view" data in which multiple panoramas are taken to better capture a single room. These kinds of data are commonly seen in public indoor datasets such as ZInD [3], Matterport3D [1], Gibson [27], and Structure3D [31] in which photographers often take multiple panoramas to better capture complex, non-convex rooms that are partially occluded from just a single location.

This multi-view layout estimation problem is recently tackled by a work named PSMNet [26]. The idea of PSMNet is to build an architecture that first registers two panoramas in their ceiling view projections and then jointly estimates a 2D layout segmentation. An important aspect of their architecture is that the layout estimation and registration can be trained jointly. However, a major limitation of PSMNet (also mentioned in their paper) is that the architecture relies on an initial approximate registration. The authors argued that such an approximate registration could be given either manually or computed by external methods such as Structure from Motion (SfM) methods. While a manual registration may work, the method would no longer be automatic.

When experimenting with existing methods for approximate registration, we observed that they frequently make registration errors and even fail to provide a registration in a substantial number of cases. The main reason is that the required registration mainly falls into the category of wide baseline registration with only two given images. For example, our results show that the state-of-the-art SfM method OpenMVG [14] fails to register 77% of panorama pairs from our test dataset. It is thus impractical to assume an independent algorithm that can reliably give an approximate solution to the challenging wide baseline registration problem. In addition, relying on such an algorithm moves a critical part of the problem to a pre-process.

Therefore, we set out to develop a complete stereo panorama registration and layout estimation framework that no longer relies on an approximate registration given as input as shown in Figure 1. To achieve this, we propose a novel Geometry-aware Panorama Registration Network, or GPR-Net, based on the following design ideas. First, our experiments indicate that a global (pixel-space) registration that directly regresses pose parameters (i.e., translation and rotation) is too ambitious. Instead, we propose to support the direct pose regression by first learning more fine-grained

[1]https://github.com/ericsujw/GPR-Net

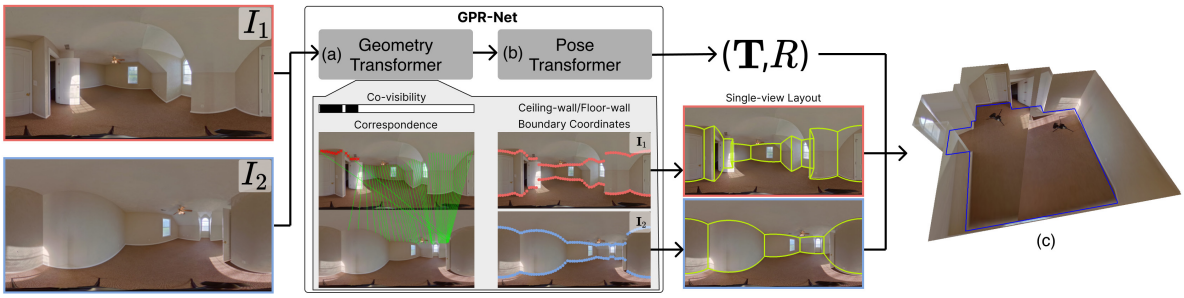


Figure 1. **The proposed geometry-aware panorama registration and layout estimation framework.** Given two panorama images, we introduce GPR-Net that aims to learn fine-grained geometric cues, including boundary coordinates, correspondence, and co-visibility, via a *geometry transformer*, and then directly regresses the relative pose (translation and rotation) via a *pose transformer*. The final layout is then obtained by registering the two layouts using the estimated pose and taking the union of the two individual layouts derived from the estimated layout boundary maps.

correspondences in a different space. We devise a *geometry transformer* that conceptually samples the layout boundaries of two input layouts and estimates rich geometric cues on the sampled locations (Figure 1a). Specifically, for each boundary sample in each of the two panoramas, it estimates the ceiling-wall and floor-wall boundary coordinates. In addition, it estimates the correspondence map from the samples in the first panorama to the second panorama and a co-visibility map describing if a sample in the first panorama is visible in the second panorama. Each of these maps (layout boundary, correspondence, and co-visibility) is a 1D sequence of values. This representation has the advantage of having more supervision signal for fine-grained estimation, thus leading to better learning performance. Then, we feed the learnt latent features of the geometry transformer into a *pose transformer* that directly regresses the relative pose (translation and rotation) (Figure 1b). In addition to the direct pose regression, we alternatively evaluate on a RANSAC-based pose computed by the estimated layouts, correspondence and co-visibility. To obtain the final 3D layout, we first compute a 3D layout for each panorama based on the estimated ceiling-wall and floor-wall boundary maps followed by taking the union of two layouts registered with the estimated pose (Figure 1c).

We extensively validate our model by comparing with the state-of-the-art panorama registration and layout estimation methods on a large-scale indoor panorama dataset ZInD [3]. The experimental results demonstrate that our model is superior to competing methods by achieving a significant performance boost in both panorama registration accuracy (Rotation error@2.5°: +62.27%, Translation error@2.5°: +45.98%) and reconstruction accuracy (2D IoU +6.77%).

In summary, our contributions are as follows:

- We propose the first complete stereo panoramic layout estimation framework. Our architecture jointly learns the layout and registration from data, is end-to-end trainable, and does not rely on a pose prior.
- We devise a novel panorama registration framework to effectively tackle the wide baseline registration problem by exploiting the layout geometry and computing a fine-grained correspondence of samples on the layout boundaries.
- We achieve state-of-the-art performance on ZInD [3] dataset for both the panorama registration and layout reconstruction tasks.

## 2. Related Work

### 2.1. Single-view room layout estimation

There exist many methods to estimate the room layouts from just a single image taken inside an indoor environment. Methods that take only one perspective image include earlier attempts that relied on image clues and optimization [7, 8, 19] and later neural networks [13, 28]. Capturing the increasing availability and popularity of full 360° panoramic images, the seminal work by Zhang et al. [30] proposed to take panoramas as native inputs for scene understanding. Recently, several methods were proposed to predict the room layouts from a single panorama using neural networks. A major difference between these methods is the assumption on the shape of the room layouts - from being strictly a cuboid [32], Manhattan world [23, 29], to general 2D layouts (Atlanta world) [16]. For our work, we choose to adopt the Manhattan assumption because more corresponding data is available. See Zou et al. [33] for a thorough survey on predicting Manhattan room layouts from a single panorama. More recent methods delivered state-of-the-art performance by transforming the problem into a depth-estimation one [26] or by leveraging powerful transformer-based network architecture [12]. Although these single-view methods perform well in the cuboid and

L-shape rooms, they tend to fail in the large-scale, complex and non-convex rooms where a single-view panorama covers only part of the whole space due to occlusion.

### 2.2. Panorama registration

Image registration, i.e., finding transformations between the cameras of two or multiple images taken of the same scene, is a key component of Structure-from-Motion (SfM). We refer to [5, 34] for a comprehensive survey and an extensive study. Summarizing the surveys, registration problems can be categorized by: 1) the assumptions about the camera model, e.g., perspective (pinhole camera), weak-perspective, or orthographic, 2) the assumptions about the transformation, e.g., rigid, affine, or general non-rigid, and 3) the types of the image inputs, e.g., perspective images or full 360° panoramas, and with/without depths. In addition, the difficulty differs greatly on whether the images are taken densely/closely or sparsely/far apart.

Modern takes on registration problems often leverage state-of-art programs/libraries such as COLMAP [21] and OpenMVG [14]. Our problem falls into a lesser-studied category: *registering rigid transforms between sparse panoramas*. While there exist methods that tackle sparse perspective image inputs [4,20] and methods that handle panoramas natively [10, 15, 24], our results show that we can improve upon the state-of-the-art panorama registration methods in our sparse view setting. A key bottleneck was that traditional SfM methods often fail to handle the wide baseline registration problem where the views are far apart from each other. Chen et al. [2] factorize the continuous 5-dimensional solution space of relative camera poses into discrete probability distributions with a novel four unit-vector parameterization, and directly learn the relative camera pose of two wide baseline perspective images. Shabani et al. [22] proposed an extreme SfM framework that utilizes semantics (i.e., room type, doors, windows, etc) to match indoor panoramas with few visual overlaps. In contrast to previous methods that perform the registration in the global pixel-space, we propose a novel learning-based panorama registration framework that directly compute the registration between two panoramas without any prior knowledge.

CoVisPose [9] shares the same goal with ours in the registration part. We argue that there are 4 differences. First, we are the first to jointly handle pose and layout estimation which could mutually benefits each of the two tasks, however the CoVisPose [9] focuses on the panorama registration only. Table 1 shows an improvement in layout estimation when using GT registration indicates that layout estimation improves only due to joint training. Second, CoVisPose [9] uses the floor boundary of the visible area. By contrast, we estimate the floor and the ceiling of room boundary. Therefore, we can derive the full 3D layout directly, but CoVisPose [9] can not. Third, our method can increasing the number of output tokens, however the CoVisPose [9] can not. In addition, the ceiling coordinates and increasing tokens show the improvements in Table 3, and Table 4 and Table 6 respectively. Fourth, we will make the code for training and testing available. The code of CoVisPose [9] will likely not become available.

### 2.3. Scene reconstruction using sparse panoramas

Attempts to reconstruct indoor scenes using just a handful of RGB panoramas as inputs [17, 18] are nascent but promising since photographers are adapting 360° cameras into their workflows (e.g., Matterport 3D capture system [1]) and it is awkward to capture dense panoramic inputs due to camera/tripod setups. While previous methods assume that all the input panoramas are already registered, PSMNet [26] introduces the first learning-based framework that jointly estimates the room layout and registration given a pair of panoramas. However, it still has a major bottleneck that an initial approximate (noisy) registration must be given (e.g., either manually specified or computed by external methods) during both the training and inference stages. Our GPR-Net is also an end-to-end deep neural network that jointly learns the room layout and panorama registration. Most importantly, our model does not rely on a pose prior and is thus suitable for real-world application scenarios.

## 3. Methodology

### 3.1. Network architecture

Figure 2 illustrates the GPR-Net architecture, which consists of two main blocks, geometry transformer and pose transformer. The geometry transformer learns a set of 1D horizon tokens that encode geometric cues (layout boundary, correspondence and co-visibility) sampled on the panorama, while the pose transformer directly regresses the relative pose using features learned from the geometry transformer.

**Geometry transformer.** First, we feed two (vertically) axis aligned panoramas $\mathbf{I}_1$ and $\mathbf{I}_2$ into a ResNet-50 [6] feature extractor and generate two feature maps of resolution $16\times8$. Following the encoder-decoder transformer architecture, we feed the extracted feature maps into the transformer encoder block using the two sets of $16 \times 8$ pixels as input tokens. The output tokens of the transformer encoder block will be used for cross attention in the transformer decoder block. We use a 2D $UV$ coordinate system to parameterize a panorama image with $(u, v)$ in $[0, 1] \times [-1, 1]$. The $u$ coordinate describes the horizontal position and the $v$ coordinate the vertical position. We want to query multiple values (ceiling-wall boundary, floor-wall boundary, correspondence, co-visibility) for different $u$ coordinates. We therefore uniformly sample the $u$ coordinate $\in [0, 1]$ with $N$

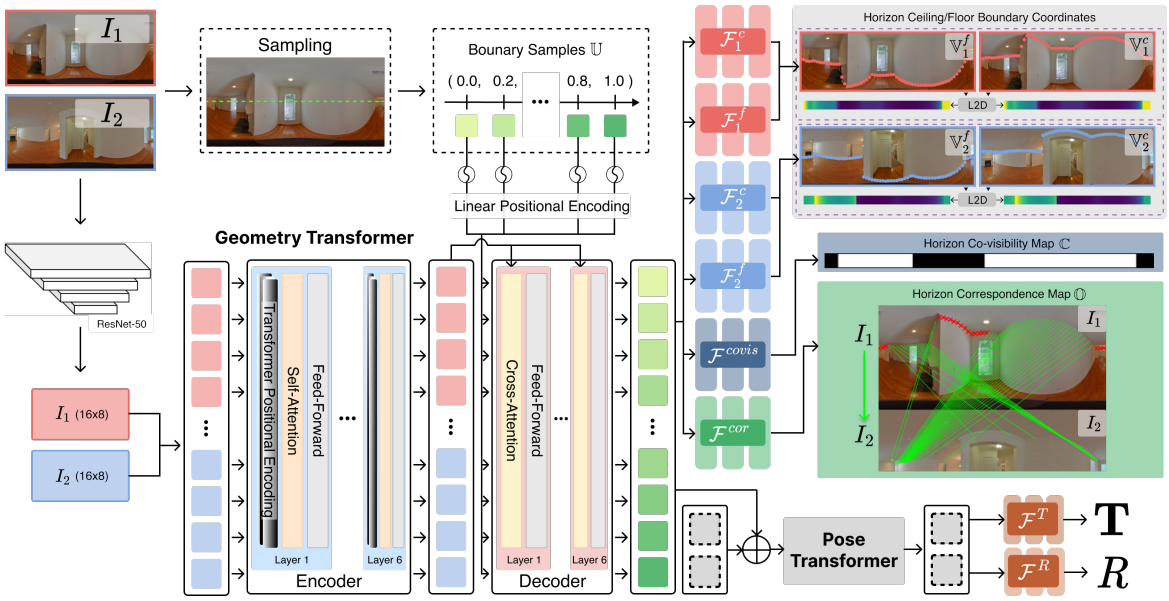


Figure 2. **GPR-Net architecture.** The network consists of two main blocks. The geometry transformer takes two panoramas as input and extracts feature maps as tokens by ResNet-50. These tokens are fed into an encoder-decoder transformer for learning rich geometric cues. We then sample the layout boundary to get a sequence of query tokens as inputs to the transformer decoder. The output tokens of the transformer decoder encode multiple types of information about the boundary samples and are further processed using multiple auxiliary MLP heads to generate supervision signals, including ceiling/floor boundary coordinates $\mathbb{V}$, correspondence map $\mathbb{O}$, and co-visibility map $\mathbb{C}$. The pose transformer takes input as the output tokens of the geometry transformer decoder with two extra query tokens for predicting the translation $\mathbf{T} \in \mathbb{R}^2$ and rotation $R \in SO(2)$ of the relative pose.

evenly distributed samples. We obtain the set $\mathbb{U} = \{u_i\}_{i=1}^{N}$. The query samples are encoded by linear positional encoding and are the input tokens of the transformer decoder. The output tokens of the transformer decoder encode multiple types of information about the boundary samples and are used as inputs to the pose transformer. To enable detailed supervision during training, the output tokens are further processed using multiple auxiliary MLP heads. Specifically, we use: (i) *layout* MLP heads $\mathcal{F}_k^c$ and $\mathcal{F}_k^f$ with $k = (1, 2)$. The outputs of $\mathcal{F}_k^c$ and $\mathcal{F}_k^f$ are the $v$ coordinates of the ceiling and floor boundaries in the panorama image $\mathbf{I}_k$, respectively. We denote the $v$ coordinates of the ceiling boundaries as $\mathbb{V}_k^c$ and the floor boundaries, $\mathbb{V}_k^f$. For each $\mathbb{V}_k^{\{c,f\}}$, we further exploit the Layout-to-Depth (L2D) transformation [25] to generate a corresponding horizon depth map $\mathbb{D}_k^{\{c,f\}}$ to provide a better supervision on the layout depth; (ii) *correspondence* MLP head $\mathcal{F}^{cor}$ that outputs a horizon correspondence map $\mathbb{O} = \{o_i\}_{i=1}^{N}$, where $o_i$ indicates the correspondence between $u_i \in \mathbf{I}_1$ and $o_i \in \mathbf{I}_2$; and (iii) *co-visibility* MLP head $\mathcal{F}^{covis}$ that outputs a horizon co-visibility map $\mathbb{C} = \{c_i\}_{i=1}^{N}$, where $c_i$ is a value $\in [0, 1]$, encoding whether the $i-th$ element in $\mathbb{O}$ should be considered ($c_i = 1$) or not ($c_i = 0$) in the pose estimation.

**Pose transformer.** The input to the pose transformer are the output tokens of the geometry transformer decoder with two extra query tokens for predicting the translation and rotation components of the relative pose. The transformer network follows a standard transformer encoder architecture. We would like to remark that we only consider 3-DoF transformations in ZInD [3] following [26]. Therefore, the outputs are two tokens representing latent features of relative pose and are further processed with *translation* MLP head $\mathcal{F}^T$ and *rotation* MLP head $\mathcal{F}^R$ to extract a translation vector $\mathbf{T} \in \mathbb{R}^2$ and rotation $R \in SO(2)$, respectively.

### 3.2. Loss functions

Here we elaborate on the layout, correspondence, co-visibility, cycle-consistency, and pose loss functions used for training our network.

**Layout loss** calculates the low-level geometry loss between the predicted horizon depth maps $\mathbb{D}_{\{1,2\}}^{\{c,f\}}$ of input panoramas $\mathbf{I}_1$ and $\mathbf{I}_2$ and the corresponding ground-truth $\overline{\mathbb{D}}_{\{1,2\}}^{\{c,f\}}$. The loss is defined as follows:

$$\mathcal{L}_{layout} = \frac{1}{M} \sum_{j=(c,f)} \sum_{k=(1,2)} \|\mathbb{D}_k^j - \overline{\mathbb{D}}_k^j\|_1, \qquad (1)$$

where $M$ is the dimension of the horizon depth maps.

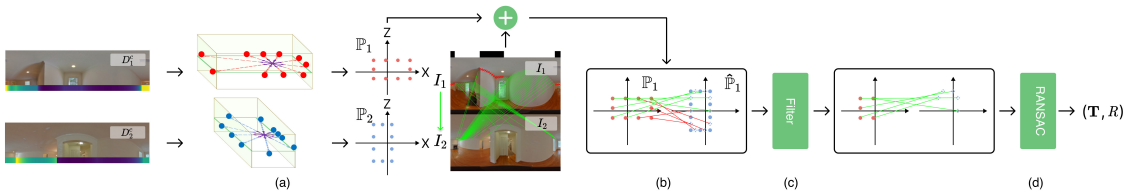


Figure 3. **Extracting boundary information**. We (a) utilize the elementary geometric transformations [25] to obtain: 1) two horizon-depth maps $\mathbb{D}_1^c$ and $\mathbb{D}_2^c$, and 2) two 2D point sets $\mathbb{P}_1$ and $\mathbb{P}_2$. (b) Considering the non-uniformly distributed horizon correspondence map $\mathbb{O}$, we extract the one-to-one point-wise correspondence $\mathbb{P}_1$ and $\hat{\mathbb{P}}_1$ by interpolating $\mathbb{P}_2$ with $\mathbb{O}$. Optionally (for ablation studies), (c) we filter out some of the in-covisible pairs via horizon co-visibility map $\mathbb{C}$ and (d) robustly estimate final pose parameters by RANSAC.

**Co-visibility loss** evaluates the predicted normalized horizon co-visibility map $\mathbb{C} = \{c_i\}_{i=1}^N$ with respect to the corresponding ground-truth $\overline{\mathbb{C}} = \{\bar{c}_i\}_{i=1}^N$. The loss is defined as follows:

$$\mathcal{L}_{covis} = \frac{1}{N}\sum_{i=1}^{N} \alpha \bar{c}_i \cdot \log(c_i) + (1-\bar{c}_i)\cdot\log(1-c_i), \quad (2)$$

where $\alpha$ is the hyperparameter for weighting the positive samples. We use $\alpha$ because there are a lot more positive than negative samples in the ground truth.

**Correspondence loss** calculates the difference between the predicted horizon correspondence map $\mathbb{O} = \{o_i\}_{i=1}^N$ and the corresponding ground-truth $\overline{\mathbb{O}} = \{\bar{o}_i\}_{i=1}^N$. The loss is defined as follows:

$$\mathrm{L}_{cor} = \frac{1}{N}\sum_{i=1}^{N} \begin{cases} \min(\|o_i - \bar{o}_i\|_1, \|1 - \bar{o}_i + o_i\|_1), & \text{if } \bar{c}_i \geq 0.5 \\ 0, & \text{otherwise} \end{cases}, \quad (3)$$

where we use a cyclic loss instead of the simple L1 loss between the predicted and ground-truth correspondence to adopt the coordinate system in equirectangular projection.

**Cycle-consistency loss [11]** enforces the network outputs to be cycle-consistent and adapts the network to different ray casting positions in contrast to the uniformly sampled ray casting positions. We reverse the order of the two panoramas $\mathbf{I}_1$ and $\mathbf{I}_2$, and treat the ground-truth horizon correspondence map $\{\bar{o}_i\}_{i=1}^N$ as input and the original input $\{u_i\}_{i=1}^N$ as target correspondence. Here, we denote the predicted horizon correspondence as $o'_i = \mathcal{F}^{cor}(\mathcal{F}^{geo}(\bar{o}_i))$ and predicted co-visibility map $c'_i = \mathcal{F}^{covis}(\mathcal{F}^{geo}(\bar{o}_i))$, where $\mathcal{F}^{geo}$ is our geometry transformer. We separate the cycle-consistency loss into two parts as follows:

$$\mathrm{L}_{cycle}^{cor} = \frac{1}{N}\sum_{i=1}^{N} \begin{cases} \min(\|o'_i - u_i\|_1, \|1 - u_i + o'_i\|_1), & \text{if } \bar{c}_i \geq 0.5 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

and

$$\mathrm{L}_{cycle}^{covis} = \frac{1}{N}\sum_{i=1}^{N} \alpha \bar{c}_i \cdot \log(c'_i) + (1-\bar{c}_i)\cdot\log(1-c'_i). \quad (5)$$

Here, we still use the ground-truth horizon co-visibility map $\{\bar{c}_i\}_{i=1}^N$ as the target in $\mathcal{L}_{cycle}^{covis}$ since the order of $\{\bar{o}_i\}_{i=1}^N$ and $\{o_i\}_{i=1}^N$ is the same.

**Pose loss** calculates the difference between the predicted transformation $\mathbf{T}$ and $R$ with ground-truth transformation $\bar{\mathbf{T}}$ and $\bar{R}$. The loss is defined as follows:

$$\mathrm{L}_{pose} = \begin{cases} \|R - \bar{R}\|_1 + 0.5(\mathbf{T} - \bar{\mathbf{T}})^2, & \text{if } \|\mathbf{T} - \bar{\mathbf{T}}\|_1 < 1 \\ \|R - \bar{R}\|_1 + \|\mathbf{T} - \bar{\mathbf{T}}\|_1 - 0.5, & \text{otherwise} \end{cases} \quad (6)$$

Finally, the overall loss function used in our network is defined as follows:

$$\begin{aligned} \mathcal{L}_{total} =& \lambda_1\mathcal{L}_{layout} + \lambda_2\mathcal{L}_{cor} + \lambda_3\mathcal{L}_{covis} + \\ & \lambda_4\mathcal{L}_{cycle}^{cor} + \lambda_5\mathcal{L}_{cycle}^{covis} + \lambda_6\mathcal{L}_{pose}, \end{aligned} \quad (7)$$

where $\lambda_{\{1,\ldots,6\}}$ are the hyperparameters for weighting the loss functions.

### 3.3. Extracting boundary and correspondence information

We extract an explicit representation of the layout boundaries and correspondence information between them as follows (See also Figure 3). We use a 3D Cartesian coordinate system to perform the registration process where the $y$-axis is the up axis, the camera center is the origin, and the XZ-plane is parallel to floor and ceiling. We reuse the elementary geometric transformations described by Wang et al. [25] to obtain the following: (i) the horizon depth maps $\mathbb{D}_1^c = \{d_i^1 \in \mathbb{R}^1\}_{i=1}^M$ and $\mathbb{D}_2^c = \{d_i^2 \in \mathbb{R}^1\}_{i=1}^M$, which are derived from the predicted ceiling layout boundaries $\mathbb{V}_1^c$ and $\mathbb{V}_2^c$, and represent the $M$ evenly sampled distances from the origin to the predicted layout boundary in the XZ-plane; (ii) two sets of 2D points $\mathbb{P}_1 = \{p_i^1 \in \mathbb{R}^2\}_{i=1}^M$ and $\mathbb{P}_2 = \{p_i^2 \in \mathbb{R}^2\}_{i=1}^M$. These points are on the layout boundary in the XZ-plane. Each point corresponds to a depth value in $\mathbb{D}_1^c$ or $\mathbb{D}_2^c$, respectively (Figure 3a). Note that the conversion of [25] includes a resampling step from $N$ to $M$ boundary samples. In order to compute a one-to-one point-wise correspondence between $\mathbb{P}_1$ and $\mathbb{P}_2$, we exploit the estimated horizon correspondence map $\mathbb{O}$. Since $\mathbb{O}$ is not necessary a uniform distribution, for each $p_i^1 \in \mathbb{P}_1$, we compute its corresponding point via interpolating $\mathbb{P}_2$ with $\mathbb{O}$ (Figure 3(b)). We denote the corresponding boundary points as $\hat{\mathbb{P}}_1 = \{\hat{p}_i^1\}_{i=1}^M$. Using this explicit boundary description, we can alternatively register the two panoramas

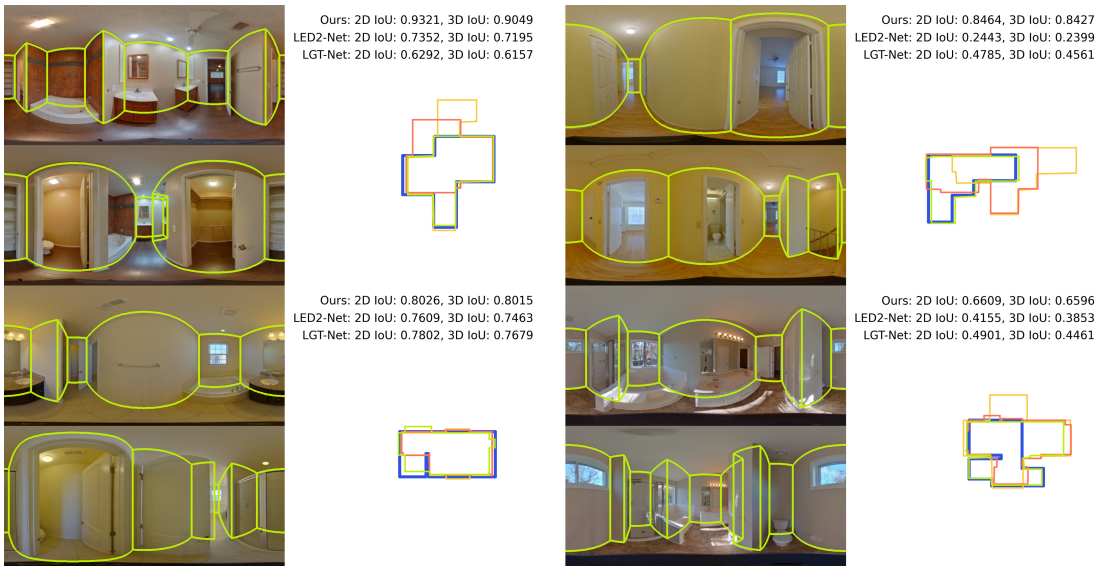


Figure 4. **Visual comparisons.** We show visual comparisons with other competing methods categorized by difficulty. We select cases where our reconstruction accuracy is within the range of top 10%(top-left), 20% (top-right), 50%(bottom-left), and bottom 10%(bottom-right) in our test set. The first column shows two input panoramas with their estimated layouts. The second column shows the stacked ground-truth layout, our layout, LED$^2$-Net's layout, and LGT-Net's layout in blue, green, yellow, red, respectively.

using RANSAC for ablation studies. To apply RANSAC, we filter out the matched pair in $\mathbb{P}_1$ and $\hat{\mathbb{P}}_1$ according to the horizon co-visibility map $\mathbb{C}$ (Figure 3(c)). Then the final pose parameters, i.e., translation ($\mathbf{T} \in \mathbb{R}^2$) and rotation ($R \in SO(2)$), are computed using the RANSAC algorithm (Figure 3(d)).

### 3.4. Layout fusion

Given the relative camera pose between input panoramas, we combine two individual partial layouts into a unified one as follows. First, for each input panorama, we adopt the same post-processing procedure as [25] to convert the estimated ceiling-wall boundary coordinates $\mathbb{V}_k^c$ and floor-wall boundary coordinates $\mathbb{V}_k^f$ into a 2D layout map $L_k$ and a layout height $H_k$. Then, we register two 2D layout maps using the estimated relative camera pose and then combine them into a complete 2D layout map via a union operation $L_{final} = L_1 \cup L_2$. The final 3D layout is obtained by extruding $L_{final}$ with the average layout height $(H_1 + H_2)/2$.

## 4. Results

### 4.1. Experimental Settings

**Dataset.** We conduct all the experiments on the Zillow Indoor Dataset (ZInD), which contains 67,448 indoor panoramas. We follow the same procedure as PSMNet to select the panorama pair instances and obtain training (105256), validation (12376), and test (12918) pairs. While we tried to match the PSMNet test protocol as closely as possible and exchanges multiple emails with the authors to that effect, we are still waiting for the authors of PSMNet to release their testing code and dataset split.

**Baselines.** We compare our method with the following state-of-the-art layout reconstruction models, LED$^2$-Net [25], LGT-Net [12], and PSMNet [26]. Since LED$^2$-Net and LGT-Net are single-view layout estimation methods, we first estimate the layout for each view, register two input panoramas using OpenMVG, and then perform a union operation to obtain the final result. Note that in cases where OpenMVG fails to produce a registration, we use average ground-truth pose of the training datasets. To evaluate the performance on the panorama registration, we compare our GPR-Net with OpenMVG, a popular Structure-from-Motion library that supports stereo panorama matching. We followed the official settings for the feature extractor and correspondence matching and applied 'incrementalv2' mode for SfM operation to achieve the best reconstruction rate. Please refer to the supplementary for additional comparisons. Regarding to the concurrent work CoVisPose [9], we are not able to provide either quantitative and qualitative comparison because the code or details on the testing setup or dataset splits are not released yet.

**Evaluation metrics.** To measure the quality of layout reconstruction, we adopt the 2D and 3D IoU metrics. We also use $\delta^i$ to measure accuracy in panorama pixel space. To measure the accuracy of relative pose estimation, we cal-

Table 1. **Quantitative evaluation on layout reconstruction.** We categorize the test dataset according to the spatial overlap ratio and highlight the best results in yellow. The * symbol means that the numbers are reported in PSMNet [26].

| Pose | Method | Overall | | | Overlap-High | | | Overlap-Medium | | | Overlap-Low | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 2D IoU↑ | $\delta^i$ ↑ | 3D IoU↑ | 2D IoU↑ | $\delta^i$ ↑ | 3D IoU↑ | 2D IoU↑ | $\delta^i$ ↑ | 3D IoU↑ | 2D IoU↑ | $\delta^i$ ↑ | 3D IoU↑ |
| w/ GT | LED$^2$-Net [25] | 0.8364 | 0.9557 | 0.8131 | 0.8555 | 0.9682 | 0.8295 | 0.8430 | 0.9595 | 0.8206 | 0.8076 | 0.9374 | 0.7854 |
| | LGT-Net [12] | 0.8388 | 0.9537 | 0.8126 | 0.8622 | 0.9686 | 0.8309 | 0.8445 | 0.9591 | 0.8206 | 0.8068 | 0.9307 | 0.7822 |
| | PSMNet* [26] | 0.8101 | 0.9238 | - | 0.8571 | 0.9349 | - | 0.8013 | 0.9253 | - | 0.7693 | 0.9074 | - |
| | GPR-Net (Direct) | 0.8449 | 0.9603 | 0.8211 | 0.8641 | 0.9702 | 0.8364 | 0.8515 | 0.9633 | 0.8295 | 0.8158 | 0.9458 | 0.7932 |
| w/o GT | LED$^2$-Net [25] | 0.5783 | 0.8737 | 0.5637 | 0.6185 | 0.8948 | 0.6011 | 0.5808 | 0.8724 | 0.5674 | 0.5344 | 0.8544 | 0.5209 |
| | LGT-Net [12] | 0.5733 | 0.8742 | 0.5569 | 0.6202 | 0.8956 | 0.5991 | 0.5773 | 0.8740 | 0.5627 | 0.5207 | 0.8531 | 0.5060 |
| | PSMNet* [26] | 0.7577 | 0.9217 | - | 0.8480 | 0.9371 | - | 0.7473 | 0.9210 | - | 0.6673 | 0.9040 | - |
| | GPR-Net (Direct) | 0.8254 | 0.9568 | 0.8023 | 0.8541 | 0.9686 | 0.8268 | 0.8340 | 0.9592 | 0.8126 | 0.7838 | 0.9414 | 0.7624 |

Table 2. **Quantitative evaluation on panorama registration.** We categorize the test dataset according to the spatial overlap ratio and highlight the best results in yellow.

| Overlap | Method | Success (% ↑) | Rotation | | | | | Translation angle | | | | | Translation vector | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Mn (° ↓) | Med (° ↓) | 2.5° ↑ | 5° ↑ | 10° ↑ | Mn (° ↓) | Med (° ↓) | 2.5° ↑ | 5° ↑ | 10° ↑ | Mn ($m.$ ↓) | Med ($m.$ ↓) | 0.5$m.$ ↑ |
| Overall | GPR-Net (RANSAC) | 99.94 | 4.4552 | 1.6138 | 0.6733 | 0.8791 | 0.9464 | 6.3809 | 1.8979 | 0.6026 | 0.8107 | 0.9051 | 0.3203 | 0.1172 | 0.8907 |
| | GPR-Net (Direct) | 100 | 2.0687 | 0.7670 | 0.9661 | 0.9822 | 0.9858 | 5.3703 | 2.3710 | 0.5238 | 0.7953 | 0.9296 | 0.2365 | 0.1468 | 0.9176 |
| | OpenMVG | 22.93 | 72.1806 | 66.9710 | 0.2138 | 0.2138 | 0.2138 | 73.6517 | 68.4432 | 0.1006 | 0.1369 | 0.1627 | - | - | - |
| | DirectionNet | 100 | 26.7651 | 4.3237 | 0.3434 | 0.5435 | 0.7172 | 26.3579 | 9.8102 | 0.1428 | 0.2876 | 0.5074 | - | - | - |
| High | GPR-Net (RANSAC) | 100 | 2.1114 | 1.3080 | 0.7806 | 0.9599 | 0.9897 | 5.9692 | 1.6772 | 0.6511 | 0.8494 | 0.9317 | 0.1260 | 0.0788 | 0.9783 |
| | GPR-Net (Direct) | 100 | 1.2105 | 0.7811 | 0.9794 | 0.9935 | 0.9951 | 6.8568 | 2.2832 | 0.5395 | 0.8082 | 0.9193 | 0.1437 | 0.1076 | 0.9821 |
| | OpenMVG | 30.50 | 69.4531 | 60.2696 | 0.2719 | 0.2719 | 0.2719 | 67.6999 | 59.4363 | 0.1381 | 0.1934 | 0.2297 | - | - | - |
| | DirectionNet | 100 | 3.3270 | 1.9398 | 0.6163 | 0.8818 | 0.9843 | 7.6541 | 4.9998 | 0.2488 | 0.5003 | 0.7935 | - | - | - |
| Medium | GPR-Net (RANSAC) | 100 | 3.9025 | 1.6306 | 0.6761 | 0.8867 | 0.9519 | 5.4238 | 1.8655 | 0.6109 | 0.8270 | 0.9168 | 0.2964 | 0.1340 | 0.8882 |
| | GPR-Net (Direct) | 100 | 1.6852 | 0.7571 | 0.9696 | 0.9866 | 0.9902 | 4.4067 | 2.3046 | 0.5396 | 0.8172 | 0.9457 | 0.2411 | 0.1624 | 0.9121 |
| | OpenMVG | 22.04 | 75.3876 | 72.5451 | 0.2067 | 0.2067 | 0.2067 | 74.0510 | 69.6638 | 0.0974 | 0.1307 | 0.1535 | - | - | - |
| | DirectionNet | 100 | 30.1753 | 5.4123 | 0.2868 | 0.4667 | 0.6903 | 27.4318 | 11.9780 | 0.1287 | 0.2345 | 0.4384 | - | - | - |
| Low | GPR-Net (RANSAC) | 99.78 | 7.6195 | 2.1033 | 0.5622 | 0.7870 | 0.8951 | 8.1646 | 2.2390 | 0.5416 | 0.7476 | 0.8611 | 0.5499 | 0.1560 | 0.8070 |
| | GPR-Net (Direct) | 100 | 3.4979 | 0.7700 | 0.9476 | 0.9643 | 0.9697 | 5.3262 | 2.6039 | 0.4843 | 0.7497 | 0.9157 | 0.3221 | 0.1866 | 0.8616 |
| | OpenMVG | 0.1957 | 74.4980 | 72.2465 | 0.1665 | 0.1665 | 0.1665 | 79.8191 | 77.3498 | 0.0681 | 0.0897 | 0.1097 | - | - | - |
| | DirectionNet | 100 | 37.9143 | 7.2713 | 0.2225 | 0.3936 | 0.5901 | 35.4612 | 14.9505 | 0.0931 | 0.1984 | 0.3806 | - | - | - |

culate angular error in both the estimated translation and rotation. Since our method is up-to-scale, we further measure translation error in meters. For the angular and translation errors, we also report the ratio of testing samples below $\{2.5°, 5°, 10°\}$ and $0.5m$ error thresholds, respectively. As non-learning based methods may fail in establishing the correspondences, we also report the success rate (%) of the registration process. To better understand the performance of our model and competing baselines when come across cases with varying difficulty level. We follow the protocol of PSMNet to split the test dataset into Overlap-High, Overlap-Medium, and Overlap-Low matching the propotions of PSMNet.

**Implementation details.** We implemented our model in PyTorch and conducted experiments on a single NVIDIA V100 with 32GB VRAM and training for 12 days. The resolution of the panoramas is resized to $512 \times 256$. We use the Adam optimizer with b1=0.9 and b2=0.999. The learning rates of the transformer and the ResNet-50 are 1e-4 and 1e-5, and the batch size is set to 8. We empirically set $\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 1, \lambda_4 = 1, \lambda_5 = 1, \lambda_6 = 1$ in Equation 7, $\alpha = 0.1$ in Equation 2, $M = 256$ in Equation 1, and $N = 256$ in Section 3.2.

## 4.2. Layout reconstruction performance

In this experiment, we evaluate both the qualitative and quantitative performance of our model on the layout reconstruction task by comparing with baselines. The qualitative results are shown in Figure 4. In short, our method produces more accurate layout reconstruction than LED$^2$-Net and LGT-Net. Please refer to the supplementary for more visual comparisons. Note that visual comparison with PSMNet is infeasible since the relevant code and data are not released. Our GPR-Net (Direct) also achieves the best performance against all the baselines across all evaluation metrics, as shown in Table 1. In the setting without ground-truth (GT) camera poses, our model shows an improvement over the PSMNet by 6.77% for 2D IoU and 0.0351 for $\delta^i$ without needing a (noisy) pose prior.

## 4.3. Panorama registration performance

In Table 2, we report the error metrics on the whole test dataset as well as subsets of different overlapping ratios. We can tell that GPR-Net outperforms OpenMVG with a significant margin across all metrics, while GPR-Net (Direct) achieves overall the best performance. The main problem of OpenMVG is that it has many failure cases where the algorithm returns no registration. In contrast, GPR-Net (Direct) always returns a valid pose, and thanks to learnt fine-grained correspondence on the layout boundary, GPR-Net (RANSAC) still return a high success rate even in the challenging category (Overlap-Low).

## 4.4. Ablation Study

**Ceiling and floor boundary maps.** In Section 3.3 and Section 3.4, we explain how to compute RANSAC-

Table 3. **Ablation study on panorama registration.** We evaluate how different (top block) design choices in ceiling/floor boundary maps and training strategies would impact pose estimation. The best results are in yellow highlight.

| Method | Success (% ↑) | Rotation | | | | | Translation angle | | | | | Translation vector | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Mn (° ↓) | Med (° ↓) | 2.5° ↑ | 5° ↑ | 10° ↑ | Mn (° ↓) | Med (° ↓) | 2.5° ↑ | 5° ↑ | 10° ↑ | Mn ($m.$ ↓) | Med ($m.$ ↓) | $0.5m.$ ↑ |
| Ceiling only | 99.94 | 4.4552 | 1.6138 | 0.6733 | 0.8791 | 0.9464 | 6.3809 | 1.8979 | 0.6026 | 0.8107 | 0.9051 | 0.3203 | 0.1172 | 0.8907 |
| Floor only | 99.94 | 4.4778 | 1.6146 | 0.6727 | 0.8783 | 0.9466 | 6.4596 | 1.9022 | 0.6027 | 0.8105 | 0.9051 | 0.3233 | 0.1176 | 0.8898 |
| Ceiling+Floor | 99.94 | 4.4760 | 1.6163 | 0.6727 | 0.8785 | 0.9457 | 6.4423 | 1.8998 | 0.6024 | 0.8102 | 0.9043 | 0.3234 | 0.1176 | 0.8901 |
| Joint training | 99.88 | 4.8402 | 1.7527 | 0.6461 | 0.8486 | 0.9344 | 6.9567 | 2.0494 | 0.5701 | 0.7882 | 0.8935 | 0.3263 | 0.1263 | 0.8729 |
| 2-stage training | 99.94 | 4.4552 | 1.6138 | 0.6733 | 0.8791 | 0.9464 | 6.3809 | 1.8979 | 0.6026 | 0.8107 | 0.9051 | 0.3203 | 0.1172 | 0.8907 |

Table 4. **Quantitative evaluation of panorama registration using the varying number of query tokens.** The best results are in yellow highlight.

| # Tokens | # Success (% ↑) | RANSAC pose: Rotation | | | | | Translation angle | | | | | Translation vector | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Mn (° ↓) | Med (° ↓) | 2.5° ↑ | 5° ↑ | 10° ↑ | Mn (° ↓) | Med (° ↓) | 2.5° ↑ | 5° ↑ | 10° ↑ | Mn ($m.$ ↓) | Med ($m.$ ↓) | $0.5m.$ ↑ |
| 256 | 99.94 | 4.4552 | 1.6138 | 0.6733 | 0.8791 | 0.9464 | 6.3809 | 1.8979 | 0.6026 | 0.8107 | 0.9051 | 0.3203 | 0.1172 | 0.8907 |
| 512 | 99.92 | 4.4565 | 1.6355 | 0.6747 | 0.8802 | 0.9491 | 6.3837 | 1.8960 | 0.6054 | 0.8117 | 0.9093 | 0.3297 | 0.1155 | 0.8935 |
| 1024 | 99.92 | 4.4065 | 1.6229 | 0.6743 | 0.8851 | 0.9498 | 6.4548 | 1.8680 | 0.6058 | 0.8111 | 0.9085 | 0.3398 | 0.1159 | 0.8939 |
| Direct pose | | | | | | | | | | | | | | |
| 256 | 100 | 2.0687 | 0.7670 | 0.9661 | 0.9822 | 0.9858 | 5.3703 | 2.3710 | 0.5238 | 0.7953 | 0.9296 | 0.2365 | 0.1468 | 0.9176 |
| 512 | 100 | 2.0970 | 0.7676 | 0.9670 | 0.9824 | 0.9850 | 5.2928 | 2.2418 | 0.5433 | 0.8057 | 0.9308 | 0.2332 | 0.1448 | 0.9175 |
| 1024 | 100 | 2.1102 | 0.7680 | 0.9667 | 0.9817 | 0.9845 | 5.2606 | 2.2021 | 0.5499 | 0.8086 | 0.9316 | 0.2319 | 0.1436 | 0.9186 |

Table 5. **Ablation study on layout reconstruction.** We evaluate how different design choices in ceiling/floor boundary maps would impact layout reconstruction. The best results are in yellow highlight.

| Method | w/ GT pose | | | w/o GT pose (Direct pose) | | |
|---|---|---|---|---|---|---|
| | 2D IoU↑ | $\delta^i$ ↑ | 3D IoU↑ | 2D IoU↑ | $\delta^i$ ↑ | 3D IoU↑ |
| Ceiling only | 0.8441 | 0.9604 | 0.8203 | 0.8249 | 0.9569 | 0.8018 |
| Floor only | 0.8449 | 0.9603 | 0.8211 | 0.8254 | 0.9568 | 0.8023 |
| Ceiling+Floor | 0.8402 | 0.9601 | 0.8166 | 0.8217 | 0.9562 | 0.7988 |

Table 6. **Quantitative evaluation of layout reconstruction using the varying number of query tokens.** The best results are in yellow highlight.

| # Tokens | Direct pose | | | RANSAC pose | | |
|---|---|---|---|---|---|---|
| | 2D IoU↑ | $\delta^i$ ↑ | 3D IoU↑ | 2D IoU↑ | $\delta^i$ ↑ | 3D IoU↑ |
| 256 | 0.8254 | 0.9568 | 0.8023 | 0.8198 | 0.9516 | 0.7969 |
| 512 | 0.8259 | 0.9571 | 0.8028 | 0.8207 | 0.9512 | 0.7978 |
| 1024 | 0.8262 | 0.9571 | 0.8030 | 0.8202 | 0.9513 | 0.7974 |

based pose and perform layout fusion using either the estimated ceiling boundary map or floor boundary map. Therefore, we conducted an experiment to evaluate the impact of using only ceiling boundary, only floor boundary, or the combination of two on the pose registration and layout reconstruction performance. As shown in Table 3 and Table 5, the setting of using only ceiling boundary map and using only floor boundary map leads respectively to a better accuracy in RANSAC-based pose estimation and layout reconstruction. Therefore, we use this setting in all the other experiments.

**Joint training vs. 2-stage training.** We train our geometry transformer first, freeze the model parameters, and then train our pose transformer. As for the joint training setting, we train the geometry transformer and pose transformer jointly. As shown in the lower part of Table 3, 2-stage training method could achieve a better registration accuracy. We believe this indicates that the fine-grained correspondences between boundaries provide much better supervision for the geometry transformer than the pose information (rotation and translation).

**The number of query tokens.** In this experiment, we start with the default setting of $N = 256$ query tokens along the $u$ coordinate. To increase $N$ without re-training the whole model, we rotate the input panoramas and re-test the network to add more query tokens progressively during the inference. As shown in Table 4, the registration accuracy gets better as we increasing the number of query tokens. We thus adopt this pose for the layout fusion part, and we get the best layout reconstruction accuracy when the number of query tokens is 1024 as shown in Table 6.

## 5. Conclusions

We present a first complete solution for room layout reconstruction from a pair of panorama images. In contrast to previous work, i.e. PSMNet, we do not rely on an approximate registration but can register the two panorama images directly. The major improvement over PSMNet comes from a novel Geometry-aware Panorama Registration Network (GPR-Net) that effectively tackles the wide baseline registration problem. We propose to exploit the layout geometry and compute fine-grained correspondences between the two layout boundaries, rather than directly computing the registration on global pixel-space. The main limitation of our method is that the layout fusion block that processes two layouts is very simple. We recommend the development of learned fusion modules as major avenue for future work.

## References


[1] 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. *International Conference on 3D Vision (3DV)*, 2017. 1, 3

[2] Kefan Chen, Noah Snavely, and Ameesh Makadia. Wide-baseline relative camera pose estimation with directional learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3258–3268, June 2021. 3

[3] Steve Cruz, Will Hutchcroft, Yuguang Li, Naji Khosravan, Ivaylo Boyadzhiev, and Sing Bing Kang. Zillow indoor dataset: Annotated floor plans with 360deg panoramas and 3d room layouts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2133–2143, June 2021. 1, 2, 4

[4] Ricardo Fabbri, Timothy Duff, Hongyi Fan, Margaret H. Regan, David da Costa de Pinho, Elias Tsigaridas, Charles W. Wampler, Jonathan D. Hauenstein, Peter J. Giblin, Benjamin Kimia, Anton Leykin, and Tomas Pajdla. Trplp - trifocal relative pose from lines at points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[5] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, USA, 2 edition, 2003. 3

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 3

[7] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1849–1856, Sept 2009. 2

[8] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, Oct 2007. 2

[9] Will Hutchcroft, Yuguang Li, Ivaylo Boyadzhiev, Zhiqiang Wan, Haiyan Wang, and Sing Bing Kang. Covispose: Co-visibility pose transformer for wide-baseline relative pose estimation in 360° indoor panoramas. In *ECCV*, 2022. 3, 6

[10] Shunping Ji, Zijie Qin, Jie Shan, and Meng Lu. Panoramic slam from a multiple fisheye camera rig. *ISPRS Journal of Photogrammetry and Remote Sensing*, 159:169–183, 2020. 3

[11] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. Cotr: Correspondence transformer for matching across images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6207–6217, October 2021. 5

[12] Zhigang Jiang, Zhongzheng Xiang, Jinhua Xu, and Ming Zhao. Lgt-net: Indoor panoramic room layout estimation with geometry-aware transformer network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1654–1663, June 2022. 2, 6, 7

[13] Chen-Yu Lee, Vijay Badrinarayanan, Tomasz Malisiewicz, and Andrew Rabinovich. Roomnet: End-to-end room layout estimation. *CoRR*, abs/1703.06241, 2017. 2

[14] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. Openmvg: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016. 1, 3

[15] A. Pagani and D. Stricker. Structure from motion using full spherical panoramic cameras. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 375–382, 2011. 3

[16] 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. 1, 2

[17] G. Pintore, F. Ganovelli, R. Pintus, R. Scopigno, and E. Gobbetti. 3d floor plan recovery from overlapping spherical images. *Computational Visual Media*, 4:367–383, 2018. 3

[18] Giovanni Pintore, Fabio Ganovelli, Alberto Jaspe Villanueva, and Enrico Gobbetti. Automatic modeling of cluttered multi-room floor plans from panoramic images. *Computer Graphics Forum*, 38(7):347–358, 2019. 3

[19] Srikumar Ramalingam and Matthew Brand. Lifting 3d manhattan lines from a single image. *2013 IEEE International Conference on Computer Vision*, pages 497–504, 2013. 2

[20] Y. Salaün, R. Marlet, and P. Monasse. Line-based robust sfm with little image overlap. In *2017 International Conference on 3D Vision (3DV)*, pages 195–204, 2017. 3

[21] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[22] Mohammad Amin Shabani, Weilian Song, Makoto Odamaki, Hirochika Fujiki, and Yasutaka Furukawa. Extreme structure from motion for indoor panoramas without visual overlaps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021. 3

[23] Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. 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 (CVPR)*, 2019. 1, 2

[24] Aparna Taneja, Luca Ballan, and Marc Pollefeys. Registration of spherical panoramic images with cadastral 3d models. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 479–486, 2012. 3

[25] 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 (CVPR)*, pages 12956–12965, June 2021. 4, 5, 6, 7

[26] 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)*, pages 8616–8625, June 2022. 1, 2, 3, 4, 6, 7

[27] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018. 1

[28] C. Yan, B. Shao, H. Zhao, R. Ning, Y. Zhang, and F. Xu. 3d room layout estimation from a single rgb image. *IEEE Transactions on Multimedia*, 22(11):3014–3024, 2020. 2

[29] 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 (CVPR)*, June 2019. 1, 2

[30] Yinda Zhang, Shuran Song, Ping Tan, and Jianxiong Xiao. Panocontext: A whole-room 3d context model for panoramic scene understanding. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI*, pages 668–686, 2014. 2

[31] 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. 1

[32] 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. 1, 2

[33] Chuhang Zou, Jheng-Wei Su, Chi-Han Peng, Alex Colburn, Qi Shan, Peter Wonka, Hung-Kuo Chu, and Derek Hoiem. Manhattan room layout reconstruction from a single 360° image: A comparative study of state-of-the-art methods. *IJCV*, Feb 2021. 2

[34] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion. *Acta Numerica*, 26:305–364, 2017. 3