

# Single-Shot Cuboids: Geodesics-based End-to-end Manhattan Aligned Layout Estimation from Spherical Panoramas

Nikolaos Zioulis

Centre for Research and Technology Hellas  
Universidad Politécnica de Madrid  
nzioulis@iti.gr

Federico Alvarez

Universidad Politécnica de Madrid  
fag@gatv.ssr.upm.es

Dimitrios Zarpalas

Centre for Research and Technology Hellas  
{zarpalas,daras}@iti.gr

## Abstract

*It has been shown that global scene understanding tasks like layout estimation can benefit from wider field of views, and specifically spherical panoramas. While much progress has been made recently, all previous approaches rely on intermediate representations and postprocessing to produce Manhattan-aligned estimates. In this work we show how to estimate full room layouts in a single-shot, eliminating the need for postprocessing. Our work is the first to directly infer Manhattan-aligned outputs. To achieve this, our data-driven model exploits direct coordinate regression and is supervised end-to-end. As a result, we can explicitly add quasi-Manhattan constraints, which set the necessary conditions for a homography-based Manhattan alignment module. Finally, we introduce the geodesic heatmaps and loss and a boundary-aware center of mass calculation that facilitate higher quality keypoint estimation in the spherical domain. Our models and code are publicly available at <https://vc13d.github.io/SingleShotCuboids/>.*

## 1. Introduction

Modern hardware advances have commoditized spherical cameras<sup>1</sup> which have evolved beyond elaborate optics and camera clusters. Affordable handheld 360° cameras are finding widespread use in various applications, with the more prominent ones being real-estate, interior design and virtual tours, with recently introduced datasets following the same trends. Realtor360 [60] contains panoramas acquired by a real-estate company, while Kujiale [28] and Structured3D [65] were rendered using a large corpus of

<sup>1</sup>We will be using the adjective terms spherical, omnidirectional and 360° for cameras and images interchangeably.

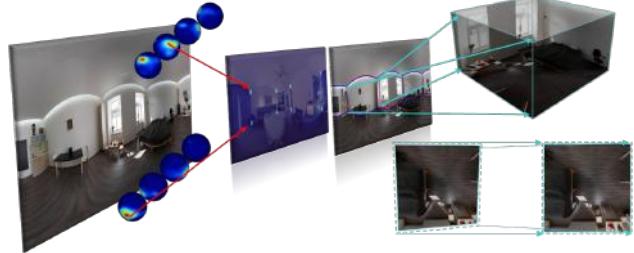


Figure 1: From a single indoor scene panorama input, we estimate a Manhattan aligned cuboid of the room’s layout, in a single-shot. To achieve this, we rely on spherical coordinate localization using geodesic heatmaps. This explicit reasoning about the corner positions in the image, allows for the integration of vertical alignment constraints that drive a differentiable homography-based cuboid fitting module.

computer-generated data from an interior design company. Further, datasets containing spherical panoramas like Matterport3D [3] and Stanford2D3D [1], were created using the Matterport camera, originally developed for virtual tours. This signifies the importance of spherical panoramas for indoor 3D capturing, as they are (re-)used in multiple 3D vision tasks [55, 50, 67].

Spherical panoramas capture the entire scene context within their field-of-view (FoV), an important trait for scene understanding. While humans can infer out of FoV information, the same cannot be said for machines, with view extrapolation methods [44] using spherical data to address this. Certain tasks like illumination or layout estimation implicitly extrapolate outside narrow FoVs. Neural Illumination [43] estimates a scene’s lighting from a single perspective image employing a perspective-to-spherical completion intermediate task within their end-to-end model. Estimating

a scene’s layout involves extrapolating structural information, and, thus, many works now resort to spherical panoramas to exploit their holistic structural and contextual information.

The seminal work of PanoContext [64], reconstructs an entire room into a 3D cuboid, fully exploiting the large FoV of omnidirectional panoramas . Its complex formulation and weak priors resulted in high computational complexity, requiring several minutes for each panorama. While modern deep priors produce higher quality results [68, 60], increasing the accuracy of their predictions and ensuring Manhattan-aligned layouts, requires postprocessing and hurts runtime efficiency.

Spherical panoramas necessitate higher resolution processing, and therefore, increased computational complexity, as evidenced by recent data-driven layout estimation models [68, 60, 47]. More efficient alternatives [15] produce irregular (*i.e.* non-Manhattan) outputs, require parameter sensitive postprocessing, and increase efficiency by lowering spatial resolution, which comes at the cost of accuracy. Moreover, data-driven spherical vision needs to address the distortion of the projective omnidirectional data formats. But distortion mitigating convolutions add a significant computational overhead as reported in [15] and [9].

In this work, we present a single-shot spherical layout estimation model. As presented in Figure 1, we employ spherical-aware corner coordinate estimation and thus, add explicit constraints that facilitate vertically aligned corners. Capitalizing on this, we further integrate full Manhattan alignment directly into the model, allowing for end-to-end training, lifting the postprocessing requirement.

## 2. Related Work

### 2.1. Layout Estimation

While an excellent review regarding the 3D reconstruction of structured indoor environments exists [40], our discussion will provide the necessary details for positioning our work. We focus on monocular layout estimation and thus, refrain from discussing works using multiple panoramas [39, 41, 37, 38], interaction [30], other types of cameras [27, 29].

PanoContext [64] showcased the expressiveness of 360° panoramas in terms of structural and contextual information. Prior to the maturation of deep data-driven methods, PanoContext relied on edge and line detection, Hough transform, and deformable part models to generate different room layout hypotheses. Similarly, low-level line segments were used in an energy minimization formulation to estimate a scene’s structural planes [17]. In Panoramix [59], the line features were supplemented by superpixel facets, and embedded as vertices in a graph for a constrained least squares problem.

Hybrid data-driven methods [16] used structural edge detection to improve the performance and runtime of [64] when using fewer hypotheses. Pano2CAD [58] used a probabilistic formulation that relied on CNN object recognition and detection. It generated a synthetic scene reconstruction but required several minutes of processing. Its computational overhead largely comes from the fusion of narrow FoV predictions from perspective 360° crops. This is common to all aforementioned methods relying on line segments and to [61], which runs various CNNs on all narrow FoV sub-views before merging them in 360°.

PanoRoom [14] and LayoutNet [68] were the first models to be trained on spherical panoramas. They both modelled layout corner and structural edge estimation as a spatial probabilistic inference task. While it is possible to extract the layout’s corners by relying on heuristically or empirically parameterized peak detection, these estimations will most likely not deliver Manhattan-aligned outputs. Consequently, joint optimization is performed using both sources of information to recover the final layout corner estimates. LayoutNet requires several seconds to infer and optimize the layout on a CPU, but PanoRoom is much faster as it uses a greedy RANSAC approach.

DuLa-Net [60] employs a novel approach for 360° layout estimation. The main insight is that spherical images can be projected in multiple ways, and different projections highlight different cues. Specifically, DuLa-Net uses a ‘ceiling-view’ that offers a more informative viewpoint with respect to the floor-plan, which is a projection of a Manhattan 3D layout. It performs feature fusion across both the equirectangular and ceiling-view branches, using a height prediction to estimate the final 3D layout. HorizonNet [47] is yet another novel take at omnidirectional layout estimation. Instead of image localised predictions, it encodes the boundaries and intersections in one-dimensional vectors, which are then used to reconstruct the scene’s corners. This allows HorizonNet to exploit the expressiveness of recurrent models (LSTM [22]) to offer globally coherent predictions. After a postprocessing step involving peak detection and height optimization, the final Manhattan-aligned layout is computed. A recent thorough comparison between LayoutNet, DuLa-Net and HorizonNet was presented in [69]. Unified encoding models and training scripts were used to fairly evaluate these approaches. Their findings indicate that the PanoStretch data augmentation proposed in [47], as well as its heavier encoder backbone lead to improved performance for the other models as well. The Corners-for-Layout (CFL) [15] model is currently the most efficient approach for 360° layout estimation in terms of runtime, but at the expense of accuracy and Manhattan alignment. While an end-to-end model is discussed, an empirically or heuristically parameterized postprocessing image peak detection step is still required.

Compared to these approaches, our model is end-to-end trainable, producing Manhattan aligned corners in a single-shot. We approach the layout estimation task as a keypoint localization one and use an efficiently designed spherical model.

## 2.2. Learning on the Sphere

There are multiple representations for spherical images with the more straightforward being the cube-map. Traditional CNN models can be applied to the cube faces [33], and then warped back to the sphere. This was used in [64] and [59] to detect lines on each cube’s faces [53], while [58] and [61] used CNN inference on each face. Still, cube-maps suffer from distortion as well, and additionally require face-specific padding [4] to deal with the faces’ discontinuities. Yet, to capture the global context these approaches need to expand their receptive field to connect all faces continuously, which leads to inefficient models.

A novel line of research pursues model adaptation from the perspective domain to the equirectangular one [45]. The follow-up work, Kernel Transformer Networks [46], adapt traditional kernels to the spherical domain in a learned manner, also discussing two important aspects. First, the accuracy-resolution trade-off for spherical images, which necessitates the user of higher resolutions. Indeed, most aforementioned data-driven layout estimation methods from 360° images operate on  $1024 \times 512$  images, which are unusually large for CNNs. Only [15] is the exception to this rule, which further supports this point, taking into account its reduced performance. The second point of discussion is related to the effect that non-linearities have, when combined with kernel projection methods like [6] and [51]. It is shown that the assumption that needs to hold for no error to accumulate when using kernel projection, only holds for the first layers of the network, and as it deepens, the accumulated error becomes even larger. Still, [15] shows that their EquiConv offer more robust predictions. A generalization of this concept, Mapped Convolutions [9], decouple the sampling operation from the filtering one, and demonstrate increased performance in dense estimation tasks. Still, runtime performance is greatly reduced as reported in both [15] and [9].

This is also the main drawback of frequency-based spherical convolutions as presented in the concurrent works of [5] and [11]. They are also highly inefficient in terms of memory, allowing for training and inference in very low resolution images only. DeepSphere [8] and [25] present another approach to handle distortion and discontinuity by leveraging graph convolutions and lifting the sphere representation to a graph. Nonetheless, this requires a graph generation step and loses efficacy compared to traditional convolutions, whose implementations are highly optimized to exploit the memory regularity of image representations.

The most efficient way to handle the discontinuity is circular padding [54, 47, 7], which is partly our approach as well, taking into account the inefficiency of distorted kernels. It should also be noted that model adaptation methods would not transfer well for the layout estimation task. While an object detection task parses a scene in a local manner, layout estimation requires to reason about the global context, with perspective methods typically needing to extrapolate the scene’s structure. However, as first proven by PanoContext [64], the availability of the entire scene is much more informative, and this would hinder the applicability of transferring models like RoomNet [27] to the 360° domain using such techniques [45, 46].

## 2.3. Coordinate Regression

Regressing coordinates in an image has been shown to be an intriguingly challenging problem [31]. The proposed solution was to offer the coordinate information explicitly. Yet, most keypoint estimation works in the literature initially used fully connected layers to regress coordinates. The counter-intuition is that convolutions are inherently spatial, and should be more well-behaved in spatial prediction tasks. This is how data-driven layout estimation models have addressed this problem up to now ([68], [15]), transforming coordinates into spatial configurations, using smoothing kernels to approximate coordinates, and leverage dense supervision. Keypoint localisation tasks with semantic inter-correlated structures, typically use one heatmap per keypoint. However, an issue that has recently received attention [62], is the way the final coordinate is estimated from each dense prediction. Indeed the spatial maxima might not always best approximate the coordinate, and thus, heuristic approaches have persisted. Specifically for layout estimation, where the corners are predicted on the same map, manually-set peak detection thresholds are used.

The overlapping works of [32], [48] and [35] derive smooth operations to reduce a heatmap to single a coordinate. Using the coordinate grid and a spatial *softmax* function, they smoothly, and differentiably, transform a spatial probabilistic representation into a single location. As shown in [52], all the above operations are treating pixels as particles with masses, and estimate their center of mass.

## 3. Single-Shot Cuboids

Unlike previous works, we approach layout estimation as a keypoint localisation task, alleviating the need for post-processing and simultaneously ensure Manhattan aligned outputs. Section 3.1 formulates our coordinate regression objective and its adaption to the spherical domain, Section 3.2 introduces the geodesic heatmaps and loss function and then, Section 3.3 provide insights into our model’s design, and the techniques to achieve end-to-end Manhattan alignment.

### 3.1. Spherical Center of Mass

The center of mass (CoM)  $\mathbf{c}_P$  for a collection of particles  $P : \{\mathbf{p}_0, \dots, \mathbf{p}_N\} \in \mathbb{R}^3$  is defined as:

$$\mathbf{c}_P = \frac{\sum_i^N m_i \mathbf{p}_i}{M}, \quad M = \sum_i^N m_i, \quad (1)$$

with  $m_i$  being the mass of particle  $\mathbf{p}_i$  and  $M$  the system's total mass. The CoM  $c_P$  represents a concentration of the particle system's mass and does not necessarily lie on an existing particle. This way, when considering a sparse keypoint estimation task in a structured grid, we can reformulate it as a dense prediction task by instead inferring the mass of each grid point. Using Eq. (1) we can directly supervise it with the keypoint coordinates, instead of relying on a surrogate objective as commonly done in pose estimation [62] or facial landmark detection [13].

For spherical layout estimation, the set of particles  $P$  for which we seek to individually estimate their per particle mass, lies on a sphere. Each layout corner is considered as the CoM of a distinct particle system defined on the sphere. Each particle  $\mathbf{p} = (\phi, \theta)$  on the sphere is represented by its longitude  $\phi$  and latitude  $\theta$ . While there are ways for learning directly on the 2-sphere  $S^2$  manifold, as explained in Section 2.2, they are very inefficient. Consequently, we consider the equirectangular projection of the sphere which preserves the angular parameterization of each particle. The equirectangular projection is an equidistant planar projection of the sphere, where the pixels in the image domain  $\Omega : (u, v) \in [0, W] \times [0, H]$  are linearly mapped to the angular domain<sup>2</sup>  $\mathcal{A} : (\phi, \theta) \in [0, 2\pi] \times [0, \pi]$ . Nevertheless, this format necessitates a different approach to overcome its weaknesses, namely, image boundary discontinuity, and planar projection distortion.

The discontinuity arises at the horizontal panorama boundary, where the particles, even though at the opposite sides of the image, are actually neighboring on the sphere. For traditional images, the (normalized) grid coordinates are typically defined in  $[0, 1]$  or  $[-1, 1]$ , and thus, the particles at the boundary would be maximally distant. However, for spherical panoramas, the longitudinal coordinate  $\phi$  is periodic and wraps around, with the particles at the boundaries being proximal (*i.e.* minimally distant). To address this, we split the CoM calculation for the longitude and latitude coordinates, and adapt the former to consider each point as lying on a circle. Therefore, for each panorama row, which represents a circle of (equal) latitude, we define new particles  $\mathbf{r} \in \mathcal{R}$  with

$$\mathbf{r}(\phi) = (\lambda, \tau) = (\cos \phi, \sin \phi), \quad (2)$$

while lie on a unit circle. We can then calculate the CoM

<sup>2</sup>We transition between these terms flexibly given their linear mapping.

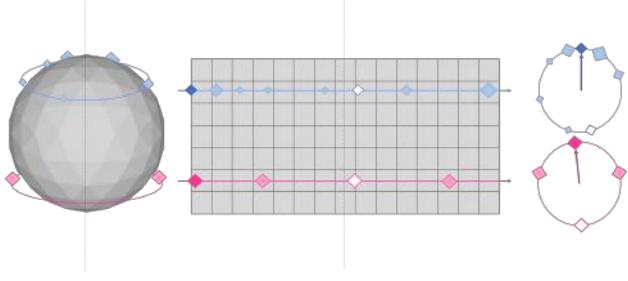


Figure 2: Spherical Center of Mass calculation. **Left:** Two sets of particles distributed on two circles of latitude (blue and pink). **Middle:** Their equirectangular projection grid coordinates. **Right:** Lifting the problem to the unit circle allows for continuous CoM estimation. Darker points illustrate the CoMs calculated using our lifting approach, and white ones the erroneous estimates when directly estimating CoM on the grid.

$\mathbf{c}_{\mathcal{R}}$ :

$$\mathbf{c}_{\mathcal{R}} = (\bar{\lambda}, \bar{\tau}) = \frac{\sum_i^N m_i \mathbf{r}_i}{M}. \quad (3)$$

This estimates, exactly and continuously, the CoM of the circle. To map this back to the original domain, we extract the angle  $\bar{\phi}$ :

$$\bar{\phi} = \text{atan2}(-\bar{\tau}, -\bar{\lambda}) + \pi, \quad (4)$$

which represents the longitudinal CoM across the discontinuity. Figure 2 shows a toy example of CoM calculations along two circles of latitude on the sphere, with the erroneous estimates acquired on the equirectangular projection and the correct ones when considering the boundary.

Although the equirectangular projection maps circles of latitude (longitude) to horizontal (vertical) lines of constant spacing, the same does not apply for its sampling density. Indeed, while it samples the sphere with a constant density vertically, it stretches each circle of latitude to fit the same constant horizontal line. Thus, its sphere sampling density is not uniform in all planar pixel locations. The sampling density is  $1/\sin \theta$  [49] and it approaches infinity near the pole singularities. When calculating the CoM in the equirectangular domain, we need to compensate for it by re-weighting the contribution of each pixel  $\mathbf{p}$  by  $\sigma(\mathbf{p}) = \sin \theta$  [66].

Essentially, given a dense mass prediction  $\mathbf{M}(\mathbf{p}), \mathbf{p} \in \mathcal{A}$ , we calculate the spherical CoM by first estimating a three-dimensional coordinate  $\mathbf{c}_a$ :

$$\mathbf{c}_a = (\bar{\lambda}, \bar{\tau}, \bar{\theta}) = \frac{\sum_{\mathbf{p}}^{\mathcal{A}} \mathbf{M}(\mathbf{p}) \sigma(\mathbf{p}) \mathbf{a}(\mathbf{p})}{\sum_{\mathbf{p}}^{\mathcal{A}} \mathbf{M}(\mathbf{p}) \sigma(\mathbf{p})}, \quad (5)$$

with  $\mathbf{a}(\mathbf{p}) = (\mathbf{r}(\phi), \theta) = (\cos \phi, \sin \phi, \theta)$ , and then drop it to the two-dimensions again to calculate the final CoM

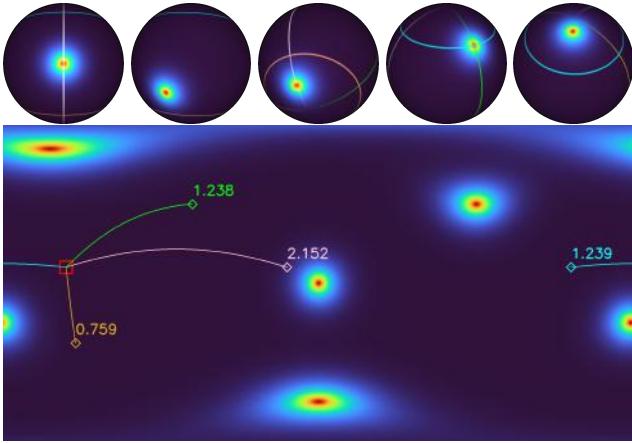


Figure 3: Geodesic heatmaps respect the horizontal boundary continuity and the equirectangular projection’s distortion. Five normal distributions on the sphere centered around different coordinates but using the same angular standard deviation are presented on the top row. Their corresponding geodesic heatmaps are aggregated on the equirectangular image on the bottom row. In addition, the geodesic distance between the red square and the colorized diamond coordinates are also presented on the same image. The geodesic distance similarly respects the boundary and distortion of the equirectangular projection as seen by the great circles drawn on the image that correspond to each pair’s angular distance.

$$\mathbf{c}_m = (\bar{\phi}, \bar{\theta}) = (\text{atan}2(-\bar{\tau}, -\bar{\lambda}) + \pi, \theta) \text{ of } \mathbf{M} \text{ in the equirectangular domain.}$$

### 3.2. Geodesic Heatmaps

Accordingly, predicting the sparse coordinates of a corner comes down to predicting the dense mass map  $\mathbf{M}$ , or otherwise heatmap, which is the terminology we will be using hereafter. Previous approaches complemented the sparse objective with a dense regularisation term [35]. The reason was that CoM regression is not constrained in any way as to the shape of its dense prediction. This was addressed by adding a distribution loss over the predicted heatmap and a Gaussian centered at the groundtruth coordinate.

Yet while extracting the CoM, as presented in Section 3.1, takes the spherical domain into account, traditional (flat) Gaussian heatmaps do not. A spatial normal distribution  $\mathcal{N}(\mathbf{c}, \mathbf{s})$  centered around a coordinate  $\mathbf{c} = (u, v)$ , using a standard deviation  $\mathbf{s} = (s_x, s_y)$  would consider the equirectangular image as a flat one, with a discontinuous boundary and no distortion.

To overcome this, we construct geodesic heatmaps, which are reconstructed directly on the equirectangular do-

main using a shifted angular coordinate grid  $\mathcal{A}_s$ <sup>3</sup> defined on the panorama:

$$\mathcal{G}(\mathbf{c}_m, \alpha) = \frac{1}{\alpha\sqrt{2\pi}} e^{-\frac{g(\mathbf{c}_m, \mathbf{p}_s)}{2\alpha^2}}, \mathbf{p}_s \in \mathcal{A}_s, \quad (6)$$

where  $\alpha$  is the angular standard deviation around the distribution’s center  $\mathbf{c}_m$ , and  $g(\cdot)$  is the geodesic distance:

$$g(\mathbf{p}_1, \mathbf{p}_2) = 2 \arcsin \sqrt{\sin^2 \frac{\Delta\theta}{2} + \cos \theta_1 \cos \theta_2 \sin^2 \frac{\Delta\phi}{2}}, \quad (7)$$

where  $\Delta\phi = \phi_1 - \phi_2$  and  $\Delta\theta = \theta_1 - \theta_2$ . As illustrated in Figure 3, using the geodesic distance between two angular coordinates on the equirectangular panorama, we reconstruct geodesic heatmaps that simultaneously take into account both the continuous boundary, as well as the projection’s distortion.

### 3.3. End-to-end Manhattan Model

Our model infers a set of heatmaps  $\mathbf{M}^j$ , one for each layout corner  $j \in [1, J]$  (or junction, given that 3 planes intersect), with  $J = 8$  for cuboid layouts. It operates in a single-shot manner, as these predictions are directly mapped into layout corners  $\mathbf{c}_m^j$ . Apart from removing the post-processing step, another advantage of our single-shot approach is the sub-pixel level accuracy that it allows for, as the CoM of the particles is not necessarily one of the particles themselves. This translates to a reduction of the input and working resolution of the model.

We choose a light-weight stacked hourglass (SH) architecture [34]. It is designed for multi-scale feature extraction and merging, that enables the effective capturing of spatial context. It suits spherical layout estimation very well as it is a global scene understanding task that benefits from spatial context aggregation, which is achieved by lowering the spatial dimension of the features. Still, it also requires precise localisation of specific keypoints, which needs higher spatial fidelity, (*i.e.* resolution) predictions.

#### 3.3.1 Stacked Hourglass Model Adaptation

We made several modifications to the original SH model stemming mainly from recent advances made in the field. While we preserve the original residual block [20] in the feature preprocessing block, we replace the hourglass residual blocks with preactivated ones [21]. Essentially, this adds direct identity mappings between the stack of hourglasses, allowing for immediate information propagation from the output to the earlier hourglass modules. We also use anti-aliased max-pooling [63], which preserves shift equivariance and leads to smoother activations across downsampled layers. Finally, unlike some state-of-the-art spherical layout estimation methods [68, 60, 69], we address feature map

<sup>3</sup> $\phi$  and  $\theta$  are shifted by  $-\pi$  and  $-\pi/2$  respectively.

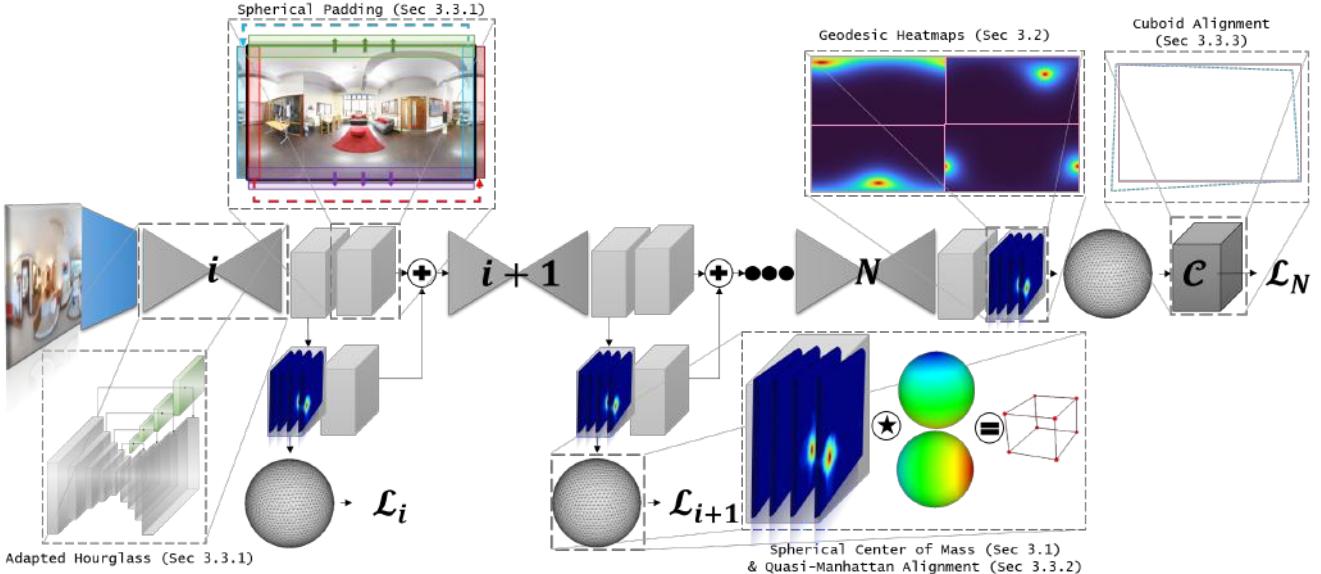


Figure 4: Our model stacks  $N$  hourglasses which embed recently developed CNN modules for direct inter-hourglass information flow, spherically padded convolutions, and smoother multi-scale feature flow. The predicted geodesic heatmaps get transformed directly to panoramic layout coordinates through a spherical CoM module. Since we regress coordinates, we explicitly enforce quasi-Manhattan alignment. This sets the ground for a homography-based cuboid alignment head that ensures the Manhattan alignment of our estimates. The  $\star$  symbol denotes a global multiply-accumulate operation, reducing the predicted dense representation to a set of sparse coordinates. Color-graded spheres indicate coordinate-based distance from the origin.

discontinuity by using spherical padding. For the horizontal image direction, we apply circular padding, as also done in [54] and [47], and for the vertical one at the pole singularities, we resort to replication padding.

### 3.3.2 Quasi-Manhattan Alignment

Since we are directly regressing coordinates, we can explicitly ensure quasi-Manhattan alignment during training and inference alike. Previous approaches either use post-processing to ensure the Manhattan alignment of their predictions [68, 60, 47], or simply forego it and produce non-Manhattan outputs [15]. While this relaxation is sometimes presented as an advantage, most man-made environments are Manhattan-aligned, with walls being orthogonal to ceiling and floors, and therefore, same edge wall corners are vertically aligned. For each wall-to-ceiling junction, there exists a wall-to-floor junction, effectively splitting our heatmaps in two groups, the *top*  $M_t^j$  and *bottom*  $M_b^j$  heatmaps (*i.e.* ceiling and floor junctions respectively). We enforce quasi-Manhattan alignment by averaging the longitudinal coordinates of each wall’s vertical edge, guaranteeing a consistent longitudinal coordinate for both the top and bottom junction.

### 3.3.3 Homography-based Full Manhattan Alignment

This quasi-Manhattan alignment ensures that wall edges are vertical to the floor, but does not enforce their orthogonality. To achieve this, we introduce a differentiable operation that transforms the predicted corners so as to ensure the orthogonality between adjacent walls. While the estimated corners are up-to-scale, with a single center-to-floor/ceiling measurement/assumption we can extract metric 3D coordinates for each corner as in [64]<sup>4</sup>, by fixing the ceiling/floor vertical distance to the corresponding average height.

We extract the  $\mathbf{f} = (\mathbf{x}, \mathbf{y})$  horizontal coordinates coordinates, corresponding to an orthographic floor view projection, which comprise a general trapezoid. This is transformed to a unit square by estimating the projective transformation  $\mathcal{H}$  (planar homography) mapping the former to the latter [18]. Using the trapezoid’s edge norms  $\|\mathbf{v}\|_2$ , with  $\mathbf{v} = \mathbf{f}^{j+1} - \mathbf{f}^j$ , we calculate the average opposite edge distances and use them to scale the unit square to a rectangle, after translating it for their centroids to align. Then, we rotate and translate the rectangle to align with the original trapezoid using orthogonal Procrustes analysis [42]. Finally, the rectangle gets lifted to a cuboid using the vertical ( $z$ ) ceiling and floor coordinates. The resulting cuboid

<sup>4</sup>See the supplementary material [56]

vertices can be transformed back to angular coordinates for loss computation, with the overall process presented in Figure 5. We use this cuboid alignment transform  $\mathcal{C}$  as the final block of our model to ensure full Manhattan alignment in an end-to-end manner.

We supervise the junction angular coordinates using the geodesic distance of Eq.(7):

$$\mathcal{L}_G = \frac{1}{J} \sum_j g(\mathbf{c}_m^j, \hat{\mathbf{c}}_m^j), \quad (8)$$

with  $\mathbf{c}_m^j$  and  $\hat{\mathbf{c}}_m^j$  being the groundtruth and predicted coordinates. The geodesic distance smoothly handles the continuous boundary and provides a more appropriate distance metric on the sphere, instead of the equirectangular projection. We additionally supervise the spatially normalized heatmaps  $\mathbf{H}^j = \text{spatial\_softmax}(\mathbf{M}^j)$  predicted by our model with Kullback Leibler divergence:

$$\mathcal{L}_D = \sum_{\mathcal{A}_f, j} \mathcal{D}_{KL}(\mathbf{H}^j, \tilde{\mathcal{G}}(\mathbf{c}_m^j)), \quad (9)$$

where  $\tilde{\mathcal{G}}(\cdot)$  is the spatially normalized geodesic heatmap  $\mathcal{G}(\cdot)$ . Apart from regularizing the predicted heatmaps, this loss allows for stable end-to-end training with the cuboid alignment transform, as pure coordinate supervision destabilized the model during early training, which prevented convergence as a consequence of the double solve required in the homography and Procrustes analysis. Our final loss is defined as:

$$\mathcal{L} = \sum_{n=1}^N \frac{\lambda_G}{N} \mathcal{L}_G^n + \frac{\lambda_D}{N} \mathcal{L}_D^n, \quad (10)$$

with  $\lambda_G$  and  $\lambda_D$  being weighting factors between the geodesic distance and KL loss, applied on each of the  $N$  hourglass predictions.

The higher level SH architecture allows for global processing without relying on heavy bottlenecks [68], computational expensive feature fusion [60] or recurrent models [47]. It also requires no post-processing as it can produce a Manhattan aligned layout in a single-shot with high accuracy albeit operating at lower than typical resolutions.

## 4. Results

### 4.1. Implementation Details

The input to our model is a single upright<sup>5</sup>, *i.e.* horizontal floor,  $512 \times 256$  spherical panorama. We use 128 features for each hourglass’s residual block, with a  $128 \times 64$  heatmap resolution, and initialize our SH model using [19]. We use the Adam [26] optimizer with a learning rate of 0.002 and

<sup>5</sup>Traditional [68, 23], or data-driven methods [24] can be used.

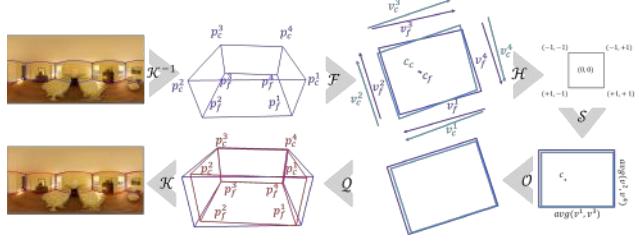


Figure 5: Starting from quasi-Manhattan corner estimates, these get first deprojected ( $\mathcal{K}^{-1}$ ) to 3D coordinates. Then, keeping only the horizontal coordinates ( $\mathcal{F}$ ), we get a floor view trapezoid, which depending on the measurement and coordinates (floor/ceiling) our projection operated on, is slightly different (cyan for the ceiling, and blue for the floor). Using these floor view horizontal coordinates, we estimate a homography  $\mathcal{H}$  to transform them to an axis aligned, unit square. This gets translated and scaled ( $\mathcal{S}$ ) using the average opposite edge lengths and centroid of the original untransformed floor view coordinates. An orthogonal Procrustes analysis ( $\mathcal{O}$ ) is used to align the rectangle to the trapezoid, which then gets lifted to a cuboid ( $\mathcal{Q}$ ) using the original heights, taking into account the quasi-Manhattan alignment of our estimates. The cuboid’s 3D coordinates then get projected ( $\mathcal{K}$ ) back to equirectangular domain corners. Apart from the ceiling and floor starting corners, we also consider a joint approach where the horizontal floor view coordinates get averaged from both 3D estimates, before proceeding to estimate the homography. For this approach to work, we rescale the ceiling coordinates so that their camera to floor distances align, therefore removing any scale difference from the camera’s position deviation from the true center.

default values for the other parameters, no weight decay, and a batch size of 8. Further, after an empirical greedy search, we use a fixed  $\alpha = 2^\circ$  and  $\mathbf{s} = (3.5, 3.5)$  for our Geodesic and Isotropic Gaussian distribution reconstructions respectively, which are created using the encoding of [62], and set the loss weights to  $\lambda_G = 1.0$  and  $\lambda_D = 0.15$ . For cuboid alignment we use the joint approach and use a floor distance of  $-1.6m$ . We implement our models using PyTorch [36, 12], setting the same seed for all random number generators. Further, each parameter update uses the gradients of 16 samples.

We apply heavy data augmentation during training, as established in prior work [69, 47, 15]. Apart from photometric augmentations (random brightness, contrast, and gamma [2]), following [15], we further apply random erasing, with a uniform random selection between 1 and 3 blocks erased per sample. We also probabilistically apply a set of 360° panorama specific augmentations in a cascaded manner: **i**) uniformly random horizontal rotations spanning the full an-



Figure 6: Qualitative results on the PanoContext (top) and Stanford2D3D (bottom) datasets. On each panorama, we overlay the reconstructed layout from the groundtruth **red** and predicted **blue** junctions. The next row showcases the overlaid aggregated heatmap predictions, with the following one illustrating the resulting 3D mesh. Finally, two orthographic floor views are presented, showing the full Manhattan (left), and quasi-Manhattan aligned (right) estimations.

gle range, **ii**) left-right flipping, and **iii**) PanoStretch augmentations [47] using the default stretching ratio ranges. All augmentation probabilities are set to 50%.

## 4.2. Datasets

Prior work up to now has experimented with small scale datasets. PanoContext [64] manually annotated a total of 547 panoramas from the Sun360 dataset [57] as cuboids. Additionally, LayoutNet manually annotated 552 panoramas from the Stanford2D3D dataset [1], which are not complete spherical images as their vertical FoV is narrower. Similar to previous works, we use the common train, test and validation splits as used in [15] and [68] for the PanoContext and Stanford2D3D datasets respectively. Taking into account their small scale, we jointly consider them as a single *real* dataset and train all our models for 150 epochs.

More recently, layout annotations have been provided in newer computer-generated datasets, the Kujiale dataset used in [28] and the Structured3D dataset [65], totaling 3550 and 21835 annotated images respectively. Albeit synthetic, they offer a much more expanded data corpus than what is currently available for real datasets. Given their synthetic nature, these datasets offer different room styles for the same scene. In particular, they provide empty rooms as well as rooms filled with furniture by interior designers. For the Kujiale dataset we use both types of scenes, while for Structured3D we only use full scenes and follow their respective official dataset splits. Our models are trained for 30 and 125 epochs respectively on Structured3D and Kujiale.

## 4.3. Metrics

For the quantitative assessment of our approach against prior works we use a set of standard metrics found in the literature [69], complemented by another set of accuracy metrics. The standard metrics include 2D and 3D intersection over union (IoU2D and IoU3D), normalized corner error (CE), pixel error (PE), and the depth-based RMSE and  $\delta_1$  accuracy [10]. For all 3D calculations a fixed floor distance at  $-1.6m$  is used. We also use junction ( $J_d$ ) and wireframe ( $W_d$ ) accuracy metrics, defined as correct when the closest groundtruth junction or line segment respectively is within a pixel threshold  $d$ . More specifically, we use the thresholds  $d = [5, 10, 15]$ . Finally, since we regress sub-pixel coordinates, all metric calculations are evaluated on a  $1024 \times 512$  panorama resolution, and the arrows next to each metric denote the direction of better performance.

## 4.4. Performance Analysis

First, we focus on the latest results reported in [69], where three data-driven cuboid panoramic layout estimation methods ([68, 60, 47]) were adapted for fairer compar-

ison. Similar to [69], we train a 3 stack (HG-3) single-shot cuboid (SSC) model using the real dataset. We present results tested on real (combined and single) datasets in Table 1 where our model compares favorably with the state-of-the-art<sup>6</sup>, offering robust performance and end-to-end Manhattan aligned estimates, a trait no other state-of-the-art method offers currently. For these results, we report the same metrics as those reported in [69]. Furthermore, Figure 6 presents a set of qualitative results for our HG-3 model on these two datasets.

With the recent availability of large scale synthetic datasets, we additionally train a model using Structured3D [65]. Since only HorizonNet offers a pretrained model using the same data, we present results on the Structured3D test dataset for two HorizonNet variants and our model in Table 2. Apart from the standard model that includes postprocessing, we also assess a single-shot variant of HorizonNet. For this, we only perform peak detection on the predicted wall-to-wall boundary vector and directly sample the heights at the detected peaks to reconstruct the layout. While this saves an amount of processing, the postprocessing scheme used by HorizonNet improves the results when applied to Structured3D’s test set. On the other hand, our model produces accurate layout corner estimates without any postprocessing. While SSC outperforms HorizonNet in the established metrics, HorizonNet offers higher accuracy in the junction and wireframe metrics. This is also the case for the cross-validation experiment that we present in Table 3. We test the models trained using Structured3D on the test set of Kujiale, using only the full rooms. The difference is this setting is that the single-shot variant of HorizonNet provides more accurate layout estimates than the postprocessed one. This exposes the weakness of postprocessing approaches, which require empiric or heuristic tuning. Nonetheless, this HorizonNet model is trained for general layout estimation, and the performance deviation might be related to this extra trait. Qualitative results for our end-to-end model for both synthetic datasets are presented in Figure 7.

## 4.5. Ablation Study

We perform an ablation study across all datasets. Tables 4, 2 and 5 present the results on the real and synthetic datasets<sup>7</sup>. Our baseline is the model as presented in Section 3.3 without the end-to-end Manhattan alignment homography module (Section 3.3.3), but with the quasi-Manhattan alignment (Section 3.3.2) offered by aligning the longitude of top and bottom corners. Apart from adding the end-to-end Manhattan alignment module, we also ablate the effect of the geodesic heatmap and loss (Section 3.2), the SH model adaptation (spherical padding, pre-

<sup>6</sup>Best three performances are denoted with bold red, orange and yellow.

<sup>7</sup>Our supplement offers results for each of the real datasets.



Figure 7: Qualitative results on the Structured3D (top) and Kujiale (bottom) datasets. Same scheme as Figure 6 applies.

Table 1: Quantitative results on the real domain datasets for each model variant.

Model			PanoContext			Stanford2D3D			Real (Combined)		
Name	Variant	Parameters ↓	CE ↓	IoU3D ↑	PE ↓	CE ↓	IoU3D ↑	PE ↓	CE ↓	IoU3D ↑	PE ↓
LayoutNet v2	ResNet-18	15.57M	0.65%	84.13%	1.92%	0.77%	83.53%	2.30%	0.71%	83.83%	2.11%
LayoutNet v2	ResNet-34	25.68M	<b>0.63%</b>	<b>85.02%</b>	1.79%	0.71%	84.17%	2.04%	0.67%	84.60%	<b>1.92%</b>
LayoutNet v2	ResNet-50	91.50M	0.75%	82.44%	2.22%	0.83%	82.66%	2.59%	0.79%	82.55%	2.41%
DuLa-Net v2	ResNet-18	25.64M	0.83%	82.43%	2.55%	0.74%	84.93%	2.56%	0.79%	83.68%	2.56%
DuLa-Net v2	ResNet-34	45.86M	0.82%	83.41%	2.54%	0.66%	<b>86.45%</b>	2.43%	0.74%	<b>84.93%</b>	2.49%
DuLa-Net v2	ResNet-50	57.38M	0.81%	83.77%	2.43%	0.67%	86.6%	2.48%	0.74%	85.19%	2.46%
HorizonNet	ResNet-18	23.49M	0.83%	80.27%	2.44%	0.82%	80.59%	2.72%	0.83%	80.43%	2.58%
HorizonNet	ResNet-34	33.59M	0.76%	81.30%	2.22%	0.78%	80.44%	2.65%	0.77%	80.87%	2.44%
HorizonNet	ResNet-50	81.57M	0.74%	82.63%	2.17%	0.69%	82.72%	2.27%	0.72%	82.68%	2.22%
SSC	HG-3	<b>6.35M</b>	<b>0.63%</b>	83.97%	<b>1.78%</b>	<b>0.51%</b>	<b>87.80%</b>	<b>1.62%</b>	<b>0.57%</b>	<b>85.89%</b>	<b>1.70%</b>

Table 2: Quantitative results and ablation on the synthetic Structured3D synthetic dataset.

Model	Variant	CE ↓	IoU2D ↑	IoU3D ↑	PE ↓	$J_5 \uparrow$	$J_{10} \uparrow$	$J_{15} \uparrow$	$W_5 \uparrow$	$W_{10} \uparrow$	$W_{15} \uparrow$	RMSE ↓	$\delta_1 \uparrow$
HNet	Single-Shot	0.57%	93.10%	91.17%	1.53%	78.20%	90.69%	95.09%	<b>56.67%</b>	77.50%	86.53%	0.0712	97.84%
	Postprocessed	0.75%	93.49%	91.82%	1.46%	<b>78.61%</b>	<b>91.64%</b>	95.75%	<b>56.67%</b>	<b>78.22%</b>	<b>87.57%</b>	0.0756	98.58%
SSC	Quasi-Manhattan w/ Homography (joint)	<b>0.39%</b>	93.97%	92.00%	1.25%	75.18%	90.96%	<b>95.82%</b>	49.27%	74.22%	86.29%	0.0667	<b>98.80%</b>
	w/o Geodesics	0.40%	<b>94.27%</b>	<b>92.33%</b>	1.26%	75.35%	90.90%	95.74%	48.16%	74.35%	85.68%	<b>0.0626</b>	98.76%
	w/o Model Adaptation	<b>0.39%</b>	93.94%	92.03%	1.25%	73.95%	90.14%	95.35%	49.26%	73.20%	84.98%	0.0671	98.71%
	w/o Quasi-Manhattan	0.45%	93.15%	91.04%	1.44%	71.10%	88.19%	94.35%	43.01%	69.59%	82.74%	0.0800	98.20%
HG-3	Quasi-Manhattan	<b>0.39%</b>	93.89%	92.00%	<b>1.23%</b>	73.43%	90.42%	95.63%	47.38%	72.63%	85.00%	<b>0.0651</b>	98.73%

Table 3: Cross-validation results on the Kujiale dataset using the Structured3D trained model.

Model	Variant	CE ↓	IoU2D ↑	IoU3D ↑	PE ↓	$J_5 \uparrow$	$J_{10} \uparrow$	$J_{15} \uparrow$	$W_5 \uparrow$	$W_{10} \uparrow$	$W_{15} \uparrow$	RMSE ↓	$\delta_1 \uparrow$
HNet	Single-Shot	0.61%	91.68%	89.53%	1.83%	<b>72.27%</b>	<b>87.91%</b>	94.09%	<b>46.27%</b>	<b>71.30%</b>	81.88%	0.0899	98.02%
	Postprocessed	1.04%	90.97%	88.96%	1.82%	71.18%	86.59%	92.95%	45.45%	69.67%	<b>81.39%</b>	0.0967	98.29%
SSC	Quasi-Manhattan w/ Homography (joint)	0.45%	92.83%	90.55%	1.46%	70.95%	86.86%	93.41%	41.55%	68.58%	81.88%	0.0811	98.40%
	w/o Geodesics	<b>0.42%</b>	<b>93.37%</b>	<b>91.21%</b>	<b>1.38%</b>	71.82%	87.36%	<b>94.86%</b>	44.12%	70.73%	<b>82.06%</b>	<b>0.0706</b>	<b>98.45%</b>

activated residual blocks and anti-aliased maxpooling - Section 3.3.1), and the quasi-Manhattan alignment itself by training a model with unrestricted, traditional (*i.e.* not spherical as presented in Section 3.1) CoM calculation for each corner.

These offer a number of insights. While the end-to-end model provides the more robust performance across all datasets, its performance is uncontested in the IoU and depth related metrics. However, on the remaining projective metrics, the unrestricted coordinate regression approaches usually perform better. This is reasonable as the homography fits a cuboid on the predictions, while the un-/semi-constrained approaches can freely localise the corners, even though at the expense of unnatural/Manhattan outputs, which manifests at an IoU3D drop. Overall, we observe that the additional of explicit Manhattan constraints (quasi and homography-based) offer increased performance compared to directly regressing the corners. The same applies to spherical (periodic CoM and geodesics) and model adaptation that consistently increase performance.

We also ablate the three approaches (floor/ceiling/joint) that use different starting coordinates for the homography estimation in Tables 4 and 5. We find that the joint approach

produces higher quality results, as it enforces both the top and bottom predictions to be consistent between them. This way, the cuboid misalignment errors are backpropagated to all corner estimates through the homography.

## 5. Conclusion

Our work has focused on keypoint estimation on the sphere and in particular on layout corner estimation. Through coordinate regression we integrate explicit constraints in our model. Moreover, while we have also shown that end-to-end single-shot layout estimation is possible, our approach is rigid as it is based on a frequent and logical assumption, that the underlying room is, or can be approximated by, a cuboid. Nonetheless, this rigidity comes from the structured predictions that CNN enforce, with the number of heatmaps that will be predicted being strictly defined at the design phase. Future work should try to address this limitation to fully exploit the potential that single-shot approaches offer, mainly stemming from end-to-end supervision. Finally, as with all prior layout estimation works, predictions are up to a scale, which hinders applicability. Even so, structured scene layout estimation is an important task that can even be used as an intermediate task to improve

Table 4: Ablation study on the real dataset.

Variant	CE ↓	IoU2D ↑	IoU3D ↑	PE ↓	$J_5 \uparrow$	$J_{10} \uparrow$	$J_{15} \uparrow$	$W_5 \uparrow$	$W_{10} \uparrow$	$W_{15} \uparrow$	RMSE ↓	$\delta_1 \uparrow$
Quasi-Manhattan	<b>0.55%</b>	87.90%	85.02%	1.74%	<b>61.54%</b>	<b>84.04%</b>	91.02%	<b>30.61%</b>	<b>57.97%</b>	<b>74.93%</b>	0.1734	96.14%
w/ Homography (joint)	0.57%	<b>88.39%</b>	85.89%	<b>1.70%</b>	55.01%	80.62%	<b>91.75%</b>	20.98%	52.63%	71.07%	<b>0.1557</b>	<b>97.93%</b>
w/ Homography (floor)	0.68%	88.25%	<b>85.97%</b>	1.91%	47.01%	76.66%	88.65%	16.30%	43.11%	63.24%	0.1591	97.52%
w/ Homography (ceil)	0.63%	87.63%	85.25%	1.88%	52.79%	<b>81.58%</b>	<b>91.43%</b>	18.50%	<b>51.64%</b>	69.73%	<b>0.1671</b>	<b>97.64%</b>
w/o Geodesics	0.79%	84.40%	81.08%	2.31%	33.78%	70.05%	86.80%	4.66%	26.33%	53.39%	0.2233	95.48%
w/o Model Adaptation	0.65%	86.60%	82.94%	1.98%	<b>55.41%</b>	78.16%	88.55%	<b>22.73%</b>	50.21%	66.49%	0.2033	95.61%
w/o Quasi-Manhattan	<b>0.61%</b>	87.24%	84.09%	<b>1.81%</b>	54.15%	80.21%	91.00%	16.26%	50.04%	<b>69.62%</b>	0.1874	96.31%

Table 5: Ablation study on the synthetic Kujiale dataset.

Variant	CE ↓	IoU2D ↑	IoU3D ↑	PE ↓	$J_5 \uparrow$	$J_{10} \uparrow$	$J_{15} \uparrow$	$W_5 \uparrow$	$W_{10} \uparrow$	$W_{15} \uparrow$	RMSE ↓	$\delta_1 \uparrow$
Quasi-Manhattan	<b>0.53%</b>	91.13%	88.43%	1.74%	<b>65.00%</b>	82.14%	90.50%	<b>37.27%</b>	<b>62.30%</b>	<b>74.91%</b>	0.0979	96.99%
w/ Homography (joint)	<b>0.53%</b>	<b>91.40%</b>	<b>89.01%</b>	<b>1.70%</b>	63.36%	<b>82.55%</b>	<b>90.55%</b>	35.91%	61.79%	74.24%	<b>0.0872</b>	97.09%
w/ Homography (floor)	0.57%	91.28%	88.72%	1.79%	62.09%	80.55%	89.68%	34.24%	58.97%	72.42%	0.0945	<b>97.44%</b>
w/ Homography (ceil)	0.56%	90.92%	<b>88.55%</b>	1.78%	61.68%	80.82%	89.59%	<b>35.55%</b>	60.42%	72.48%	0.0925	97.13%
w/o Geodesics	0.59%	90.81%	88.31%	1.81%	59.55%	79.36%	89.64%	27.64%	56.39%	71.24%	0.0998	96.92%
w/o Model Adaptation	0.59%	90.42%	87.52%	1.82%	61.36%	79.14%	88.68%	29.61%	57.27%	70.82%	0.1026	96.65%
w/o Quasi-Manhattan	<b>0.54%</b>	90.92%	88.42%	1.73%	<b>62.59%</b>	80.91%	<b>90.36%</b>	33.03%	59.33%	<b>73.36%</b>	0.0962	97.07%

other tasks, as shown in [28]. With metric scale inference, it has the potential for significant interplay with other 3D vision tasks like depth or surface estimation.

## Supplement

Supplementary material including additional ablation experiments and qualitative results are appended after the references.

## Acknowledgements

This work was supported by the EC funded H2020 project ATLANTIS [GA 951900].

## References

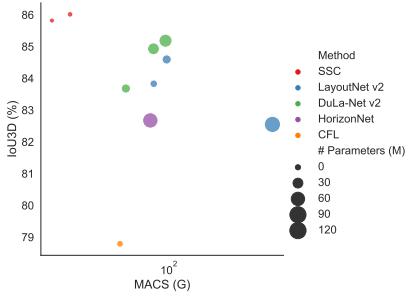
- [1] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017.
- [2] Alexander Buslaev, Vladimir I Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A Kalinin. Albumentations: fast and flexible image augmentations. *Information*, 11(2):125, 2020.
- [3] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgbd data in indoor environments. In *7th IEEE International Conference on 3D Vision (3DV)*, pages 667–676. Institute of Electrical and Electronics Engineers Inc., 2018.
- [4] Hsien-Tzu Cheng, Chun-Hung Chao, Jin-Dong Dong, Hao-Kai Wen, Tyng-Luh Liu, and Min Sun. Cube padding for weakly-supervised saliency prediction in 360 videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1420–1429, 2018.
- [5] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. In *International Conference on Learning Representations*, 2018.
- [6] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–533, 2018.
- [7] Thiago LT da Silveira and Claudio R Jung. Dense 3d scene reconstruction from multiple spherical images for 3-dof+ vr applications. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 9–18. IEEE, 2019.
- [8] Michaël Defferrard, Nathanaël Perraudin, Tomasz Kacprzak, and Raphael Sgier. Deepsphere: towards an equivariant graph-based spherical cnn. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.
- [9] Marc Eder, True Price, Thanh Vu, Akash Bapat, and Jan-Michael Frahm. Mapped convolutions. *arXiv preprint arXiv:1906.11096*, 2019.
- [10] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.
- [11] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning so(3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
- [12] WA Falcon. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 2019.
- [13] Zhen-Hua Feng, Josef Kittler, Muhammad Awais, Patrik Huber, and Xiao-Jun Wu. Wing loss for robust facial landmark localisation with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2235–2245, 2018.
- [14] Clara Fernandez-Labrador, José Falcí, Alejandro Perez-Yus, Cédric Demonceaux, and Jose Guerrero. Panoroom: From the sphere to the 3d layout. In *ECCV 2018 Workshops*, 2018.
- [15] Clara Fernandez-Labrador, Jose M Facil, Alejandro Perez-Yus, Cédric Demonceaux, Javier Civera, and Jose J Guer-

- rero. Corners for layout: End-to-end layout recovery from 360 images. *IEEE Robotics and Automation Letters*, 5(2):1255–1262, 2020.
- [16] Clara Fernandez-Labrador, Alejandro Perez-Yus, Gonzalo Lopez-Nicolas, and Jose J Guerrero. Layouts from panoramic images with geometry and deep learning. *IEEE Robotics and Automation Letters*, 3(4):3153–3160, 2018.
- [17] Kosuke Fukano, Yoshihiko Mochizuki, Satoshi Iizuka, Edgar Simo-Serra, Akihiro Sugimoto, and Hiroshi Ishikawa. Room reconstruction from a single spherical image by higher-order energy minimization. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 1768–1773. IEEE, 2016.
- [18] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.
- [22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [23] Jinwoong Jung, Beomseok Kim, Joon-Young Lee, Byung-moon Kim, and Seungyong Lee. Robust upright adjustment of 360 spherical panoramas. *The Visual Computer*, 33(6-8):737–747, 2017.
- [24] Raehyuk Jung, Aiden Seuna Joon Lee, Amirsaman Ashtari, and Jean-Charles Bazin. Deep360up: A deep learning-based approach for automatic vr image upright adjustment. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1–8. IEEE, 2019.
- [25] Renata Khasanova and Pascal Frossard. Graph-based classification of omnidirectional images. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 869–878, 2017.
- [26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [27] Chen-Yu Lee, Vijay Badrinarayanan, Tomasz Malisiewicz, and Andrew Rabinovich. Roomnet: End-to-end room layout estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4865–4874, 2017.
- [28] Jin Lei, Xu Yanyu, Zheng Jia, Zhang Junfei, Tang Rui, Xu Shugong, Yu Jingyi, and Gao Shenghua. Geometric structure based and regularized depth estimation from 360° indoor imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [29] Mingyang Li, Yi Zhou, Ming Meng, Yuehua Wang, and Zhong Zhou. 3d room reconstruction from a single fisheye image. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.
- [30] Niantao Liu, Bingxian Lin, Linwang Yuan, Guonian Lv, Zhaoyuan Yu, and Liangchen Zhou. An interactive indoor 3d reconstruction method based on conformal geometry algebra. *Advances in Applied Clifford Algebras*, 28(4):73, 2018.
- [31] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Advances in Neural Information Processing Systems*, pages 9605–9616, 2018.
- [32] Diogo C Luvizon, Hedi Tabia, and David Picard. Human pose regression by combining indirect part detection and contextual information. *Computers & Graphics*, 85:15–22, 2019.
- [33] Rafael Monroy, Sebastian Lutz, Tejo Chalasani, and Aljosa Smolic. Salnet360: Saliency maps for omni-directional images with cnn. *Signal Processing: Image Communication*, 69:26–34, 2018.
- [34] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.
- [35] Aiden Nibali, Zhen He, Stuart Morgan, and Luke Prendergast. Numerical coordinate regression with convolutional neural networks. *arXiv preprint arXiv:1801.07372*, 2018.
- [36] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
- [37] Giovanni Pintore, Fabio Ganovelli, Ruggero Pintus, Roberto Scopigno, and Enrico Gobbetti. 3d floor plan recovery from overlapping spherical images. *Computational Visual Media*, 4(4):367–383, 2018.
- [38] Giovanni Pintore, Fabio Ganovelli, Alberto Jaspe Vilanueva, and Enrico Gobbetti. Automatic modeling of cluttered multi-room floor plans from panoramic images. In *Computer Graphics Forum*, volume 38, pages 347–358. Wiley Online Library, 2019.
- [39] Giovanni Pintore, Valeria Garro, Fabio Ganovelli, Enrico Gobbetti, and Marco Agus. Omnidirectional image capture on mobile devices for fast automatic generation of 2.5 d indoor maps. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.
- [40] Giovanni Pintore, Claudio Mura, Fabio Ganovelli, Lizeth Fuentes-Perez, Renato Pajarola, and Enrico Gobbetti. State-of-the-art in automatic 3d reconstruction of structured indoor environments. *STAR*, 39(2), 2020.
- [41] Giovanni Pintore, Ruggero Pintus, Fabio Ganovelli, Roberto Scopigno, and Enrico Gobbetti. Recovering 3d existing-conditions of indoor structures from spherical images. *Computers & Graphics*, 77:16–29, 2018.
- [42] Peter H Schönenmann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.
- [43] Shuran Song and Thomas Funkhouser. Neural illumination: Lighting prediction for indoor environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6918–6926, 2019.

- [44] Shuran Song, Andy Zeng, Angel X Chang, Manolis Savva, Silvio Savarese, and Thomas Funkhouser. Im2pano3d: Extrapolating 360 structure and semantics beyond the field of view. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3847–3856, 2018.
- [45] Yu-Chuan Su and Kristen Grauman. Learning spherical convolution for fast features from 360 imagery. In *Advances in Neural Information Processing Systems*, pages 529–539, 2017.
- [46] Yu-Chuan Su and Kristen Grauman. Kernel transformer networks for compact spherical convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9442–9451, 2019.
- [47] 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 Conference on Computer Vision and Pattern Recognition*, pages 1047–1056, 2019.
- [48] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 529–545, 2018.
- [49] Yule Sun, Ang Lu, and Lu Yu. Weighted-to-spherically-uniform quality evaluation for omnidirectional video. *IEEE signal processing letters*, 24(9):1408–1412, 2017.
- [50] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7199–7209, 2018.
- [51] 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)*, pages 707–722, 2018.
- [52] C. Tensmeyer and T. Martinez. Robust keypoint detection. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 5, pages 1–7, 2019.
- [53] Rafael Grompone Von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd: a line segment detector. *Image Processing On Line*, 2:35–55, 2012.
- [54] Tsun-Hsuan Wang, Hung-Jui Huang, Juan-Ting Lin, Chan-Wei Hu, Kuo-Hao Zeng, and Min Sun. Omnidirectional cnn for visual place recognition and navigation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2341–2348. IEEE, 2018.
- [55] 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.
- [56] Jianxiong Xiao. 3d geometry for panorama, 2012.
- [57] Jianxiong Xiao, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Recognizing scene viewpoint using panoramic place representation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2695–2702. IEEE, 2012.
- [58] Jiu Xu, Björn Stenger, Tommi Kerola, and Tony Tung. Pano2cad: Room layout from a single panorama image. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 354–362. IEEE, 2017.
- [59] Hao Yang and Hui Zhang. Efficient 3d room shape recovery from a single panorama. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5422–5430, 2016.
- [60] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3363–3372, 2019.
- [61] Yang Yang, Shi Jin, Ruiyang Liu, Sing Bing Kang, and Jingyi Yu. Automatic 3d indoor scene modeling from single panorama. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3926–3934, 2018.
- [62] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7093–7102, 2020.
- [63] Richard Zhang. Making convolutional networks shift-invariant again. In *International Conference on Machine Learning*, pages 7324–7334, 2019.
- [64] Yinda Zhang, Shuran Song, Ping Tan, and Jianxiong Xiao. Panocontext: A whole-room 3d context model for panoramic scene understanding. In *European conference on computer vision*, pages 668–686. Springer, 2014.
- [65] 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.
- [66] 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.
- [67] 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.
- [68] 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.
- [69] Chuhang Zou, Jheng-Wei Su, Chi-Han Peng, Alex Colburn, Qi Shan, Peter Wonka, Hung-Kuo Chu, and Derek Hoiem. 3d manhattan room layout reconstruction from a single 360 image. *arXiv preprint arXiv:1910.04099*, 2019.

## A. Supplementary Material

In this supplementary material we present additional information regarding runtime and floating point operations, with the data offered in Table 6, and illustrated in Figure 8. Apart from the models presented in the main document, we also add efficient CFL models for completeness. In addition, we provide evaluation results for the Stanford2D3D and PanoContext datasets separately, in Tables 7 and 8 respectively. Further, in Tables 9, 10, and 11, we offer a decomposed model ablation for the Stanford2D3D, the PanoContext, and both datasets (averaged) respectively, where each individual component is ablated (namely, pre-activated bottlenecks, spherical padding, and anti-aliased max pooling). The pre-activated residual blocks offer the larger gains, followed by the padding and finally, the anti-aliased max pooling. Nonetheless, each different component is contributing to increased performance, with their combined effect being the most significant as observed by the model without all of these components together. Figures 9, 10, 11 and 12 present additional qualitative results of our single-shot, end-to-end Manhattan aligned layout estimation model using the joint homography head module in Stanford2D3D, PanoContext, Structured3D and Kujiale datasets respectively. Finally, Figures 13 and 14 present the qualitative samples from the real and synthetic datasets respectively, which are included in the main manuscript in animated 3D views (can only be viewed in recent Adobe Acrobat Reader versions).



**Figure 8: Model Size vs Accuracy vs Complexity.** Visual comparison of spherical layout estimation models in terms of parameters (denoted by each bullet’s size), computational complexity ( $x$  axis, in log scale, billions of multiply-accumulate operations) and accuracy ( $y$  axis, average IoU3D accuracy). Our model (SSC) is the most lightweight and offers a good comprise between complexity and accuracy, surpassing most other approaches. It also provides an end-to-end layout prediction in a single-shot, compared to all other approaches that require postprocessing. Different variants of each model are depicted. The exact data of this plot can be found in Table 6.

Table 6: This table presents model complexity measures (multiply-accumulate giga-operations per inference, millions of parameter counts, runtime performance) as well as accuracy (IoU3D) on real domain datasets. This table's reported values are used to generate Figure 8.

Method	Variant	MACS	Parameters	CPU	GPU	IoU3D
SSC	HG-3	<b>17.61G</b>	<b>6.35M</b>	1.78s	0.085s	<b>85.89%</b>
LayoutNet v2	ResNet18	76.12G	15.57M	11.65s	0.034s	83.83%
LayoutNet v2	ResNet34	95.48G	25.68M	12.97s	0.044s	84.60%
LayoutNet v2	ResNet50	607.43G	91.50M	34.63s	0.130s	82.55%
DuLa-Net v2	ResNet18	46.76G	25.64M	4.99s	0.037s	83.68%
DuLa-Net v2	ResNet34	75.79G	45.86M	6.46s	0.049s	<b>84.93%</b>
DuLa-Net v2	ResNet50	93.53G	57.38M	7.22s	0.072s	<b>85.19%</b>
HorizonNet	ResNet18	<b>23.03G</b>	23.49M	N/As	N/As	80.43%
HorizonNet	ResNet34	42.38G	33.59M	N/As	N/As	80.87%
HorizonNet	ResNet50	71.70G	81.57M	3.21s	0.063s	82.68%
CFL	EfficientNet	<b>42.19G</b>	<b>11.69M</b>	<b>0.074s</b>	<b>0.028s</b>	N/A%
CFL	ResNet50	N/A	N/A	0.420s	0.052s	78.79%

Table 7: Ablation results on the Stanford2D3D dataset.

Variant	CE ↓	IoU2D ↑	IoU3D ↑	PE ↓	$J_5$ ↑	$J_{10}$ ↑	$J_{15}$ ↑	$W_5$ ↑	$W_{10}$ ↑	$W_{15}$ ↑	RMSE ↓	$\delta_1$ ↑
Quasi-Manhattan	0.56%	88.18%	85.16%	1.83%	<b>64.82%</b>	83.41%	89.82%	<b>33.55%</b>	<b>62.32%</b>	75.81%	0.1787	95.59%
w/ Homography (joint)	<b>0.51%</b>	<b>89.83%</b>	<b>87.80%</b>	<b>1.62%</b>	64.27%	<b>85.07%</b>	<b>92.70%</b>	27.65%	62.02%	<b>77.36%</b>	<b>0.1402</b>	<b>98.28%</b>
w/ Homography (floor)	0.59%	89.51%	87.56%	1.80%	54.87%	81.86%	90.27%	23.01%	52.73%	70.35%	0.1474	97.95%
w/ Homography (ceil)	0.58%	89.04%	87.04%	1.82%	60.29%	82.74%	91.81%	27.88%	57.52%	74.04%	0.1539	97.36%
w/o Geodesics	0.80%	84.72%	81.73%	2.34%	32.19%	67.70%	87.28%	4.13%	25.15%	49.71%	0.2213	95.31%
w/o Model Adaptation	0.62%	87.77%	84.47%	1.85%	59.40%	79.20%	89.60%	25.96%	54.35%	69.76%	0.1815	97.10%
w/o Quasi-Manhattan	0.60%	87.50%	84.72%	1.86%	58.30%	79.76%	<b>90.49%</b>	19.32%	54.79%	71.17%	0.1825	95.97%

Table 8: Ablation results on the PanoContext dataset.

Variant	CE ↓	IoU2D ↑	IoU3D ↑	PE ↓	$J_5$ ↑	$J_{10}$ ↑	$J_{15}$ ↑	$W_5$ ↑	$W_{10}$ ↑	$W_{15}$ ↑	RMSE ↓	$\delta_1$ ↑
Quasi-Manhattan	<b>0.53%</b>	<b>87.63%</b>	<b>84.89%</b>	<b>1.65%</b>	<b>58.25%</b>	<b>84.67%</b>	<b>92.22%</b>	<b>27.67%</b>	<b>53.62%</b>	<b>74.06%</b>	<b>0.1682</b>	96.68%
w/ Homography (joint)	0.63%	86.95%	83.97%	1.78%	45.75%	76.18%	90.80%	14.31%	43.24%	64.78%	0.1711	97.58%
w/ Homography (floor)	0.76%	<b>87.00%</b>	84.39%	2.02%	39.15%	71.46%	87.03%	9.59%	33.49%	56.13%	<b>0.1708</b>	97.09%
w/ Homography (ceil)	0.68%	86.22%	83.47%	1.93%	45.28%	<b>80.42%</b>	<b>91.04%</b>	9.12%	<b>45.75%</b>	65.41%	0.1803	<b>97.92%</b>
w/o Geodesics	0.78%	84.08%	80.43%	2.27%	35.38%	72.41%	86.32%	5.19%	27.52%	57.08%	0.2252	95.64%
w/o Model Adaptation	0.68%	85.42%	81.41%	2.11%	<b>51.42%</b>	77.12%	87.50%	19.50%	<b>46.07%</b>	63.21%	0.2250	94.13%
w/o Quasi-Manhattan	0.61%	<b>86.99%</b>	83.46%	<b>1.77%</b>	50.00%	80.66%	91.51%	13.21%	45.28%	<b>68.08%</b>	0.1922	96.66%

Table 9: Model ablation results on the Stanford2D3D dataset.

Variant	CE ↓	IoU2D ↑	IoU3D ↑	PE ↓	$J_5$ ↑	$J_{10}$ ↑	$J_{15}$ ↑	$W_5$ ↑	$W_{10}$ ↑	$W_{15}$ ↑	RMSE ↓	$\delta_1$ ↑
Quasi-Manhattan	<b>0.56%</b>	<b>88.18%</b>	<b>85.16%</b>	1.83%	<b>64.82%</b>	83.41%	89.82%	<b>33.55%</b>	<b>62.32%</b>	<b>75.81%</b>	0.1787	95.59%
w/o Model Adaptation	0.62%	87.77%	84.47%	1.85%	59.40%	79.20%	89.60%	25.96%	54.35%	69.76%	0.1815	97.10%
w/o Pre-activated	0.58%	87.93%	85.09%	1.86%	63.94%	83.08%	89.82%	28.24%	61.73%	74.34%	0.1748	96.05%
w/o Padding	<b>0.56%</b>	88.10%	85.07%	<b>1.73%</b>	<b>64.82%</b>	<b>83.85%</b>	<b>90.71%</b>	30.60%	62.02%	75.22%	0.1788	96.61%
w/o Anti-aliasing	<b>0.56%</b>	<b>88.06%</b>	85.15%	1.76%	62.50%	82.52%	90.60%	28.32%	60.03%	75.29%	<b>0.1721</b>	<b>97.11%</b>

Table 10: Model ablation results on the PanoContext dataset.

Variant	CE ↓	IoU2D ↑	IoU3D ↑	PE ↓	$J_5$ ↑	$J_{10}$ ↑	$J_{15}$ ↑	$W_5$ ↑	$W_{10}$ ↑	$W_{15}$ ↑	RMSE ↓	$\delta_1$ ↑
Quasi-Manhattan	<b>0.53%</b>	<b>87.63%</b>	<b>84.89%</b>	<b>1.65%</b>	<b>58.25%</b>	<b>84.67%</b>	<b>92.22%</b>	<b>27.67%</b>	<b>53.62%</b>	<b>74.06%</b>	<b>0.1682</b>	96.68%
w/o Model Adaptation	0.68%	85.42%	81.41%	2.11%	51.42%	77.12%	87.50%	19.50%	46.07%	63.21%	0.2250	94.13%
w/o Pre-activated	0.62%	87.48%	83.68%	1.81%	55.90%	79.48%	88.92%	20.60%	51.10%	68.40%	0.1793	95.85%
w/o Padding	0.61%	85.96%	82.84%	1.96%	56.13%	82.31%	87.97%	24.69%	50.94%	70.28%	0.2043	95.18%
w/o Anti-aliasing	0.55%	87.48%	84.41%	1.76%	55.42%	82.78%	91.98%	27.52%	51.10%	69.50%	0.1745	<b>96.72%</b>

Table 11: Average model ablation results on both the real datasets.

Variant	CE ↓	IoU2D ↑	IoU3D ↑	PE ↓	$J_5$ ↑	$J_{10}$ ↑	$J_{15}$ ↑	$W_5$ ↑	$W_{10}$ ↑	$W_{15}$ ↑	RMSE ↓	$\delta_1$ ↑
Quasi-Manhattan	<b>0.55%</b>	<b>87.90%</b>	<b>85.02%</b>	<b>1.74%</b>	<b>61.54%</b>	<b>84.04%</b>	91.02%	<b>30.61%</b>	<b>57.97%</b>	<b>74.93%</b>	0.1734	96.14%
w/o Model Adaptation	0.65%	86.60%	82.94%	1.98%	55.41%	78.16%	88.55%	22.73%	50.21%	66.49%	0.2033	95.61%
w/o Pre-activated	0.60%	87.70%	84.39%	1.84%	59.92%	81.28%	89.37%	24.42%	56.41%	71.37%	0.1771	95.95%
w/o Padding	0.59%	87.03%	83.95%	1.84%	60.48%	83.08%	89.34%	27.65%	56.48%	72.75%	0.1916	95.90%
w/o Anti-aliasing	0.56%	87.77%	84.78%	1.76%	58.96%	82.65%	<b>91.29%</b>	27.92%	55.57%	72.40%	<b>0.1733</b>	<b>96.92%</b>



Figure 9: Additional qualitative results on the Stanford2D3D dataset.



Figure 10: Additional qualitative results on the PanoContext dataset.

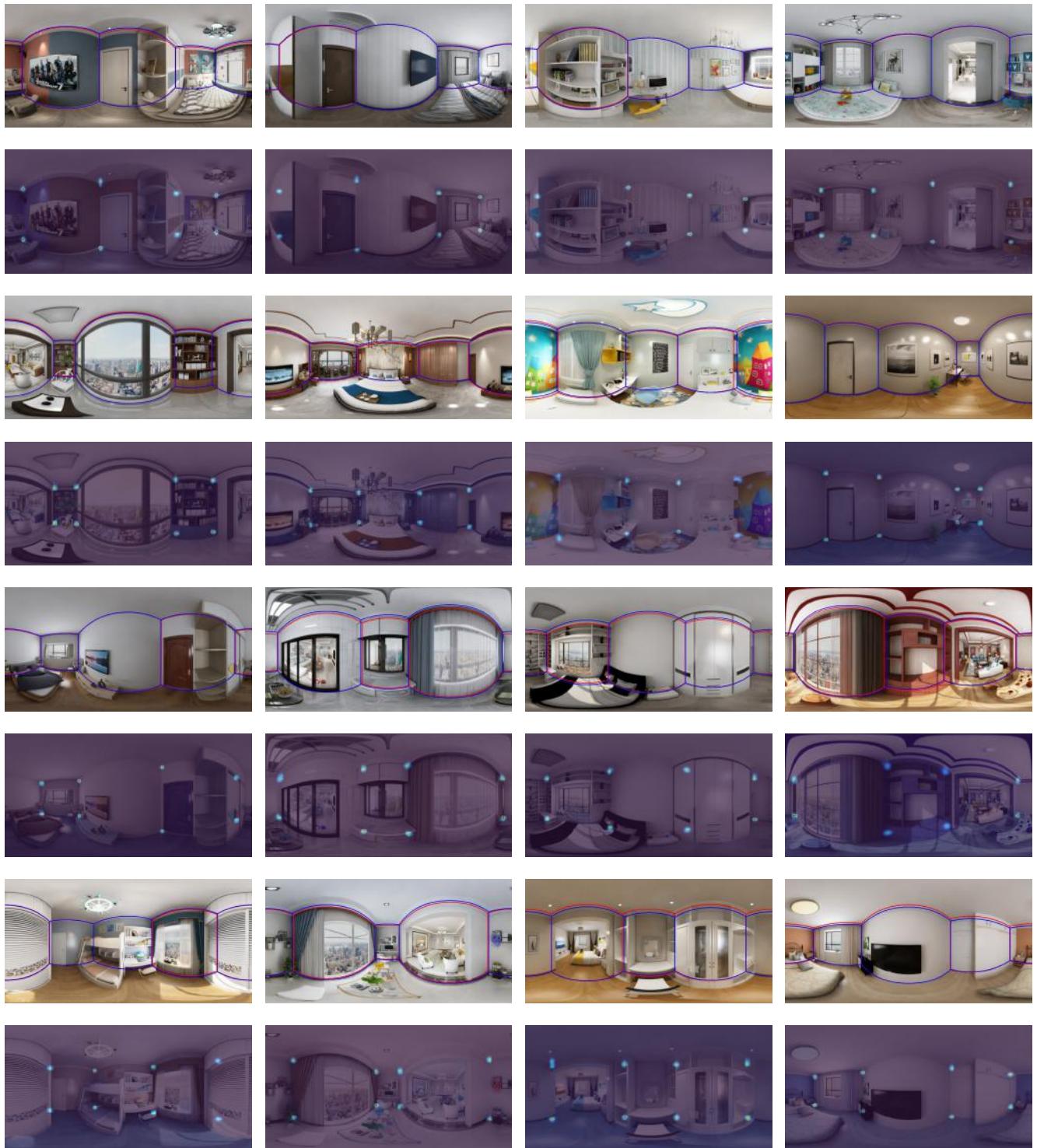


Figure 11: Additional qualitative results on the Structured3D dataset.



Figure 12: Additional qualitative results on the Kujiale dataset.

Figure 13: Animated renderings of the 3D qualitative results of the real datasets as presented in the figures of the main manuscript. Top row samples are from PanoContext, bottom row samples are from Stanford2D3D. (animations are only playable in recent Adobe Acrobat Reader versions).

Figure 14: Animated renderings of the 3D qualitative results of the synthetic datasets as presented in the figures of the main manuscript. Top rows samples are from Structured3D, bottom row samples are from Kujiale. (animations are only playable in recent Adobe Acrobat Reader versions).