OmniLayout: Room Layout Reconstruction from Indoor Spherical Panoramas

Shivansh Rao [*] Vikas Kumar [*] Daniel Kifer C. Lee Giles Ankur Mali
The Pennsylvania State University, University Park
PA, USA 16802
{shivanshrao,vuk160,duk17,clg20,aam35}@psu.edu

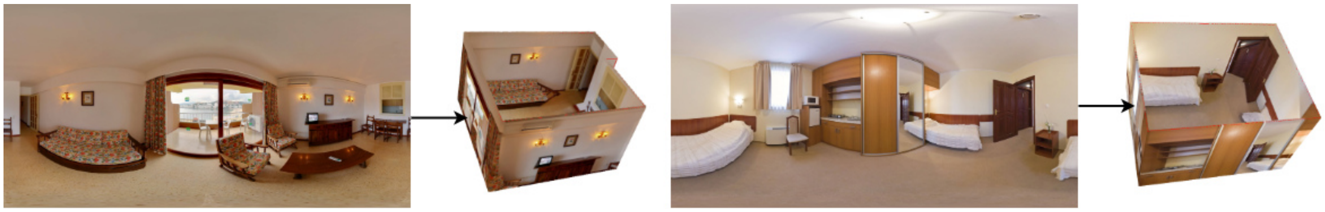

Figure 1. **Illustration:** Our OmniLayout can predict both non-cuboid layout as well as cuboid layout from the input RGB panorama.

Abstract

Given a single RGB panorama, the goal of 3D layout reconstruction is to estimate the room layout by predicting the corners, floor boundary, and ceiling boundary. A common approach has been to use standard convolutional networks to predict the corners and boundaries, followed by post-processing to generate the 3D layout. However, the space-varying distortions in panoramic images are not compatible with the translational equivariance property of standard convolutions, thus degrading performance. Instead, we propose to use spherical convolutions. The resulting network, which we call OmniLayout performs convolutions directly on the sphere surface, sampling according to inverse equirectangular projection and hence invariant to equirectangular distortions. Using a new evaluation metric, we show that our network reduces the error in the heavily distorted regions (near the poles) by $\approx$ 25% when compared to standard convolutional networks. Experimental results show that OmniLayout outperforms the state-of-the-art by $\approx$4% on two different benchmark datasets (PanoContext and Stanford 2D-3D). Code is available at https://github.com/rshivansh/OmniLayout.

1. Introduction

Estimating the 3-dimensional layout of the room from a single RGB image has received considerable attention in the last decade. Layout estimation can present useful information (height, corner positions, and orientation of the room) for holistic scene understanding applications such as robotics and augmented/virtual reality [27, 10]. Most of the previous works [26, 30, 34] tackle room layout estimation problem by using artificial neural networks (ANNs). They capture the salient features from the image while considering the manhattan room layout [7]. These approaches have shown impressive results not just in terms of the quantitative evaluation but also qualitatively by generating both cuboid-shaped room layouts as well as non-cuboid-shaped general layouts. Since conventional cameras have a limited field of view leading to several ambiguities, existing literature [26, 30, 34, 11] directly operates on 360° panoramas, exploiting the wider field of view.

Although existing work are heavily dependent on standard convolution layers, showing impressive results on few panoramic benchmarks [26, 30, 34]. We believe standard convolution often fail to capture features in panoramic images, thus leading to sub-optimal representation. Prior works also argue that standard convolutions are not well suited for processing panoramic images [6, 9]. This is because equirectangular images, which are considered a common example for spherical image representation, have heavy distortions in them (especially towards the poles) which cannot be addressed by standard convolutions [6, 9]. Thus leading to bottleneck in all the prior approaches while dealing with room layout estimation. To address this problem and towards building robust representation for

[*] equal contribution

panoramic image we present OmniLayout: a deep neural network that estimates the room layout while accounting for these common distortions pattern. Inspired from [6], our model tackles the distortions in the given equirectangular image by changing where the convolutional kernel samples from the image in a location-dependent manner. Instead of performing the convolution operation on the regular image domain, our network performs convolutions on the sphere surface where the omnidirectional images can be represented without any distortions. We show that our methodology significantly boosts the performance in the complex regions of the images (i.e., the polar regions containing most of the distortions) while maintaining equal or better performance in the least complex regions of the images (near the equator).

While Coors *et al.* [6] uses gnomic projection to map the sphere onto a tangent plane, we argue that this is not an accurate projection for equirectangular images. Instead, we perform sampling using inverse equirectangular projection which leads to better representation across wide variety of networks, as shown in later section. Our main idea is to use more principled approach and replace the standard convolutions with spherical convolutions, which we believe are well-suited for the task of room layout estimation from a panorama. We build our network on top of HorizonNet [26] and replace standard convolution operation with spherical convolution for enhanced representation and reduce computational complexity by replacing Bi-LSTM with Bi-GRU. We validate our hypothesis by conducting several experiments across two large-scale benchmarks [32, 1]. Finally, we conduct an ablation study across each model component to highlight their significance and contribution resulting in better estimation over panoramic images.

2. Related Work

Room layout estimation from a single RGB image has been an active area of research in the last decade. The existing literature differs in mainly two different aspects: 1) input image type, and 2) proposed methodology. In this section, we review several lines of related work falling in each of the categories.

In terms of input image type, prior work differ on the basis of the field of view (FoV), ranging from the normal perspective images to 360° panorama images. Delage *et al.* [8], Hedau *et al.* [14] and Lee *et al.* [18] operate only on the perspective images, while Zhang *et al.* [32] estimates the room layout directly from a single 360° panorama and proposes the PanoContext dataset. Xu *et al.* [29] combines surface normal estimates, 2D object detection, and 3D object pose estimation to estimate the room layout and 3D pose of the object. There are some other works that use more information than just a single image, such as using multiple images [3] or using the depth information as well (RGB-D data) [19, 12, 31].

Most of the recently proposed methodologies incline towards adopting deep neural networks to improve layout estimation. These approaches use dense models to predict the semantic label of each pixel. Some of these approaches [20, 24, 33] operate on the perspective images. Mallya *et al.* [20] learns to predict informative edge probability maps whereas Zhao *et al.* and Ren *et al.* [33, 24] predict for the boundary classes. Since the recent increase in omnidirectional sensors, there have been a few deep learning approaches that directly operate on panoramas. Zou *et al.* [34] presents a method that can generate both cuboid layout and general layout directly from the given panorama. Yang *et al.* [30] uses two different projections of the panorama at the same time (front-view panorama and top-view perspective) showing the advantages of additional information from the ceiling-view image. Sun *et al.* [26] presents a new approach by representing the room layout as a 1D representation.

Although existing works show impressive performance for both cuboid as well as non-cuboid layouts [26, 30, 34], none of them considers the distortions that the equirectangular images contain. There is an incongruence between the panoramic images and standard convolutional networks. A few recent approaches have proposed to overcome the distortions by using spherical convolutions. Su *et al.* [25] proposes to increase the kernel size of the standard convolution filters towards the polar regions. However, this results in a significant increase in the model parameters, since the weights now can only be shared along each row. Cohen *et al.* [5] proposes to use spherical CNNs that encodes full rotational invariance. However, assuming that the camera is not tilted while capturing 360° images, full rotational invariance is an undesired property for our task and reduces the discriminative power of the model. In concurrent work, Coors *et al.* [6] addresses the issue by capturing rotational invariance only in one dominant orientation and is compatible with modern CNN architectures. Additionally, it allows the transfer of pre-trained object detectors to omnidirectional inputs. Results show that SphereNet [6] performs better than the other methods that handle omnidirectional inputs on benchmark datasets Omni-MNIST and Flying-Cars [6].

To our knowledge, none of the previous work in room layout estimation deal with the shortcoming that standard convolutions have in terms of panoramic images, except the work in [11]. While Clara *et al.* [11] has attempted to reduce the distortions in the equirectangular representation, it considers the convolutional kernel as a tangent plane to a sphere making use of the inverse gnomic projection which is not the accurate projection type for equirectangular images. In our work, we use the inverse equirectangular projection which we believe is the accurate projection type to eliminate the existing distortions in equirectangular images.

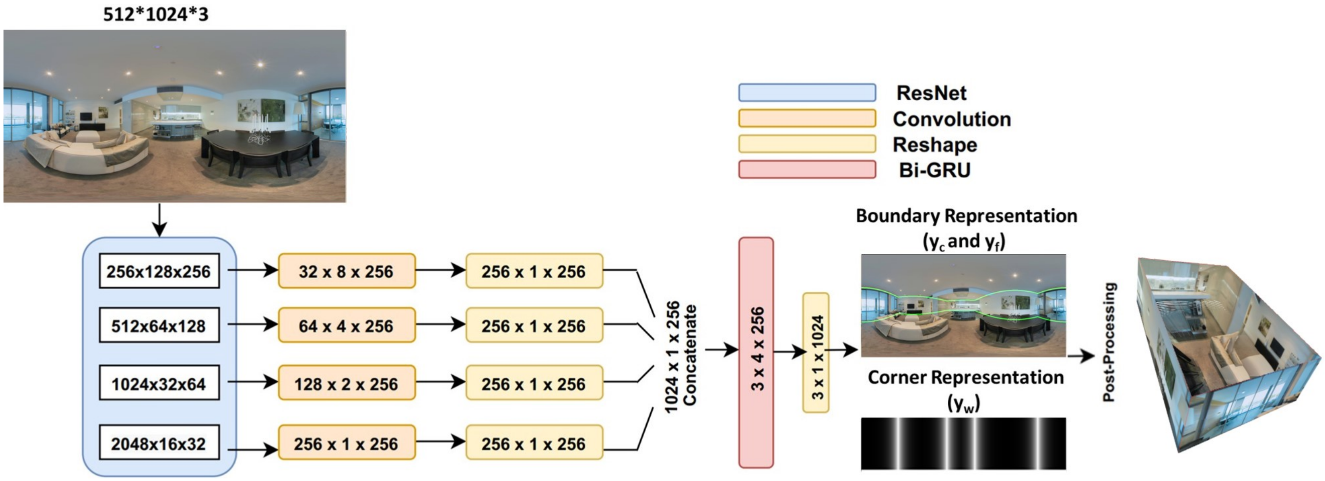

Figure 2. **OmniLayout architecture:** Our model is built on ResNet followed by Bi-Directional GRU that predicts the positions of corners of the room. We have replaced the standard convolution from each block of ResNet with the sphere convolution (inverse equirectangular projection). The output of the network is a 1D representation map of shape 3 × 1 × 1024. Since the width of the panorama is 1024 the output map has 3 values per column: y_c (ceiling-wall), y_f (floor-wall) and y_w (wall-wall / corners).

3. Approach

In this section, we describe our end-to-end network for generating the 3D room layout from a single RGB panorama. We first give a brief overview of our architecture (Sec. 3.1), followed by the description of inverse equirectangular projection for spherical convolutions (Sec. 3.2). Then we describe our model's architecture (Sec. 3.3, 3.4). Finally, we describe the post-processing details for generating the 3D room layout from the model's predictions (Sec. 3.5).

3.1. Network Architecture

An overview of OmniLayout is illustrated in Fig. 2. The proposed architecture consists of a ResNet-50 [13] encoder with proposed spherical convolutions. We remove the final fully-connected layer and concatenate the features from different levels and pass it to a Bi-Directional Gated Recurrent Unit (Bi-GRU) [4] that predicts the layout floor-wall boundary (y_f), ceiling-wall boundary (y_c), and wall-wall boundary (y_w).

3.2. Convolution for Panoramic Images

Omnidirectional sensors have gained huge popularity in the last few years due to their wider field of view with several applications in virtual/augmented reality and robotics. Due to an increase in omnidirectional sensors, spherical imagery is receiving increased attention as well. The most common representation of spherical images is the equirectangular projection in which the longitude and latitude of a spherical image are mapped to vertical and horizontal coordinates. However, this mapping comes with heavy distortions, especially near the poles. Standard convolutions are not a good choice for such images. From Fig. 3 we can observe how the proposed kernel deforms itself near the poles in order to account for the distortions.

One of the simplest examples of a covariant neural network one can consider are traditional $s+1$ layers CNN used for image recognition and other vision-related tasks. Traditionally neurons in each layer of CNN are arranged in a rectangular grid. Let us consider a network with a single channel, then the activation of layer s can be regarded as a function $f^s\colon \mathbb{Z}^2 \to \mathbb{R}$, with f^0 being the input image [5, 16]. We now adopt notations and definitions proposed in prior work [16] and define the overall flow for spherical CNNs. As noted earlier the neurons in our network compute f^s by taking the cross-correlation of the previous hidden layer's output f^{s-1} with a learnable filter or kernel h^s as follows:

$$(h^s \star f^{s-1})(x) = \sum_{y} h^s(y-x)\, f^{s-1}(y), \qquad (1)$$

then we apply nonlinear activation function σ, such as the ReLU or other variants operator [1]:

$$f^s(x) = \sigma(h^s \star f^{s-1})(x). \qquad (2)$$

Defining $T_x(h^s)(y) = h^s(y-x)$, which is nothing but h^s translated by x, allows us to equivalently write Eq. 1 as follows:

$$(h^s \star f^{s-1})(x) = f^{s-1}, T_x(h^s), \qquad (3)$$

[1] Better results can be obtained by using variants of ReLU or other functions such as SELU which might lead to better gradient flow across model.

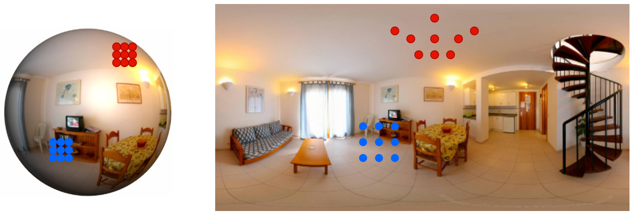

Figure 3. **Spherical Convolution on panoramic images:** We show two different kernel positions, one at the center of the image (blue) and one toward the poles (red). Equirectangular images usually has more distortions near the poles. The proposed kernel deforms itself near the poles accounting for the distortions in that region when compared to the region near the equator.

where the inner product is $f^{s-1}, T_x(h^s) = \sum_y f^{s-1}(y)\, T_x(h^s)(y)$. This formulation as noted in prior works indicates that each layer in CNN are doing some kind of pattern matching: $f^s(x)$ is an indicatior of how well the part of f^{s-1} around x matches the filter or kernel h^s. Equation 3 is the natural starting point for generalizing convolution to the unit sphere, S^2.

A number of authors have addressed the issue of discretizing S^2 by a regular arrangement of points, which is often convenient when dealing with planes [2, 25]. Instead of following the aforementioned approaches, similarly to recent work on manifold CNNs [21, 22], one can simply treat each f^s and the corresponding filter h^s as continuous functions on the sphere [16], $f^s(\theta, \phi)$ and $h^s(\theta, \phi)$, where θ and ϕ are the polar and azimuthal angles. Thus, we perform the convolutions operations directly on the sphere surface instead of the image domain, giving use advantage when compared with competitors. We allow both functions to be complex-valued which is argued to provide better generalization [16].

Finally the correct way to generalize cross-correlations on a sphere while considering the rotation around a third axis [16] can be established by defining $h \star f$ as a function, that is represented as follows:

$$(h \star f)(R) = 4\pi \int_0^{2\pi} \int_{-\pi}^{\pi} h_R(\theta, \phi)^* \, f(\theta, \phi) \, \cos\theta \, d\theta \, d\phi R(3), \tag{4}$$

where h_R is h rotated by R. Further we can express these terms as follows:

$$h_R(x) = h(R^{-1}x), \tag{5}$$

with x being the point on the sphere at position (θ, ϕ) .

This formulation offers one key advantage by efficiently encoding equirectangular projection into the kernel's sampling function, thus allowing better estimation over the spherical surface as opposed to standard convolutions. We formulate the kernel over a cylindrical patch available in the spherical surface and then sample the equirectangular projection. The positions of the kernel locations on the cylindrical patch are calculated similarly to [6], while ensuring that we use equirectangular projection instead of gnomic projection. The equirectangular projection is described as follows:

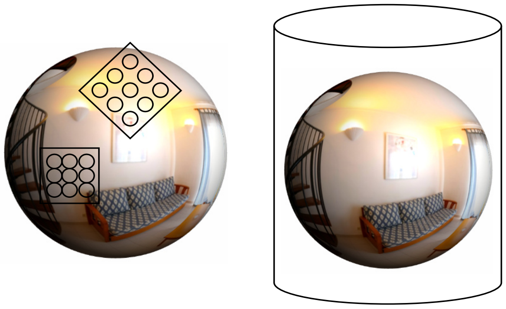

Figure 4. Gnomic projections are azimuthal projection (left) that project sphere to tangent planes, and Equirectangular projections are cylindrical projection (right) that project sphere to a cylinder.

$$\begin{aligned} \theta &= v_0 + \Delta v_{(i,j)}, \\ \phi &= u_0 + \Delta u_{(i,j)} \sec\theta, \end{aligned} \tag{6}$$

where the sphere is parameterized in terms of its polar (θ) and azimuthal angles (ϕ).u$_0$ and v_0 represents the center of the kernel, $\Delta u_{(i,j)}$ and $\Delta v_{(i,j)}$ represents the angular distance at index (i,j) from the kernel center in the x and y direction respectively. The approach proposed by Clara *et al.* [11] and Coors *et al.* [6] instead utilizes the inverse gnomic projection which maps the sphere to a tangent plane (See Fig. 4). Since equirectangular images are cylindrical projections that project sphere to a cylinder, the distortions produced by them are different which can not be handled by the gnomic projections.

3.3. Encoder

To be comparable with current state-of-the-art model, we adopt the same feature extractor - ResNet [13] as the HorizonNet [26]. The input panorama is of shape 3 × 512 × 1024. ResNet initially has a convolution layer of 7 × 7 kernel with stride 2 and padding 3. This is followed by four blocks, each block consists of a sequence of convolution layers reducing the channels and height by a factor of 8 (i.e. for first block 256 / 8 = 32) and 16 (i.e. for first block 128 / 16 = 8) respectively. Precisely, there are three convolution layers in each block. The features from different blocks help to capture both low-level, as well as high-level information [17] from the given panorama. The output feature from each block is reshaped to $256 \times 1 \times 256$ tensors and concatenated to form a single tensor of shape $1024 \times 1 \times 256$. In the base architecture of ResNet, we convert all the standard convolutions to spherical convolutions with inverse equirectangular projections (see section 3.2). In the ablation study (see section 4.4), we show that the spherical convolution shows improvement across the entire family of ResNet and is not restricted to ResNet-50.

3.4. Recurrent Neural Network

Due to the geometry of a room, a corner can be approximately predicted from the position of other corners of the room. Assum-

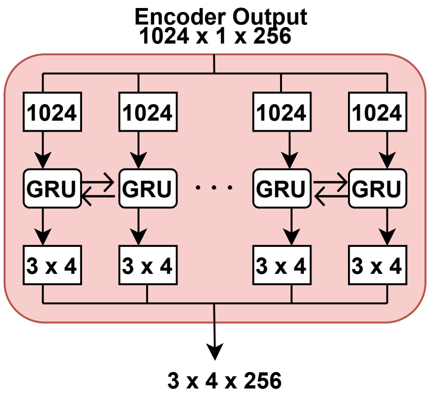

Figure 5. Illustration of Bi-GRU used for predicting corners, floor boundary and ceiling boundary.

ing this we feed the concatenated feature map from encoder as the input sequence to a recurrent neural network (RNN), more specifically to a bi-directional gated recurrent unit (Bi- GRU). RNN's are stateful models better known for capturing long-range dependencies. Non-local neural networks [23, 28] are another alternative and are faster in comparison to RNN's, however we leave this for future investigation [2]. The input sequence is of shape $1024 \times 1 \times 256$ and the Bi-GRU produces the output sequence of shape $3 \times 4 \times 256$ which is later reshaped to $3 \times 1 \times 1024$ (see Fig. 2 and Fig. 5). Thus the room layout is represented as three 1D predictions similar to [26]. This formulation leads to computational efficiency model while training.

We set Bi-GRU sequence length equal to the width of the image (1024) and predicts three values for each column of the image (y_c, y_f, and y_w). To reduce the computational time the Bi-GRU predicts for four columns at any given time instead of a single column thus the output is of shape $3 \times 4 \times 256$. We use the bidirectional nature of GRU since it offers flexibility of incorporating left and right context values (y_w, y_f, and y_c) offering enhanced representation for our model.

3.5. 3D Layout Generation

For recovering the 3D layouts from the predictions, we follow the methodology described by Sun *et al.* [26] and make the following assumptions: (a) all rooms follow the Manhattan world assumptions, (b) the camera height is 1.6 meters [32] above the floor, and (c) pre-processing [34] correctly aligns the floor perpendicular to the y-axis. There are two broad steps in the layout recovery, the first is to recover the floor plane and ceiling plane, while the second is to recover the wall-wall planes. First the model's predictions provide the locations of floor boundaries (y_f), and ceiling boundaries (y_c) for every column, we can project them from image coordinates to 3D XYZ coordinates. Second the ceiling wall boundaries share the same positions as the floor wall boundary (X and Z). We then subtract the ceiling and floor 3D coordinates for each image column and take the average over all the image columns to get the height h of the room.

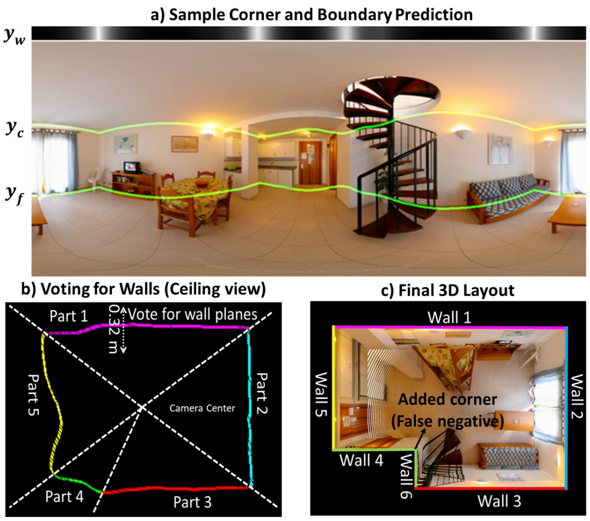

Figure 6. Our model predicts (top) the ceiling-wall (y_c) boundary, floor-wall (y_w) boundary and the wall-wall (y_w) probability map. The post-processing is done in ceiling view (bottom-left). This helps in enforcing the orthogonality of adjacent walls and helps in detecting false negative corners (bottom-right).

Later the wall planes are recovered by selecting peak points from the predicted wall-wall probability map (y_w) which have the peak signal strength in its 5° field of view and minimum signal strength of 0.05. While the prediction of boundaries and corners are done using the equirectangular view (See Fig. 6a), the post-processing is done using the ceiling view. To correct the horizontal alignment of the 3D layout, the ceiling wall boundary is divided into parts ($p_1, p_2, ..., p_n$) using the prominent peaks (see Fig. 6b). It then gives a higher score to the vector line with more pixel points within 0.16 meters and selects the vector that obtains the highest score as the wall for every part p_i (see Fig. 6b).

Finally we force adjacent walls to be orthogonal to each other, however the wall whose adjacent walls have not been constructed yet are free to choose the orthogonality type. We also consider special cases, where two adjacent walls for a part p_i are already constructed, but their vector lines are orthogonal to each other (instead of being parallel). This may occur in rare cases of occluded or undetected corners, hence its important to add an additional corner to the layout based on the position of the adjacent walls with respect to part p_i. For example, in Figure 6, we see that Wall 5 is orthogonal to Wall 3 which violates the Manhattan properties leading to a new corner being added to the layout. The ceiling-wall corner points are established using the intersection of 3 perpendicular planes (2 adjacent walls and a ceiling). The floor-wall corner points are found using the ceiling-wall corner points and the height of the room. More details can be found in [26].

4. Experiments

In this section, we first introduce the datasets used in the experiments (Sec. 4.1). Then we describe our experimental setup and some implementation details (Sec. 4.2). At the end, we present the experimental results and compare with other state-of-the-art

[2] We performed experiments with both LSTM [15] as well as GRU [4] and finalized Bi-GRU for our network since it trains faster and gives approximately the similar performance as a Bi-LSTM.

Method	Pixel	Corner	3D IoU
	Error (%)	Error (%)	
PanoContext [32]	4.55	1.60	67.23
CFL [11]	2.49	0.79	78.79
LayoutNet [34]	3.34	1.06	74.48
DuLa-Net [30]	-	-	77.42
HorizonNet [26]	2.7	0.82	79.8
Ours	**2.2**	**0.75**	**83.02**

Table 1. Cuboid layout estimation evaluation on PanoContext Dataset [32] (Training set - PanoContext [32]).

Method	Pixel	Corner	3D IoU
	Error (%)	Error (%)	
LayoutNet [34]	3.18	1.02	75.12
HorizonNet [26]	2.6	0.79	80.2
Ours	**2.10**	**0.69**	**84.5**

Table 2. Cuboid layout estimation evaluation on PanoContext Dataset [32] (Training set - PanoContext [32] + Stanford 2D-3D [1]).

Method	Pixel	Corner	3D IoU
	Error (%)	Error (%)	
LayoutNet [34]	2.70	1.04	76.33
DuLa-Net [30]	-	-	79.63
HorizonNet [26]	2.5	0.97	77.2
Ours	**2.37**	**0.78**	**81.2**

Table 3. Cuboid layout estimation evaluation on Stanford 2D-3D Dataset [1] (Training set - Stanford 3D-3D [1]).

Method	Pixel	Corner	3D IoU
	Error (%)	Error (%)	
LayoutNet [34]	2.42	0.92	77.51
HorizonNet [26]	2.36	0.77	80.8
Ours	**2.14**	**0.68**	**83.4**

Table 4. Cuboid layout estimation evaluation on Stanford 2D-3D Dataset [1] (Training set - PanoContext [32] + Stanford 2D-3D [1]).

results (Sec. 4.3 and Sec. 4.4).

4.1. Dataset

We conduct experiments on two benchmark datasets: PanoContext [32] and Stanford 2D-3D [1] extended by [34].

PanoContext: This dataset consists of 500 annotated cuboid room layouts. We perform the same experimental protocol as [26] and [34] by splitting 10 % validation images from the training set to make sure similar rooms do not appear in the training set. The panoramas are captured from indoor settings such as living rooms and bedrooms.

Stanford 2D-3D: This dataset consists of 571 RGB panoramas with room layout annotations provided by [34]. The panoramic images are captured from large-scale indoor environments such as offices, classrooms, and corridors. This is a more challenging dataset since it has more occlusions on the floor boundaries and the images have a smaller vertical field of view.

4.2. Setup and Implementation Details

We follow the same train/val/test split as LayoutNet [34] for both the datasets PanoContext [32] and Stanford 2D-3D [1] and use the same experimental protocol described in [26] for training the baseline method. Training and test set images are preprocessed by the panoramic image alignment method proposed in [34]. PanoStretch data augmentation [26] is used to augment the training data by stretching the panorama images along the axes in 3D space. The main idea of PanoStretch data augmentation [26] is to convert the pixels of the equirectangular image to 3D space and multiply their X, Y, Z coordinates with separate hyperparameters (augmentation parameters). The stretched points can then be projected back to form the final image. Exact details can be found in [26].

To predict the position of the floor-wall boundary (y_f) and ceiling-wall boundary (y_c) we use L1 loss for the learning, whereas for the prediction of the wall-wall boundary (y_w) we use binary cross-entropy loss. The optimizer used is Adam with a learning rate of 0.0003. We train all our networks by using the pretrained Imagenet weights for 150 epochs with a batch size of 4. It takes around 12 hours to finish the training on a single NVIDIA GTX 1080 GPU. [3]

4.3. Experimental Results

We present the results on two benchmark datasets described in Sec. 4.1 and compare our model with the current state-of-the-art models in Table. 1 - Table. 4. We perform both quantitative (Sec. 4.3.1) and qualitative evaluation (Sec. 4.3.1). Finally in Sec. 4.4, we conduct ablation studies to highlight the gain in performance due to the proposed sphere convolution formulation.

4.3.1 Quantitative Evaluation

We measure the quantitative performance based on three standard metrics: 3D intersection over union between the predicted 3D reconstructed layout and ground truth, pixel error between predicted and ground truth surface class, and corner error which measures the euclidean distance between predicted and ground truth corners. Apart from the architectural difference, existing literature differ in the input resolution and augmentation techniques. Clara *et al.* [11] and Yang *et al.* [30] use input image of resolution 256 × 512 whereas Zou *et al.* [34], Sun *et al.* [26] and our method

[3] Training time can be further optimized by optimizing spherical CNN implementation on CUDA. The training time in the case of standard convolutions is 3x faster.

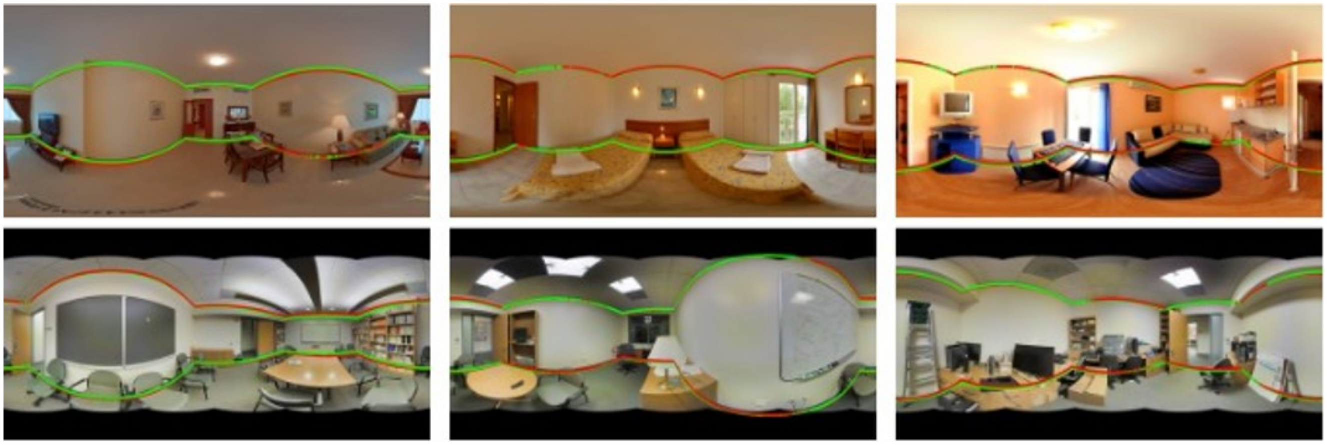

Figure 7. Qualitative results for room layout estimation on PanoContext [32] (top) and Stanford 2D-3D [1] (bottom). Each image was randomly sampled from the dataset. Our model's prediction is highlighted in red color whereas the ground truth is highlighted in green color. Best viewed in color.

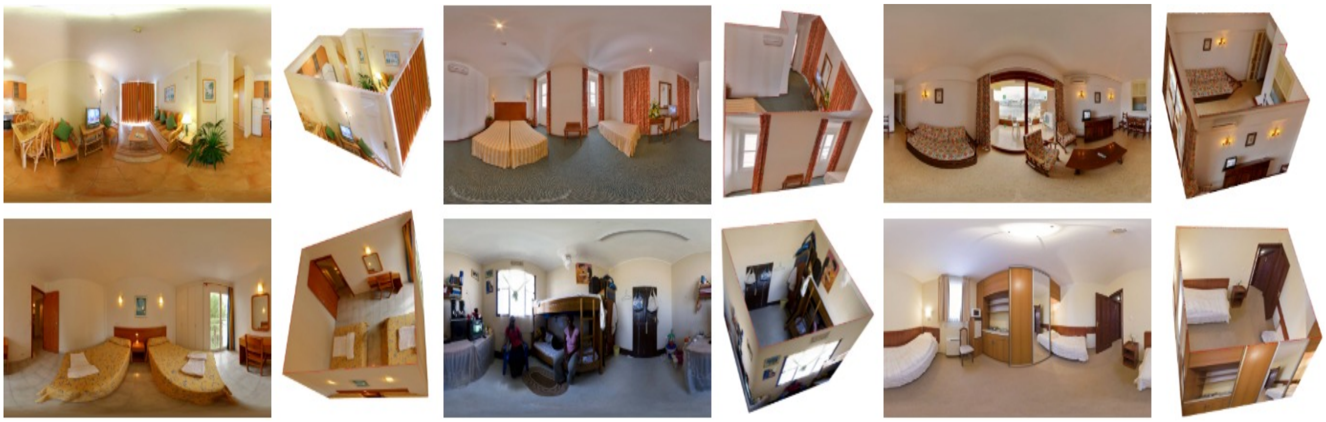

Figure 8. Qualitative results for the non-cuboid layout prediction (top row) and the cuboid layout prediction (bottom row). Best viewed in color.

uses input resolution of 512 × 1024. Clara *et al.* [11] is trained with random erase augmentation technique, while our method and Sun *et al.* [26] are both trained with the PanoStretch augmentation technique. Throughout our evaluation from Table 1 - Table 4, we compare with the state-of-the-art results that we were able to reproduce from the open-source codes.

Based on the evaluation metrics pixel error (%), corner error (%), and 3D IoU we can see that our method is the new state-of-the-art approach and outperforms all prior methods by $\approx 4\%$ on both benchmarks PanoContext [32] and Stanford 2D-3D [1]. The comparison with more relevant approach [11] validates our hypothesis since they are only other work using spherical convolutions for layout estimation. However Clara *et al.* [11] use spherical convolution with inverse gnomic projection and reports 3D IoU, which is $\approx$ 5% lower compared to our method on both PanoContext [32] and Stanford 2D-3D [1] benchmarks.

Although our method uses a different projection type for spherical convolution, in ablation studies we show our network's results with the inverse gnomic projection are similar to the projection type used in [11] despite this our network achieves $\approx 4\%$ better performance than Clara *et al.* [11]. Therefore we hypothesis that the boost in our performance is due to incorporation of spherical convolution with better representation architecture. To validate the importance of spherical convolution alone, we perform several experiments as described in Sec. 4.4.

4.3.2 Qualitative Evaluation

We present the qualitative results in the form of room layout maps (Fig.7) and 3D layouts of both non-cuboid and cuboid-shaped rooms (Fig. 8). The non-cuboid rooms in Stanford 2D-3D [1] and PanoContext [32] are labelled as cuboids, thus making it difficult for our model to learn non-cuboid rooms. To overcome this, we use the 65 general room layouts re-labeled by Sun *et al.* [26] in the train split to fine-tune our network. The samples in Fig. 8 show that our network is capable of generating both non-cuboid ("L-shaped") room layouts as well as cuboid room layouts.

From Fig. 7 we can observe the obvious similarity between model's predictions and ground truth. One important aspect of our model is the capability to detect the corner while estimating the boundary line even when the corner is hidden (such as corner hidden behind a door in Fig. 7). We believe that our model

Method	Pixel Error (%)	Corner Error (%)	3D IoU
Standard Conv	2.6	0.79	81.4
Sphere Conv (Inv. gnomic proj.)	2.09	0.669	84.65
Ours	**2.06**	**0.662**	**86.15**

Table 5. Comparison between standard convolution, spherical convolution (with inverse gnomic projection) and spherical convolution (with inverse equirectangular projection) on the PanoContext [32] + Stanford 2D-3D dataset [1]

representation combined with Bi-GRU to understand context over longer horizon, leads to better prediction of corners.

4.4. Ablation Study

In Table. 5, we compare the results with standard convolution, spherical convolution with inverse gnomic projection [6] and our proposed spherical convolution with inverse equirectangular projection. It is evident from our results that both the spherical convolutions (inverse gnomic and inverse equirectangular projection) are better when compared with standard convolution. Thus validating the hypothesis that spherical convolutions are well suited for this problem and can efficiently handle the distortions in the equirectangular images. Since the property of equirectangular images are more inclined towards cylindrical projections rather than projections over tangent plane, inverse equirectangular projection offers rich representation and leads to improved performance (≈ 2 %) than inverse gnomic projection (Table. 5).

While dealing with equirectangular images metric such as pixel error offer least information and fail to capture significance of the model, since error is calculated over entire dataset. It is important to know the various regions (i.e simple or complex regions) in image which lead to improvement or degradation in model performance. To incorporate this scenario we propose a new metric, which identifies the region or groups where spherical convolution performs better than standard convolution. In Fig. 9, we plot the pixel error (%) observed in the test set for standard convolution and spherical convolution against row groups. The panoramic images are divided into different row groups based on distance from the poles (See Fig. 9), where each row group has a width of 25 rows. As hypothesised the difference in the pixel space is highest (≈25%) when we are closer to the poles and the image regions where the ceiling-wall and floor-wall boundaries are likely to appear in majority of samples. The difference gradually decreases as we go towards the equator of the image (i.e simple region). This confirms to our assumption that majority of the distortion that our method removes are near the poles of the image, which we categorize as the difficult or complex regions for standard convolutions.

Finally, we input our proposed sphere convolution to the following networks of the ResNet family: ResNet-34 [13], ResNet-101 [13], and ResNet-151 [13] and report the results in Table 6. It is evident that our approach is not restricted to any architecture and can improve performance across any convolution architecture. The proposed method is independent of model parameters and depth of the network, hence for complex tasks can also be extended to work with very deep networks.

Method	Pixel Error (%)	Corner Error (%)	3D IoU
Standard Conv (ResNet34)	2.56	0.79	81.4
Spherical Conv (ResNet34)	**2.2**	**0.67**	**85.7**
Standard Conv (ResNet101)	2.53	0.77	82.1
Spherical Conv (ResNet101)	**2.05**	**0.65**	**86.3**
Standard Conv (ResNet151)	2.52	0.76	82.4
Spherical Conv (ResNet151)	**2.04**	**0.65**	**86.5**

Table 6. Comparison between standard convolution and proposed spherical convolution for 3 different networks of the ResNet family. Evaluation done on both PanoContext [32] and Stanford 2D-3D [1].

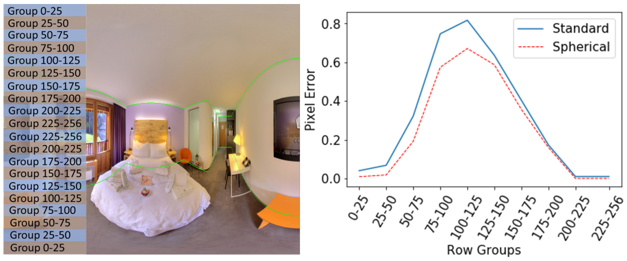

Figure 9. Input images are divided into row groups (left image shows a sample for reference) based on distance from the poles. We calculate pixel error in these row groups to show the difference in performance across different regions of the image. Results evaluated on PanoContext [32] and Stanford 2D-3D [1].

5. Conclusions

We proposed a novel state-of-the-art approach which reduces the distortions in equirectangular images for the task of 360° room layout recovery. In our knowledge this is the first work in room layout estimation that uses the equirectangular projection function to reduce the distortions. The proposed method, OmniLayout is computationally efficient and can also recover both cuboid shaped layouts as well non-cuboid shaped layouts ("L-shaped"). The experimental analysis and ablation study shows that OmniLayout significantly improves the performance on two room layout benchmark datasets, especially in the distortion heavy regions of the input panoramic images.

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, 6, 7, 8

[2] Wouter Boomsma and Jes Frellsen. Spherical convolutions and their application in molecular modelling. In *NIPS*, volume 2, page 6, 2017. 4

[3] Ricardo Cabral and Yasutaka Furukawa. Piecewise planar and compact floorplan reconstruction from images. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 628–635. IEEE, 2014. 2

[4] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 3, 5

[5] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018. 2, 3

[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. 1, 2, 4, 8

[7] James M Coughlan and Alan L Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 941–947. IEEE, 1999. 1

[8] Erick Delage, Honglak Lee, and Andrew Y Ng. A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 2418–2428. IEEE, 2006. 2

[9] Marc Eder, True Price, Thanh Vu, Akash Bapat, and Jan-Michael Frahm. Mapped convolutions. *arXiv preprint arXiv:1906.11096*, 2019. 1

[10] Clara Fernandez-Labrador. *Indoor Scene Understanding using Non-Conventional Cameras*. PhD thesis, Université de Bourgogne Franche-Comté (COMUE)(UBFC), FRA.; Universidad . . . , 2020. 1

[11] Clara Fernandez-Labrador, Jose M Facil, Alejandro Perez-Yus, Cédric Demonceaux, Javier Civera, and Jose J Guerrero. Corners for layout: End-to-end layout recovery from 360 images. *IEEE Robotics and Automation Letters*, 5(2):1255–1262, 2020. 1, 2, 4, 6, 7

[12] Ruiqi Guo, Chuhang Zou, and Derek Hoiem. Predicting complete 3d models of indoor scenes. *arXiv preprint arXiv:1504.02437*, 2015. 2

[13] 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. 3, 4, 8

[14] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. In *2009 IEEE 12th international conference on computer vision*, pages 1849–1856. IEEE, 2009. 2

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 5

[16] Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch-gordan nets: a fully fourier space spherical convolutional neural network, 2018. 3, 4

[17] Vikas Kumar, Shivansh Rao, and Li Yu. Noisy student training using body language dataset improves facial expression recognition. In *European Conference on Computer Vision*, pages 756–773. Springer, 2020. 4

[18] David C Lee, Martial Hebert, and Takeo Kanade. Geometric reasoning for single image structure recovery. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2136–2143. IEEE, 2009. 2

[19] Chen Liu, Pushmeet Kohli, and Yasutaka Furukawa. Layered scene decomposition via the occlusion-crf. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–173, 2016. 2

[20] Arun Mallya and Svetlana Lazebnik. Learning informative edge maps for indoor scene layout prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 936–944, 2015. 2

[21] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015. 4

[22] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5115–5124, 2017. 4

[23] Shivansh Rao, Peng Cao, Tanzila Rahman, Mrigank Rochan, and Yang Wang. Non-local attentive temporal network for video-based person re-identification. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8. IEEE, 2019. 5

[24] Yuzhuo Ren, Shangwen Li, Chen Chen, and C-C Jay Kuo. A coarse-to-fine indoor layout estimation (cfile) method. In *asian conference on computer vision*, pages 36–51. Springer, 2016. 2

[25] Yu-Chuan Su and Kristen Grauman. Learning spherical convolution for fast features from 360 {\deg} imagery. *arXiv preprint arXiv:1708.00919*, 2017. 2, 4

[26] 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*, pages 1047–1056, 2019. 1, 2, 4, 5, 6, 7

[27] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Hohonet: 360 indoor holistic understanding with latent horizontal features. *arXiv preprint arXiv:2011.11498*, 2020. 1

[28] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 5

[29] 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. 2

[30] Shang-Ta Yang, Fu-En Wang, Chi-Han Peng, Peter Wonka, Min Sun, and Hung-Kuo Chu. Dula-net: A dual-projection network for estimating room layouts from a single rgb panorama. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3363–3372, 2019. 1, 2, 6

[31] Jian Zhang, Chen Kan, Alexander G Schwing, and Raquel Urtasun. Estimating the 3d layout of indoor scenes and its clutter from depth sensors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1273–1280, 2013. 2

[32] 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. 2, 5, 6, 7, 8

[33] Hao Zhao, Ming Lu, Anbang Yao, Yiwen Guo, Yurong Chen, and Li Zhang. Physics inspired optimization on semantic transfer features: An alternative method for room layout estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 10–18, 2017. 2

[34] 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, 5, 6