Floorplan Restoration by Structure Hallucinating Transformer Cascades

Sepidehsadat Hosseini
sepidh@sfu.ca

Yasutaka Furukawa
furukawa@sfu.ca

GrUVi Lab, VML Lab
Simon Fraser University
British Colombia, BC

Abstract

This paper presents a novel floorplan restoration task, a new benchmark for the task, and a neural architecture as a solution. Given a partial floorplan reconstruction inferred from panorama images, the task is to restore a complete floorplan including invisible architectural structures. The proposed neural network 1) encodes an input partial floorplan into a set of latent vectors by convolutional neural networks and a Transformer; and 2) recovers an entire floorplan while hallucinating invisible rooms and doors by cascading Transformer decoders. Qualitative and quantitative evaluations demonstrate the effectiveness of our approach over the benchmark of 701 houses, outperforming the state-of-the-art reconstruction techniques. We will publish code and data.

1 Introduction

Indoor panorama photography is exploding. Pioneered by Ricoh Theta [2], consumer-grade panorama cameras are prevalent on the market, whose applications range from real estate to entertainment and surveillance. Ricoh Theta cameras have collected 100 million panoramas from residential houses, which allow house renters/buyers/realtors to browse immersive 360 views for tens of millions of houses. However, these panorama collections are extremely sparse (i.e., one panorama per room with little visual overlaps) and even leave some rooms invisible, posing fundamental challenges to existing techniques.

This paper presents a new floorplan restoration task, a benchmark, and a solution, which could exploit 100 million panoramas to create floorplans for tens of millions of houses. The potential applications range from real estate and construction industries (*e.g.*, building code verification and property value assessment) to virtual and augmented reality. Concretely, the task is to take a partial floorplan inferred or manually curated from panorama images, and restore a complete floorplan. Note that the task is named "restoration", which is too ambiguous as reconstruction, but has a unique solution, and is more constrained than inpainting.

The technical challenge lies in the restoration or "hallucination" of invisible rooms and doors. Inferring invisible image or geometry data has been studied in computer vision in the context of image inpainting [32], illumination inference [22], amodal segmentation[34, 57], and surface reconstruction [28]. To our knowledge, this paper is the first to tackle the hallucination of architectural components, such as rooms and doors, at the scale of an entire house.

Our contribution is three fold: 1) A floorplan restoration task; 2) A benchmark with 701 houses with ground-truth vector floorplan images; and 3) A neural architecture, whose cascading Transformer decoders restore an entire floorplan, including invisible rooms/doors.

Qualitative and quantitative evaluations demonstrate the effectiveness of our approach over the existing techniques. We will share our code, and data.

2 Related Work

We review related work in 1) floorplan reconstruction, 2) content hallucination, and 3) extreme pose estimation.

Floorplan reconstruction: Floorplan reconstruction has a long history in computer vision with strong ties to the real estate and construction industries. With the success of commodity depth sensors, current methods reconstruct floorplans from 3D point clouds, often via a combination of deep neural networks and optimization [15, 18, 20, 23]. The emergence of panorama cameras meets the growing demand for floorplan reconstruction from images alone [4]. The Zillow indoor dataset provides panorama images and ground-truth floorplans obtained with manual interactions [7]. Although floorplan reconstruction has been extensively studied in recent years, there is still significant demand among companies such as Ricoh, Zillow, and MagicPlan [1] for floorplan restoration from real production data collected by real users, despite the incompleteness and sparse coverage of such data. However, most of these companies rely on human annotators to perform the task. This paper presents a novel contribution by addressing the challenge of fully automated floorplan restoration from real production data collected by real users, despite its incompleteness and sparse coverage.

Content hallucination: Techniques such as image inpainting, scene completion, image restoration, and object removal has created realistic structures and textures in missing areas [3, 5, 9, 10, 13, 31, 33]. Neural illumination has been used to infer a spherical illumination image from a single perspective image, while implicit neural surface representation can infer an entire object surface from a single image [21, 28]. Moreover, Room layout estimation often infers invisible wall layouts behind objects [17, 24, 29]. This paper infers invisible architectural structures at a house scale (i.e., rooms and doors).

Extreme pose estimation: Given two RGB-D images with little to no visual overlaps, relative pose estimation is possible by inferring and aligning complete scene structures [30]. Room-layout estimations are registered to complete a floorplan [14]. Extreme structure from motion (SfM) algorithm estimates the camera poses of panorama images with little to no visual overlaps by learning spatial arrangements of architectural components [19, ?]. Instead of camera pose estimation, this paper takes aligned panorama images as a partial floorplan model and reconstructs a complete floorplan, including invisible rooms and doors.

3 Floorplan Restoration Problem

Input: The input is a partial floorplan reconstruction, consisting of rooms and doors, each of which is a 14-channel 800×800 segmentation image in a top-down view. The raw input data is a set of panorama images in an equirectangular projection with a camera pose, a room layout, door/window detections, and a room type. Panoramas are acquired with a mono-pod and have roughly the same height from the floor, which allows us to produce an input partial floorplan as a 14-channel image from room layout estimations and camera poses. There are 10 room types (living room, kitchen, western style room, bathroom, balcony, corridor, Japanese style room, washroom, toilet, and closet) and 4 door types (standard-door, entrance-
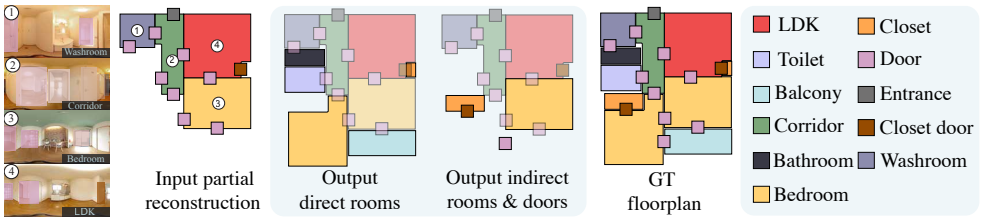
Figure 1: An extreme floorplan reconstruction dataset consists of 701 single-floor apartments/houses with ground-truth vector-graphics floorplan images. The technical challenge lies in the reconstruction of invisible rooms and doors.

door, closet-door, and open-portal). A door is represented as a 2x2 pixel region by the annotators. For the extreme-SfM reconstructions, we identify the door center and replace it by a 2x2 pixel region. Input data come from an extreme Structure from Motion algorithm [19] or human annotators. At testing, we fit an axis-aligned bounding box to the room/door masks in a 14-channel image, uniformly scale to fit at the center of a 200×200 square, then add a padding of 300 pixels around, resulting in an 800×800 image. At training, we use only ground-truth (GT) samples. We uniformly scale an image to fit at the center of a 100×100 square and add a padding of 76 pixels all around to make a 256×256 image. We apply the same augmentation process in DETR [5] (i.e., cropping and resizing), followed by 1) 50% chance of horizontal flipping and 2) 50% chance of rotation by 90 or -90 degrees.

Output: The output is a complete floorplan in a similar format as the input (i.e., a component-wise 800×800 raster mask). Figure 1 shows a sample, where the house has 10 rooms and 12 doors. 10 doors are visible and given in the input, leaving 2 "invisible doors" to be reconstructed. 4 rooms are visible, leaving 6 "invisible rooms" to be reconstructed; 1) 5 of which are adjacent to the input reconstruction and are dubbed "direct invisible rooms"; and 2) the last of which is dubbed "indirect invisible room" (*e.g.*, an invisible closet in an invisible room). Note that the output of our neural network is a raster floorplan image, which is converted to a vector-graphics floorplan by post-processing.

Dataset: The dataset consists of 701 houses with 2,355 panorama images captured by Ricoh Theta series in a production pipeline. The number of panoramas per house ranges from 1 to 7. Each house has a GT floorplan image, converted to a vector format by manual annotation. Input partial floorplans are inferred by extreme Structure-from-Motion system [19] or created from the GT floorplans (i.e., dropping rooms that do not contain panorama centers). Randomly sampled 651 houses are used for training while 50 houses are used for testing. To evaluate the robustness across different datasets, we created a synthetic one from the widely-used RPLAN dataset [26], while dropping some rooms/doors. We refer to the supplementary for more details on the datasets and comparison with other available datasets.

Metrics: We use precision/recall metrics of a floorplan reconstruction paper [6] for missing rooms. [1] When the input is a damaged GT, there is no need to align reconstructions. We match reconstructed component-wise room masks with the GT, and calculate the precision/recall. We declare that a reconstructed room matches a GT, when the room types match, and the intersection over union (IoU) score is above 0.7, we provided result for other thresholds in the supplementary material. A GT room is matched at most once. In practice, we greedily find matches by: 1) Identifying the match with the highest IoU score; 2) Removing

---

[1]We borrow the room metrics but not the corner/edge metrics that are a bit too harsh in our setting.
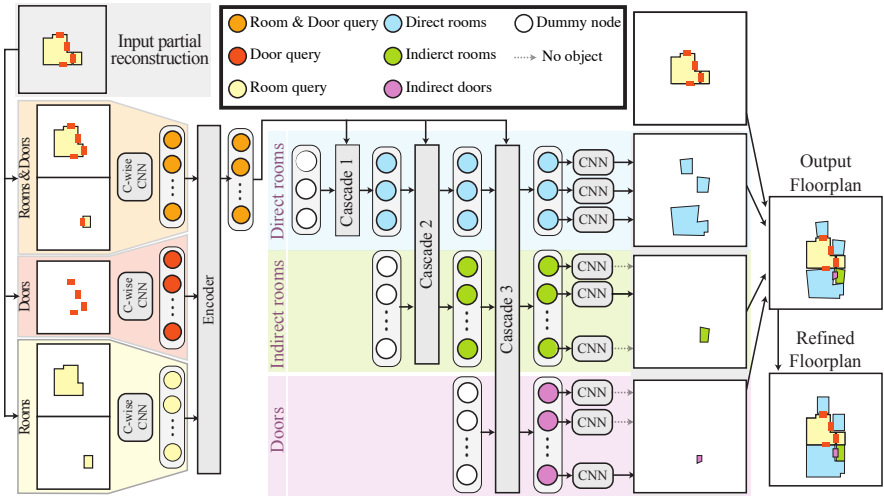
Figure 2: System overview. Category-wise CNNs with a Transformer block encodes an input partial reconstruction into a set of embedding vectors. The cascading Transformer decoders reconstruct invisible rooms and doors in three steps

the matched pair; and 3) Repeat. The process is exactly the same for the door metrics except that the IoU threshold is 0.5, because door segments are small. When the input results from the extreme-SfM system, we need to align the reconstruction with the GT before calculating the metric. We exhaustively try all possible translations (at the granularity of a pixel in a top-down image space), then use the result with the best F1 score. We also have reported LPIPs [55] and FID [12] in ther supplements.

4 Neural Floorplan Restoration

Our end-to-end neural architecture consists of a CNN/Transformer based encoder and cascading Transformer decoders. The architecture takes a partial reconstruction and produces a complete floorplan as component-wise raster segmentation masks, which is refined to a vector-graphics floorplan (See Fig. 2). The section explains the key ideas and the design choices. We refer to the supplementary for the full architecture specification.

4.1 Visible room/door encoder

Convolutional neural networks (CNNs) with a Transformer encode an input partial reconstruction as a set of $(W \times H \times 14)$ images into a $(\frac{W}{32} \times \frac{H}{32} \times 256)$ feature map in two steps. At testing, $W = H = 800$. At training, W and H depend on the augmentation and are around 600. In the first step, a standard Res-Net [11] (Res-Net-50) processes input images per architectural component category in three branches (i.e., room, door, or both). For example, in the room-branch, we feed each room-image to Res-Net, take the last layer of the last conv-block $(\frac{W}{32} \times \frac{H}{32} \times 2048)$, and apply a 1x1 convolution to change the depth to 256. Element-wise maximum across all the rooms results in a $(\frac{W}{32} \times \frac{H}{32} \times 256)$ feature map.

We treat the output as $(\frac{W}{32} \times \frac{H}{32})$ tokens with 256-dim embedding for the next Transformer block. ResNet produces the same number of tokens from the other two branches. The both-branch is the same as the room-branch except that the input image of a room contains masks of its incident doors. In the door-branch, the input is a $(W \times H \times 14)$ image containing all the door masks (instance unaware). In the second step, a self-attention block from standard Transformer [25] (6 layers w/ 8 heads) takes the tokens from the three branches.

Following the work by Zhou et al. [36], we add standard frequency position encoding to distinguish the branch-type and the X/Y position in the feature map:

$$\overrightarrow{f_{xy}} \leftarrow \overrightarrow{f_{xy}} + \left[\overrightarrow{P}_{128}(x), \overrightarrow{P}_{128}(y) \right] + \overrightarrow{P}_{256}(type) \tag{1}$$

$$\overrightarrow{P}_d(t) = \left[\{\cos(10^{8i/d}t)\}, \{\sin(10^{8i/d}t)\} \right], \ (i = 1, 2, \cdots d/2). \tag{2}$$

$\overrightarrow{f_{xy}}$ denotes the 256-dim feature vector of a token at position (x, y), where $x \in \left[1, \frac{W}{32}\right]$ and $y \in \left[1, \frac{H}{32}\right]$. $\overrightarrow{P}_d(t)$ is a d-dimensional standard frequency position encoding. $(type)$ is a scalar, indicating the branch-type (both=1001, room=1002, and door=1003). The tokens from the "both" branch are passed to the decoder.

4.2 Invisible room/door cascading decoders

Three cascaded transformer decoders reconstruct direct invisible rooms, indirect invisible rooms, and invisible doors (See Fig. 1).

Direct invisible room decoder: Direct invisible rooms are behind the doors detected in the panoramas. Let $N_{vis \cdot d}$ be the number of detected doors, which is the expected number of direct invisible rooms. Following DETR [5], we pass $N_{vis \cdot d}$ query tokens with learnable embeddings to a self-attention block, while feeding the tokens from the encoder via cross-attention. Each query token will contain an embedding of a direct invisible room to be reconstructed. The number of detected doors (i.e., query embeddings) varies. We prepare 20 embedding vectors, which is large enough during training. An output embedding at a query token is used to 1) classify the room-type as a 15-dim vector by a fully connected layer with soft-max; 2) regress the bounding box parameters (i.e., the center, the width, and the height) in the normalized image coordinate $(x, y \in [0, 1])$ by a 3-layer MLP with ReLU (hidden dimension 256); and 3) estimate a binary segmentation mask by the panoptic segmentation head in DETR [5]. Note that the type classification labels consist of 10 room types, 4 door types, and "no room/door". The last label indicates that nothing is reconstructed. At testing, we use the category with the greatest value and keep pixels above the positive value for segmentation. The bounding box is used for a loss during training, but is not during testing.

Indirect invisible room decoder: The second cascaded decoder reconstructs indirect invisible rooms via query tokens while passing the encoder tokens via cross-attention in the same way as the first cascade. The only difference is that the self-attention block also incorporates the direct room tokens from the first cascade. Both the direct and the indirect room tokens are used to predict the room-type, the bounding box parameters, and the segmentation mask by exactly the same network modules with the same loss functions. We assume 15 indirect invisible rooms at maximum and pass 15 query tokens with learnable embedding.

Invisible door decoder: The third decoder reconstructs invisible doors via query tokens in the same way. The difference is that self-attention incorporates all the tokens (direct rooms, indirect rooms, and doors). A complete floorplan is reconstructed after the cascade.

4.3 Loss Functions

Our neutral network reconstructs a floorplan in three cascades. In each cascade, we take the reconstructed components, match them with the corresponding ground-truth components, and inject loss functions. The matching is done in exactly the same way as DETR [5], based on the type classification and the bounding box estimation. The first cascade is for invisible direct rooms, and we match against only the GT invisible direct rooms. For the second cascade, we match against the GT invisible direct/indirect rooms. For the third cascade, we match against the GT invisible direct/indirect rooms and invisible doors. We descried the loss fuctions in more details in the supplements.

4.4 Floorplan refinement and vectorization

Cascading decoders reconstruct a floorplan as raster images, where room boundaries are curvy, and door shapes are irregular. We use a floorplan generative model (House-GAN++[16] by Nauata *et al.*) to refine our output and convert to a vector format. House-GAN++ is designed to produce floorplans from noise vectors, but can also be used to refine a design without changing the overall arrangement by specifying an entire floorplan as an input constraint. House-GAN++ learns a bias to prefer straight shape boundaries and we use the same post-processing in the original work to produce a vector graphics floorplan. Note that House-GAN++ alone, without our constraint fails to infer an accurate floorplan, as shown in the experimental results next. We refer to the supplementary for more details.

5 Experimental Results

We have implemented the proposed system using PyTorch 1.10.0 and Python 3.9.7, and used a workstation with a 3.70GHz Intel i9-10900X CPU (20 cores) and an NVIDIA RTX A6000 GPU. To accommodate the varying number of visible doors/rooms, we use a batch size of 1 during training, while accumulating gradients over 16 samples for a parameter update. We use AdamW optimizer, while setting the learning rate to 10^{-5} for the CNN module in the encoder and 10^{-4} for the rest of the network (i.e., a Transformer block in the encoder and the three cascading Transformer decoders). We divide the learning rate by 10 after every 100 epochs. We train for 240 epochs, which takes roughly 22 hours with the workstation.

Competing methods: We compare against the four competing methods.
• The first one is one of the state of the art inpainting methods "MAT" with a official of implementation [13].
• The second one is Mask-RCNN with an official implementation from Meta AI Research [27].
• The third one is Housegan++ [16], requiring a full bubble diagram where rooms are nodes in that bubble diagram to produce the house layout. However, Housegan++ is not able to predict missing rooms in our problem. We use our model which works the best in missing room prediction to predict missing rooms and create input bubble diagrams for Housegan++. In all iterations we pass visible rooms segmentation masks to Housegan++ and from second iteration we also pass 50% of invisible room predicted segmentation mask from previous iteration to the network too. We continue iterations until 10th iteration.
• The fourth one is DETR via an official implementation [5].

These methods take a single image as an input, while our input is a set of 14-channel images (instance-aware). We perform pixel-wise max-pooling over the images and produce a single

14-channel image as their input. Our output is a set of architectural components, each of which is a binary segmentation mask (i.e., a probability distribution over the room/door types) and the bounding box parameters, which Mask-RCNN and DETR directly produce.

For MAT, the output is a single instance-unaware 14-channel image. We use flood fill algorithm to find connected components, and discard small components whose areas are less than 4 pixels to produce a floorplan. For Mask-RCNN, we set a threshold of 0.6 on the confidence prediction as that was giving the best performance. We vary the parameter and pick the one with the highest average F1 score. DETR does not require a threshold: A room/door is not generated when the probability of type "no room/door" is the greatest.

For fair evaluation, we have used 1) the same category based weighing in Sect. 4.3 for the type classification loss for Mask-RCNN and DETR; 2) the same parameter update schedule (once every 16 samples); 3) the same data augmentation steps; and 4) the same learning rate.

Table 1: The main quantitative evaluations. We compare against MAT, Mask-RCNN (M_{rcnn}), DETR, and Housegan++ (H-PP) Input partial floorplans are either ground-truth (left) or outputs from the extreme SfM system [19] (right), * means refinement .

| Method | Manual (GT) | | | | | | Extreme-SfM output | | | | | |
| | Rooms | | | Doors | | | Rooms | | | Doors | | |
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
MAT	20.3	43.2	27.6	10.6	11.3	11.0	18.2	36.5	24.2	10.7	9.1	9.8
M_{rcnn}	38.1	24.2	29.6	12.0	13.4	12.7	37.5	23.2	28.6	12.3	11.4	11.8
H-PP	33.7	33.1	33.4	10.3	9.9	10.1	25.0	24.4	24.7	12.6	12.2	12.4
DETR	30.6	47.3	37.1	13.0	13.5	13.2	27.1	41.5	32.8	11.0	13.9	12.3
Ours	49.2	50.8	50.0	20.2	19.3	19.7	37.1	41.0	38.9	14.0	18.4	15.9
Ours*	56.2	53.1	54.6	21.0	20.4	21.1	40.6	44.9	42.6	15.6	19.1	17.1

Table 2: Quantitative evaluations on larger synthetic dataset, RPLAN [26]. We compare against MAT, Mask-RCNN (M_{rcnn}), House-GAN++, and DETR. Input partial floorplans are ground-truth. The precision, the recall, and the F1 score are reported for the rooms and the doors.

| Method | Rooms | | | Doors | | |
	Pre.	Rec.	F1	Pre.	Rec.	F1
MAT	43.5	52.4	47.5	10.1	13.5	11.5
M_{rcnn}	46.4	37.6	41.5	11.5	10.3	10.8
House-GAN++	47.6	45.3	46.4	15.8	12.7	14.0
DETR	50.5	52.1	51.3	15.4	16.3	15.8
Ours	64.7	51.3	57.2	20.8	18.7	19.1
Our*	65.8	53.0	58.7	22.9	22.3	22.6

Quantitative evaluations: Table 1 provides our main quantitative evaluations, comparing against three competing methods. The best, the second best, and the third best results are shown in cyan, orange, and magenta, respectively. Note that the last row is our result with the post-processing refinement step (Sect. 4.4), which is not an end-to-end system and given as a reference. Our system achieves the highest F1 score in all the settings. Table 2 presents results on larger synthetic dataset [26], where the proposed method outperforms all the other baselines. As it can be seen, shape and type information alone is not sufficient for inpainting methods to handle structured geometry. Scene graph based generative models such as Housegan++ alone need significant modifications and extra infromation such as full input
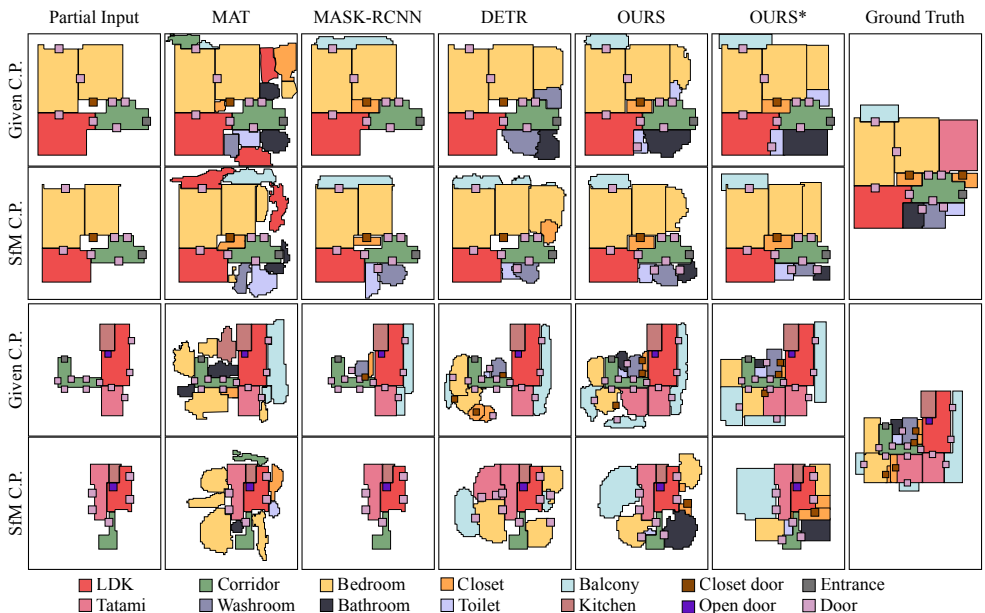
Figure 3: Main qualitative evaluations. The figure shows reconstruction results before and after the final refinement for four houses, (Ours, Ours*). For each method, we show both when the input partial reconstruction is from the ground-truth or the extreme SfM system [19]. At the right, we show both our ground-truth vector-graphics floorplans as well as the original raster floorplan images by an architect.

Table 3: Ablation study on the category-wise CNN encoder. The left three columns shows how the input partial reconstructions are passed to each of the three CNN branches. (m) indicates that a branch receives multiple images (instance-aware) as an input. (s) indicates that a branch receives a single image (instance-unaware) that contains masks of all the instances. No-mark indicates that a branch is not used.

both	room	door	Direct Rooms (First Cas.)			Rooms (Second Cas.)			Rooms (Full Pipeline)			Doors (Full Pipeline)		
			Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
s			56.3	62.6	59.3	40.8	45.1	42.8	43.8	46.5	42.8	14.8	14.5	14.6
m			67.8	63.5	65.5	34.6	43.9	38.7	35.8	48.2	41.8	6.2	7.5	6.7
m	m		63.7	62.5	63.1	35.3	44.6	39.4	36.0	48.6	41.3	7.1	6.9	7.0
m		s	60.7	61.1	60.8	47.6	45.5	46.3	40.6	47.6	43.9	14.1	15.6	14.8
m	m	m	66.2	64.7	65.8	46.7	44.9	45.8	48.4	49.2	48.8	10.6	11.2	10.9
s	s	s	57.2	54.2	55.6	37.6	39.7	38.6	38.4	40.1	39.2	17.8	18.9	18.3
m	m	s	68.2	64.2	66.1	49.6	46.6	48.0	49.2	50.8	50.0	20.2	19.3	19.7

graph to be adapted to this problem. Our Transformer cascades provide a more effective solution, performing higher-order reasoning and accurately composing layouts.

Table 3 provides an ablation study on the category-wise CNN encoder. The bottom row is our final system. In the fifth entry, the door branch receives multiple images (instance aware), which shows comparable room metrics but worse performance for doors. This suggests that doors are already cleanly separated and an instance unaware single image representation is more efficient and effective. The last two rows suggest that the room information should be

Table 4: Ablation study on the cascading decoders. We turn on and off each of the three cascades (indicated by the left columns) and retrain the entire model. The table also reports the metrics of reconstructions by the first and the second cascades. Note that the first (resp. second) cascade reconstructs only direct rooms (resp. direct and indirect rooms), respectively, whose metrics are reported.

Cascades			Direct Rooms (First Cas.)			Rooms (Second Cas.)			Rooms (Full Pipeline)			Doors (Full Pipeline)		
1st	2nd	3rd	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
		✓	x	x	x	x	x	x	36.3	41.1	38.5	14.5	14.6	14.5
	✓	✓	x	x	x	39.4	46.3	42.6	43.1	47.0	44.9	14.9	15.2	15.0
✓		✓	67.5	60.3	63.6	x	x	x	36.2	41.4	38.6	16.3	16.1	16.2
✓	✓		44.8	46.3	45.5	40.9	42.0	41.5	40.9	42.0	41.5	x	x	x
✓	✓	✓	68.2	64.2	66.1	49.6	46.6	48.2	49.2	50.8	50.0	20.2	19.3	19.7

passed as the instance-aware representation.

Table 4 provide an ablation study on the cascading decoders. The middle three rows show that dropping any of the cascades downgrades performance. While the third cascade alone (first row) produces reasonable results, the proposed system with the three cascades achieves the best result in every metric. The last two rows show an interesting phenomena where the third cascade improves the performance of the first two cascades via gradient propagation.

Qualitative evaluation: Figure 3 provides the main qualitative evaluations against Mask-RCNN and DETR. "Ours*" shows the final floorplan models after the refinement by House-GAN++, The refinement process successfully turns raw floorplan reconstructions with many artifacts (*e.g.*, curvy room boundaries and gaps between rooms) into clean floorplan models. Reconstructed floorplans by other different methods before and after the refinement are presented in supplements.

The input and the corresponding ground-truth reveal that this is a challenging reconstruction task, unlike any existing problems. The MAT method produced reasonable pixel-level output but it lacks the ability produce instance-aware output. As a consequence, the results may contain divided or missing rooms. Mask-RCNN is often capable of inferring closets or balconies that are direct neighbors of the input partial reconstruction. However, it fails to infer indirect invisible rooms or doors that are far away from the input reconstruction. With the power of the Transformer, DETR reconstructs indirect rooms and doors more than Mask-RCNN. Nonetheless, our approach infers many invisible structures successfully, in particular when manual (ground-truth) partial reconstructions are given. Please see the supplementary for more reconstruction examples.

Concluding marks: This paper presents a new floorplan restoration task, a new benchmark, and a neural architecture as a solution. The task is challenging and our results are not always satisfactory. Major failure modes are 1) missing rooms; 2) inaccurate room shapes, in particular corridors; and 3) inaccessible rooms without any doors. We hope that this paper starts an avenue of new research toward an ultimate extreme floorplan reconstruction system capable of reconstructing accurate and realistic floorplans for tens of millions of houses with sparse panorama coverage out there. We will share all our code, model, and data, except the panorama images, for privacy concerns.

References

[1] Magicplan. hhttps://www.magicplan.app.

[2] Ricoh360. https://www.ricoh360.com/tours/.

[3] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. volume 28, August 2009.

[4] 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, 2014. doi: 10.1109/CVPR.2014.546.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58452-8.

[6] Jiacheng Chen, Chen Liu, Jiaye Wu, and Yasutaka Furukawa. Floor-sp: Inverse cad for floorplans by sequential room-wise shortest path. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

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

[8] Qiaole Dong, Chenjie Cao, and Yanwei Fu. Incremental transformer structure enhanced image inpainting with masking positional encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[9] Andreas Lugmayr et al. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022.

[10] James Hays and Alexei A Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (ToG)*, 26(3):4–es, 2007.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. NIPS'17. Curran Associates Inc., 2017. ISBN 9781510860964.

[13] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[14] Cheng Lin, Changjian Li, and Wenping Wang. Floorplan-jigsaw: Jointly estimating scene layout and aligning partial scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5674–5683, 2019.

[15] Chen Liu, Jiaye Wu, and Yasutaka Furukawa. Floornet: A unified framework for floorplan reconstruction from 3d scans. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[16] Nelson Nauata, Sepidehsadat Hosseini, Kai-Hung Chang, Hang Chu, Chin-Yi Cheng, and Yasutaka Furukawa. House-gan++: Generative adversarial layout refinement network towards intelligent computational agent for professional architects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13632–13641, 2021.

[17] Giovanni Pintore, Marco Agus, and Enrico Gobbetti. AtlantaNet: Inferring the 3D indoor layout from a single 360 image beyond the Manhattan world assumption. In *Proc. ECCV*, August 2020. URL http://vic.crs4.it/vic/cgi-bin/bib-page.cgi?id='Pintore:2020:AI3'.

[18] 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. In *Computer Graphics Forum*, volume 39, pages 667–699. Wiley Online Library, 2020.

[19] Mohammad Amin Shabani, Weilian Song, Makoto Odamaki, Hirochika Fujiki, and Yasutaka Furukawa. Extreme structure from motion for indoor panoramas without visual overlaps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5703–5711, 2021.

[20] Mohammad Amin Shabani, Sepidehsadat Hosseini, and Yasutaka Furukawa. Housediffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising. *arXiv preprint arXiv:2211.13287*, 2022.

[21] Shuran Song and Thomas Funkhouser. Neural illumination: Lighting prediction for indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6918–6926, 2019.

[22] Shuran Song and Thomas A. Funkhouser. Neural illumination: Lighting prediction for indoor environments. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6911–6919, 2019.

[23] Sinisa Stekovic, Mahdi Rad, Friedrich Fraundorfer, and Vincent Lepetit. Montefloor: Extending mcts for reconstructing accurate large-scale floor plans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16034–16043, October 2021.

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

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

[26] Wenming Wu, Xiao-Ming Fu, Rui Tang, Yuhan Wang, Yu-Hao Qi, and Ligang Liu. Data-driven interior plan generation for residential buildings. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 38(6), 2019.

[27] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[28] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *Advances in Neural Information Processing Systems*, 32, 2019.

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

[30] Zhenpei Yang, Jeffrey Z Pan, Linjie Luo, Xiaowei Zhou, Kristen Grauman, and Qixing Huang. Extreme relative pose estimation for rgb-d scans via scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4531–4540, 2019.

[31] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Freeform image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4471–4480, 2019.

[32] Yingchen Yu, Fangneng Zhan, Rongliang Wu, Jianxiong Pan, Kaiwen Cui, Shijian Lu, Feiying Ma, Xuansong Xie, and Chunyan Miao. Diverse image inpainting with bidirectional and autoregressive transformers. *Proceedings of the 29th ACM International Conference on Multimedia*, 2021.

[33] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5728–5739, June 2022.

[34] Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised scene de-occlusion. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, June 2020.

[35] Haoran Zhang, Zhenzhen Hu, Changzhi Luo, Wangmeng Zuo, and Meng Wang. Semantic image inpainting with progressive generative networks. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, page 1939–1947, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356657. doi: 10.1145/3240508.3240625. URL https://doi.org/10.1145/3240508.3240625.

[36] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference*, volume 35, pages 11106–11115. AAAI Press, 2021.

[37] Yan Zhu, Yuandong Tian, Dimitris Mexatas, and Piotr Dollár. Semantic amodal segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.