IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

360-DFPE: Leveraging Monocular 360-Layouts for Direct Floor Plan Estimation

Bolivar Solarte*[1], Yueh-Cheng Liu*[1], Chin-Hsuan Wu[1], Yi-Hsuan Tsai[2], and Min Sun[1] .

***Abstract*—We present 360-DFPE, a sequential floor plan estimation method that directly takes 360-images as input without relying on active sensors or 3D information. Our approach leverages a loosely coupled integration between a monocular visual SLAM solution and a monocular 360-room layout approach, which estimate camera poses and layout geometries, respectively. Since our task is to sequentially capture the floor plan using monocular images, the entire scene structure, room instances, and room shapes are unknown. To tackle these challenges, we first handle the scale difference between visual odometry and layout geometry via formulating an entropy minimization process, which enables us to directly align 360-layouts without knowing the entire scene in advance. Second, to sequentially identify individual rooms, we propose a novel room identification algorithm that tracks every room along the camera exploration using geometry information. Lastly, to estimate the final shape of the room, we propose a shortest path algorithm with an iterative coarse-to-fine strategy, which improves prior formulations with higher accuracy and faster run-time. Moreover, we collect a new floor plan dataset with challenging large-scale scenes, providing both point clouds and sequential 360-image information. Experimental results show that our monocular solution achieves favorable performance against the current state-of-the-art algorithms that rely on active sensors and require the entire scene reconstruction data in advance. The code and dataset are available at the project page: https://github.com/EnriqueSolarte/direct_360_FPE.**

***Index Terms*—SLAM, Mapping, Omnidirectional Vision**

I. Introduction

THE underlying 3D structures of an indoor scene, such as a floor plan of a building, play a crucial role for a holistic robot perception as detailed in [1]–[6]. Despite their simplicity, these high-level geometry abstractions can complement challenging tasks such as obstacle avoidance [3], robot localization [4], path planning [5], and scene understanding [6]; hence, a handy and direct estimation of a floor plan geometry is our primary motivation.

Current state-of-the-art solutions for floor plan estimation [7]–[9] rely on dense point clouds as input, which require active sensors (e.g., LiDAR, depth cameras) for data collection and pre-processing steps beforehand; hence, a direct estimation from a sequence of observation is not supported. On the other hand, approaches that rely only on imagery [10], [11] generally leverage structure-from-motion (SfM), multi-view stereo (MVS), and pixel semantic estimations to project geometric clues into 3D/2D space from where the floor plan is inferred. However, these solutions heavily depend on the estimated point cloud's quality and sparsity. Additionally, they require the whole data in advance to successfully assess multiple rooms in the scene.

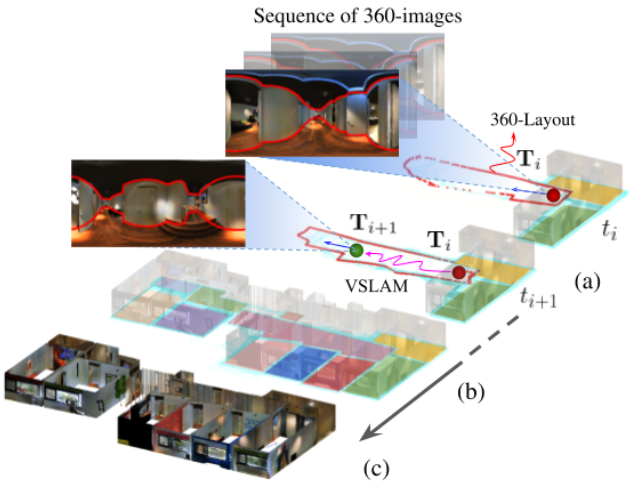


Fig. 1. **360-DFPE: Direct Floor Plan Estimation using sequence of 360-images.** Our goal is to sequentially estimate the floor plan by aggregating the 360 layout estimations (red-line) directly from the images. (a) We illustrate the process of 360-DFPE under two camera positions ($\mathbf{T}_i$, $\mathbf{T}_{i+1}$) at consecutive times (t_i, t_{i+1}). (b) A complete estimation of the floor plan is presented as multiple room shapes (denoted by different colors). (c) A visually pleasing 3D scene aligned with the estimated floor plan is rendered using the 360-images as texture. Note that only a passive 360-camera is required and the process can be completed in a sequential fashion.

Other solutions [12]–[14] leverage Visual Simultaneous Localization and Mapping (VSLAM) by estimating the 3D scene structure using point, line, plane, and depth features from a sequence of images. However, this scene structure is usually represented by a set of piece-wise planes that include all furniture and texture details, without clearly defining room layouts or multiple room instances; hence, it cannot directly represent a floor plan structure. To the best of our knowledge, an approach that directly uses images to estimate the floor plan geometry sequentially has not been addressed yet.

One way to utilize sequential monocular images for floor plan estimation is to directly apply room layout estimation methods [15]–[21]. However, even by leveraging large field-of-view images, i.e., 360-images, some issues remain unsolved. For example, the room layout estimation may differ with different camera positions, or it may fail to capture large rooms or corridors. Furthermore, in the context of a monocular frame-

Manuscript received: December 11, 2021; Revised: March 28, 2022; Accepted: April 22, 2022.

This paper was recommended for publication by Editor Javier Civera upon evaluation of the Associate Editor and Reviewers' comments.

Digital Object Identifier (DOI): see top of this page.

[1] Bolivar Solarte, Yueh-Cheng Liu, Chin-Hsuan Wu, and Min Sun are with National Tsing Hua University enrique.solarte.pardo@gamil.com

[2] Yi-Hsuan Tsai is with Phiar Technologies.

* The authors contribute equally to this paper.

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

work, the scale difference between estimated camera poses and the layout projection is a challenge for reconstructing the floor plan using multiple layout estimations directly.

In this paper, we present a direct floor plan estimation algorithm (360-DFPE) that does not rely on active sensors but directly takes a sequence of 360-images as input. In contrast to previous approaches, we value the ability of sequential reconstruction as an application. Therefore, we design a novel pipeline that registers multiple 360-layout geometries frame-by-frame and reconstructs the floor plan room-by-room. In Fig. 1, we exemplify our solution with estimated 360-layouts and camera poses at two timestamps.

Our solution leverages a VSLAM system [22] and a 360-layout estimation algorithm [20] to estimate camera poses and 360-layout geometries, using both off-the-shelf implementations. To handle the scale difference between estimated camera poses and 360-layouts, we use a few initial keyframes to recover the unknown relative scale at the beginning of our system. Additionally, we propose a novel sequential procedure that identifies room instances without knowing the entire scene in advance. Lastly, we filter out the invalid layout geometries and optimize the final room shape in an iterative coarse-to-fine fashion.

Since existing floor plan estimation datasets [7], [8] do not provide sequential frame information, we collect a new dataset, MP3D-FPE, using the Matterport3D dataset [23] rendered by the Minos [24] framework. Our experimental results show favorable performance against current state-of-the-art floor plan estimation solutions, i.e., FloorNet [8] and Floor-SP [7], despite the fact that these methods use registered point clouds of the entire scene captured from active sensors.

Our main contribution are the follows:

- We propose 360-DFPE which aggregates 360-layout geometries and estimates the floor plan sequentially under a monocular framework.
- We propose an entropy optimization method that directly regresses the visual odometry scale relative to layout geometry.
- We design a sequential room identification procedure that tracks the corresponding room for every keyframe using only geometric cues.
- We propose an iterative coarse-to-fine Shortest Path Algorithm (iSPA) to estimate the room shape, which advances previous formulations [7], [10] with faster speed and better accuracy.
- Our experimental results on our newly collected MP3D-FPE dataset show that our monocular 360-DFPE outperforms state-of-the-art methods that require the point cloud of the entire scene as input.

II. Related Work

A. Single Room Layout Estimation

Early works in room layout estimation [25]–[28] mainly leverage hand-engineering features on images (e.g., vanishing points, detected objects, and textured regions) to estimate wall-surfaces and wall-contours. In particular, [26] defines the room-layout problem as boundary estimation using dominant surface orientations inferred from estimated vanished points, with a predefined room geometry to identify occlusions. Revisiting [26], [27] uses additional sensor modalities to aggregate spatial geometry reasoning. In [28], multiple room boundary hypotheses are aggregated using a Bayesian formulation to achieve real-time room layout reconstruction. Later, [25] further considers other cues in the image, such as people, objects, and their geometry relationship. However, all these approaches are constrained by the narrow FoV of the pinhole cameras and usually rely on the assumption that at least two walls are presented in an image.

With panorama images (i.e., 360-images), pioneered works like [29]–[31] leverage the whole FoV of a scene by projecting multiple 3D room layout hypotheses, estimating vanishing points [29], room boundaries [30], and semantic pixels [31]. However, this 3D projections usually assume some prior geometry constraints such as the camera height, room shape, or the size of some detected objects.

Deep learning approaches like [17], [32]–[34] robustly estimate room-layout geometry by defining the underlying problem as semantic segmentation, corner junction, or edge map estimation. However, they require a predefined topology to infer the final room-shape structure. Current state-of-the-art solutions [18]–[21] directly estimate the room layout boundary from a single 360-image. For instance, HorizonNet [20] and HoHoNet [21] regress the ceiling-wall, and floor-wall boundaries, assuming that a vertically aligned image can be encoded as a sequence of image-column data. On the other hand, AtlantaNet [19], and LED^2-Net [18], additionally constraint the problem by adding information from the layout projected into a 2D map.

B. Floor Plan Estimation

Estimating a floor plan geometry can only be accomplished by using multiple measurements at different scene locations. To this end, early works like [10], [11] leverage a standard SfM, MVS, and pixel semantic segmentation to project labeled 3D points into a 2D map. Upon this 2D projection, [10] formulates the floor plan reconstruction as a shortest path problem, and [11] formulates it as non-linear optimization, using projected floor and ceiling pixels as internal space of the room, and wall pixels as a room boundary evidence. However, these methods rely on the quality and sparsity of the projected point cloud, which may vary for scenes with poor textures.

The most stable and robust solutions for floor plan estimation [7]–[9], [35], [36] mainly rely on active sensors (e.g., LiDAR or depth cameras) to directly acquire 3D information of the scene. For instance, early works such as [35] leverage multiple 3D LiDAR-based scans by estimating primitive geometries from a layered-up formulation. In contrast, works like [8], [36] define the problem by acquiring 3D data from dense depth images. Floorplan-Jigsaw [36] constraints the problem as a set of data with small overlapping, searching for the most coherent alignment in 2D; FloorNet [8] estimates low-level geometry with multiple Convolutional Neural Network branches and aggregates the outputs with integer programming, assuming a predefined room-shape.

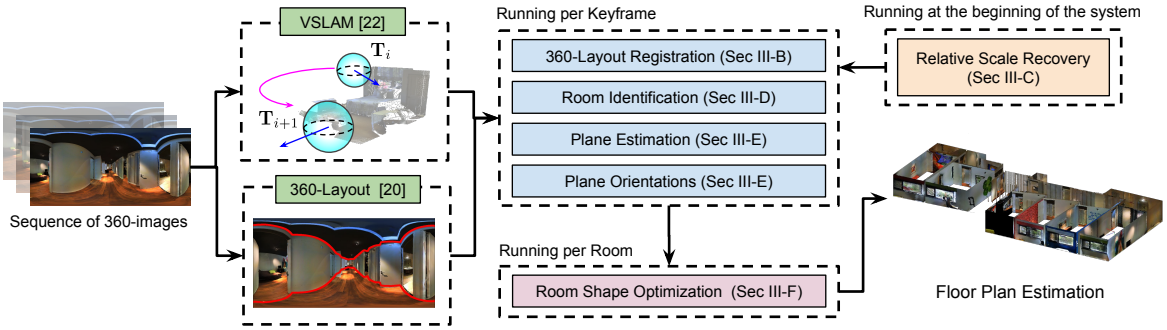


Fig. 2. **System Pipeline**. Given a sequence of monocular 360-images, our pipeline starts with a visual SLAM and an 360-layout estimation using off-the-shelf implementations. We use the initial few keyframes to recover the unknown relative scale (Section III-C). Then, for every keyframe, we sequentially apply 360-layout registration (Section III-B), room identification (Section III-D), plane estimation and plane orientations filtering (Section III-E). Once we detect that a room is completed, we apply the room shape optimization (Section III-F) to estimate the room shape.

Current state-of-the-art solutions [7], [9] assume that the whole scene has been scanned and registered into a 3D point cloud as input, which is later projected into a 2D density map. From this 2D map, Floor-SP [7] views the floor plan reconstruction as multiple single room estimations, identifying room instances with Mask R-CNN [37], then formulating the room geometry estimation as the shortest path problem similar to [10]. Recently, MonteFloor [9] adopts the pipeline similar to Floor-SP and introduce Monte Carlo Tree Search (MCTS) for searching and evaluating estimated room candidates.

III. Method

A. Overview

We present 360-DFPE, a direct floor plan estimation algorithm that takes a sequence of 360-images as input to estimate a floor plan geometry sequentially. For illustration purposes, our pipeline is presented in Fig. 2, where our proposed solution starts with estimating cameras poses and 360-layouts using [20], [22] with a loosely-coupled integration.

Since both mentioned methods are monocular solutions, their estimations are defined up to different and unknown scales. Therefore, at the beginning of the system, a relative scale recovery estimates the missing odometry scale with respect to the layout geometries (see Section III-C).

Then, running every keyframe, the estimated layout is aligned using VSLAM-based camera poses by a 360-Layout Registration (see Section III-B). Next, to identify every room in the scene, an identification algorithm tracks the current camera location sequentially (see Section III-D). In addition, each estimated layout is decomposed into a set of plane geometries to remove noisy estimations. Upon these plane geometries, we also compute the plane orientations of each room (see Section III-E). Lastly, once the camera leaves each room, a room shape optimization procedure estimates the room geometry as a set of corners, which vectorizes each room and the whole floor plan structure (see Section III-F).

B. 360-Layout Registration

The underlying problem in this section is to register multiple 360-layout estimations using VSLAM-based camera poses, where both are defined under unknown and different scales. For this purpose, we first define a layout boundary as a set of 3D points, which are projected from a 360-layout estimation [20], defined by W points on an equirectangular image with size $H \times W$. A point $\mathbf{x}$ of this layout boundary is defined as follows:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{h}{\sin\phi}\cos\phi\sin\theta \\ -h \\ \frac{h}{\sin\phi}\cos\phi\cos\theta \end{bmatrix}, \tag{1}$$

where h is the current camera height (usually assumed or measured in advance [10], [11], [18], [20]), and (θ, ϕ) is its spherical coordinate on the equirectangular image.

We represent the layout boundary of the i-th keyframe as the set of points $\mathbf{P}_i = \{\mathbf{x}_1, \cdots, \mathbf{x}_W\}$. Additionally, we define the first estimated camera pose as the world coordinate references and set its distance to the floor (i.e., camera height) to an arbitrary constant value, i.e., $h = 1$, which defines the layout projection scale.

To define the registration of $\mathbf{P}_i$ into the world coordinate, we leverage its VSLAM-based camera pose as follows:

$$\mathbf{P}_i^W = \{\mathbf{R}_i \cdot \mathbf{x} + s\ \mathbf{t}_i \mid \forall \mathbf{x} \in \mathbf{P}_i\}, \tag{2}$$

where $\mathbf{T}_i = [\mathbf{R}_i|\mathbf{t}_i] \in SE(3)$ is the estimated camera pose with rotation and translation, and s is the unknown visual odometry scale for a monocular VSLAM solution [22], [38]. This visual odometry scale and the layout projection scale are not necessarily equal. Note that by changing the scale s in (2), we can modify the translation of $\mathbf{P}_i$ in the world coordinate. For reference purpose, in Fig. 3-(a), the registration of two layout boundaries are illustrated.

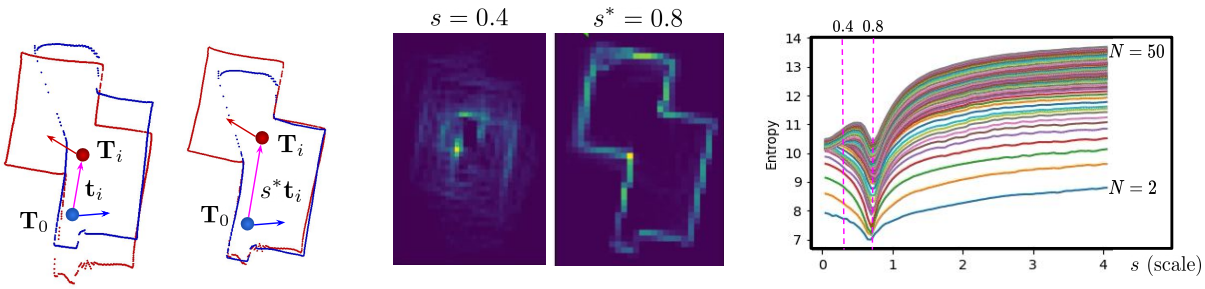
C. Relative Scale Recovery

To recover the unknown visual odometry scale s relative to the 360-layout estimates, we aggregate a sequence of registered layout boundaries and optimize over s, constrained under our predefined layout projection scale (i.e., $h = 1$, see Section III-B). We formulate an optimization problem with N layout boundaries as a function of s:

$$F_s = \Phi(\mathbf{P}_0^W \cup \cdots \cup \mathbf{P}_{N-1}^W) \tag{3}$$

$$s^* = \underset{s}{\operatorname{argmin}} \sum_{u,v} -F_s(u,v) \cdot \log F_s(u,v), \tag{4}$$

where $\Phi(*)$ is a top-down projection function that computes a 2D normalized histogram for all the input points, defining



(a) Example of Two Layouts (b) Density Map with Multiple Layouts (c) Entropy Minimization

Fig. 3. **360-Layout Registration with Relative Scale Recovery.** (a) We illustrate registering two 360-layout boundaries (red and blue lines) from layout estimations by (2). By scaling the relative translation $\mathbf{t}_i$ with s^*, we can align them properly. (b) We visualize multiple layout estimations with 2D density maps using different s values, e.g., 0.4 and 0.8. The density map using $s^* = 0.8$ will yield a lower entropy due to better alignment. (c) We plot the entropy values under different scale s using different number of layouts N. Note that we can find the optimal relative scale value for the scene through entropy minimization. See Section III-C for more details.

Algorithm 1 Room Identification

Input: New camera pose $\mathbf{T}_i = [\mathbf{R}_i|\mathbf{t}_i]$ and a set of room density maps $\mathbf{H} = \{H_j\}_{j=1}^{m}$

$\mathbf{Q}_i \leftarrow$ Bounded 2D area enclosed by $\mathbf{B}_i$ using (5)
$(u_i, v_i) \leftarrow$ 2D projected coordinate of $s^*\ \mathbf{t}_i$
if $H_m(u_i, v_i) \geq 0.5$ **then**
  $H_m \leftarrow H_m + \Phi(\mathbf{Q}_i)$ ▷ In current room m
  $H_m \leftarrow H_m\ /\ \max_{u,v} H_m(u, v)$
else if $\max_{j<m} H_j(u_i, v_i) \geq 0.5$ **then**
  $H_j \leftarrow H_j + \Phi(\mathbf{Q}_i)$ ▷ In previous rooms j
  $H_j \leftarrow H_j\ /\ \max_{u,v} H_j(u, v)$
else
  $H_{m+1} \leftarrow \Phi(\mathbf{Q}_i)$ ▷ Create a new room $m+1$
  $H_{m+1} \leftarrow H_{m+1}\ /\ \max_{u,v} H_{m+1}(u, v)$
  $\mathbf{H} \leftarrow \mathbf{H} + \{H_{m+1}\}$
end if

the 2D probabilistic density map F_s, and (4) defines an entropy minimization for s. The intuition is that only when s is correctly estimated, the layout boundaries in multiple consecutive time frames can be well-aligned on the 2D density map, which will yield the lowest entropy value.

To solve (4), we apply a linear search in a predefined range as studied in [39]. Ideally, using all the keyframes will bring the optimal s^* for the scene. However, considering the trade-off between data usage and performance, we devise a warm-up stage in the beginning of the system that takes the initial 20% keyframes to recover a stable s^*. We use a 3-level coarse-to-fine search with $\{0.5, 0.1, 0.01\}$ as the step size and a sliding-window with size $N = 10$.

In Fig. 3-(b), we illustrate the density function F_s, showing how s^* defines a better alignment of multiple layout geometries; Fig. 3-(c) shows that the minimum entropy value is at the optimal s^* scale under different sliding-window sizes.

D. Room Identification

In a sequential framework, the entire scene structure is unknown in advance. Therefore, we formulate the problem of identifying room instances as tracking the belonging room for every incoming keyframe. We propose using the layout geometries to construct a 2D room density function that evaluates the likelihood of a camera position in the room.

Our assumption is that the layout boundaries should change smoothly along the camera trajectory unless the camera crosses the room edge. Hence, to build the density function H, we sequentially aggregate the 2D area inside the clipped layout boundary $\mathbf{B}_i$ around every camera position. This boundary is defined as follows:

$$\mathbf{B}_i = \{\mathbf{R}_i \cdot \rho_r(\mathbf{x}) + s^*\ \mathbf{t}_i \mid \forall \mathbf{x} \in \mathbf{P}_i\}, \tag{5}$$

$$\rho_r(\mathbf{x}) = \begin{cases} \mathbf{x}, & \text{if } \|\mathbf{x}\| \leq r \\ r\frac{\mathbf{x}}{\|\mathbf{x}\|}, & \text{otherwise.} \end{cases} \tag{6}$$

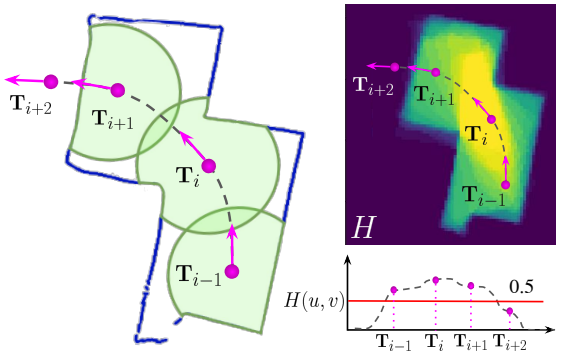
where $\rho_r(\mathbf{x})$ is a clipping function with a radius r in order to avoid over-trusting the original layout boundary $\mathbf{P}_i$, especially those regions far from the camera.

The proposed room identification procedure is detailed in Algorithm 1. Given an input keyframe with its camera pose $\mathbf{T}_i$, we first determine if the camera position is inside the current room or whether it is entering one of the previously visited rooms, using the H function of each room. If one condition is satisfied, we update the belonging room's density function with the area bounded by $\mathbf{B}_i$. Otherwise, we will trigger room creation and initialize a new density map.

For illustration purpose, in Fig. 4-(a), we present a sequence of keyframes with their corresponding clipped layout boundary $\mathbf{B}_i$. In Fig. 4-(b), we depict the density value $H(u, v)$ for several keyframes. Note that the keyframe at $\mathbf{T}_{i+2}$ is identified outside the room since its evaluation in $H(u, v)$ is lower than a predefined threshold. In our experiments, we find that setting this threshold equal to 0.5 strikes a good balance between false negatives and false positives for room identification.

E. Plane Estimation from Layout Geometry

Considering that a 360-layout estimation may contain noise and even invalid geometries, we decompose every set $\mathbf{P}_i$ (i.e, a layout boundary, see Section III-C) into subsets of points $\mathbf{S}_0 \cup \cdots \cup \mathbf{S}_k = \mathbf{P}_i$ to identify and remove the undesired geometry regions. In our implementation, we directly leverage the estimated wall corners from [20], splitting $\mathbf{P}_i$ into several subsets such that each $\mathbf{S}_k$ represents a wall, namely the plane-wall feature.



(a) Clipped Layout Boundary (b) 2D Density Function

Fig. 4. **Room Identification.** (a) The green circles are the clipped layout boundaries (see (5)). To track if the camera is inside a room, we sequentially aggregate them and build a 2D density function for the room. (b) If the new camera pose $\mathbf{T}_{i+2}$ has a density value lower than the threshold 0.5, we consider that the camera has crossed the room boundary.

For every plane-wall feature, we formulate a plane estimation problem by using RANSAC [40]. To this end, we sample points from $\mathbf{S}_k$ to compute a plane geometry $(\mathbf{n}, d)$, where $\mathbf{n}$ is the normal vector of the plane, and d is the distance value measured from the camera to the plane. Then, every estimated plane is evaluated up to a maximum residual error equal to 3×10^{-2}, where every subset $\mathbf{S}_k$ that describes an inliers ratio lower than 0.9 is removed as a reliable geometry evidence. These hyper-parameters were chosen empirically.

In addition, we estimate the likely plane orientations for each room with Bayesian filtering, which is later used in the room shape optimization as a geometry constraint (see Section III-F). This filtering formulation is presented as:

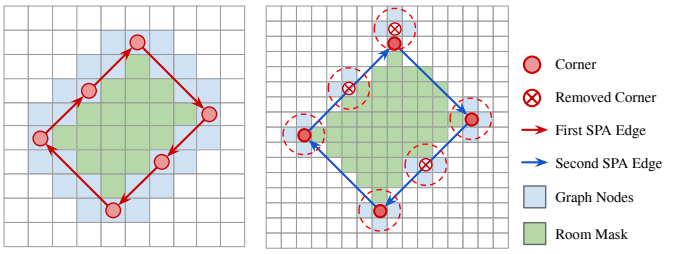
$$\begin{aligned} P(\hat{\theta}_n, \hat{\sigma}_\theta) &= \eta \cdot P(\hat{\theta}_n, \hat{\sigma}_\theta)\ \mathcal{N}(\theta_n, \sigma_\theta), \\ \sigma_\theta &= \sigma_0 + \lambda \cdot d, \end{aligned} \tag{7}$$

where $\mathcal{N}(\theta_n, \sigma_\theta)$ is a normal distribution defined by θ_n as the angular orientation directly computed from $\mathbf{n}$ and σ_θ as the variance associated with θ_n (i.e., uncertainty in the plane orientation); d is the plane distance; λ is a constant term that adjust σ_θ as function of d; σ_0 is the constant inherent noise in the measurement; lastly, η is a term that normalizes the values into a probability function.

In practice, given a new measurement θ_n, we update the associated $P(\hat{\theta}_n, \hat{\sigma}_\theta)$ that satisfies $|\hat{\theta}_n - \theta_n| \leq \pi/3$. If no posterior distribution satisfies the condition, we initialize $P(\hat{\theta}_n, \hat{\sigma}_\theta)$ with $\mathcal{N}(\theta_n, 2\pi)$. In the end, only estimations with $\hat{\sigma}_\theta < \pi/10$ are considered as plane orientations for the room.

F. Room Shape Optimization

At the final stage, we introduce the room shape optimization to convert previous estimations into room shapes (i.e., a set of room corners). To optimize over the number of corners and their locations, we introduce an iterative coarse-to-fine strategy in the shortest path algorithm (SPA) proposed in prior works [7], [10], where our method reduces the run-time and increases the accuracy. We name it as the iterative shortest path algorithm (iSPA). The main idea of the shortest path algorithm for room shape estimation is to represent the room structure as a graph of possible corners as nodes and edges weighted



(a) First SPA (b) Second SPA

Fig. 5. **Iterative Shortest Path Algorithm (iSPA).** We show two SPA evaluations as an example. (a) The first SPA estimates room corners around the room mask (green) under a coarse grid constrained by a maximum edge length, which will significantly speed up the optimization but also introduce redundant corners. (b) The second SPA evaluation searches over the neighboring areas around previous corner predictions (blue) with a finer grid size, which will help reduce redundant corners and refine the corner position. See Section III-F for details.

by a cost function based on geometry evidence (e.g., plane-wall features, plane orientations, room masks). Therefore, by optimizing the shortest path, the resulting nodes are the corner positions of the room.

In our design, after the camera leaves a room, we collect all the point sets $\mathbf{S}_0, \ldots, \mathbf{S}_k$ in the room that define valid plane-wall features (see Section III-E). Then, we project them into a 2D density map M_P. We also binarize the room density function H into a room mask M_H with a threshold 0.5 (see Section III-D). Lastly, we construct a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ are the nodes composed by the neighboring pixels around M_H, and $\mathcal{E}$ are the edges.

For the weight of an edge from $\mathbf{p} = (u_1, v_1)$ to $\mathbf{q} = (u_2, v_2)$, we define three cost functions:

$$\mathcal{L}_{\text{ori}}(\mathbf{p}, \mathbf{q}) = \min_{\theta \in \mathbf{M}_\theta} \left| \arccos\left(\mathbf{n}_\theta \cdot \frac{\overrightarrow{\mathbf{pq}}}{\|\overline{\mathbf{pq}}\|}\right)\right|, \tag{8}$$

$$\mathcal{L}_{\text{plane}}(\mathbf{p}, \mathbf{q}) = \sum_{\mathbf{z} \in E(\mathbf{p}, \mathbf{q})} (1 - M_P(\mathbf{z})), \tag{9}$$

$$\mathcal{L}_{\text{mask}}(\mathbf{p}, \mathbf{q}) = \sum_{\mathbf{z} \in E(\mathbf{p}, \mathbf{q})} M_H(\mathbf{z}), \tag{10}$$

where $\mathbf{M}_\theta$ is the set of likely plane orientations for the room (see Section III-E), $\mathbf{n}_\theta$ is the normal vector of the orientation, and $E(\mathbf{p}, \mathbf{q})$ is a set containing pixels on the line segment $\overline{\mathbf{pq}}$. Finally, we define our edge weight by the following:

$$\begin{aligned} \mathcal{L}(\mathbf{p}, \mathbf{q}) = \lambda_{\text{ori}} \mathcal{L}_{\text{ori}}(\mathbf{p}, \mathbf{q}) + \lambda_{\text{plane}} \mathcal{L}_{\text{plane}}(\mathbf{p}, \mathbf{q}) \\ + \lambda_{\text{mask}} \mathcal{L}_{\text{mask}}(\mathbf{p}, \mathbf{q}) + \lambda_{\text{complex}}, \end{aligned} \tag{11}$$

where λ_* are weighting constants, and λ_{complex} controls the model complexity (i.e., number of the room corners). To prevent SPA from finding trivial solutions, we adopt the containment constraint proposed in [7]. By solving the shortest path problem, we can estimate the room corners; hereinafter, we refer to this estimation as a SPA evaluation.

The grid resolution of the density maps is correlated with the number of nodes and edges in the graph, which controls the computation time. Therefore, we propose a first SPA evaluation that uses a coarse grid and constrains edges with a maximum length. This limited length strategy significantly reduces the time complexity of building graph: from $O(N^2)$ to

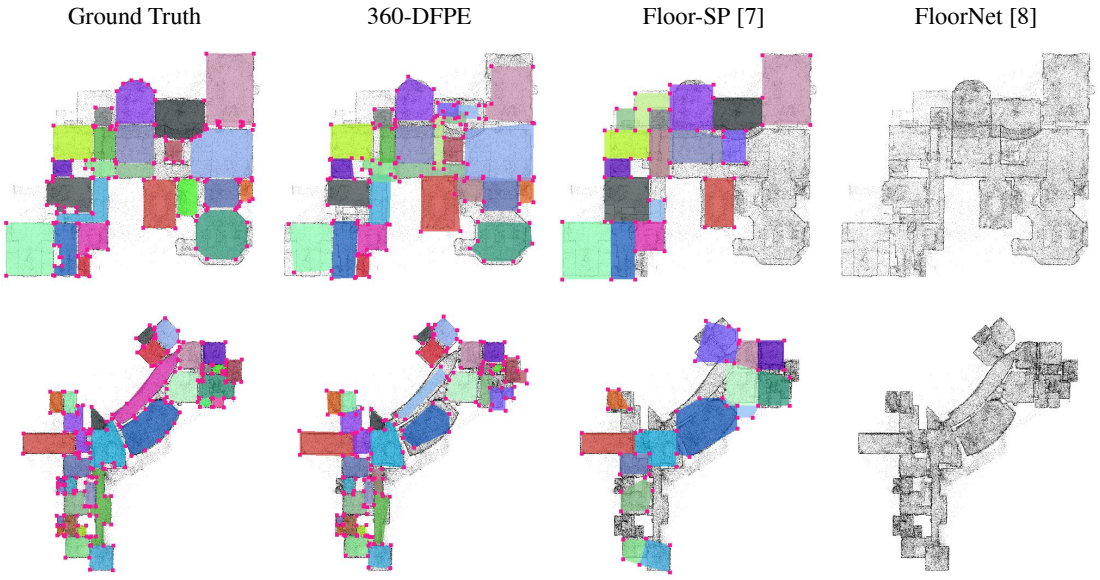


Fig. 6. **Qualitative Comparison on MP3D-FPE.** Compared with FloorNet [8] and Floor-SP [7], our 360-DFPE performs significantly better in two large scenes, which consist of a lot of rooms. 360-DFPE is able to successfully identify and reconstruct complex room shapes. We use colors to represent room instances and magenta dots to highlight room corners.

$O(\alpha N)$ where $\alpha \ll N$. However, it also introduces redundant estimated corners (see Fig. 5-(a)).

To reduce the redundant corner estimations, as shown in Fig. 5-(b), we apply additional SPA evaluations using finer resolutions without the edge-length constraint, in where the nodes in $\mathcal{V}$ are now defined by neighboring pixels around the previous estimated corners. In our implementation, we use 3 rounds of SPA with grid sizes $\{64, 96, 96\}$ in a coarse-to-fine fashion for room shape optimization.

In practice, iSPA not only removes redundant corner estimation from previous results but also reduces quantization error (i.e., the error due to discretization over a coarse grid), and increases the corner precision while having a faster run-time than the original SPA [7]. To further elaborate, we provide an ablation study of iSPA in Section IV-D.

IV. Experiments

In this section, we first introduce our experimental setups, including datasets, evaluation metrics, and implementation details. Then, we provide our experimental results comparing with baselines and the ablation study.

A. Dataset

Since image sequences are not available in previous floor plan datasets [7], [8], [10], [35], [36], we collect and annotate a new dataset, MP3D-FPE. We synthesize users scanning through the scenes in Matterport3D [23] with the Minos simulator [24]. MP3D-FPE contains 50 scenes and 687 rooms in total. For each scene, we provide the point cloud, sequence of panorama images with depth, camera pose information, and the floor plan annotation. This enables us to evaluate methods consuming various input modalities. As shown in Table I, MP3D-FPE has larger scenes and more rooms per scene, making it more challenging compared to the previous floor plan dataset proposed by Floor-SP [7].

TABLE I
Statistics of MP3D-FPE Dataset.

Dataset	No. Scenes	No. Rooms		Area (m^2)	
		avg	std	avg	std
Floor-SP [7]	**99**	7.07	2.26	104.52	44.36
MP3D-FPE (Ours)	50	**13.74**	**9.00**	**685.72**	**760.52**

B. Metrics

Following FloorNet [8] and Floor-SP [7], we evaluate the floor plan results with corners and room metrics, which are defined as the follows:

1) Corner: For the corner metric, we evaluate the recall and precision of the estimated corners. First, we project all corners in the scene onto a 2D grid with size 256×256. Then, each predicted corner is determined successfully reconstructed if there is a ground-truth corner within 10 pixels. When multiple predictions are around one ground truth, only the closest one is counted as true-positive; the rest are treated as false-positive cases.

2) Room: Like the corner metric, we report the recall and precision values in terms of the room area. An estimated room is considered successfully reconstructed if the intersection-over-union (IoU) with the ground-truth is above a certain IoU threshold.

C. Experimental Results

We compare 360-DFPE with two state-of-the-art baselines[1], Floor-SP [7] and Floor-Net [8], on our MP3D-FPE dataset. For reference purposes, we present two different implementations, using only initial 20% of keyframes and using all keyframes for scale recovery (see Section III-C). Moreover, since the ground-truth floor plans are labeled on top of the point cloud, we align our predicted floor plan to the real scale of the point

[1]We do not compare with MonteFloor [9] since the official code is not available during the submission of this manuscript.

TABLE II
EVALUATION RESULT ON MP3D-FPE. SR-*% REPRESENTS DIFFERENT RATIO OF KEYFRAMES USED IN RELATIVE SCALE RECOVERY.

Method	Input	Room $IoU_{0.3}$		Room $IoU_{0.5}$		Room $IoU_{0.7}$		Corner	
		Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
FloorNet [8]	Point Cloud	19.82	48.61	11.60	20.84	6.53	12.37	15.23	66.69
Floor-SP [7]	Point Cloud	72.07	76.16	61.39	61.87	46.20	43.34	53.58	**76.27**
360-DFPE (SR-20%)	360-Images	87.80	81.53	79.09	73.02	60.16	55.54	74.02	73.53
360-DFPE (SR-100%)	360-Images	**90.45**	**91.19**	**83.34**	**83.96**	**66.73**	**67.23**	**78.64**	72.75

cloud for a fair comparison. Therefore, during the evaluation, we align the estimated poses with ground truth poses to recover the real scale.

The quantitative result of floor plan estimation is presented in Table II, where our 360-DFPE significantly outperforms the two baselines. For the room metric, our method achieves the best recall and precision results under different IoU thresholds, which shows that our proposed pipeline with novel room identification and iSPA can estimate the room instances with higher success rates and more accurate room shapes. Note that our better room geometry estimation also leads to significantly higher corner recall.

On the other hand, comparing the number of keyframes used in scale recovery, using all the data performs slightly better than using only the initial 20% in both corner precision and room metrics. This is because using all data considers dynamic factors (e.g., different room sizes, changes in the camera height, and small drift in odometry estimation) and hence estimates a more robust scale. Nevertheless, using only the initial 20% keyframes can already acquire a good scale s, which achieves better results than the baselines without requiring the whole scene data in advance.

We show visualization examples in Fig. 6. These scenes are large, consist of numerous rooms, and contain rooms with complex shapes. As a result, FloorNet [8] suffers from producing valid rooms in the floor plan due to its strong assumptions about the room shape. On the other hand, Floor-SP [7] predicts room instances on 2D density map with fixed size based on ground-truth point clouds. In some cases, it may be hard to detect room boundaries on the density map. Therefore, in both the first and second rows, Floor-SP fails to reconstruct several small rooms. On the contrary, our proposed room identification successfully differentiates the rooms by reasoning the room layout geometries and the camera poses within the scene. Moreover, our method does not enforce prior assumptions such as Manhattan-world or require axis-aligned point clouds as input. Hence, 360-DFPE has the ability to handle rooms with more complex shapes compared to the baselines (see the second row of Fig. 6).

D. Ablation Study

To verify the effectiveness of iSPA, we provide a study in Table III. As shown in the first row, the original SPA, which is similar to the way in Floor-SP [7], suffers from a long run-time. Enforcing fixed grid size can reduce the run-time (see the second row). In the third row, although constraining maximum edge-length further speeds it up, it will introduce redundant corners and lower the corner precision. In the fourth and fifth row, additional SPA evaluations in a coarse-to-fine fashion increase corner precision.

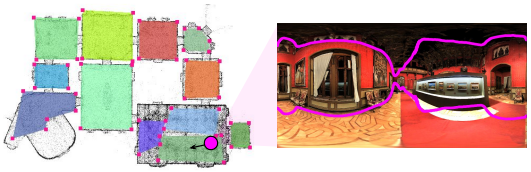


Fig. 7. **Failure Case.** We show that the noise from the 360-layout estimation could cause the room identification to fail, making a large room split into small parts. Note that the floor is occluded by a large furniture.

TABLE III
ABLATION STUDY OF ISPA. WE SELECT THE FOURTH ROW (3X ITERATIONS) FOR FINAL VERSION AS A BALANCE BETWEEN CORNER PRECISION AND RUN-TIME.

Fixed Size	Limited Edge Length	Iteration	Room $IoU_{0.5}$		Corner		Time (Sec. / Room)
			Recall	Precision	Recall	Precision	
		1x	70.51	76.66	70.48	71.03	80.71
v		1x	72.28	78.84	69.44	74.20	29.83
v	v	1x	**72.39**	**78.91**	**74.83**	54.01	**9.92**
v	v	3x	72.01	78.21	67.66	78.90	34.28
v	v	5x	71.83	78.22	67.58	**80.10**	49.03

V. FAILURE CASES AND LIMITATIONS

One of the common failure cases of 360-DFPE is due to error and drift produced by a monocular VSLAM, which may lead to a wrong layout registration in extreme cases, e.g., drastic movements and textureless regions. In addition, noisy 360-layout estimations, due to the occlusion of large furniture (e.g., Fig. 7) or atypical scenes (e.g., large halls or semi-open spaces), may affect our final estimation. Lastly, since the proposed pipeline relies on a deep-learning method for layout estimation, it requires enough computation resources to run the pre-trained model.

We believe that the mentioned limitations and failure cases may be addressed by using more robust solutions for both VSLAM and layout estimation. Moreover, closer integration between the VSLAM and floor plan estimation may increase the robustness of our current proposed pipeline.

VI. CONCLUSIONS

In this work, we present 360-DFPE, a new direct floor plan estimation that uses a sequence of 360-images without relying on active sensors. Our proposed solution can perform sequentially without strictly requiring the entire scene in advance. In particular, 360-DFPE consists of a novel 360-layout registration and a scale recovery, a novel room identification that sequentially tracks the corresponding room for each camera

location leveraging only the surrounding geometry, and lastly, a novel coarse-to-fine SPA formulation that enhances previous solutions [7], [10] achieves higher speed and precision. Our 360-DFPE outperforms current state-of-the-art solutions by reconstructing floor plan structures regardless of the size, axis-alignment, and room topology of the scene.

ACKNOWLEDGMENTS

This work is supported by the MOST Joint Research Center for AI Technology and All Vista Healthcare, Taiwan Computing Cloud, and MOST 110-2634-F-007-016. We thanks Professor Wei-Chen Chiu for his advises on this project.

REFERENCES

[1] A. J. Davison, "Futuremapping: The computational structure of spatial ai systems," *arXiv preprint arXiv:1803.11288*, 2018.

[2] F. Boniardi, T. Caselitz, R. Kümmerle, and W. Burgard, "Robust lidar-based localization in architectural floor plans," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3318–3324.

[3] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone, "3D Dynamic Scene Graphs: Actionable Spatial Perception with Places, Objects, and Humans," in *Proceedings of Robotics: Science and Systems*, July 2020.

[4] F. Boniardi, A. Valada, R. Mohan, T. Caselitz, and W. Burgard, "Robot localization in floor plans using a room layout edge extraction network," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5291–5297.

[5] Z. Ravichandran, L. Peng, N. Hughes, J. Griffith, and L. Carlone, "Hierarchical representations and explicit memory: Learning effective navigation policies on 3D scene graphs using graph neural networks," *arXiv preprint arXiv: 2108.01176*, 2021.

[6] S. Hampali, S. Stekovic, S. D. Sarkar, C. S. Kumar, F. Fraundorfer, and V. Lepetit, "Monte carlo scene search for 3d scene understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 804–13 813.

[7] J. Chen, C. Liu, J. Wu, and Y. Furukawa, "Floor-sp: Inverse cad for floorplans by sequential room-wise shortest path," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2661–2670.

[8] C. Liu, J. Wu, and Y. Furukawa, "Floornet: A unified framework for floorplan reconstruction from 3d scans," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 201–217.

[9] S. Stekovic, M. Rad, F. Fraundorfer, and V. Lepetit, "Montefloor: Extending mcts for reconstructing accurate large-scale floor plans," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 16 034–16 043.

[10] R. Cabral and Y. Furukawa, "Piecewise planar and compact floorplan reconstruction from images," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 628–635.

[11] G. Pintore, F. Ganovelli, R. Pintus, R. Scopigno, and E. Gobbetti, "3d floor plan recovery from overlapping spherical images," *Computational visual media*, vol. 4, no. 4, pp. 367–383, 2018.

[12] S. Yang and S. Scherer, "Monocular object and plane slam in structured environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3145–3152, 2019.

[13] P.-H. Le and J. Košecka, "Dense piecewise planar rgb-d slam for indoor environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4944–4949.

[14] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "Pl-slam: Real-time monocular visual slam with points and lines," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 4503–4508.

[15] C. Yan, B. Shao, H. Zhao, R. Ning, Y. Zhang, and F. Xu, "3d room layout estimation from a single rgb image," *IEEE Transactions on Multimedia*, vol. 22, no. 11, pp. 3014–3024, 2020.

[16] C.-W. Hsiao, C. Sun, M. Sun, and H.-T. Chen, "Flat2layout: Flat representation for estimating layout of general room types," *arXiv preprint arXiv:1905.12571*, 2019.

[17] C.-Y. Lee, V. Badrinarayanan, T. Malisiewicz, and A. Rabinovich, "Roomnet: End-to-end room layout estimation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4865–4874.

[18] F.-E. Wang, Y.-H. Yeh, M. Sun, W.-C. Chiu, and Y.-H. Tsai, "Led2-net: Monocular 360deg layout estimation via differentiable depth rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 956–12 965.

[19] G. Pintore, M. Agus, and E. Gobbetti, "Atlantanet: Inferring the 3d indoor layout from a single 360 image beyond the manhattan world assumption," in *European Conference on Computer Vision*. Springer, 2020, pp. 432–448.

[20] C. Sun, C.-W. Hsiao, M. Sun, and H.-T. 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*, 2019, pp. 1047–1056.

[21] C. Sun, M. Sun, and H.-T. Chen, "Hohonet: 360 indoor holistic understanding with latent horizontal features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2573–2582.

[22] S. Sumikura, M. Shibuya, and K. Sakurada, "Openvslam: a versatile visual slam framework," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 2292–2295.

[23] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *International Conference on 3D Vision (3DV)*, 2017.

[24] M. Savva, A. X. Chang, A. Dosovitskiy, T. Funkhouser, and V. Koltun, "MINOS: Multimodal indoor simulator for navigation in complex environments," *arXiv:1712.03931*, 2017.

[25] Y.-W. Chao, W. Choi, C. Pantofaru, and S. Savarese, "Layout estimation of highly cluttered indoor scenes using geometric and semantic cues," in *International Conference on Image Analysis and Processing*. Springer, 2013, pp. 489–499.

[26] A. Flint, C. Mei, D. Murray, and I. Reid, "A dynamic programming approach to reconstructing building interiors," in *European conference on computer vision*. Springer, 2010, pp. 394–407.

[27] A. Flint, D. Murray, and I. Reid, "Manhattan scene understanding using monocular, stereo, and 3d features," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 2228–2235.

[28] G. Tsai, C. Xu, J. Liu, and B. Kuipers, "Real-time indoor scene understanding using bayesian filtering with motion cues," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 121–128.

[29] Y. Zhang, S. Song, P. Tan, and J. Xiao, "Panocontext: A whole-room 3d context model for panoramic scene understanding," in *European conference on computer vision*. Springer, 2014, pp. 668–686.

[30] H. Yang and H. Zhang, "Efficient 3d room shape recovery from a single panorama," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5422–5430.

[31] J. Xu, B. Stenger, T. Kerola, and T. Tung, "Pano2cad: Room layout from a single panorama image," in *2017 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2017, pp. 354–362.

[32] S. Dasgupta, K. Fang, K. Chen, and S. Savarese, "Delay: Robust spatial layout estimation for cluttered indoor scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 616–624.

[33] A. Mallya and S. Lazebnik, "Learning informative edge maps for indoor scene layout prediction," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 936–944.

[34] M. Hirzer, V. Lepetit, and P. ROTH, "Smart hypothesis generation for efficient and robust room layout estimation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 2912–2920.

[35] J. Xiao and Y. Furukawa, "Reconstructing the world's museums," *International journal of computer vision*, vol. 110, no. 3, pp. 243–258, 2014.

[36] C. Lin, C. Li, and W. Wang, "Floorplan-jigsaw: Jointly estimating scene layout and aligning partial scans," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5674–5683.

[37] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[38] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE robotics & automation magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[39] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[40] M. Fischler and R. Bolles, "Random sample paradigm for model consensus: A apphcatlons to image fitting with analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.