Supplementary Material: Deep Polycuboid Fitting for Compact 3D Representation of Indoor Scenes

Gahye Lee Hyejeong Yoon Jungeon Kim Seungyong Lee

POSTECH

{gahye0509, hjyoon02, jungeonkim, leesy}@postech.ac.kr

In this supplemental material, we provide detailed explanations of the preprocessing steps and score function used in our rectilinear mesh reconstruction (Section 1), implementation details of transformer network and GCN (Section 2), and the synthetic dataset generation process (Section 3). We also present additional qualitative and quantitative results to demonstrate the effectiveness of our method (Section 4).

1. Rectilinear Mesh Reconstruction Details

1.1. Preprocessing Step

We details the preprocessing mentioned in Sec. 3.2 of the main paper, which computes the rotation matrix. A polycuboid instance consists of a set of detected faces, where each face is represented as a set of points with an inferred face label. Each face label corresponds to one of six global axes (e.g., ±x, ±y, ±z). We utilize this information to compute the rotation matrix. We first fit a plane to each detected face to obtain its normal vector. We then compute the angular differences between the plane normals and the corresponding global axes derived from the inferred face labels. Finally, we determine the rotation matrix that minimizes these angle differences using a least-square approach.

Using the optimized rotation matrix, we rotate the polycuboid instance to be aligned with the global frame. After completing the rectilinear mesh reconstruction process, we transform the reconstructed polycuboid mesh back to its original coordinate by applying the inverse of the rotation matrix.

1.2. Score Function

In Sec. 3.2 of the main paper, to reconstruct a polycuboid instance, we select a set of 3D boxes from a 3D non-uniform grid using a heuristic score function. Each box b in the grid is represented as a polygonal mesh of a cuboid parameterized by its center and extents, and the score function for b is defined as:

$$S(b) = \sum_{n=1}^{6} sgn(f_n, P_n) \cdot w(f_n, P_n), \tag{1}$$

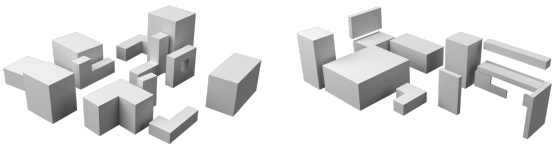
where f_n represents each face of the box b, and P_n is the set of points from detected faces located within an 0.05m of f_n. The sign function $sgn(f_n, P_n)$ returns 1 if the cuboid face label of f_n matches the most frequent face label in P_n and -1 otherwise. The weight factor $w(f_n, P_n)$ mitigates the impacts of partially detected faces due to incomplete scan data. It is defined as the ratio of the estimated surface area of P_n to the area of f_n. The area of P_n is computed using the area of the 2D bounding box of projected P_n onto the face f_n. A box with a positive score is classified as inside the polycuboid instance, whereas a negative score is classified as outside.

2. Implementation Details

We use Transformer and Graph convolutional Network (GCN) in our framework. This section provides implementation details of these networks with model architectures.

For the face labeling task, we adopt the model named Stratified Transformer [1]. The model first encodes the input point cloud using a hierarchical tokenization strategy, where local features are extracted at multiple levels of granularity. Theses features are then processed by five transformer layers, where each layer refines the point features based on attention mechanism. Once the point features are fully embedded, point-wise classification and offset regression are performed using two separate heads, each composed of a two-layer MLP.

For spatial relationship prediction, we adopt the GCN model proposed by Wald et al. [3], which is designed to learn structured scene graphs from 3D indoor scene data. The model first encodes input data using two separate PointNet encoders, one for node features and another for edge features to transform the raw information into higher-dimensional feature representation. The node and edge fea-



(a) Random configuration (b) Contextual configuration

Figure 1. Two types of configurations of polycuboids are used to generate our synthetic dataset. (a) Polycuboids are randomly placed. (b) Polycuboids are positioned to replicate the object configurations derived from the ScanNet dataset.

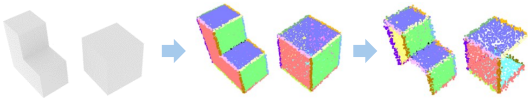


Figure 2. A simplified example of the data generation process. Points are sampled from polycuboid and cuboid meshes, followed by adding noise and holes.

tures are then fed into a GCN that consists of five graph convolution layers and iteratively refines the node and edge features through message passing. Node classification and edge classification are finally performed using two separate heads composed of three-layer MLP.

3. Synthetic Dataset Generation

We provide further details on the dataset generation process described in Sec. 4.1 of the main paper, along with visual examples.

Each scene is constructed using a combination of polycuboid and cuboid meshes, following two configuration types: 1) Random configuration: Scenes are generated by randomly placing polycuboid and cuboid meshes with varying numbers, scales, and orientations while ensuring that no meshes overlap (Fig. 1a). Each scene contains 5 to 20 meshes whose scales range from 0.4m to 2.4m. 2) Contextual configuration: To simulate more realistic configurations, we utilize object bounding boxes from the ScanNet dataset, which are parameterized by their center coordinates and extents. In these scenes, existing bounding boxes are randomly replaced by polycuboid meshes (Fig. 1b).

After composing the scenes, we sample points at 1cm interval on each mesh face. To better simulate real-world scanning conditions, we add Gaussian noise to the sampled points and create holes by randomly removing 0 to 3 faces from each mesh, along with their corresponding sampled points (Fig. 2).

4. Additional Results

In this section, we present additional qualitative and quantitative results to further evaluate our method. In Fig. 3, we illustrate examples of input point clouds alongside our final results for 'room0' from the Replica dataset. Despite the noisy input, scene components are plausibly represented with polycuboids.

We also demonstrate two applications, scene editing and virtual room tours, in Figs. 4 and 5, respectively. These examples highlight the ease of manipulating scene components and the seamless virtual exploration experience, which would be valuable for interior design and AR/VR applications.

Further quantitative and qualitative evaluations are provided on the ScanNet and Replica datasets. Visual results are shown in Figs. 6 and 7, with corresponding quantitative results in Tables 1 and 2, respectively.

References

[1] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3D point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8500–8509, 2022. 1

[2] Michaël Ramamonjisoa, Sinisa Stekovic, and Vincent Lepetit. Monteboxfinder: Detecting and filtering primitives to fit a noisy point cloud. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, pages 161–177. Springer, 2022. 3, 4, 5

[3] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3D semantic scene graphs from 3D indoor reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3961–3970, 2020. 1

Figure 3. Example of polycuboid abstraction on Replica dataset for 'room0'. The two images on the left show the input point cloud and our final result, while the two images on the right provide close-up views.

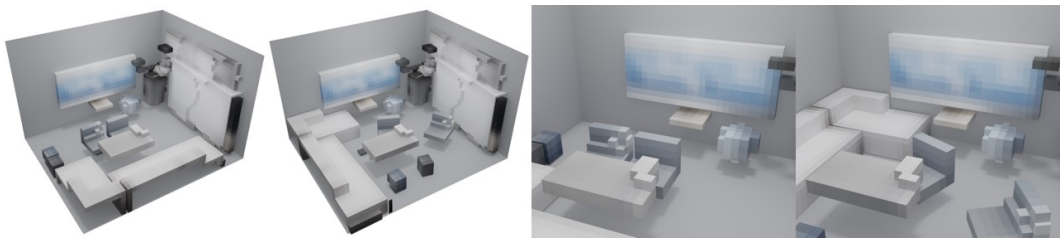


Figure 4. Editing examples of reconstructed polycuboids on Replica dataset for 'office 0'. The first and third images show the original arrangements of polycuboids representing the scene, while the second and fourth images show rearranged polycuboids, including an L-shaped sofa and two chairs.

Polycuboids with textures Room tour Scene editing

Figure 5. Application examples on the dataset scanned with an iPhone. Our compact polycuboid representation of an indoor scene enables practical applications, such as virtual room tour and scene editing. For scene editing, a grey sofa created by a designer is used to replace the original scanned sofa.

Table 1. For four scenes from ScanNet dataset, we use Chamfer Distance (CD) to measure geometric discrepancy between input points and points sampled from our output polycuboids. Average: average CD score for four examples. Average*: average CD score for all validation data of ScanNet.

	0006_00	0035_00	0273_01	0276_00	Average	Average*
MBF[2]	0.078	0.046	0.069	0.065	0.065	0.066
Ours	0.047	0.034	0.034	0.087	0.050	0.040

Table 2. For nine scenes from Replica dataset, we use Chamfer Distance (CD) to measure geometric discrepancy between input points and points sampled from our output polycuboids. We apply our framework to points categorized under 'layout' and 'non-layout', respectively.

	type	hotel 0	office 0	office 1	office 2	office 3	office 4	room 0	room 1	room 2	Average
MBF[2]	non-layout	0.068	0.151	0.068	0.074	0.076	0.071	0.060	0.078	0.056	0.078
Ours	non-layout	0.040	0.037	0.051	0.040	0.033	0.036	0.048	0.075	0.036	0.044
MBF[2]	layout	0.078	0.062	0.049	0.036	0.043	0.027	0.037	0.049	0.036	0.047
Ours	layout	0.131	0.090	0.073	0.042	0.116	0.045	0.064	0.082	0.084	0.081

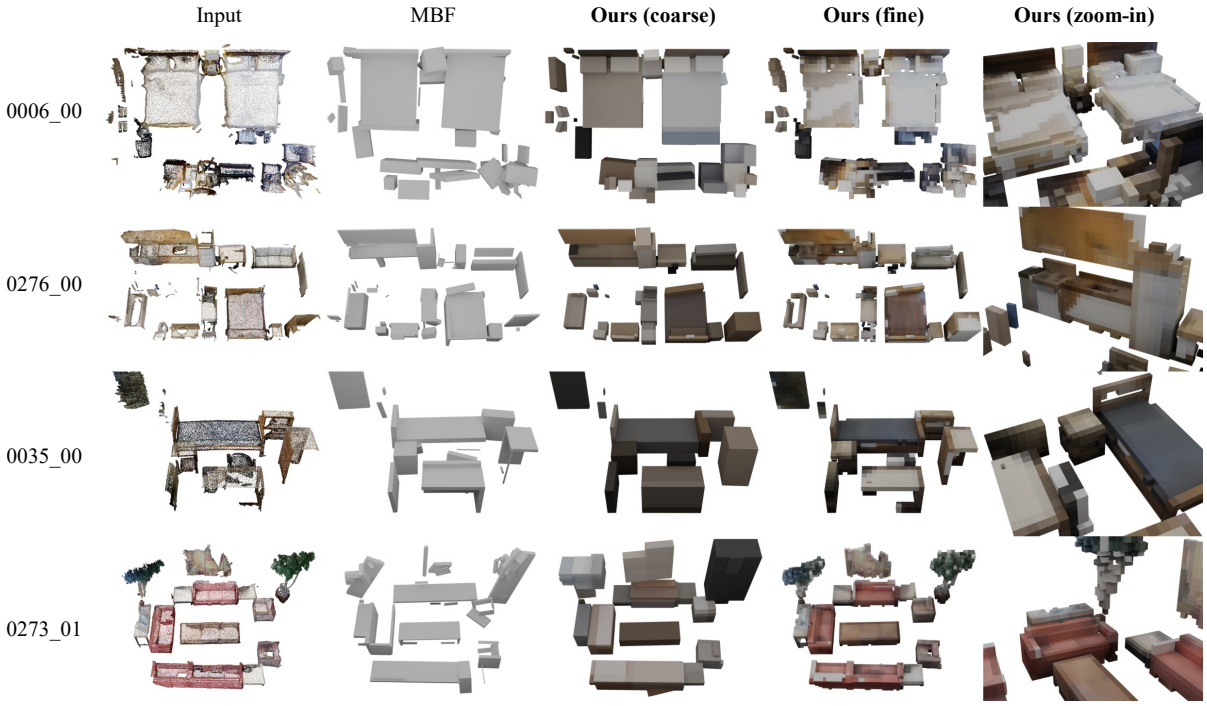


Figure 6. Qualitative comparison on four scenes from ScanNet dataset. From left to right, columns show the input point cloud, results from MBF [2], results from our method with coarse and fine detail levels, and zoomed-in views of our fine level results. Our method faithfully captures the underlying geometry, even when the inputs are very noisy and incomplete.

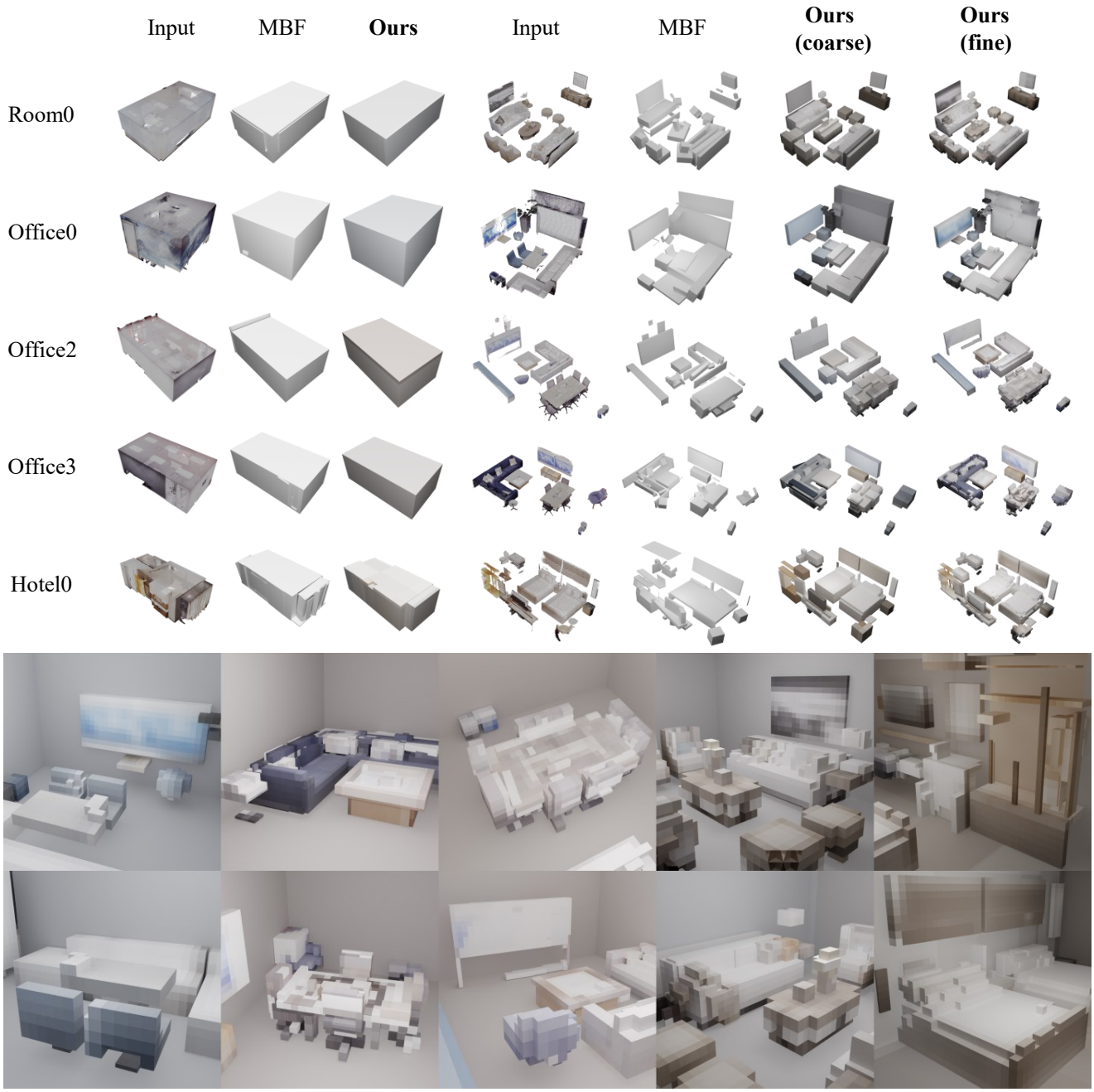


Figure 7. Qualitative evaluations on five rooms from Replica dataset, showing the separate results for layout and objects. From left to right, columns show the input point cloud for layout, results from MBF [2], results from our method, the input point cloud for objects, results from MBF, results from our method with coarse and fine detail levels. The bottom two rows provide zoomed-in views of our results constructed using polycuboids for both layout and objects.