

RandomRooms: Unsupervised Pre-training from Synthetic Shapes and Randomized Layouts for 3D Object Detection

Supplementary Material

A. Details on Random Room Generation

To more clearly show the generation process of random rooms, we provide pseudo-code and explanatory comments of our room generation method in Algorithm 1.

B. Visualization of Random Rooms

We show more examples of the generated random room pairs in Figure 1.

Algorithm 1 Pseudo-code of Random Room Generation

```
1 # objects: list of object point clouds
2
3 # object level data augmentation
4 objects = object_augmentation(objects)
5
6 # sort objects by their areas
7 objects, object_ind, obj_area = sort_object(objects)
8
9 # set the overall area of the rectangular room
10 overall_area = sum(obj_area) * 2 * 10000 * (random.random() * 0.4 + 0.6)
11 a_value = np.sqrt(overall_area)
12 a = random.randint(int(a_value*0.75), int(a_value*1.25))
13 b = int(overall_area) // a
14 a_m = float(a) / 100.
15 b_m = float(b) / 100.
16 room_state = np.zeros((a, b), dtype=np.float)
17
18 final_layout = []
19 instance_label = []
20
21 # place object to the room
22 for i in range(len(objects)):
23     obj = objects[i]
24     x, y, z = get_object_size(obj)
25
26     for _ in range(max_iter):
27         # generate the position from beta distribution
28         pos_x = np.random.beta(0.5, 0.5) * (a_m - x)
29         pos_y = np.random.beta(0.5, 0.5) * (b_m - y)
30         state_part = room_state[int(pos_x*100):int((pos_x+x)*100), int(pos_y*100): int((pos_y+y)*100)]
31         max_height = state_part.max()
32         if (max_height + z < 2.0 and max_height < 0.5) or max_height < 1e-3:
33             break
34
35         room_state[int(pos_x * 100):int((pos_x + x) * 100), int(pos_y * 100): int((pos_y + y) * 100)] += z
36         obj[:, 0] += pos_x
37         obj[:, 1] += pos_y
38         obj[:, 2] += max_height
39         final_layout.append(obj)
40         instance_label.append(np.ones((obj.shape[0],), dtype=int) * (object_ind[i] + 1))
41
42 # add floor and walls
43 final_layout, instance_label = add_floor_wall(final_layout, instance_label)
44
45 # form the final scene point cloud
46 final_layout = np.concatenate(final_layout, axis=0)
47 instance_label = np.concatenate(instance_label, axis=0)
48
49 # normalize coordinates
50 final_layout[:, 0:2] = final_layout[:, 0:2] - final_layout[:, 0:2].mean(axis=0, keepdims=True)
```



Figure 1: Visualization of the pairs of *Random Rooms*.