

A APPENDIX

A.1 THE FRAMEWORK

Octree Construction. As shown in Figure 1 (c), octree is a point cloud storage structure, which is beneficial to compression. To construct the octree, we first need to surround the point cloud with the smallest cube. Then the smallest cube will be split into eight sub-cube of the same size. For each sub-cube, if there is no point in the cube, this cube is recorded empty. Otherwise, the cube is recorded nonempty, which means there are some points in this cube. After that, for the nonempty sub-cube, we repeat the above split process to reduce the size of the cube until the depth of the octree reaches the predefined maximum depth value. In the constructed octree, a non-leaf node stands for one cube and the nonempty non-leaf node has eight child nodes that stand for the sub-cubes.

Encoder. The encoder compresses the octree into the bit-stream. All octrees are encoded into the bit-stream from the low depth level to the high depth level, as shown in Figure 1(c). Therefore, we can divide the full bit-stream into two parts according to the selected octree depth.

Decoder and Point Cloud Reconstruction. The decoder will restores the octree from the bit-stream. And the point cloud reconstruction module reconstructs the point cloud coordinates from the octree. The reconstruct point cloud coordinates is the coordinate of the center point of the smallest nonempty cubes. The point cloud coordinates can not only be used in machine vision tasks but also be easily visualized for human vision.

Data Processing. In each octree, all points in one smallest cube will be combined to one point. So the number of points in the reconstructed point cloud will have less number of points than the raw point cloud. Additionally, the reduced number of different point clouds is different, so the output point clouds from the point cloud reconstruction module have different number of points. However, our framework need the size of batch size more than 1 (*e.g.*, 32 or 48). And the point clouds with different number of points can not directly concatenate together to one batch. So we random sample the point cloud based on the predefined number of points to unify the size of the point cloud. As we know, the octree will combine some points to one point. However, each point corresponds to one target for the segmentation task. So if all points in the smallest cube have the same label, we use this label for the new combined point. If the points in one smallest cube have the different label, we use the label of the point which is closest to the combined point.

A.2 HUMAN VISION RESULT

The experimental results of our SPC-Net for human vision are shown in Table 2. In this table, we observe that our SPC-Net achieves the same performance as the VoxelContext-Net.

Table 2: The compression performance of our SPC-Net for human vision on the ModelNet10, ModelNet40, ShapeNet and ScanNet datasets.

Dataset	method	bpp	PSNR	CD
ModelNet10	SPC-Net	6.8508	48.9000	0.003604
	VoxelContext-Net	6.8508	48.9000	0.003604
ModelNet40	SPC-Net	6.4124	48.8835	0.003612
	VoxelContext-Net	6.4124	48.8835	0.003612
ShapeNet	SPC-Net	4.2866	49.1161	0.003528
	VoxelContext-Net	4.2866	49.1161	0.003528
ScanNet	SPC-Net	5.9525	55.2217	0.001740
	VoxelContext-Net	5.9526	55.2236	0.001740

A.3 VISUALIZATION

The visualization results of the segmentation task are shown in Figure 5. From the results of the table and the mug in the first two rows of Figure 5, point clouds reconstructed from the octree with 5 depth levels can achieve similar segmentation performance when compared with the point clouds reconstructed from the octree with more depth levels. Therefore, our octree depth level predictor

prefers the octree with 5 depth levels for the segmentation task in this two cases to save bits. From the results of the car and airplane in the last two rows of Figure 5, the point clouds reconstructed from the octrees with 7 depth levels achieve much better segmentation performance when compared with those octrees with less depth levels. Therefore, our octree depth level predictor selects the octree with 7 depth levels for achieving better segmentation performance in this two cases.

The visualization results of the detection task is shown in Figure 6. In the first row, the point cloud reconstructed from the octree with 7 depth levels achieves the same mAP@0.25 performance when compared with the point clouds reconstructed from the octrees with higher depth levels. Therefore, our octree depth level predictor selects the 7 depth levels in this case to save bits. In the second row, the point cloud reconstructed from the octree with 9 depth levels has much better mAP@0.25 performance than the point cloud reconstructed from the octrees with less depth level. Therefore, our octree depth level predictor selects the octree with 9 depth levels for better detection performance in this case.

It is observed that our proposed octree depth level predictor can select the optimal depth levels of the octrees for different cases, which demonstrate the effectiveness of our proposed octree depth level predictor.

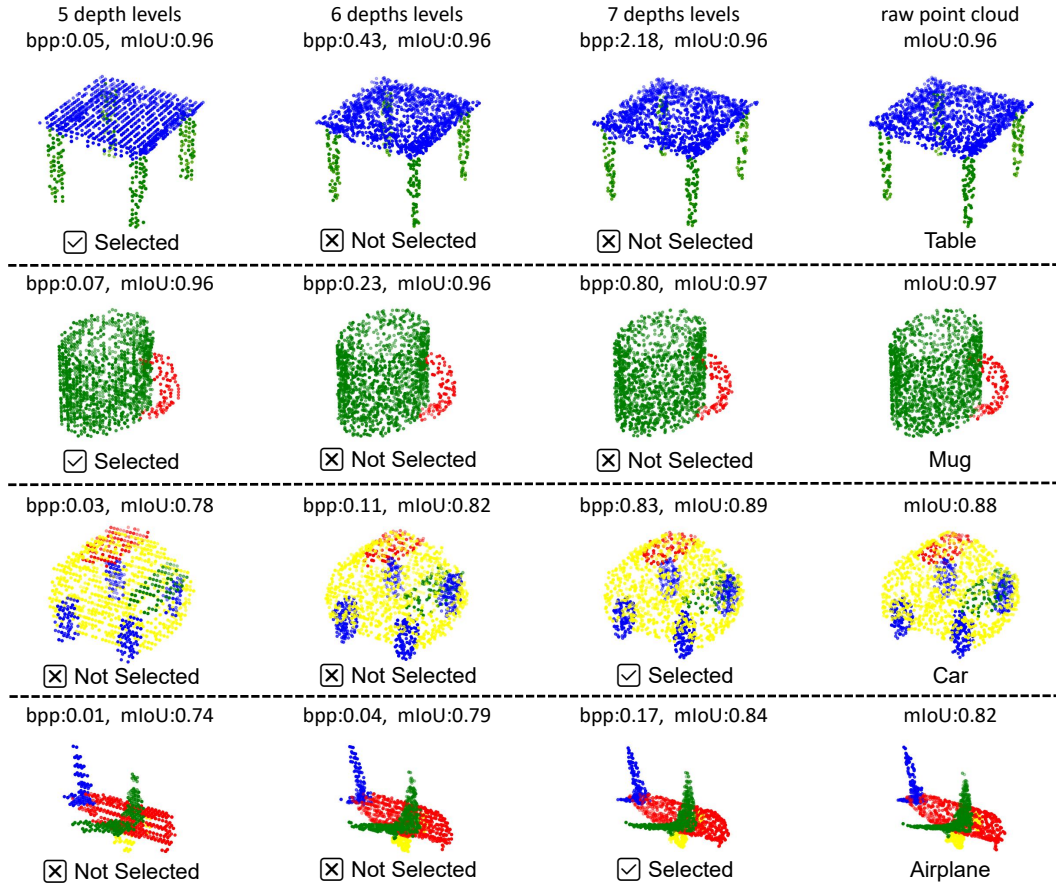


Figure 5: Different qualitative results of segmentation task on ShapeNet dataset. "5 depth levels", "6 depth levels" and "7 depth levels" denote that the point cloud is reconstructed by the octree with 5, 6 and 7 depth levels. The λ is predefined as 0.02 in the loss function 3.

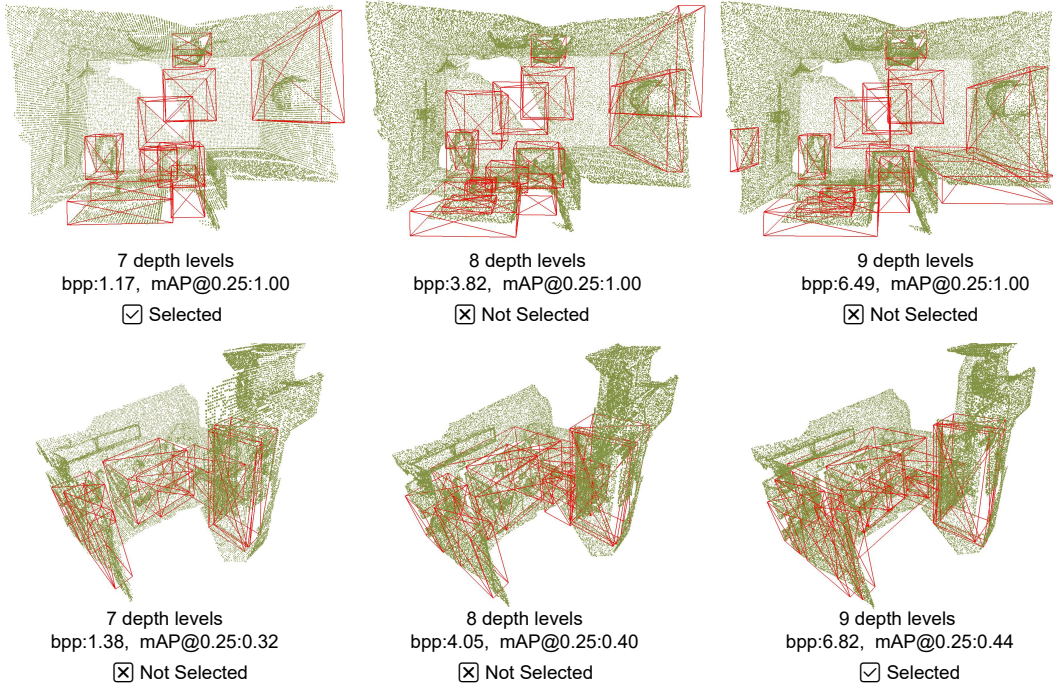


Figure 6: Different qualitative results of detection task on ScanNet dataset. "7 depth levels", "8 depth levels" and "9 depth levels" mean the point cloud is reconstructed from the octree with 7 depth levels, 8 depth levels and 9 depth levels. The λ predefine as 0.6 in the loss function 3 for the octree depth level predictor to select.