Supplementary: SGPN: Similarity Group Proposal Network for 3D Point Clouds Instance Segmentation

Weiyue Wang[1] Ronald Yu[2] Qiangui Huang[1] Ulrich Neumann[1]
[1]University of Southern California, Los Angeles, California
[2]University of California, San Diego, San Diego, California
{weiyuewa,qianguih,uneumann}@usc.edu ronaldiscool@gmail.com

1. Network Architecture

In our experiments, we use both PointNet and PointNet++ as our baseline architectures. For the S3DIS dataset, we use PointNet as our baseline for fair comparison with the 3D object detection system described in the PointNet paper [2]. The network architecture is the same as the semantic segmentation network as stated in PointNet except for the last two layers. Our F is the last 1×1 *conv* layer with BatchNorm and ReLU in PointNet with 256 output channels. F_{SIM}, F_{CF}, F_{SEM} are 1×1 *conv* layers with output channels $(128, 128, 128)$, respectively.

For the NYUV2 dataset, we use PointNet++ as our baseline. We use the same notations as PointNet++ to describe our architecture:

$SA(K, r, [l_1, ..., l_d])$ is a set abstraction (SA) level with K local regions of ball radius r using a PointNet architecture of d 1×1 *conv* layers with output channels $l_i (i = 1, ..., d)$. $FP(l_1, ..., l_d)$ is a feature propagation (FP) level with d 1×1 *conv* layers. Our network architecture is:

$$
\begin{aligned}
&SA(1024, 0.1, [32, 32, 64]),\\
&SA(256, 0.2, [64, 64, 128]),\\
&SA(128, 0.4, [128, 128, 256]),\\
&SA(64, 0.8, [256, 256, 256]),\\
&SA(16, 1.2, [256, 256, 512]),\\
&FP(512, 256),\\
&FP(256, 256),\\
&FP(256, 256),\\
&FP(256, 128),\\
&FP(128, 128, 128, 128).
\end{aligned}
$$

F_{SIM}, F_{CF}, F_{SEM} are 1×1 *conv* layers with output channels $(128, 128, 128)$ respectively.

For our experiments on the ShapeNet part dataset, PointNet++ is used as our baseline. We use the same network architecture as in the PointNet++ paper [3].

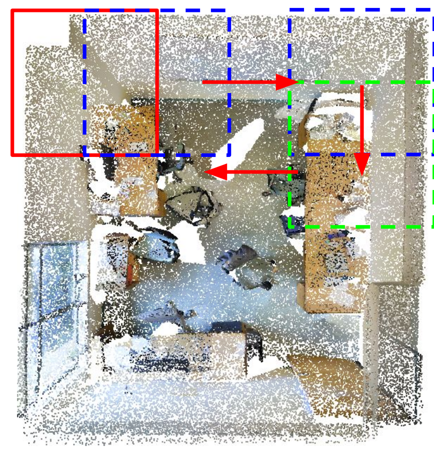

Figure 1: Dividing scene into blocks with overlap (top view).

F_{SIM}, F_{CF}, F_{SEM} are 1×1 *conv* layers with output channels $(64, 64, 64)$, respectively.

2. Experiment Settings

2.1. S3DIS Dataset

Block Merging We divide each scene into $1m \times 1m$ blocks with overlapping sliding windows in a snake pattern of stride $0.5m$ as is shown in Figure 1. The entire scene is also divided into a $400 \times 400 \times 400$ grid V. V_k is used to indicate the instance label of cell k where $k \in [0, 400 \times 400 \times 400)$. Given V and point instance labels for each block PL where PL_{ij} represents the instance label of jth point in block i, a *BlockMerging* algorithm (refer to Algorithm 1) is derived to merge object instances from different blocks.

In Figure 2, we show more qualitative results of instance segmentation with SGPN.

	Mean	wall	floor	chair	table	desk	bed	book shelf	sofa	sink	bath tub	toilet	cur-tain	coun-ter	door	win-dow	shower curtain	fridge	pic-ture	cabi-net
Seg-Cluster	30.9	**49.3**	77.1	57.1	38.8	17.6	39.4	17.2	**37.0**	29.4	40.0	43.1	52.9	26.7	0.0	0.0	18.0	15.7	0.0	28.8
SGPN	**35.1**	46.9	**79.0**	**63.6**	**40.7**	**22.8**	**43.8**	**22.4**	36.8	**35.8**	**46.2**	**60.5**	**61.1**	**26.9**	0.0	0.0	**21.7**	**24.5**	0.0	**34.1**

Table 1: Instance segmentation results on ScanNet. The metric is AP (%) with IoU threshold 0.5. We observe 0 percent AP on items that appear on the wall (door, window, picture) as they contain very little depth information and are almost all incorrectly semantically labeled as the wall. Future works can explore addressing this problem.

Algorithm 1: BlockMeriging

Input : V, PL
Output: Point instance labels for the whole scene L
Initialize V with all elements -1;
$GroupCount \leftarrow 0$;
for *every block* i **do**
  if i *is the* $1st$ *block* **then**
    for *every point* P_j *in block* i **do**
      Define k where P_j is located in the kth cell of V;
      $V_k \leftarrow PL_{1j}$;
    end
  else
    for *every instance* I_j *in block* i **do**
      Define V_{I_j} points in I_j are located in cells V_{I_j};
      $V_t \leftarrow$ the cells in V_{I_j} that do not have value -1;
      if *the frequency of the mode in* $V_t < 30$ **then**
        $V_{I_j} \leftarrow GroupCount$;
        $GroupCount \leftarrow GroupCount + 1$;
      else
        $V_{I_j} \leftarrow$ the mode of V_t;
      end
    end
  end
end
for *every point* P_j *in the whole scene* **do**
  Define k where P_j is located in the kth cell of V;
  $L_j \leftarrow V_k$;
end

3. More Experiments

3.1. ScanNet

We provide more experimental results on ScanNet [1]. This dataset contains 1513 scanned and reconstructed indoor scenes. We use the official split with 1201 scenes for training and 312 for testing. Following the same *Block-Merging* procedure, each scene is divided into $1.5m \times 1.5m$ blocks and each block is uniformly sampled into 4096 points for training. All points in the block are used at test time. Each point is represented by a $9D$ vector (XYZ, RGB, and normalized location with respect to the room scene). PointNet++ is used as the baseline. The network architecture is:

$$SA(1024, 0.1, [32, 32, 64]),$$
$$SA(256, 0.2, [64, 64, 128]),$$
$$SA(64, 0.4, [128, 128, 256]),$$
$$SA(16, 0.8, [256, 256, 512]),$$
$$FP(256, 256),$$
$$FP(256, 256),$$
$$FP(256, 128),$$
$$FP(128, 128, 128, 128).$$

And F_{SIM}, F_{CF}, F_{SEM} are 1×1 $conv$ layers with output channels $(128, 128, 128)$ respectively. Table 1 illustrates the quantitative comparison results with Seg-Cluster. The metric is average precision (AP) with IoU threshold 0.5. Figure 3 shows instance segmentation results on ScanNet.

References

[1] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 1

[2] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CVPR*, 2017. 1

[3] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 1

Figure 2: Instance segmentation results on S3DIS with SGPN. Different colors represent different instances. The colors of the same object in ground truth and prediction are not necessarily the same.

Figure 3: Instance segmentation results on ScanNet with SGPN. Different colors represent different instances. The colors of the same object in ground truth and prediction are not necessarily the same.