Supplemental Material
3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans

Ji Hou Angela Dai Matthias Nießner
Technical University of Munich

In this supplemental document, we describe the details of our 3D-SIS network architecture in Section 1. In Section 2, we describe our training scheme on scene chunks to enable inference on entire test scenes, and finally, in Section 3, we show additional evaluation on the ScanNet [1] and SUNCG [3] datasets.

1. Network Architecture

small anchors	big anchors
(8, 6, 8)	(12, 12, 40)
(22, 22, 16)	(8 , 60, 40)
(12, 12, 20)	(38, 12, 16)
	(62, 8 , 40)
	(46, 8 , 20)
	(46, 44, 20)
	(14, 38, 16)

Table 1: Anchor sizes (in voxels) used for SUNCG [3] region proposal. Sizes are given in voxel units, with voxel resolution of ≈ 4.69cm

small anchors	big anchors
(8, 8, 9)	(21, 7, 38)
(14, 14, 11)	(7, 21, 39)
(14, 14, 20)	(32, 15, 18)
	(15, 31, 17)
	(53, 24, 22)
	(24, 53, 22)
	(28, 4, 22)
	(4, 28, 22)
	(18, 46, 8)
	(46, 18, 8)
	(9, 9, 35)

Table 2: Anchor sizes used for region proposal on the ScanNet dataset [1]. Sizes are given in voxel units, with voxel resolution of ≈ 4.69cm

Table 3 details the layers used in our detection backbone, 3D-RPN, classification head, mask backbone, and mask prediction. Note that both the detection backbone and mask backbone are fully-convolutional. For the classification head, we use several fully-connected layers; however, due to our 3D RoI-pooling on its input, we can run our entire instance segmentation approach on full scans of varying sizes.

We additionally list the anchors used for the region proposal for our model trained on the ScanNet [1] and SUNCG [3] datasets in Tables 2 and 1, respectively. Anchors for each dataset are determined through k-means clustering of ground truth bounding boxes. The anchor sizes are given in voxels, where our voxel size is ≈ 4.69cm.

2. Training and Inference

In order to leverage as much context as possible from a input RGB-D scan, we leverage fully-convolutional detection and mask backbones to infer instance segmentation on varying-sized scans. To accommodate memory and efficiency constraints during training, we train on chunks of scans, i.e. cropped volumes out of the scans, which we use to generalize to the full scene at test time (see Figure 1). This also enables us to avoid inconsistencies which can arise with individual frame input, with differing views of the same object; with the full view of a test scene, we can more easily predict consistent object boundaries.

The fully-convolutional nature of our methods allows testing on very large scans such as entire floors or buildings in a single forward pass; e.g., most SUNCG scenes are actually fairy large; see Figure 2.

3. Additional Experiment Details

We additionally evaluate mean average precision on SUNCG [3] and ScanNetV2 [1] using an IoU threshold of 0.5 in Tables 5 and 4. Consistent with evaluation at an IoU threshold of 0.25, our approach leveraging joint color-geometry feature learning and inference on full scans enables significantly better instance segmentation perfor-

layer name	input layer	type	output size	kernel size	stride	padding
geo_1	TSDF	conv3d	(32, 48, 24, 48)	(2, 2, 2)	(2, 2, 2)	(0, 0, 0)
geo_2	geo_1	conv3d	(32, 48, 24, 48)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
geo_3	geo_2	conv3d	(32, 48, 24, 48)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)
geo_4	geo_3	conv3d	(32, 48, 24, 48)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
geo_5	geo_4	conv3d	(32, 48, 24, 48)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
geo_6	geo_5	conv3d	(32, 48, 24, 48)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)
geo_7	geo_6	conv3d	(32, 48, 24, 48)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
geo_8	geo_7	conv3d	(64, 24, 12, 24)	(2, 2, 2)	(2, 2, 2)	(0, 0, 0)
geo_9	geo_1	conv3d	(32, 24, 12, 24)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
geo_10	geo_2	conv3d	(32, 24, 12, 24)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)
geo_11	geo_3	conv3d	(64, 24, 12, 24)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
geo_12	geo_4	conv3d	(32, 24, 12, 24)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
geo_13	geo_5	conv3d	(32, 24, 12, 24)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)
geo_14	geo_6	conv3d	(64, 24, 12, 24)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
color_1	projected 2D features	conv3d	(64, 48, 24, 48)	(2, 2, 2)	(2, 2, 2)	(0, 0, 0)
color_2	color_1	conv3d	(32, 48, 24, 48)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
color_3	color_2	conv3d	(32, 48, 24, 48)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)
color_4	color_3	conv3d	(64, 48, 24, 48)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
color_5	color_4	maxpool3d	(64, 48, 24, 48)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)
color_6	color_5	conv3d	(64, 24, 12, 24)	(2, 2, 2)	(2, 2, 2)	(0, 0, 0)
color_7	color_6	conv3d	(32, 24, 12, 24)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
color_8	color_7	conv3d	(32, 24, 12, 24)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)
color_9	color_8	conv3d	(64, 24, 12, 24)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
color_10	color_9	maxpool3d	(64, 24, 12, 24)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)
concat_1	(geo_14, color_10)	concat	(128, 24, 12, 24)	None	None	None
combine_1	concat_1	conv3d	(128, 24, 12, 24)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)
combine_2	combine_1	conv3d	(64, 24, 12, 24)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
combine_3	combine_2	conv3d	(64, 24, 12, 24)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)
combine_4	combine_3	conv3d	(128, 24, 12, 24)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
combine_5	combine_4	conv3d	(64, 24, 12, 24)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
combine_6	combine_5	conv3d	(64, 24, 12, 24)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)
combine_7	combine_6	conv3d	(128, 24, 12, 24)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
combine_8	combine_7	maxpool3d	(128, 24, 12, 24)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)
rpn_1	combine_7	conv3d	(256, 24, 12, 24)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)
rpn_cls_1	rpn_1	conv3d	(6, 24, 12, 24)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
rpn_bbox_1	rpn_1	conv3d	(18, 24, 12, 24)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
rpn_2	combine_5	conv3d	(256, 24, 12, 24)	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)
rpn_cls_2	rpn_2	conv3d	(22, 24, 12, 24)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
rpn_bbox_2	rpn_2	conv3d	(66, 24, 12, 24)	(1, 1, 1)	(1, 1, 1)	(0, 0, 0)
cls_1	combine_7	FC	128x4x4x4 $\rightarrow$ 256	None	None	None
cls_2	cls_1	FC	256 $\rightarrow$ 256	None	None	None
cls_3	cls_2	FC	256 $\rightarrow$ 128	None	None	None
cls_cls	cls_3	FC	128 $\rightarrow N_{cls}$	None	None	None
cls_bbox	cls_3	FC	128 $\rightarrow N_{cls} \times 6$	None	None	None
mask_1	TSDF	conv3d	(64, 96, 48, 96)	(3, 3, 3)	(1,1,1)	(1,1,1)
mask_2	mask_1	conv3d	(64, 96, 48, 96)	(3, 3, 3)	(1,1,1)	(1,1,1)
mask_3	mask_2	conv3d	(64, 96, 48, 96)	(3, 3, 3)	(1,1,1)	(1,1,1)
mask_4	mask_3	conv3d	(64, 96, 48, 96)	(3, 3, 3)	(1,1,1)	(1,1,1)
mask_5	mask_4	conv3d	(64, 96, 48, 96)	(3, 3, 3)	(1,1,1)	(1,1,1)
mask_6	mask_5	conv3d	(N_{cls}, 96, 48, 96)	(1, 1, 1)	(1,1,1)	(0,0,0)

Table 3: 3D-SIS network architecture layer specifications.

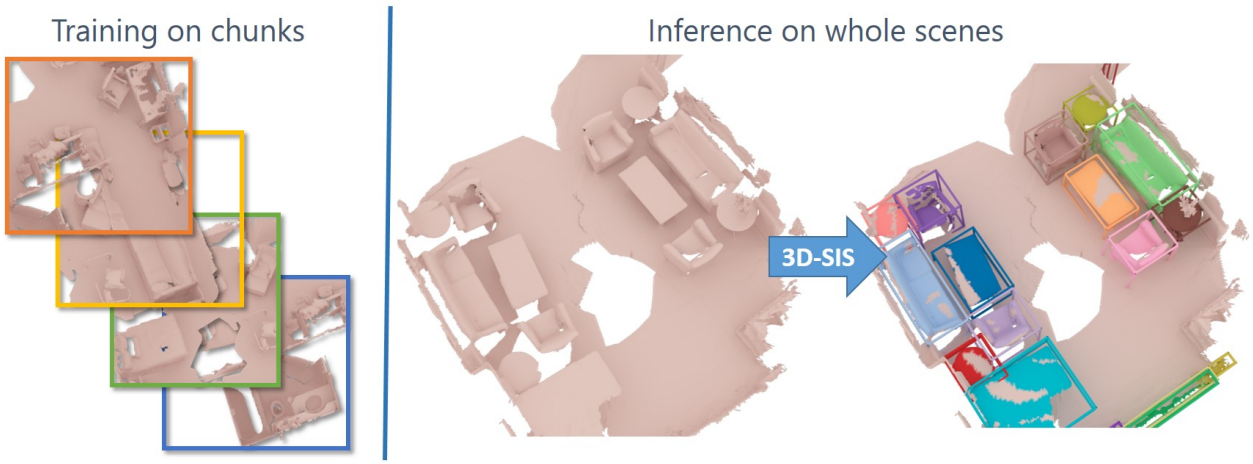

Figure 1: 3D-SIS trains on chunks of a scene, and leverages fully-convolutional backbone architectures to enable inference on a full scene in a single forward pass, producing more consistent instance segmentation results.

	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn	**avg**
Seg-Cluster	10.4	11.9	15.5	12.8	12.4	10.1	10.1	10.3	0.0	11.7	10.4	11.4	0.0	13.9	17.2	11.5	14.2	10.5	10.8
Mask R-CNN [2]	11.2	10.6	10.6	11.4	10.8	10.3	0.0	0.0	**11.1**	**10.1**	0.0	10.0	12.8	0.0	18.9	13.1	11.8	11.6	9.1
SGPN [4]	10.1	16.4	20.2	20.7	14.7	11.1	**11.1**	0.0	0.0	10.0	10.3	12.8	0.0	0.0	48.7	16.5	0.0	0.0	11.3
Ours(geo only)	11.5	17.5	18.0	26.3	0.0	10.1	0.0	10.3	0.0	0.0	0.0	0.0	**24.4**	21.5	25.0	**17.2**	34.9	10.1	12.6
Ours(geo+1view)	12.5	15.0	17.8	23.7	0.0	**19.0**	0.0	11.0	0.0	0.0	10.5	11.1	13.0	19.4	22.5	14.0	**40.5**	10.1	13.3
Ours(geo+3views)	14.4	19.9	**48.4**	**37.3**	**16.9**	18.3	0.0	11.0	0.0	0.0	10.5	**13.1**	16.3	15.3	**51.3**	13.0	12.9	**13.4**	17.3
Ours(geo+5views)	**19.7**	**37.7**	40.5	31.9	15.9	18.1	0.0	**11.0**	0.0	0.0	**10.5**	11.1	18.5	**24.0**	45.8	15.8	23.5	12.9	**18.7**

Table 4: 3D instance segmentation on real-world scans from ScanNetV2 [1]. We evaluate the mean average precision with IoU threshold of 0.5 over 18 classes. Our explicit leveraging of the spatial mapping between the 3D geometry and color features extracted through 2D convolutions enables significantly improved instance segmentation performance.

	cab	bed	chair	sofa	tabl	door	wind	bkshf	cntr	desk	shlf	curt	drsr	mirr	tv	nigh	toil	sink	lamp	bath	ostr	ofurn	oprop	**avg**
Seg-Cluster	10.1	10.9	10.4	10.1	10.3	0.0	0.0	12.9	10.7	15.2	0.0	0.0	10.0	0	0.0	11.2	26.1	12.1	0	16.5	0	0	10	7.7
Mask R-CNN [2]	0.0	10.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**10.8**	**11.4**	10.8	18.8	13.5	0.0	11.5	0.0	0.0	10.7	4.3
SGPN [4]	15.3	28.7	23.7	29.7	17.6	15.1	**15.4**	0.0	10.8	16.0	0.0	**10.9**	0.0	0.0	0.0	12.3	33.7	25.9	**19.2**	31.7	0.0	10.4	10.5	14.2
Ours(geo only)	12.6	**60.5**	38.6	45.8	21.8	**16.8**	0.0	0.0	10.0	18.5	10.0	0.0	14.0	0.0	0.0	**14.9**	**64.2**	30.8	17.6	35.2	10.0	0.0	**16.9**	19.1
Ours(geo+1view)	13.9	42.4	35.3	**52.9**	22	10	0.0	**35.0**	**13.4**	**21.4**	10.0	0.0	13.5	0.0	0.0	10.0	33.8	29.2	**17.7**	**48.3**	10.0	16.9	10.0	19.4
Ours(geo+3views)	15.4	58.5	35.5	34.5	**24.4**	16.6	0.0	20.0	10.0	17.6	10.0	0.0	**24.3**	0.0	10.0	10.0	34.6	28.5	15.6	40.7	10.0	**24.9**	15.5	19.8
Ours(geo+5views)	**15.5**	43.6	**43.9**	48.1	20.4	10.0	0.0	30.0	10.0	17.4	**10.0**	0.0	14.5	0.0	10.0	10.0	53.5	**35.1**	17.2	39.7	**10.0**	18.9	16.2	**20.6**

Table 5: 3D instance segmentation on synthetic scans from SUNCG [3]. We evaluate the mean average precision with IoU threshold of 0.5 over 23 classes. Our joint color-geometry feature learning enables us to achieve more accurate instance segmentation performance.

mance. We also submit our model the ScanNet Benchmark, and we achieve the state-of-the-art in all three metrics.

We run an additional ablation study to evaluate the impact of the RGB input and the two-level anchor design; see Table. 6.

	mAP@0.5	mAP@0.25
3D-SIS (only color-1view)	9.4	30.5
3D-SIS (only color-3view)	16.5	35.0
3D-SIS (only color-5view)	17.4	35.7
3D-SIS (only geometry)	16.0	27.6
3D-SIS (one anchor layer)	12.2	33.4
3D-SIS (final)	**22.5**	**40.2**

Table 6: Additional ablation study on ScanNetV2; combination of geometry and color signal complement each other, thus achieving the best performance.

4. Limitations

While our 3D instance segmentation approach leveraging joint color-geometry feature learning achieves marked performance gain over state of the art, there are still several important limitations. For instance, our current 3D bounding box predictions are axis-aligned to the grid space of the 3D environment. Generally, it would be beneficial to additionally regress the orientation for object instances; e.g., in the form of a rotation angle. Note that this would need to account for symmetric objects where poses might be ambiguous. At the moment, our focus is also largely on indoor environments as we use commodity RGB-D data such as a Kinect or Structure Sensor. However, we believe that the idea of taking multi-view RGB-D input is agnostic to this

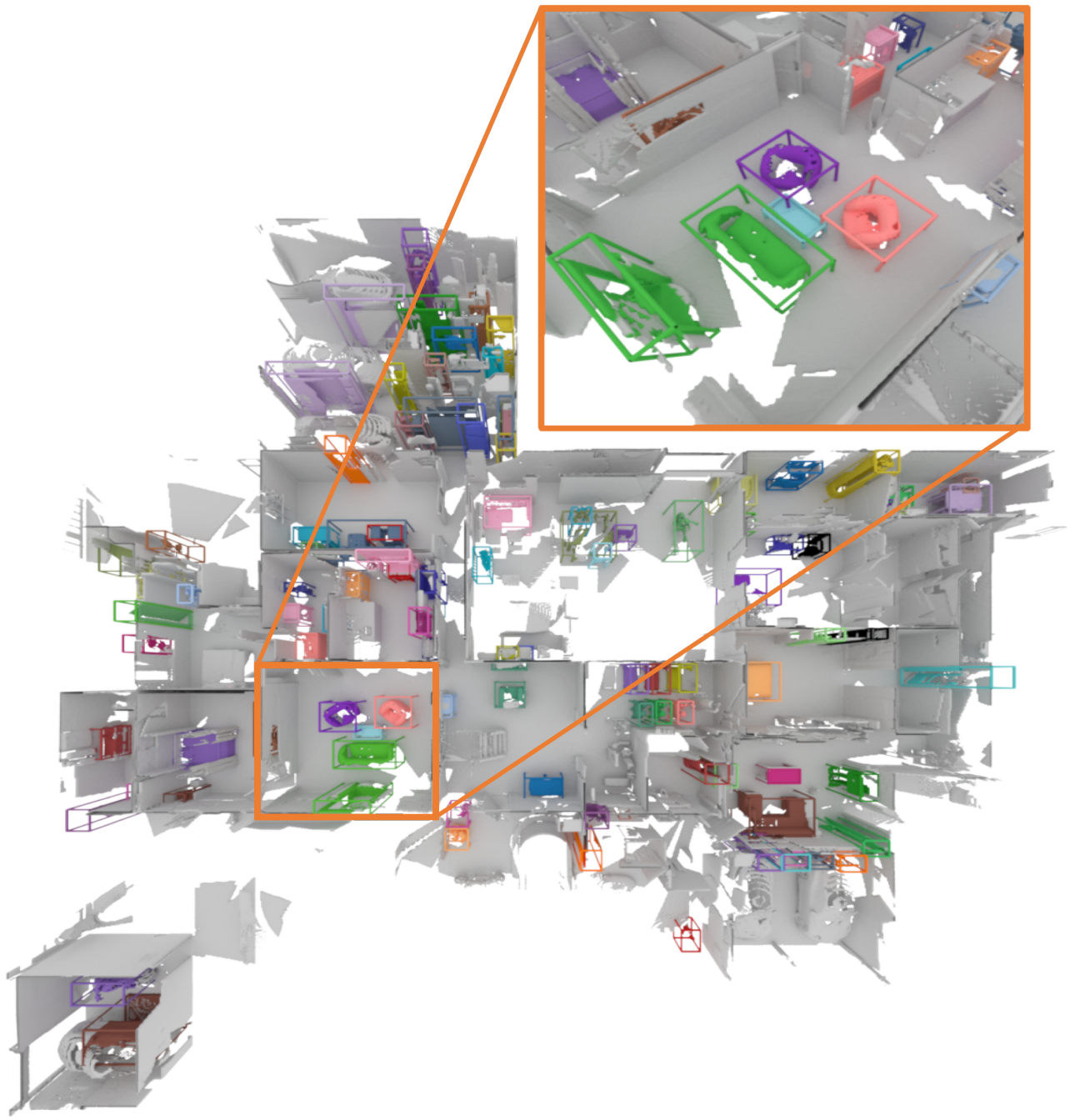

Figure 2: Our fully-convolutional architectures allows testing on a large SUNCG scene (45m x 45m) in about 1 second runtime.

specific setting; for instance, we could very well see applications in automotive settings with LIDAR and panorama data. Another limitation of our approach is the focus on static scenes. Ultimately, the goal is to handle dynamic or at least semi-dynamic scenes where objects are moving, which we would want to track over time. Here, we see a significant research opportunities and a strong correlation to tracking and localization methods that would benefit from semantic 3D segmentation priors.

References

[1] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Com-*

puter Vision and Pattern Recognition (CVPR), IEEE, 2017. 1, 3

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. 3

[3] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 3

[4] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2569–2578, 2018. 3