Supplementary Materials for DeepContext: Context-Encoding Neural Pathways for 3D Holistic Scene Understanding

Yinda Zhang[1] Mingru Bai[1] Pushmeet Kohli[2,5] Shahram Izadi[3,5] Jianxiong Xiao[1,4]

[1]Princeton University [2]DeepMind [3]PerceptiveIO [4]AutoX [5]Microsoft Research

In this supplementary material, we show detailed network structure, precision-recall curves for more object categories, examples of synthetic data, and visualization of initial alignment.

1. Network Structure

The transformation network (rotation and translation) and the scene pathway shares the similar network structure, which consists of several 3D convolution layers followed by fully connected layers. The network structure is visualized in Figure 1. Respectively for scene classification, rotation estimation, and translation estimation, the corresponding size of **fc_scene_cls** would be 8, 36, and 726. The size and receptive field of each weight and response in scene pathway is shown in Table 1.

Figure 2 shows network structure for the object pathway. For simplicity, we show a network for a scene template with only 2 objects. For a template used in our experiment which usually contains up to 15 objects, there would be more similar object pathway in parallel attached to the scene pathway. The object pathway starts from ROI pooling from **conv2** layer. Then all the pooled features for different objects will pass through 2 shared convolution layers (**conv_roi**) and 2 shared fully connected layers (**fc_object_roi**), and eventually be converted to 128-dim feature vectors. This feature will be concatenated with the scene feature (**fc6p**) and fed into each object's individual network for classification and regression. For ease of understanding, we include the scene pathway on the left start from **tsdf**. The size and receptive field of each weight and response in object pathway is shown in Table 2.

Besides the parameters shown in Table 1 and Table 2, we adopt the following setting to all the layers:

- **Convolution Layer.** The stride equals to 1, and there is no dilation. The padding is done to make sure the size of the output is the same as that of the input.
- **Pooling Layer.** The stride is set to 2. The kernel size is $2 \times 2 \times 2$. Therefore, each pooling layer reduces the resolution of feature by half.
- **Dropout Layer.** The dropout ratio is set to 0.5.

2. Precision Recall Curves

The precision recall curves of more object categories on the testing set, which contains 361 images that can be classified as one of the scene templates, are shown in Figure 3. For all the other object categories in our templates but not shown in the figure, the average precision for both the baseline and our method are close to 0. Note that some categories are not used by [1], and therefore we did not include their performance in those categories.

Figure 4 further shows precision recall curves of more cateogries on the extended testing set, which contains 2000 images randomly selected from SUNRGBD dataset [2].

3. Synthetic Data

Figure 5 shows more examples of our synthetic data. Figure 5 (a) shows multiple CAD model replacements in an image from SUN-RGBD dataset. (b) shows more examples of raw depth images with synthesized depth images.

4. Transformation Alignment

Figure 6 shows some alignment results. Below each image is the original point cloud in camera coordinates (top view), and the aligned point cloud, overlaid with ground truth in the template coordinates. The red cross marks the origin of the scene template coordinates, which is supposed to be at the center of the main object. The algorithm sometimes makes mistakes in recognizing the main object (the failure cases in lounging area and table & chair set). Special view points (the failure case for sleeping area) and situations that disobey the Manhattan world assumption (the failure case for office area) also cause confusion.

Layer	Responses	Receptive Field (m)	Receptive Gap (m)
SceneHolisticTSDF	tsdf[5]={24,3,64,128,128} scene_cls[5]={24,1,1,1,1}	[0.05,0.05,0.05]	[0.05,0.05,0.05]
conv1 weight[5]={64,3,5,5,5} relu1 pool1	conv1[5]={24,64,64,128,128} pool1[5]={24,64,32,64,64}	[0.25,0.25,0.25] [0.3,0.3,0.3]	[0.05,0.05,0.05] [0.1,0.1,0.1]
conv2 weight[5]={128,64,3,3,3} relu2 pool2	conv2[5]={24,128,32,64,64} pool2[5]={24,128,16,32,32}	[0.5,0.5,0.5] [0.6,0.6,0.6]	[0.1,0.1,0.1] [0.2,0.2,0.2]
conv3 weight[5]={256,128,3,3,3} relu3 pool3	conv3[5]={24,256,16,32,32} pool3[5]={24,256,8,16,16}	[1,1,1] [1.2,1.2,1.2]	[0.2,0.2,0.2] [0.4,0.4,0.4]
conv4 weight[5]={256,256,3,3,3} relu4 pool4	conv4[5]={24,256,8,16,16} pool4[5]={24,256,4,8,8}	[2,2,2] [2.4,2.4,2.4]	[0.4,0.4,0.4] [0.8,0.8,0.8]
conv5 weight[5]={256,256,3,3,3} relu5	conv5[5]={24,256,4,8,8}	[4,4,4]	[0.8,0.8,0.8]
fc4p_r weight[2]={1024,65536} relu4p drop4p	fc4p[5]={24,1024,1,1,1}	[6.4,9.6,9.6]	[0,0,0]
fc6p_r weight[2]={128,1024} relu6p drop6p	fc6p[5]={24,128,1,1,1}	[6.4,9.6,9.6]	[0,0,0]
fc_cls_tsl weight[2]={726,128} loss_tsl fc_cls_rot weight[2]={36,128} loss_rot fc_cls_scn weight[2]={8,128} loss_scn	fc_cls_tsl[5]={24,726,1,1,1} fc_cls_rot[5]={24,36,1,1,1} fc_cls_scn[5]={24,8,1,1,1}	[6.4,9.6,9.6]	[0,0,0]

Table 1: **Transformation and Scene Pathway Network Parameters.**

	Layer	Responses
Shared	SceneHolisticTSDF	cls_val[5]={24,1,1,1,1} reg_val[5]={24,1,1,1,1}, for wall reg_val[5]={24,6,1,1,1}, for object
	roi_pool source: conv2	roi[5]={24,128,6,6,6}
	conv_roi_1 weight[5]={128,128,3,3,3} relu_roi_1	roiconv_1[5]={24,128,6,6,6}
	conv_roi_2 weight[5]={64,128,3,3,3} relu_roi_2	roiconv_2[5]={24,64,6,6,6}
	fc_object_roi_1 weight[2]={256,13824} relu_roi_fc_1	roifc_1[5]={24,256,1,1,1}
	fc_object_roi_2 weight[2]={128,256} relu_roi_fc_2	roifc_2[5]={24,128,1,1,1}
Independent	concat with fc6p	cat[5]={24,256,1,1,1}
	fc_cat_1 weight[2]={64,256} relu_cat	fc_cat_1[5]={24,64,1,1,1}
	fc_cat_cls weight[2]={2,64} loss_cls	fc_cat_cls[5]={24,2,1,1,1}
	fc_cat_reg weight[2]={1,64}, for wall fc_cat_reg weight[2]={6,64}, for object loss_reg	fc_cat_reg[5]={24,1,1,1,1} fc_cat_reg[5]={24,6,1,1,1}

Table 2: **Object Pathway Network Parameters.**

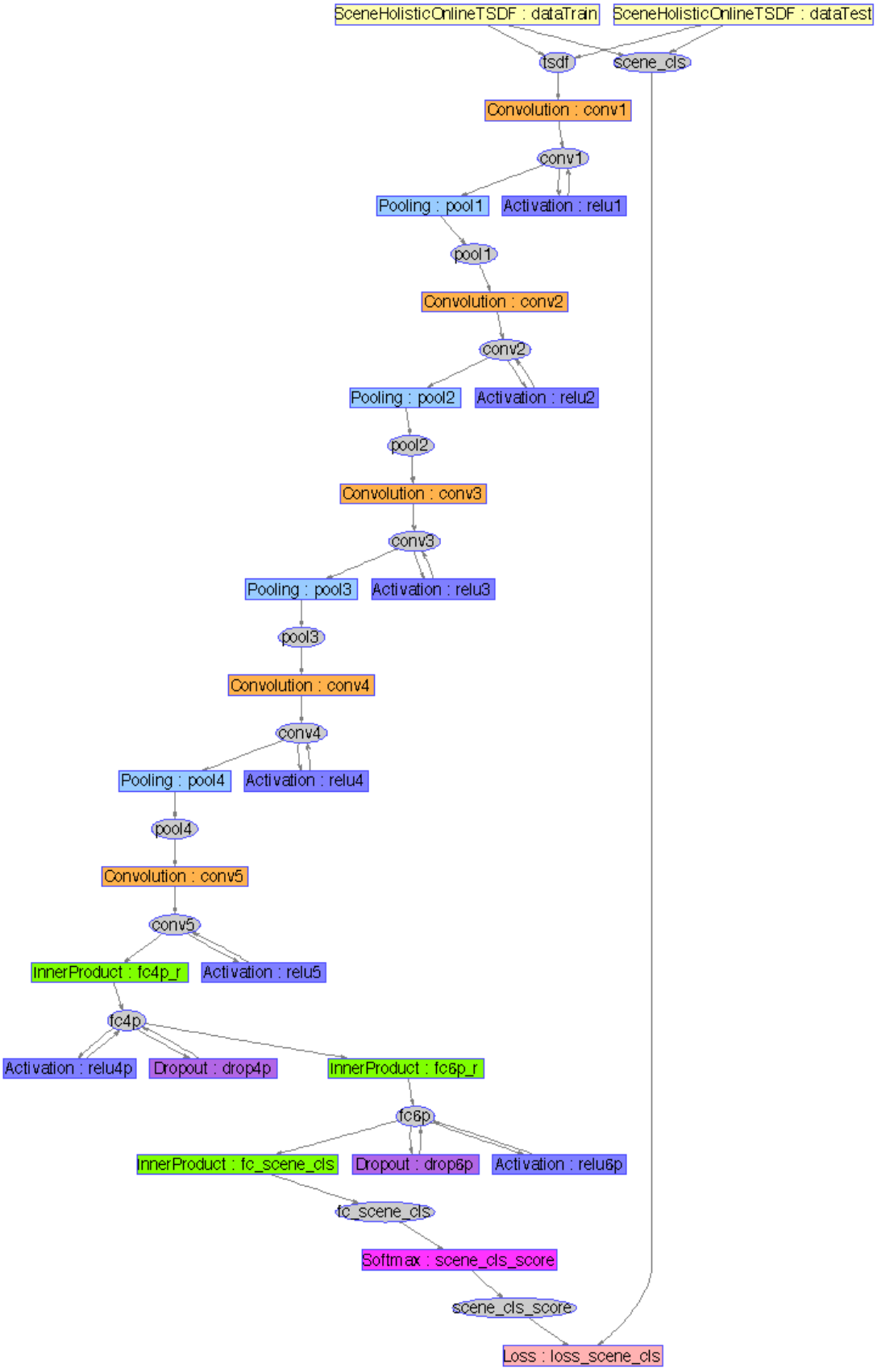


Figure 1: **Transformation and Scene Pathway Network Structure.**

Figure 2: **Object Pathway Network Structure.** We show a network for a template with only 2 objects as an example. Usually a template contains around 15 objects, which will contain more parallel object pathways. The object pathway starts from ROI pooling from **conv2** layer. Then all the pooled features for different objects will pass through 2 shared convolution layers (**conv_roi**) and 2 shared fully connected layers (**fc_object_roi**), and be converted to 128-dim feature vectors. This feature will be concatenated with scene feature (**fc6p**) and fed into each object's individual network for classification and regression.

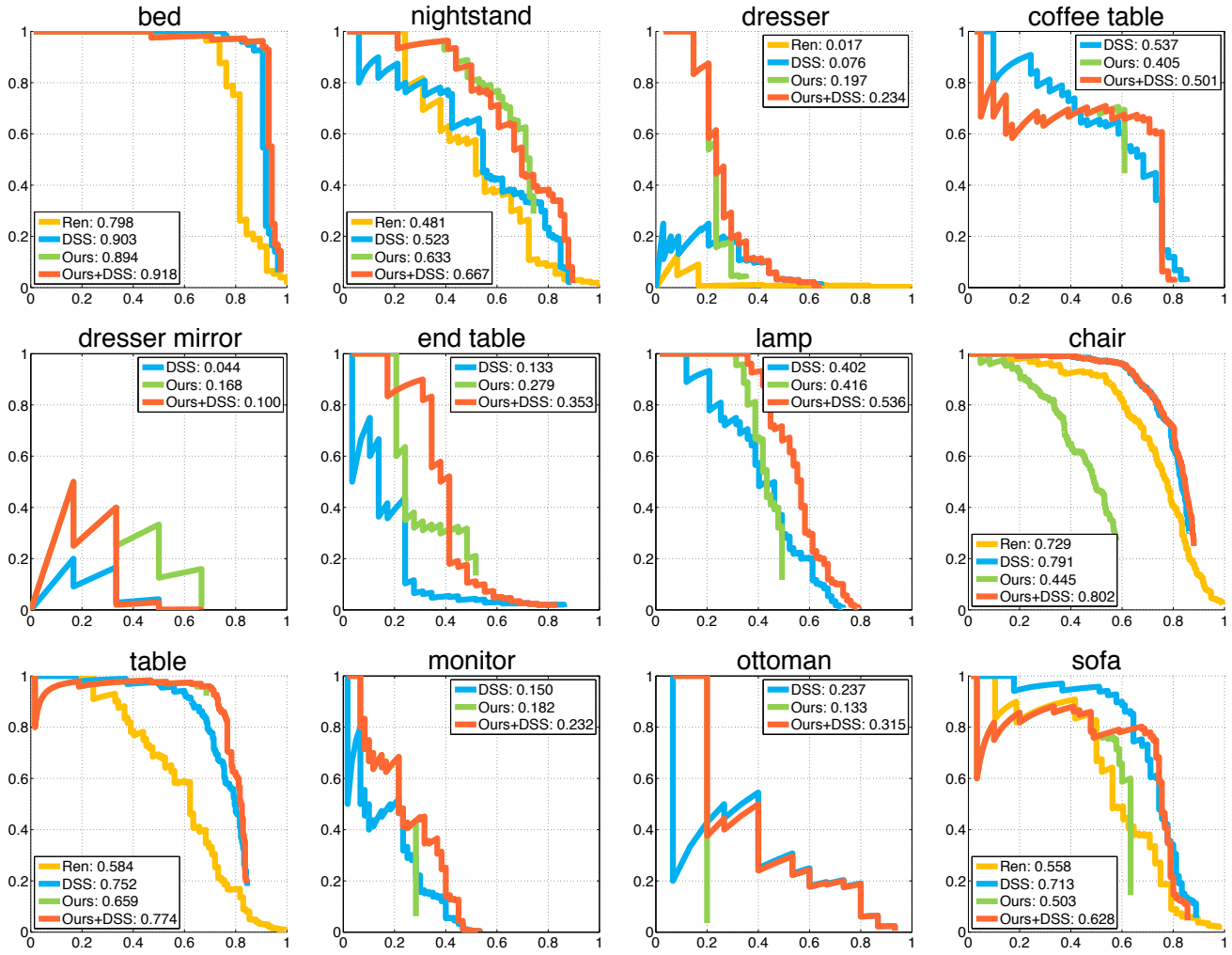


Figure 3: **Precision-recall Curve for 12 object categories on testing set containing 361 images that can be classified as one of the scene templates.** The number in the legend is the average precision. **Ren [1] & DSS [3]** are the performances of two state of the art methods mentioned in the paper. **Ours** is the performance of our method. **Ours+DSS** is the performance of merging our result with Deep Sliding Shape. We can see that: (1) our method achieves comparable performance with the state of the art methods, (2) our method provides complementary information against local appearance based detector since the merged result outperforms all the individual methods.

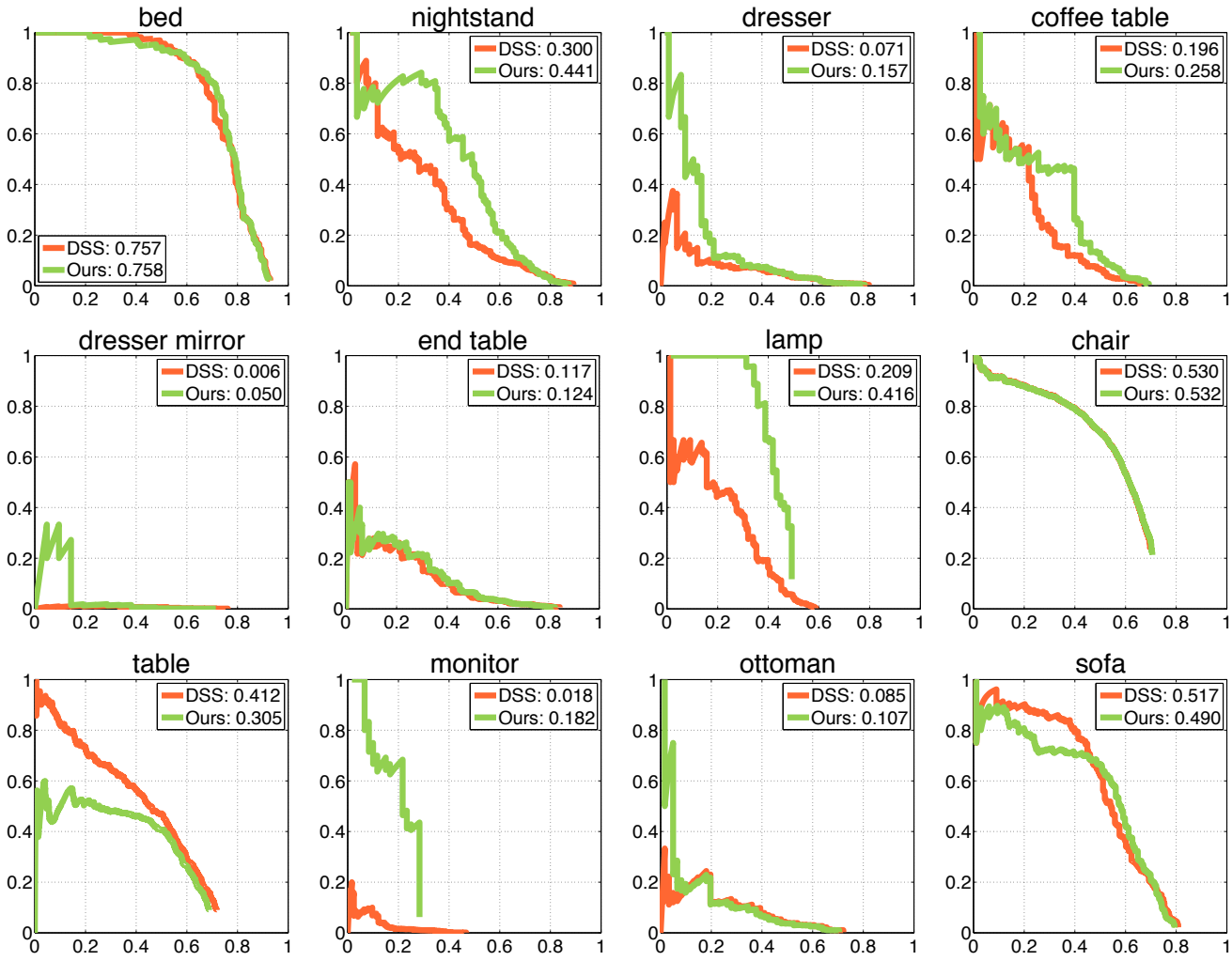
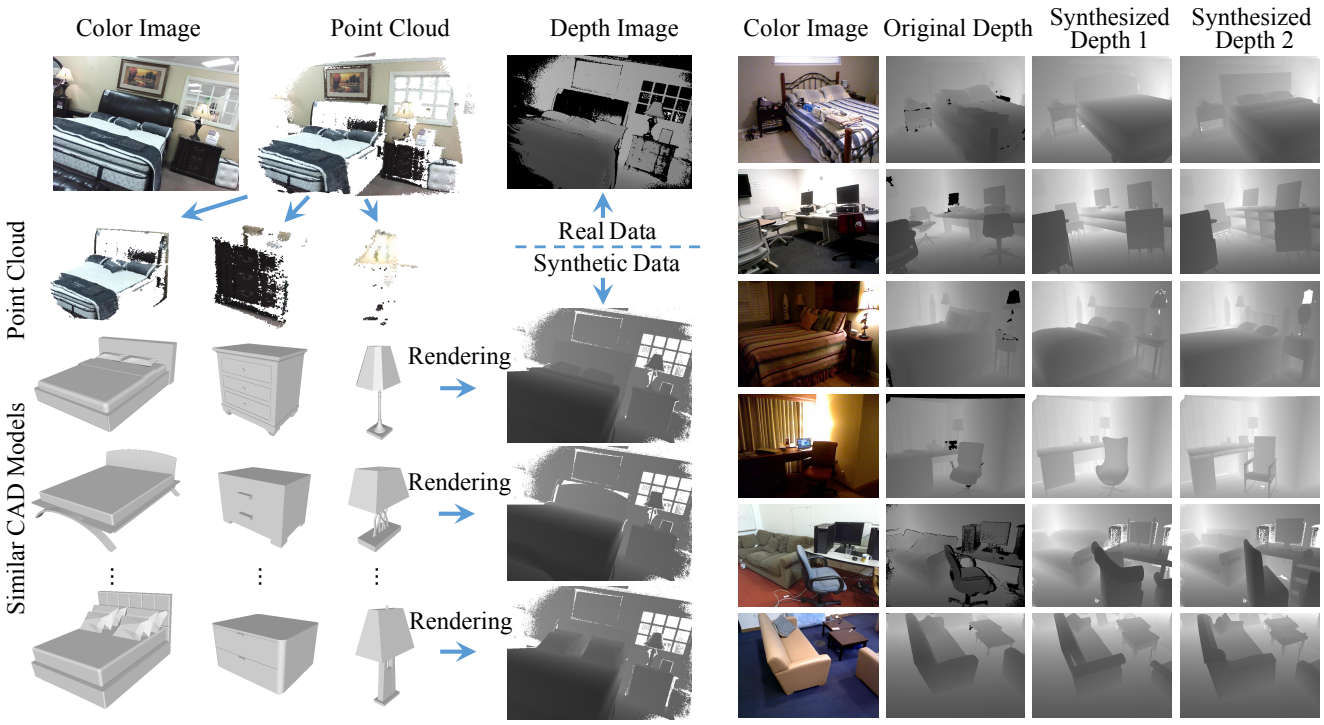


Figure 4: **Precision-recall Curve for 12 object categories on testing set containing 2000 images that are randomly chosen from SUNRGBD [2] dataset.** The number in the legend is the average precision. **DSS [3]** is the performance with Deep Sliding Shape. **Ours** is the performance of merging out result with the Deep Sliding Shape.



(a) The pipeline of synthesizing hybrid data (b) Examples of synthesized data.

Figure 5: **Synthetic Data.** We show more examples of model replacement in the image shown in paper. We also show more synthetic data generated from other scenes.

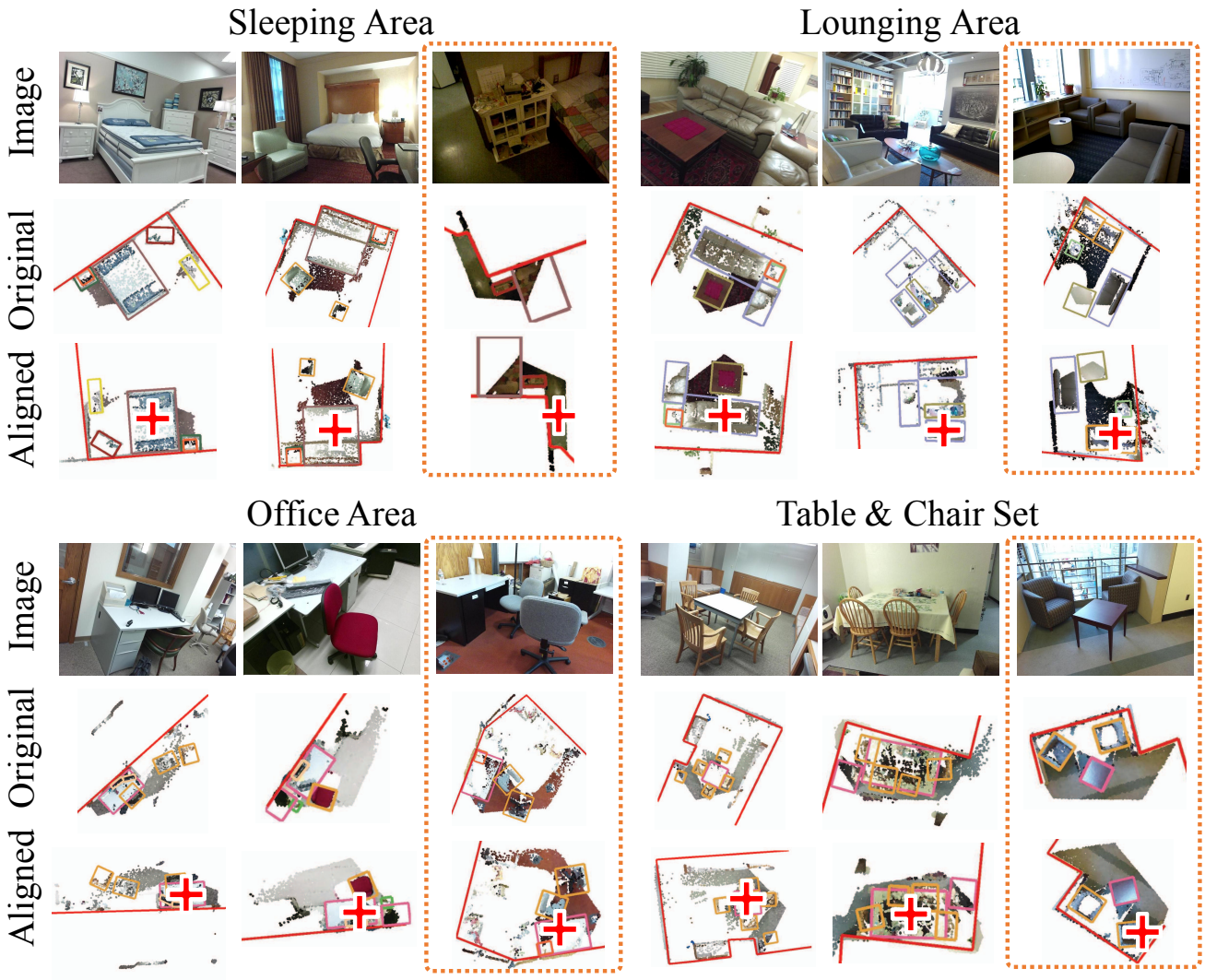


Figure 6: **Initial Alignment Result.** For each scene category, we show two successful alignment results and one failure case (in the right most dashed box). Below each image, we show the point cloud overlaid with the ground truth in original camera coordinates, followed by the aligned result according to the rotation and translation estimated by our network. The red cross marks the origin of the new coordinates, which is expected to be perfect if locates at the center of the major object of each scene.

References

[1] Z. Ren and E. B. Sudderth. Three-dimensional object detection and layout prediction using clouds of oriented gradients. In *CVPR*, 2016. 1, 6

[2] S. Song, S. Lichtenberg, and J. Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *CVPR*, 2015. 1, 7

[3] S. Song and J. Xiao. Deep Sliding Shapes for amodal 3D object detection in RGB-D images. In *CVPR*, 2016. 6, 7