Recurrent Scene Parsing with Perspective Understanding in the Loop
Supplementary Material

Shu Kong Charless Fowlkes
Department of Computer Science
University of California, Irvine
Irvine, CA 92697, USA
{skong2, fowlkes}@ics.uci.edu

project page with code and demo

In this supplementary material, we first provide additional analysis of the proposed gating module on the Cityscapes dataset [3], comparing quantitatively the performance of models with tied/untied weights, average gating or soft-weighted gating using either the learned attention or depth (ground-truth or estimated from monocular input). Then, we display more examples of SUN-RGBD dataset [4], including the depth prediction and semantic segmentation in the recurrent loops. Finally, we show more qualitative results on Cityscapes and Stanford-2D-3D datasets [1] including results of gating using learned attention map and depth adaptation within the recurrent loops.

1. Analysis of Depth-aware Gating Module

In this section, we analyze the proposed depth-aware gating module with detailed results in Table 1. We perform the ablation study on the Cityscapes dataset [3]. Specifically, we train the following models sequentially, initializing from the previous in order (except the fourth model which learns attention to gate).

1. "baseline" is our DeepLab-like baseline model which adds two convolutional (with 3×3 kernels) layers above the ResNet101 backbone.

2. "tied, avg." is the model we train based on "baseline" by using the same 3×3 kernel but with parallel branches using different dilation rates equal to $\{1, 2, 4, 8, 16\}$, respectively. The kernel weights in the five branches are tied so in order to make processing scale-invariant. We average the resulting feature maps for the final output prior to classification.

3. "gt-depth, tied, gating" uses the quantized ground-truth depth map to select which of the five branches is used at each spatial location; the pooling window size is determined according to the inverse of the ground-truth depth value.

4. "gt-depth, untied, gating" is the same structure as "gt-depth, tied, gating" but unleashing the tied kernels in the five branches. These untied kernels improve the flexibility and representation power of the network. Figure 1 (a) depicts this model.

5. "attention, untied, gating" is trained independently from the previous models and is trained without any depth supervision loss on the gating signal. Instead, the gating acts as a generic attentional signal that modulates spatially adaptive pooling. Specifically, we train an attention branch to produce a soft weighted combination of the features from multiple pooling at different scales (softmax followed by element-wise multiplication) We also adopt untied weights for the scale-specific pooling branches. The architecture is similar to that depicted in Figure 1 (b), but without any depth supervision.

6. "pred-depth, untied, gating" is our final model in which we learn a quantized depth predictor to gate the five branches which is supervised during training with the depth loss and then fine-tuned. This model determines the size of pooling window based on its predicted depth map. Figure 1 (b) shows the architecture of this model.

Quantitative evaluation is shown in Table 1 and highlight the nIoU results which specifically benchmark performance on dynamic objects. We can see that averaging multiple dilated versions of the kernel with our model "tied, avg." improves the performance noticeably over baseline. This is consistent with the observation in [2], in which the large view-of-field version of DeepLab performs better. The benefit can be explained by the large dilation rate increasing the size of the receptive field, allowing more contextual information to be captured at higher levels of the network. With the gating mechanism, either using ground-truth depth map

or the predicted one, the performance is improved further over non-adaptive pooling. The depth-aware gating module helps determine the pooling window size wisely, which is better than averaging all branches equally as in our "tied, avg." model and DeepLab. Moreover, by unleashing the tied kernels, the "gt-depth untied, gating" improves over "gt-depth, tied, gating" remarkably. We conjecture that this is because the untied kernels provide more flexibility to distinguish features at different scales and allow selection of the appropriate non-invariant features from lower in the network.

Interestingly, the attention-gating model performs well, but using the predicted depth map achieves the best among all these compared models. We attribute this to three reasons. Firstly, unlike ground-truth depth, the predicted depth is smooth without holes or invalid entries. When using ground-truth depth on Cityscapes dataset, we assign equal weight on the missing entries so that the gating actually averages the information at different scales. This average pooling might be harmful in some cases such as very small object at distance. This can be taken as complementary evidence that the blindly averaging all branches achieves inferior performance to using the depth-aware gating. Secondly, the predicted depth maps have some object-aware pattern structure, which might be helpful for segmentation. From the visualization shown later in Figure 4, we can observe such patterns, e.g. for cars. When trained without depth-supervision, the attention map (also in Figure 4) discovers a different strategy which uses small pooling regions near object boundaries. Thirdly, the depth prediction branch, as well as the attention branch, generally increases the representational power and flexibility of the whole model which is beneficial for segmentation when sufficient training data is available to avoid overfitting.

2. Results on the SUN-RGBD dataset

In Figure 2, we show the depth prediction results of several images randomly picked from the test set of SUN-RGBD. Note that the there are unnatural regions in the ground-truth depth maps, which are the result of refined depth completion by the algorithm in [4]. Visually, these regions do not always make sense and constitute bad depth completions. In contrast, our predicted depth maps are much smoother than the ground-truth. We also evaluate our depth prediction on SUN-RGBD dataset, and achieve 0.754, 0.899 and 0.961 by the three threshold metrics respectively. As SUN-RGBD is an extension of NYU-depth-v2 dataset, it has similar data statistics resulting in similar prediction performance.

In Figure 3, we show fourteen randomly selected images and their segmentation results at loops of the recurrent refining module. Visually, we can see that the our recurrent module refines the segmentation result in the loops.

3. Visualization on Large Perspective Images

In Figure 4 and 5, we visualize more results on Cityscapes and Stanford-2D-3D datasets, respectively. First, we show the segmentation prediction and the attention map after training with the unsupervised attentional mechanism in the third column. We can see the attention map appears to encode the distance from object boundaries. We hypothesize this selection mechanism serves to avoid pooling features across different semantic segments while still utilizing large pooling regions within each region. This is understandable and desirable in practice, as per-pixel feature vectors have different feature statistics for different categories.

We also compare the segmentation results and depth estimate for adaptation in the recurrent refinement loops (last three columns in Figure 4 and 5). We notice that the depth estimate for adaptation changes remarkably in the loop (the depth module is fine-tuned using the segmentation loss only in training). In the Cityscapes dataset, the depth estimate improves quantitatively over iterations, particularly for semantic objects such as cars. However, in the Stanford-2D-3D dataset, the average pooling size selected decreases over the iterations in a coarse-to-fine manner. We conjecture that this is due to the "top-down" signal from the depth estimate at the previous loop. The recurrent refinement module also fills the holes in large areas of the predicted label map, such as the light reflection regions on the car in street scene (Cityscapes) and white board in the second image (row 3 and 4) of panoramic photos (Stanford-2D-3D).

References

[1] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese. Joint 2d-3d-semantic data for indoor scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 40(4):834–848, 2018. 1

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[4] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2, 4

Table 1: Result of different depth-aware gating module deployments on Cityscapes dataset. IoU is short for intersection over union averaged over all classes, and nIoU is the weighted IoU through the pre-defined class weights provided by the benchmark.

	baseline		tied, avg.		gt-depth, tied, gating		gt-depth, untied, gating		attention, untied, gating		pred-depth, untied, gating	
	IoU	nIoU	IoU	nIoU	IoU	nIoU	IoU	nIoU	IoU	nIoU		
Score Avg.	**0.738**	**0.547**	**0.747**	**0.554**	**0.748**	**0.556**	**0.753**	**0.561**	**0.754**	**0.558**	**0.759**	**0.571**
road	0.980	–	0.981	–	0.981	–	0.982	–	0.982	–	0.982	–
sidewalk	0.849	–	0.847	–	0.849	–	0.852	–	0.853	–	0.857	–
building	0.916	–	0.917	–	0.918	–	0.919	–	0.923	–	0.920	–
wall	0.475	–	0.499	–	0.506	–	0.511	–	0.527	–	0.512	–
fence	0.596	–	0.605	–	0.605	–	0.611	–	0.618	–	0.614	–
pole	0.598	–	0.599	–	0.604	–	0.616	–	0.615	–	0.624	–
traffic light	0.684	–	0.674	–	0.678	–	0.692	–	0.689	–	0.699	–
traffic sign	0.780	–	0.776	–	0.775	–	0.782	–	0.783	–	0.790	–
vegetation	0.918	–	0.917	–	0.918	–	0.920	–	0.920	–	0.922	–
terrain	0.619	–	0.620	–	0.627	–	0.632	–	0.625	–	0.638	–
sky	0.941	–	0.937	–	0.940	–	0.942	–	0.943	–	0.944	–
person	0.803	0.635	0.803	0.631	0.804	0.639	0.808	0.648	0.804	0.641	0.814	0.659
rider	0.594	0.448	0.595	0.462	0.602	0.460	0.612	0.461	0.602	0.443	0.616	0.473
car	0.939	0.859	0.942	0.854	0.942	0.863	0.942	0.867	0.943	0.862	0.944	0.871
truck	0.631	0.398	0.666	0.421	0.679	0.407	0.679	0.417	0.666	0.424	0.674	0.421
bus	0.759	0.595	0.802	0.607	0.787	0.612	0.786	0.609	0.798	0.602	0.799	0.615
train	0.621	0.467	0.683	0.494	0.656	0.489	0.655	0.487	0.684	0.508	0.687	0.507
motorcycle	0.562	0.396	0.587	0.387	0.591	0.398	0.602	0.410	0.583	0.407	0.610	0.425
bicycle	0.755	0.582	0.747	0.387	0.753	0.575	0.761	0.586	0.760	0.575	0.765	0.594

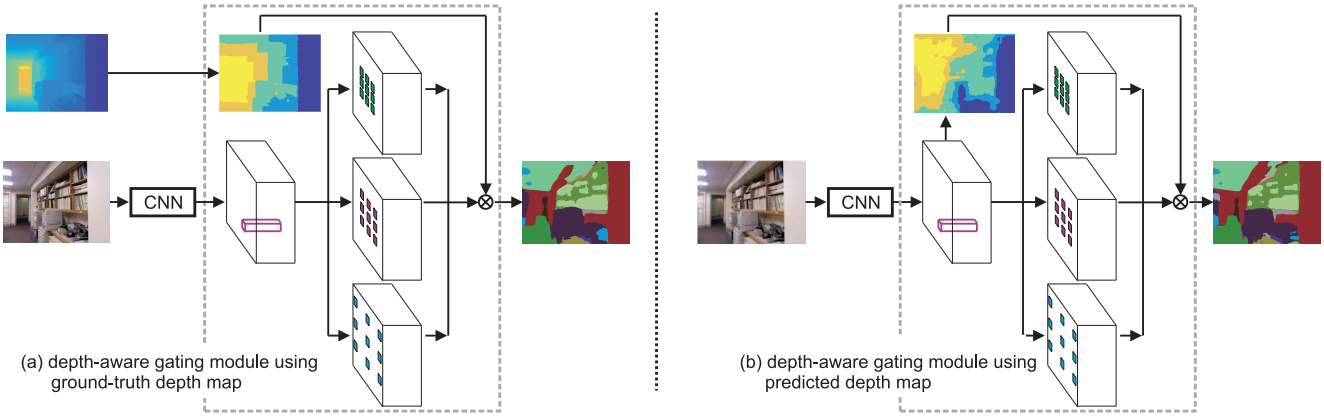

Figure 1: (a) Depth-aware gating module using the ground-truth depth map, and (b) depth-aware gating module using the predicted depth map. The grids within the feature map blocks indicate different pooling field sizes. Here we depict three different pooling window sizes while in our actual experiments we quantize the depth map into five scale bins.

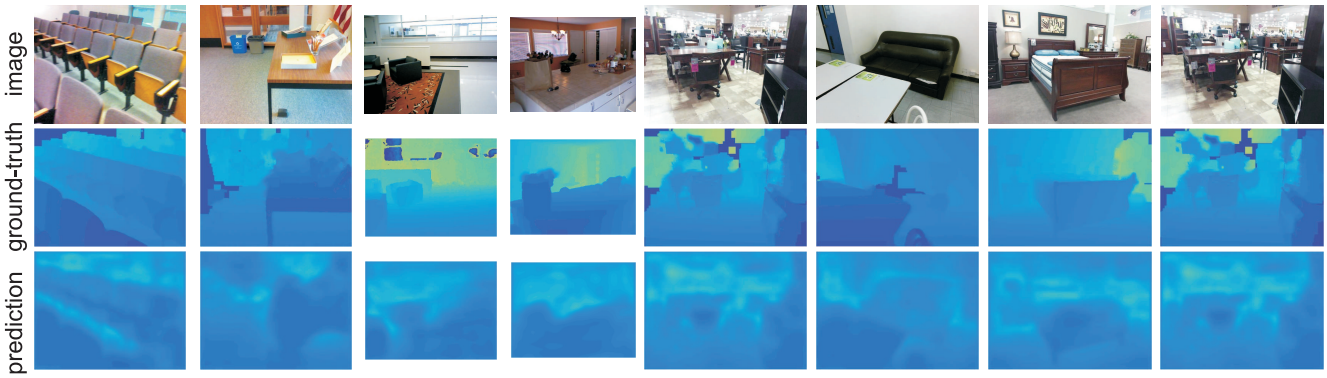

Figure 2: Visualization of images from SUN-RGBD dataset and their ground-truth depth and our predicted depth on the three rows, respectively. We scale all the depth maps into a fixed range of $[0, 10^5]$. In this sense, the color of the depth maps directly reflect the absolute physical depth. Note that there are unnatural regions in the ground-truth depth maps, which have been refined by the algorithm in [4]. Visually, these refined region do not always make sense and are incorrect depth completions. In contrast, our monocular predictions are quite smooth.

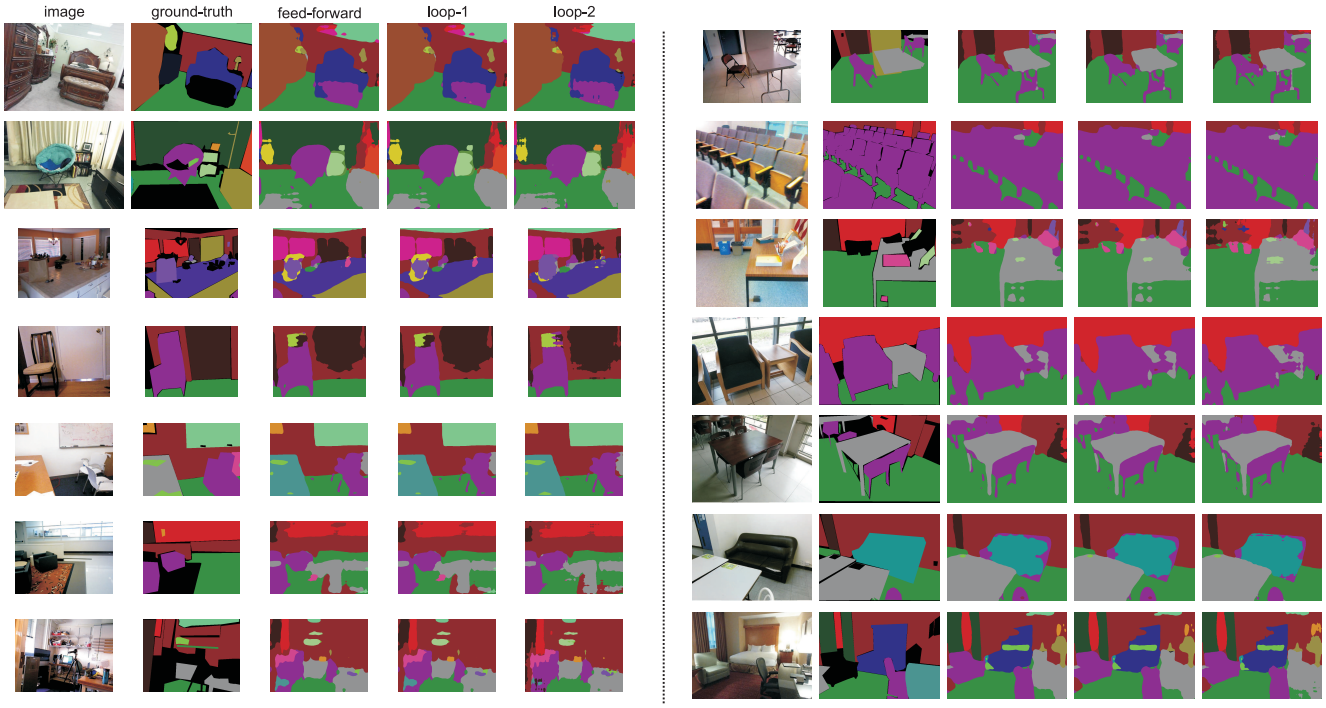

Figure 3: Visualization of the output on SUN-RGBD dataset. We show fourteen randomly selected images from the test set with their segmentation output from both feed-forward pathway and recurrent loops. In the ground-truth segmentation annotation, we can see that there are many regions (with black color) not annotated.

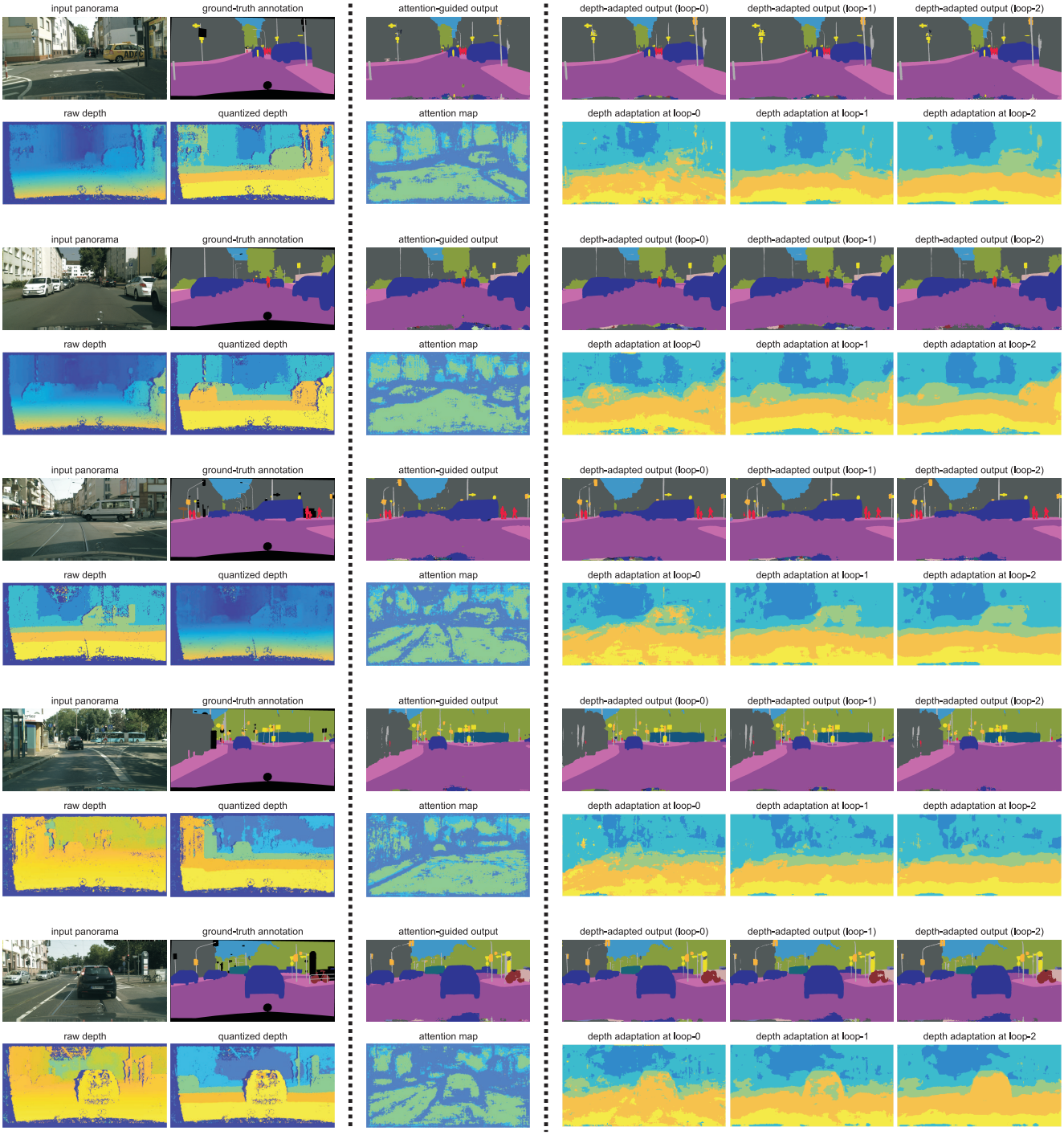

Figure 4: Visualization of the results on Cityscapes dataset. For five random images from the validation set, we show the input perspective street scene photos, ground-truth annotation, raw disparity and the five-scale quantized depth map in the leftmost two columns. Then, we show the segmentation prediction and the attention map using our unsupervised attentional mechanism in the third column. In the remaining three columns, we show the output of our depth-aware adaptation over each iteration of recurrent refinement, from loop-0 to loop-2. Note that the more yellowish the color is, the closer the object is to the camera and the finer scale of the feature maps the model selects to process. From the visualization, we can see 1) the attention map helps the model avoid pooling across semantic segments by using smaller pooling near boundaries; 2) the depth-adaptation in the recurrent refinement loops improves depth prediction for some semantic object categories like the cars. We attribute this to to the top-down signal from previous iterations.

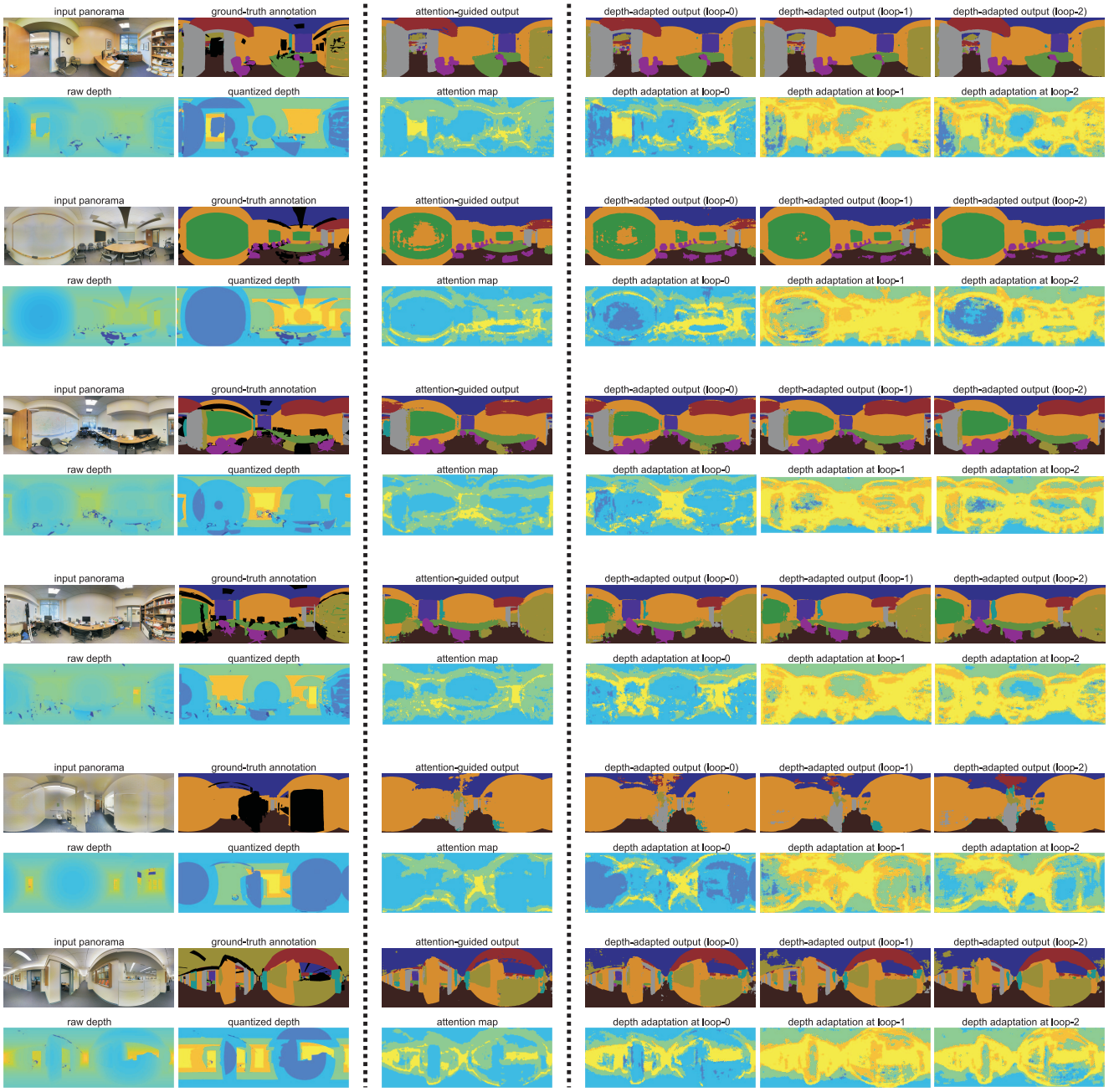

Figure 5: Visualization of the results on Stanford-2D-3D dataset. For six random images from the test set, we show the input panorama, ground-truth annotation, raw depth map and the five-scale quantized depth map in the leftmost two columns. Then, we show the segmentation prediction and the attention map using our unsupervised attentional mechanism in the third column. In the remaining three columns, we show the output of our depth-aware adaptation over each iteration of recurrent refinement, from loop-0 to loop-2. Note that the more yellowish the color is, the further away the object is to camera and the finer scale of the feature maps the model adopts to process. From the visualization, we can see 1) the attention map helps the model avoid pooling across semantic segments by using smaller pooling near boundaries; 2) the depth-adaptation in the recurrent refinement loops behave in a coarse-to-fine manner with smaller receptive fields used in later iterations due to the top-down signal from earlier semantic predictions.