StruMonoNet: Structure-Aware Monocular 3D Prediction Supplementary Materials

Zhenpei Yang[1], Li Erran Li[2,3], Qixing Huang[1]
[1]The University of Texas at Austin,[2]Columbia University,[3]Amazon

1. Technical Details

1.1. Details of Synchronization Module

To simplify the notations, let $\boldsymbol{x}$ collect all the parameters of the planes. Note that we parametrize each plane normal as $\boldsymbol{n} = \frac{\boldsymbol{n}_0+x_1\boldsymbol{t}_1+x_2\boldsymbol{t}_2}{\|\boldsymbol{n}_0+x_1\boldsymbol{t}_1+x_2\boldsymbol{t}_2\|}$, where $(\boldsymbol{n}_0, \boldsymbol{t}_1, \boldsymbol{t}_2)$ denotes a coordinate system. Let θ collect all the relevant network parameters and hyper-parameters. We rewrite the objective function as

$$\min_{\boldsymbol{x}} \sum_{i=1}^{N} w_i(\boldsymbol{x}, \theta) f_i^2(\boldsymbol{x}, \theta) \tag{1}$$

where f_i denote the objective terms that involve adjacent planes, parallel planes, perpendicular planes, and the data terms.

Starting from the initial solution $\boldsymbol{x}^{(0)}$, StruMonoNet solves (1) via reweighted alternating minimization. At iteration k, StruMonoNet first update the term weights as

$$w_i^{(k)} := w_i(\boldsymbol{x}^{(k)}, \theta).$$

It then applies one iteration of Gauss-Newton, which seeks to solve

$$\min_{d\boldsymbol{x}} \sum_{i=1}^{N} \left(f_i(\boldsymbol{x}^{(k)}, \theta) + (\frac{\partial f_i}{\partial \boldsymbol{x}})^T d\boldsymbol{x}\right)^2 \tag{2}$$

The optimal solution to (2) is given by

$$d\boldsymbol{x} := -H(\boldsymbol{x}^{(k)}, \theta)^{-1} \boldsymbol{g}(\boldsymbol{x}^{(k)}, \theta)$$

where

$$H(\boldsymbol{x}^{(k)}, \theta) = \sum_{i=1}^{N} w_i^{(k)} \frac{\partial f_i}{\partial \boldsymbol{x}} (\frac{\partial f_i}{\partial \boldsymbol{x}})^T,$$
$$\boldsymbol{g}(\boldsymbol{x}^{(k)}, \theta) = \sum_{i=1}^{N} w_i^{(k)} f_i \frac{\partial f_i}{\partial \boldsymbol{x}}.$$

This leads to the solution at $(k+1)$ as

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - H(\boldsymbol{x}^{(k)}, \theta)^{-1} \boldsymbol{g}(\boldsymbol{x}^{(k)}, \theta). \tag{3}$$

The gradient update for (3) is given by

$$\frac{\partial \boldsymbol{x}^{(k+1)}}{\partial \theta} = \frac{\partial \boldsymbol{x}^{(k)}}{\partial \theta} - H(\boldsymbol{x}^{(k)}, \theta)^{-1} \frac{\partial \boldsymbol{g}(\boldsymbol{x}^{(k)}, \theta)}{\partial \theta}$$
$$+ H(\boldsymbol{x}^{(k)}, \theta)^{-1} \cdot \frac{\partial H(\boldsymbol{x}^{(k)}, \theta)}{\partial \theta} \cdot H(\boldsymbol{x}^{(k)}, \theta)^{-1} \boldsymbol{g}(\boldsymbol{x}^{(k)}, \theta)$$

Here

$$\frac{\partial H(\boldsymbol{x}^{(k)}, \theta)}{\partial \theta} = \sum_{i=1}^{N} \Big(\frac{\partial w_i}{\partial \theta}(\boldsymbol{x}^{(k)}, \theta) \frac{\partial f_i}{\partial \boldsymbol{x}} (\frac{\partial f_i}{\partial \boldsymbol{x}})^T$$
$$+ w_i^{(k)} \frac{\partial^2 f_i}{\partial \theta \partial \boldsymbol{x}} (\frac{\partial f_i}{\partial \boldsymbol{x}})^T + w_i^{(k)} \frac{\partial f_i}{\partial \boldsymbol{x}} (\frac{\partial^2 f_i}{\partial \theta \partial \boldsymbol{x}})^T \Big)$$
$$\frac{\partial \boldsymbol{g}(\boldsymbol{x}^{(k)}, \theta)}{\partial \theta} = \sum_{i=1}^{N} \Big(\frac{\partial f_i}{\partial \boldsymbol{x}} f_i (\frac{\partial w_i}{\partial \theta}(\boldsymbol{x}^{(k)}, \theta))^T,$$
$$+ w_i^{(k)} \frac{\partial^2 f_i}{\partial \theta \partial \boldsymbol{x}} f_i + w_i^{(k)} \frac{\partial f_i}{\partial \boldsymbol{x}} (\frac{\partial f_i}{\partial \theta})^T \Big).$$

1.2. Training Details

This section provides the details of network training.

Surfel Prediction Module. Denote one training data as $\mathcal{T} = (I, \{d_i^{gt}, \boldsymbol{n_i}^{gt}, b_i^{gt}\}_{i=1}^N)$, where d_i^{gt}, $\boldsymbol{n_i}^{gt}$, b_i^{gt} collects the ground-truth depth,normal and boundary label for input image I. N is the total number of pixels that has annotations. Let d_i,$\boldsymbol{n_i}$,b_i and f_i collect the depth,normal,boundary and descriptor channels of the first module respectively. We define the training loss for the first module:

$$\mathcal{L}_1 = \lambda_d \mathcal{L}_d + \lambda_n \mathcal{L}_n + \lambda_b \mathcal{L}_b + \lambda_f \mathcal{L}_f \tag{4}$$

where

$$\mathcal{L}_d = \frac{1}{N} \sum_{i=1}^{N} \text{Smooth}L_1(d_i - d_i^{gt})$$
$$\mathcal{L}_n = \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{n_i} - \boldsymbol{n_i}^{gt}\|_{\mathcal{F}}^2 \tag{5}$$

and

$$\mathcal{L}_b = -\frac{1}{N}\sum_{i=1}^{N}(b_i^{gt}\log(b_i) + (1-b_i^{gt})\log(1-b_i)) \quad (6)$$

$$\mathcal{L}_f = \mathcal{L}_f^{\text{push}} + \mathcal{L}_f^{\text{pull}} \quad (7)$$

where

$$\mathcal{L}_f^{\text{pull}} = \sum_{\mathcal{P}_k\in\mathcal{P}}\frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}_k}\|f_i - \frac{1}{|\mathcal{P}_k|}\sum_{i\in\mathcal{P}_k} f_i\|_{\mathcal{F}}^2 \quad (8)$$

$$\mathcal{L}_f^{\text{push}} = \frac{2}{|\mathcal{P}|(|\mathcal{P}|-1)}\sum_{\mathcal{P}_k,\mathcal{P}_j\in\mathcal{P},\mathcal{P}_k\neq\mathcal{P}_j} \max(0, \delta - \|\frac{1}{|\mathcal{P}_k|}\sum_{i\in\mathcal{P}_k} f_i - \frac{1}{|\mathcal{P}_j|}\sum_{i\in\mathcal{P}_j} f_i\|_{\mathcal{F}}^2) \quad (9)$$

We set $\delta = 1.5$, $\lambda_d = 1.0$, $\lambda_n = 1.0$, $\lambda_f = 10^{-2}$, $\lambda_b = 10^{-2}$ in our experiments.

Plane Detection Module. The plane detection module uses contrastive loss on clustering output, which adopts the same formulation as (7).

Plane Synchronization Module. The loss for synchronization module is composed of two parts. The first part is depth and normal loss for planar pixels, the other part is a loss on planar-relation. Let d_i and n_i collect the depth and normal prediction(on planar region) of the synchronization module,

$$\mathcal{L}_2 = \lambda_d\mathcal{L}_d + \lambda_n\mathcal{L}_n + \lambda_r\mathcal{L}_r \quad (10)$$

$$\mathcal{L}_d = \frac{1}{N^{\text{planar}}}\sum_{i=1}^{N^{\text{planar}}}\text{Smooth}L_1(d_i - d_i^{gt}) \quad (11)$$

$$\mathcal{L}_n = \frac{1}{N^{\text{planar}}}\sum_{i=1}^{N^{\text{planar}}}\|\boldsymbol{n_i} - \boldsymbol{n_i}^{gt}\|_{\mathcal{F}}^2 \quad (12)$$

,where $\mathcal{L}_r$ adopts a similar formulation as (6) while replacing boundary prediction with relation prediction. N^{planar} is the number of planar pixels.

Refinement Module. The last module is similar to the first module where we directly regress the refined surfel geometry.

1.3. Plane Labeling Algorithm on NYUv2

The plane detection algorithm we used contains two stages, namely, the plane proposal stage and the plane merging stage. During the plane proposal stage, we first pre-compute an adjacency matrix by connecting point pairs with a distance less than $0.1m$. Then we run RANSAC for 200 iterations and record the largest plane patch. Note that we use the adjacency matrix to filter out points that do not belong to the largest connected components. This step is essential in order to filter out noise association. During the plane merging stage, we merge two planes if they meet two criteria. The first criterion is the average angle between the candidate two planes must be less than $30°$. The second criterion is the average point-to-plane distance between the fitted plane to the combined point clouds to each point cloud is less than $0.1m$. We show some of the plane annotations in Figure 1.

1.4. Details of Boundary Prediction

Boundary is needed for stitching adjacent planes. We generate the ground truth boundary by computing the analytic plane intersection line using the ground truth plane equation and draw the intersecting lines with a fixed linewidth(we use 3 pixels in our experiments). We prune the intersecting lines that are far away (in image space) from either of the two plane segment. We show in Figure 3 the ground truth boundary annotation and the predicted boundary. Noted that the boundary label is different from the edge detection. In particular, the cyan plane and the green plane in Figure 3 top right share a sharp edge in the image but do not intersect in 3D.

2. More Experimental Results

2.1. More Results on Plane Detection

Please refer to Figure 5 for more qualitative results on plane detection. We show comparisons against Liu et al. [2] and Yu et al. [4]. Instead of showing the recalls at three thresholds $0.1m, 0.3m, 0.5m$ in the main paper, here we also shows the pixel recall plot and plane recall plot in Figure 4, follow the practice as [2, 4].

2.2. More Qualitative Comparisions of Predicted Pointclouds

Please refer to Figure 6 for more qualitative results on predicted point cloud on NYUv2 dataset. Our method shows clear advantage over structural regions.

2.3. Error Distribution Analysis

We show the error distribution of depth estimator and normal estimator in Figure 2

References

[1] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 5

[2] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. Planenet: Piece-wise planar reconstruction from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2579–2588, 2018. 2, 3, 4

[3] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5684–5693, 2019. 5

[4] Zehao Yu, Jia Zheng, Dongze Lian, Zihan Zhou, and Shenghua Gao. Single-image piece-wise planar 3d reconstruction via associative embedding. In *Proceedings of the*

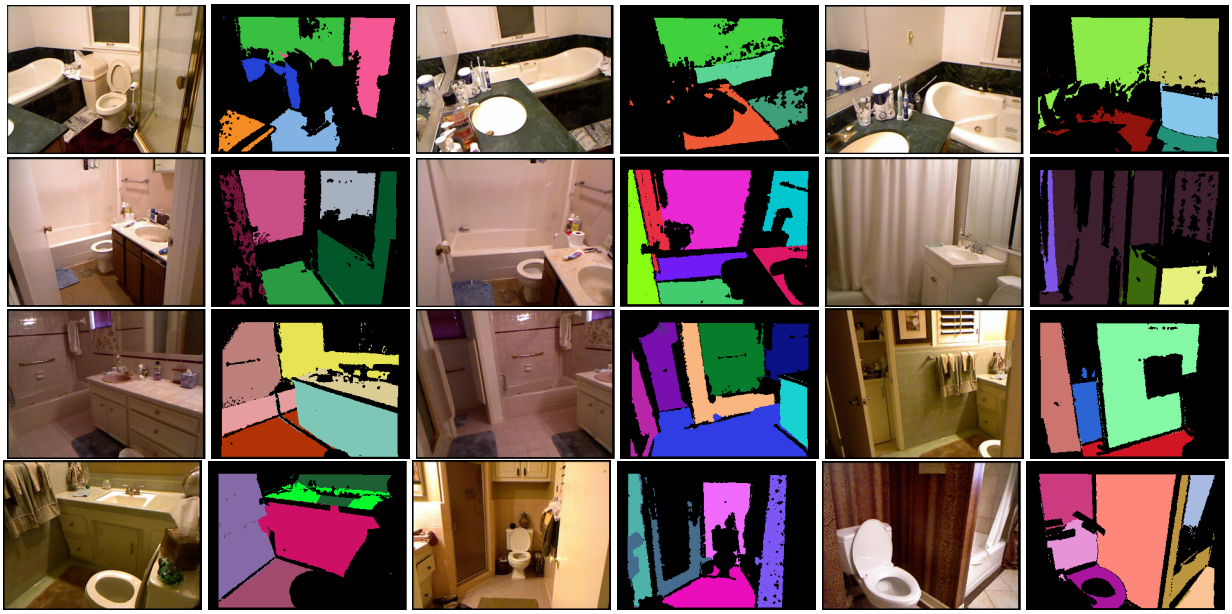

Figure 1. Visualization of the outputs of our plane annotation algorithm on NYUv2. We show input rgb image on the left and plane segmentation on the right. Pixels that corresponds to the same plane is drawn using the same color. Black corresponds to non-planar region.

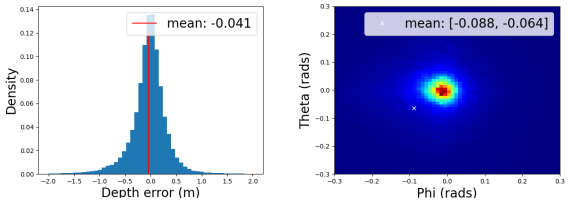

Figure 2. Error distribution on NYUv2. The left plot shows error distribution for depth. The right plot shows error distribution for normal. The legend shows the distribution mean.

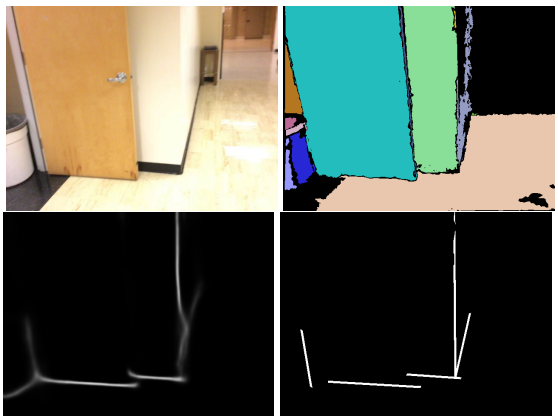

Figure 3. We show the input image (top-left), ground truth segmentation(top-right), predicted boundary(bottom-left), and ground truth boundary(bottm-right).

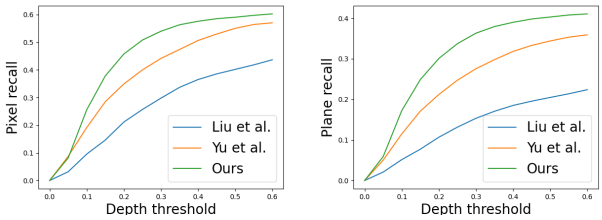

Figure 4. We show the pixel recall and plane recall when varying the depth thresholds on ScanNet. Our methods out-perform previous state-of-the-art Yu et al. [4] and Liu et al. [2] by a large margin.

IEEE Conference on Computer Vision and Pattern Recognition, pages 1029–1037, 2019. 2, 3, 4

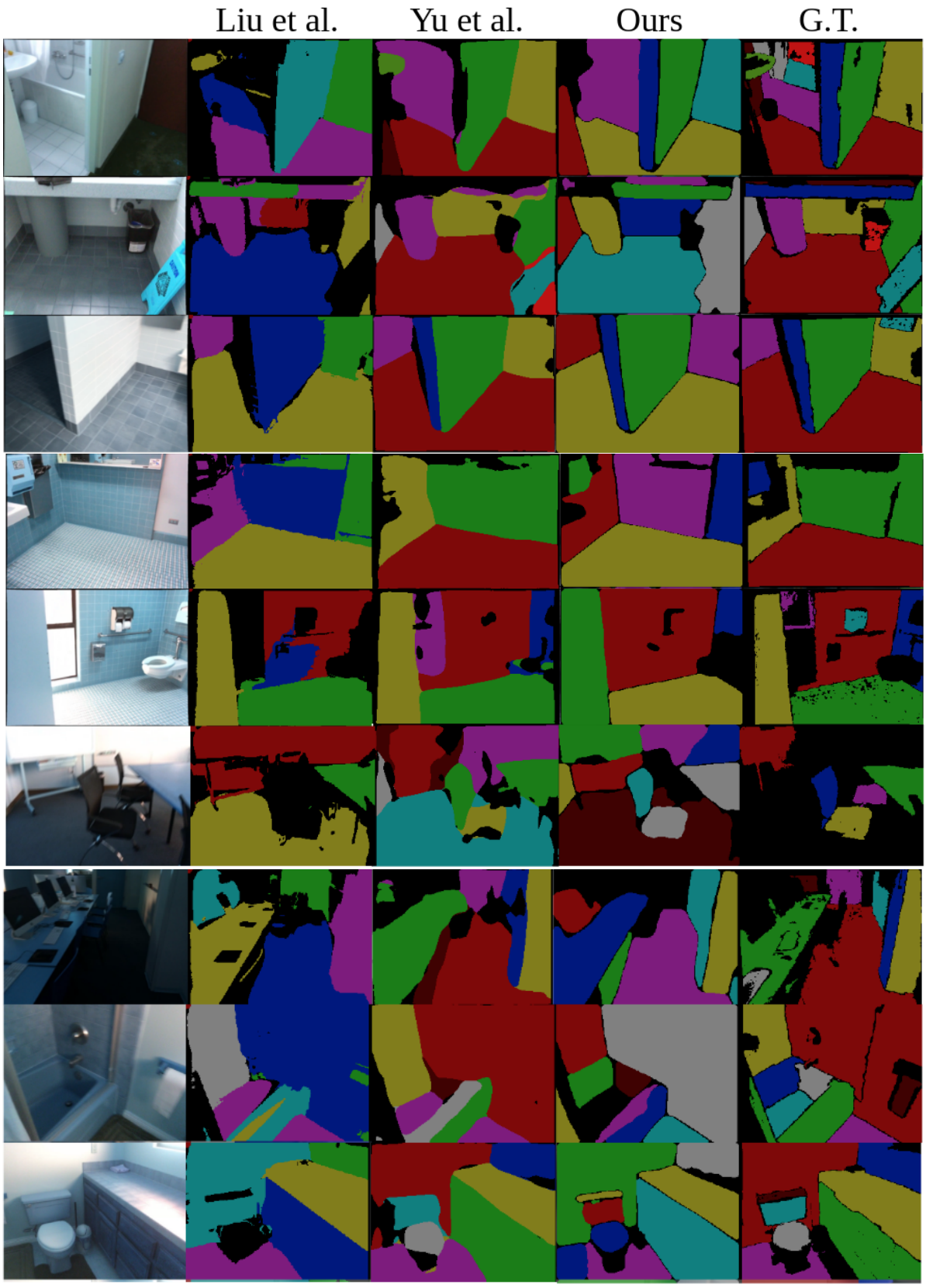

Figure 5. Qualitative comparisons between StruMonoNet (Ours) and state-of-the-art plane detection (Yu et al. [4], Liu et al. [2]) on ScanNet. The first column shows input images. the second column shows predictions of [2], the third column shows predictions of [4], the fourth column shows our predictions. The last column shows the ground-truth plane annotation provided by [2].

Figure 6. Qualitative comparisons on NYUv2. The first column shows the input images, the second and third column show results of Yin et al. [3] and Lee et al. [1] respectively. The fourth column show Ours-Baseline. The fifth column shows results of Ours. The last column shows the ground-truth. Each prediction is visualized in two views.