SUPPLEMENTARY FOR
PlanarRecon: Real-time 3D Plane Detection and Reconstruction from Posed Monocular Videos

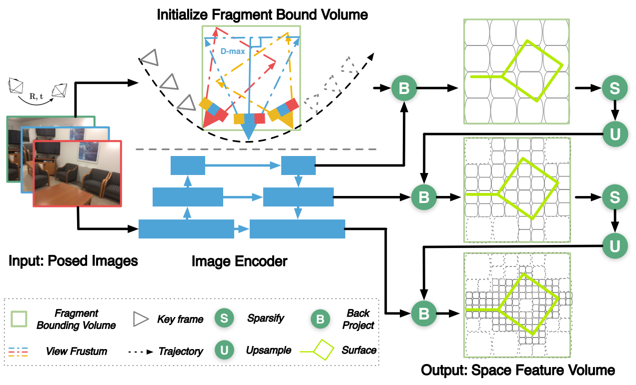

Figure 1. 2D illustration of 3D sparse feature volume construction.

1. 3D Sparse Feature Volume Construction

The 3D sparse feature volume construction is detailed in Fig. 1. Feature representations of the key frames in a fragment obtained using a 2D neural network (MnasNet [3]) are back-projected to the 3D space through camera poses. The 3D space is divided into a set of voxels, where the feature representation of each voxel is an average of the representations of its observed pixels in different key frames.

We build the 3D sparse feature volumes in a coarse-to-fine pyramid manner, following the network hierarchy when computing image representations, where the 3D space is gradually divided into finer voxels. In each pyramid, we perform occupancy classification for each voxel. Only occupied voxels will be further processed at the next pyramid level. At the finest level, we end up having a set of occupied voxels, which are used to detect planes.

2. Loss Function

Occupancy Classification. The occupancy loss L_o is defined as the binary cross-entropy (BCE) between the predicted occupancy values and the ground-truth occupancy values. The supervision is applied to all the coarse-to-fine levels.

Fragment-based Plane Detection. We use the cross-entropy loss to supervise the learning of anchor normal selection (it is a classification problem), and the smooth_L1 loss to supervise the anchor normal residual vector and plane offset distance regression. The predicted 3D displacement $\Delta\mathbf{x}'_m$ is supervised by L1 loss.

The fragment-based plane detection loss L_d is a combination of the losses for anchor normal selection L_{ac}, anchor normal residual regression L_{ar}, plane offset distance regression L_{or}, and 3D displacement to plane centroid regression L_{dc}.

$$L_d = \alpha_1 L_{ac} + \alpha_2 L_{ar} + \alpha_3 L_{or} + \alpha_4 L_{dc}. \tag{1}$$

Differentiable Matching for Plane Tracking. Denote the ground-truth matching labels $\mathcal{M}$ between the global plane reconstruction $\mathcal{P}^g_{i-1}$ and plane detections from the current fragment $\mathcal{P}_i$, we minimize the negative log-likelihood of the matching matrix $\mathbf{M}^*$. But some planes in $\mathcal{P}^g_{i-1}$ may not find their matchings in $\mathcal{P}_i$ and vice versa. Denote the unmatched labels for $\mathcal{P}^g_{i-1}$ and $\mathcal{P}_i$ as $\mathcal{I}$ and $\mathcal{J}$, respectively, the matching loss is defined as follows inspired by [2].

$$L_m = -\sum_{(i,j)\in\mathcal{M}} \log \mathbf{M}^*_{i,j} - \sum_{i\in\mathcal{I}} \log \mathbf{M}^*_{i,N+1} - \sum_{j\in\mathcal{J}} \log \mathbf{M}^*_{M+1,j},$$

where M and N are the number of planes in $\mathcal{P}^g_{i-1}$ and $\mathcal{P}_i$, respectively.

Final loss . The final loss L is defined as

$$L = \beta_o L_o + \beta_d L_d + \beta_m L_m. \tag{2}$$

3. The Tracking and Fusion Process for Baseline

We compute the cosine similarity of the normals and the difference of the plane offset from two adjacent key frames. If the cosine similarity is higher than α and the difference is smaller than β, we believe these two planes are matched. Then, we refine the plane by fitting a new plane using the points in the matched pairs. We tune all the hyper-parameters on ScanNet validation set and choose the one that leads to the best performance. We set $\alpha = 0.833$ and $\beta = 0.5$ in our experiment.

4. The Sequential RANSAC

Given the estimated 3D mesh, our RANSAC algorithm is first asked to randomly sample k vertices from the mesh. Each vertex will generate one plane hypothesis with its 3D coordinate and normal. Then, all the other vertices that are within t_d distance to a plane hypothesis and have its normal within $t_a{}^\circ$ to the plane normal are considered as inliers to the hypothesis. One vertex could be an inlier to multiple hypotheses, and the hypothesis with the largest number of inliers will be detected as a plane instance. Next, we check the connection of the inlier vertices based on their locations in the volumetric space defined by the space carving model. Finally, vertices belonging to the largest connected component will be selected as the final inliers to the detected plane instance and removed from the mesh. We repeat this process until the detected plane has less than 5 inliers. To favor this approach, we tune all the hyper-parameters on ScanNet validation set and use the ones that lead to the best performance – $k = 1000$, $t_d = 0.25$ and $t_a = 30°$.

5. Implementation Details

We set the number of key frames in a fragment N_k to 9 in our experiments. R_{max} and t_{max} are set to 15°and $0.1m$ respectively. The occupancy score o is predicted with a Sigmoid layer. The voxel size for the sparse feature volume is $4cm$ and the radius λ is set to $0.12m$. The threshold θ is set to 0.5. We use torchsparse [4] to implement the 3D sparse convolution in 3D sparse volume reconstruction. The image backbone is MnasNet [3] initialized with weights pretrained on ImageNet. We further integrate the feature pyramid network [1] in the image backbone to better capture the geometry feature. The entire network is trained end-to-end with randomly initialized weights except for MnasNet. The bandwidth in Mean-shift is set to 0.3. The iteration number of Mean-shift is set to 5 in the training, and 10 in the testing. In the learning-based tracking, the matching threshold is set to 0.01.

References

[1] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 2

[2] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 1

[3] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *CVPR*, 2019. 1, 2

[4] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. In *ECCV*, 2020. 2