Homework 2 - SexySemanticMapping USING 3 LATE DAY(S)

Zhentian Qian
Robotics Engineering
Worcester Polytechnic Institute
Worcester, Massachusetts
Email: zqian@wpi.edu

I. Building The Map

The map built based on the first thousand lidar scans is visualized in Fig. 1. Map points are colored according to their height. We use the Point-To-Plane ICP implemented in open3d [1] to estimate relative motion between scans. To do so, we first need to estimate a plane normal for every point in the point cloud. 30 nearest neighbors are used to fit a plane and determine the plane normal. Assume that we have two surfaces P and Q, and that we have already computed an initial transformation T_0. The Point-To-Plane ICP aims to minimize the following objective function [2]:

$$e^k = \sum_{i=1}^{N} d_s^2(T \circ T^{k-1}\mathbf{p}_i, S_i^k) \tag{1}$$

where

$T \circ T^{k-1} = T^k$

$S_j^k = \{s \mid \mathbf{n}_{qj}^k \cdot (\mathbf{q}_j^k - s) = 0\}$ is the tangent plane to Q at $\mathbf{q}_j^k$

$\mathbf{n}_{qj}^k$ is the normal to surface Q at $\mathbf{q}_j^k$

$\mathbf{q}_j^k = (T^{k-1}l_i) \cap Q$ is the intersection point of Q with line $T^{k-1}l_i$

$l_i = \{\mathbf{a} \mid (\mathbf{p}_i - a) \times \mathbf{n}_{pi} = 0\}$ is the line normal to P at P_i

$\mathbf{p}_i \in P$ is a point on P

d_s is the signed distance from a point to a plane.

The Point-To-Plane ICP algorithm is to find the update transformation T, which minimizes e^k in the above equation with a least squares method iteratively. The algorithm is described in Algorithm 1. The convergence of the algorithm is tested by checking

$$\delta = \frac{\|e^k - e^{k+1}\|}{N'} \le \epsilon_e, (\epsilon_e > 0) \tag{2}$$

where ϵ_e is a threshold, $N'(N' < N)$ is the actual number of $\mathbf{p}_i'$s used, since some of them may not have a counterpart in Q.

Algorithm 1: The Point-To-Plane ICP algorithm

Select a set of control points $\mathbf{p}_i \in P(i = 1, \dots, N)$ and compute the surface normals $\mathbf{n}_{pi}$ at those points. Let $T^0 = T_0$;
foreach *control point* $\mathbf{p}_i$ **do**
  Apply T^{k-1} to both the control point $\mathbf{p}_i$ and the normal $\mathbf{n}_{pi}$ to get $\mathbf{p}_i'$ and $\mathbf{n}_{pi}'$;
  Find the intersection $\mathbf{q}_i^k$ of surface Q with the normal line defined by $\mathbf{p}_i'$ and $\mathbf{n}_{pi}'$;
  Compute the tangent plane S_i^k of Q at q_i';
  Find the transformation T that minimize e^k in (1) with a least squares method, let $T^k = T \circ T^{k-1}$;
end

II. Semantic Point Painting The Map

A. Semantic Segmentation

We use the PSA [3] semantic segmentation neural network for semantic prediction. The network structure is shown in Fig. 2. Given an input image $\mathbf{I}$, its local representation is acquired through Fully Convolutional Networks (FCN) as feature map $\mathbf{X}$, which is the input of the PSA module. ResNet [4] is used as the FCN backbone. The PSA module then aggregates long-range contextual information from the local representation. It follows stage 5 in ResNet, which is the final stage of the FCN backbone.

The PSA network is trained on the Cityscapes Dataset [5]. The Cityscapes Dataset is a new large-scale dataset that contains a diverse set of stereo video sequences recorded in street scenes from 50 different cities, with high-quality pixel-level annotations of 5,000 frames in addition to a larger set of 20,000 weakly annotated frames. 30 common classes of road, person, car, etc., are annotated, and 19 of them are used for semantic segmentation evaluation.

One instance of semantic prediction is given Fig. 3. It can be see that the predicted semantic segmentation is accurate.

B. Transfer the RGB semantic labels

The lidar points $\mathbf{P}^v$ are represented in the Velodyne frame. To transfer the RGB semantic labels, we first transform the

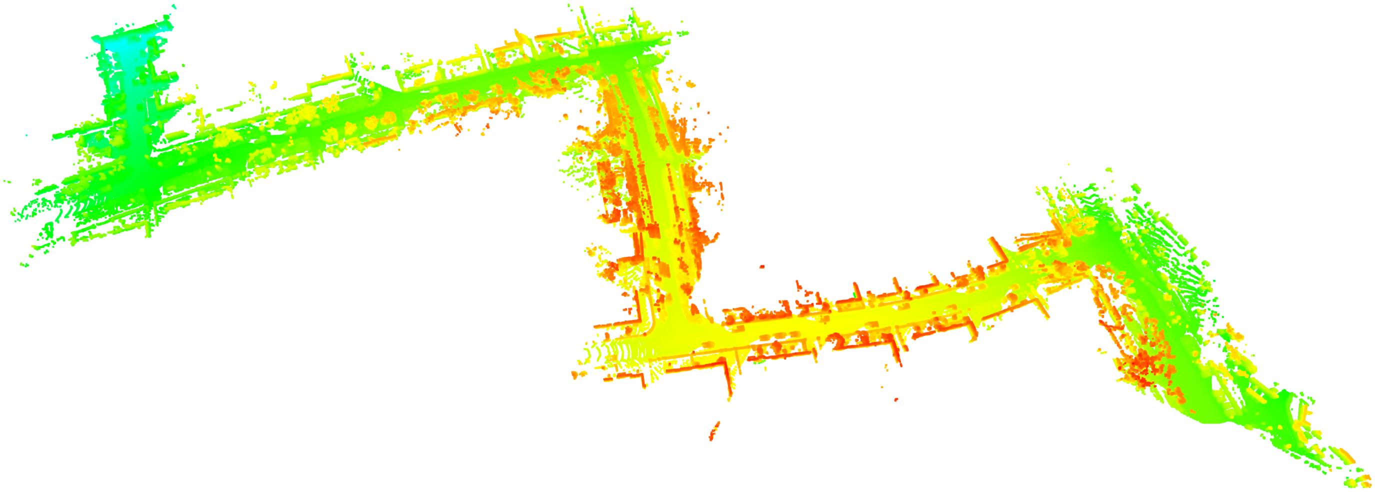


Fig. 1: LIDAR based mapping. Map points are colored according to their height.

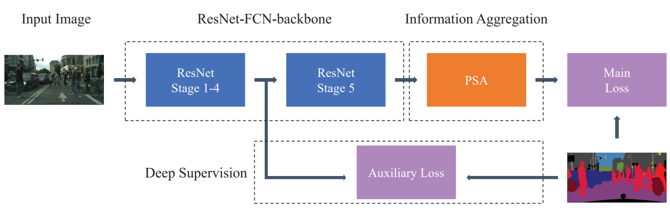
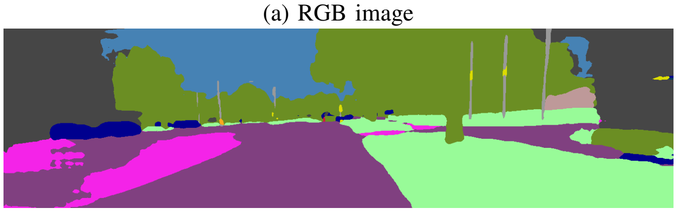


Fig. 2: Network structure of ResNet-FCN-backbone with PSA module incorporated. [3].

(a) RGB image



(b) Semantic prediction of the RGB image

Fig. 3: RGB image and its semantic prediction

points into the rectified image frame:

$$\mathbf{P}^c = T_v^c \mathbf{P}^v \tag{3}$$

Then we project the 3d points into the rectified image frame:

$$\mathbf{p}^c = [K \mid \mathbf{0}]\mathbf{P}^c \tag{4}$$

The semantic label assigned to the 2d projection is then transferred to the 3d points. Note that apart from requiring the depth of points in the camera frame to be positive, we also require the depth to be smaller than 30 meters to be considered valid. Otherwise, no semantic label is assigned to that point.

The lidar maps colored with RGB data and semantics are shown in Figs. 4 and 5. In Fig. 4, we can see that the street view has been reconstructed. In Fig. 5, the corresponding semantic label has also been appended.

References

[1] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3d: A modern library for 3d data processing," *arXiv preprint arXiv:1801.09847*, 2018.

[2] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.

[3] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, and J. Jia, "Psanet: Point-wise spatial attention network for scene parsing," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 267–283.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[5] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

Fig. 4: LIDAR map colored with RGB image data.

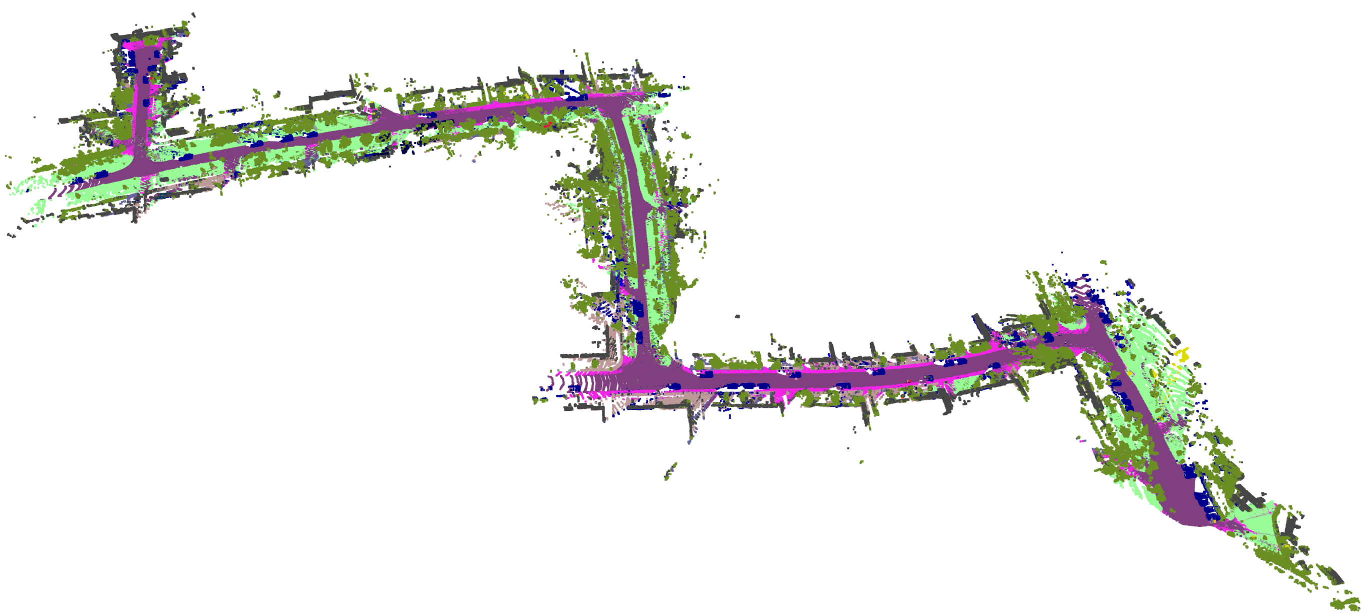


Fig. 5: LIDAR map colored with semantics from the RGB image data.