PWCLO-Net: Deep LiDAR Odometry in 3D Point Clouds Using Hierarchical Embedding Mask Optimization (Supplementary Material)

Guangming Wang[1] Xinrui Wu[1] Zhe Liu[2] Hesheng Wang[1]*
[1]Department of Automation, Insititue of Medical Robotics, Key Laboratory of System Control and Information Processing of Ministry of Education, Shanghai Jiao Tong University
[2]Department of Computer Science and Technology, University of Cambridge
{wangguangming,916806487,wanghesheng}@sjtu.edu.cn zl457@cam.ac.uk

Module	Layer type		K	Sample rate	MLP width
Siamese Point Feature Pyramid	Set conv layer for PC_1^0 and PC_2^0		32	0.25	[8,8,16]
	Set conv layer for PC_1^1 and PC_2^1		32	0.5	[16,16,32]
	Set conv layer for PC_1^2 and PC_2^2		16	0.25	[32,32,64]
	Set conv layer for PC_1^3 and PC_2^3		16	0.25	[64,64,128]
Attentive Cost Volume (on the penultimate level)	Attentive cost volume for E_{coarse}		4, 32	1	[128,64,64], [128,64]
Generation of Initial Embedding Mask and Pose	Set conv layer for E^3		16	0.25	[128,64,64]
	Shared MLP for M^3		—	1	[128,64]
	FC for q^3, FC for t^3		—	1	[4], [3]
Iterative Pose Warp-Refinement	Pose Warp-Refinement	Attentive cost volume for RE^2	4, 6	1	[128,64,64], [128,64]
		Set upconv for CE^2	8	4	[128,64], [64]
		Shared MLP for E^2	—	1	[128,64]
		Set upconv for CM^2	8	4	[128,64], [64]
		Shared MLP for M^2	—	1	[128,64]
		FC for q^2, FC for t^2	—	1	[4], [3]
	Pose Warp-Refinement	Attentive cost volume for RE^1	4, 6	1	[128,64,64], [128,64]
		Set upconv for CE^1	8	4	[128,64], [64]
		Shared MLP for E^1	—	1	[128,64]
		Set upconv for CM^1	8	4	[128,64], [64]
		Shared MLP for M^1	—	1	[128,64]
		FC for q^1, FC for t^1	—	1	[4], [3]
	Pose Warp-Refinement	Attentive cost volume for RE^0	4, 6	1	[128,64,64], [128,64]
		Set upconv for CE^0	8	4	[128,64], [64]
		Shared MLP for E^0	—	1	[128,64]
		Set upconv for CM^0	8	4	[128,64], [64]
		Shared MLP for M^0	—	1	[128,64]
		FC for q^0, FC for t^0	—	1	[4], [3]

Table 1: **Detailed network parameters in PWCLO-Net.** K points are selected in the K Nearest Neighbors (KNN) of set conv layer, set upconv layer, and attentive cost volume layer. Set conv layer uses the Farthest Point Sampling (FPS) to obtain the sampling points, and the sampling rate is less than 1. For the set upconv layer, skip connections are used to propagate the sparse features to dense features, and the sampling rate is larger than 1. MLP width means the number of output channels for each layer of MLP. The variables in the table are defined the same as the main manuscript.

1. Overview

In this supplementary material, we provide detailed network parameters and data augmentation parameters in Sec. 2. Sec. 3 contains the comparison experiments on removing and reserving the ground mentioned in the main manuscript. More ablation studies are presented in Sec. 4 to show the effectiveness of the design details of our model. Sec. 5 shows

Method	00*		01*		02*		03*		04*		05*		06*		07		08		09		10		Mean on 07-10	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
Ours (removing ground less than $0.55m$ in height)	0.78	0.42	0.67	0.23	0.86	0.41	0.76	0.44	0.37	0.40	0.45	0.27	0.27	0.22	**0.60**	0.44	1.26	0.55	0.79	0.35	1.69	**0.62**	1.085	0.490
Ours (removing ground less than $0.3m$ in height)	0.78	0.36	**0.51**	0.21	0.76	0.29	**0.58**	0.58	**0.30**	0.22	0.43	0.27	**0.25**	0.14	0.78	0.46	1.45	0.67	**0.71**	**0.34**	2.38	0.90	1.330	0.593
Ours (reserving ground)	**0.68**	**0.28**	0.68	**0.18**	**0.74**	**0.23**	**0.58**	**0.41**	0.38	**0.08**	**0.40**	**0.19**	0.28	**0.13**	**0.60**	**0.38**	**1.13**	**0.35**	0.86	0.45	**1.18**	0.74	**0.943**	**0.480**

Table 2: The LiDAR odometry experiment results of our model removing the ground less than different heights and reserving the ground on KITTI odometry dataset [1].

	Method	00*		01*		02*		03*		04*		05*		06*		07		08		09		10		Mean on 07-10	
		t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
(a)	Ours (with ℓ_2-norm for t and ℓ_2-norm for q)	**0.75**	**0.36**	**0.46**	**0.15**	**0.77**	**0.37**	0.93	0.68	**0.37**	**0.27**	0.62	0.29	0.35	**0.19**	0.84	0.48	1.35	0.66	1.03	**0.34**	**1.29**	**0.55**	1.128	0.508
	Ours (with ℓ_1-norm for t and ℓ_1-norm for q)	0.78	0.46	0.70	0.43	0.78	0.43	**0.74**	0.49	0.38	0.55	0.57	0.33	0.35	0.23	0.68	0.54	1.48	0.70	1.04	0.50	1.49	0.57	1.173	0.578
	Ours (selected, with ℓ_1-norm for t and ℓ_2-norm for q)	0.78	0.42	0.67	0.23	0.86	0.41	0.76	**0.44**	**0.37**	0.40	**0.45**	**0.27**	**0.27**	0.22	**0.60**	**0.44**	**1.26**	**0.55**	**0.79**	0.35	1.69	0.62	**1.085**	**0.490**
(b)	Ours (one Dimensional Mask)	0.84	0.56	**0.56**	0.24	0.94	0.42	**0.74**	0.70	**0.29**	0.43	0.73	0.40	0.37	**0.21**	0.87	0.72	1.58	0.66	0.98	0.46	1.89	0.99	1.330	0.708
	Ours (selected, Multidimensional Mask)	**0.78**	**0.42**	0.67	**0.23**	**0.86**	**0.41**	0.76	**0.44**	0.37	**0.40**	**0.45**	**0.27**	**0.27**	0.22	**0.60**	**0.44**	**1.26**	**0.55**	**0.79**	**0.35**	**1.69**	**0.62**	**1.085**	**0.490**

Table 3: The ablation study results of LiDAR odometry on KITTI odometry dataset [1].

more visual results on successful and failing cases. Sec. 6 shows more visual trajectory results on KITTI odometry dataset [1]. We give a video demo in Sec. 7.

2. Network Details

2.1. Network Parameters

In the training and evaluation process, the input point number N is set to be 8192.

Each layer in MLP contains the ReLU activation function, except for the FC layer. For shared MLP, 1×1 convolution with 1 stride is the implement manner. The detailed layer parameters including K values in K Nearest Neighbors (KNN), the sample rate of each sampling layer, and each linear layer width in MLP are described in Table 1.

2.2. Data Augmentation Parameters

We augment the training dataset by the augmentation matrix T_{aug}, generated by the rotation matrix R_{aug} and the translation vector t_{aug}. Varied values of yaw-pitch-roll Euler angles are generated by Gaussian distribution around $0°$. Due to the motion characteristics of the car, the standard deviation is different for each angle and each direction of the translation vector. We set the standard deviations are $0.01°$, $0.05°$ and $0.01°$ for the yaw-pitch-roll Euler angles respectively. We set the standard deviations are $0.1m$, $0.05m$, and $0.5m$ for the XYZ of the translation respectively. In these Gaussian distributions, we select the data in the range of 2 times standard deviation around the mean value for data augmentation.

The composed T_{aug} from R_{aug} and t_{aug} is then used to augment the PC_1 to obtain new point clouds $PC_{1,aug}$.

3. Comparison Experiment on Removing and Reserving Ground

To speed up the data reading and speed up the training, the ground less than $0.55m$ in height is removed in the main manuscript. We also did the ablation study on removing the ground less than different heights, including $0.55m$ (in the main manuscript) and $0.3m$ (like the scene flow estimation in [3, 2]) and reserving the ground. The comparison results are listed in Table 2. For our model, the performances of removing and reserving the ground are similar. On average, the performance of reserving the ground is a little better due to the plane characteristics of the ground.

4. Additional Ablation Experiments

In our main manuscript, we remove or change components of our model to do the ablation studies and confirm the contributions of the key components. In this section, we change some other design details of our model to do the ablation studies on the KITTI odometry dataset [1]. We analyze the effectiveness of these details in our network. The training/testing details are the same as the ablation studies in the main manuscript.

Comparisons with Wang et al. [2] Wang et al. [2] is an end-to-end learnable 3D scene flow estimation method based on PWC structure. However, Wang et al. [2] only estimate the motion of each point. Thus, to generate a transformation of two point clouds, it needs further calculation, which is influenced a lot by the presence of dynamic obstacles and other outliers. So using Wang et al. [2] in odometry tasks will lead to performance degradation. Unlike Wang et al. [2], we creatively design optimizable embedding masks and pose refinement modules to address the above challenges and successfully solve the odometry problem. The results in Table 4 show the necessity and superior performance of our end-to-end trainable LiDRA odometry.

ℓ_1-norm or ℓ_2-norm in Loss Function: Different from the LO-Net, we use the ℓ_1-norm for the translation t and the ℓ_2-norm for the quaternion q. We also test the ℓ_2-norm both for the translation t and the quaternion q, and the ℓ_1-norm both for the translation t and the quaternion q. The experiment results are listed in Table 3(a). The results show that ours with ℓ_1-norm for t and ℓ_2-norm for q has the best average performance in the three.

Method	07		08		09		10		Mean on 07-10	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
Wang et al. [2]	14.24	8.15	24.59	9.64	21.43	8.51	19.03	8.29	19.823	8.648
Wang et al. [2]-Refine	2.92	1.82	4.13	1.60	3.05	1.11	3.62	1.78	3.430	1.578
Ours	**0.60**	**0.44**	**1.26**	**0.55**	**0.79**	**0.35**	**1.69**	**0.62**	**1.085**	**0.490**

Table 4: The LiDAR odometry results on KITTI test seq. 07-10. Wang et al. [2] is trained on Flything3D scene flow dataset. Wang et al. [2]-Refine is trained on KITTI seq. 00-06 again after being trained on Flything3D dataset. This refined-training process generates the ground truth 3D scene flow from ground truth pose transformation based on the assumption that all points are static in the frame. Ours are only trained on KITTI seq. 00-06.

α_l $(l = 1, 2, 3, 4)$	$1.6, 0.8, 0.4, 0.2$	$0.2, 0.4, 0.8, 1.6$	$1.6, 0.4, 0.1, 0.025$	$0.025, 0.1, 0.4, 1.6$	$1, 1, 1, 1$	Learnable
Mean t_{rel} on 07-10	**1.085**	1.130	1.220	1.150	1.228	1.255
Mean r_{rel} on 07-10	**0.490**	0.558	0.573	0.638	0.598	0.588

Table 5: The results adopting different α_l on KITTI test seq. 07-10. Models are traned on KITTI seq. 00-06.

α_l in Loss Function: α_l represents the weight of each level in the loss function. Table 5 shows changing α_l has little effect on results. A learnable α_l is feasible, but has lower performance.

Multidimensional Mask or One Dimensional Mask: In our submitted paper, the mask has the same feature dimension as the embedding features, which means there is a different mask for each feature dimension of embedding features. (We use the mean value of multidimensional mask to obtain the visualization of mask in the main manuscript.) In this section, we test the difference between using a one-dimensional mask and our choice in the main manuscript.

Original mask generation process: The embedding features $E = \{e_i | e_i \in \mathbb{R}^c\}_{i=1}^n$ and the features F_1 of PC_1 are input to a shared MLP followed by the softmax operation along the point dimension to obtain the trainable embedding mask $M = \{m_i | m_i \in \mathbb{R}^c\}_{i=1}^n$, as follows (This is for initial mask generation. For embedding mask refinement, the coarse mask from last level is also used.):

$$M = softmax(sharedMLP(E \oplus F_1)). \tag{1}$$

We change the dimension of the mask by adding a mean-pooling along the feature dimension to the middle of the equation and obtain the one dimension mask:

$$M = softmax(meanpooling(sharedMLP(E \oplus F_1))). \tag{2}$$

The ablation study results are listed in Table 3(b). The results show that our multidimensional mask has better performance. We believe this is because different dimensions of features describe different characteristics. For local correspondence of different objects, the features of each dimension have different contribution weights.

5. Pose Transformation Visualization of Two Consecutive frames

We visualize the consecutive frames to present some successful and failing cases of pose estimation by registering the two point clouds through the ground truth pose and our estimated pose. We register the PC_2 to the first frame in this supplementary material.

The registered $PC_{gt,trans}$ is obtained by:

$$PC_{gt,trans} = T_{gt} PC_2 \tag{3}$$

where T_{gt} is the ground truth pose transfrmation between PC_1 and PC_2. The registered $PC_{ours,trans}$ is obtained by:

$$PC_{ours,trans} = T_{ours} PC_2 \tag{4}$$

where T_{ours} is the estimated pose transfrmation between PC_1 and PC_2.

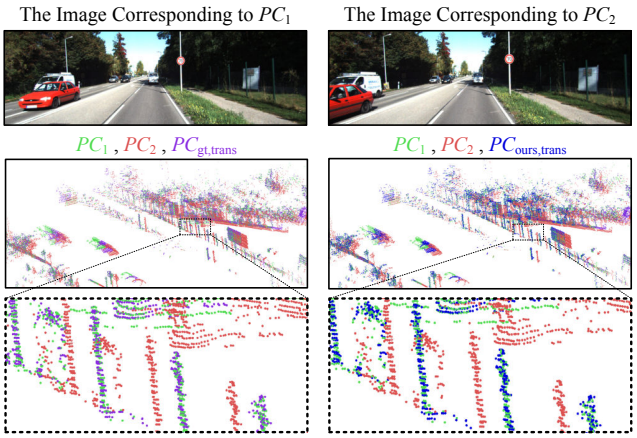
The effect of registration is visulized in Figs. 2 and 1.

In Fig. 2, the static objects all have right correspondences between registered $PC_{ours,trans}$ and PC_2. The dynamic objects between $PC_{ours,trans}$ and PC_2 are not correspondences because of the ego-motion of dynamic objects. It is noted that there is one case in the last line of Fig. 2, where the ground truth of pose is wrong while ours has the right correspondence.

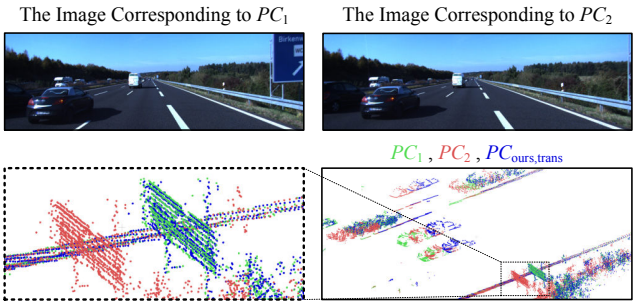
In Fig. 1, some highly dynamic scenarios are visualized to see the robustness of our model for dynamic objects. In Fig. 1(b) and (c), we visualized two images in sequences 11-21 without ground truth because there are few highly dynamic scenarios in sequences 00-10 with ground truth. In Fig. 1(a) and (b), when there are many dynamic cars, ours also have right correspondence between registered $PC_{ours,trans}$ and PC_2, which can be seen by zooming in on static objects. There is one case in Fig. 1(c), where ours has a little error because of tons of dynamic objects and few static, rigid objects. Overall, the visualization of registration in many scenes demonstrates the effectiveness of our model for dynamics.

6. More Trajectory Results on KITTI Odometry Dataset

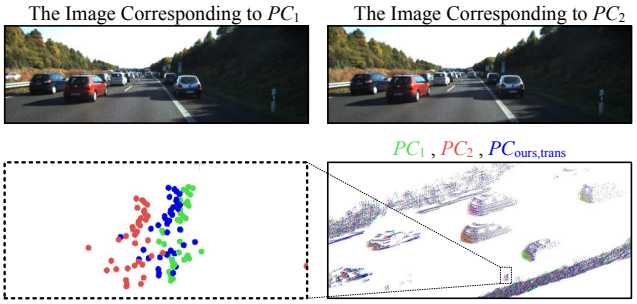
We list all visualized trajectory results on sequences 00-10 of KITTI odometry dataset [1] with the ground truth in



(a) Sequence 04 frames 257-258



(b) Sequence 21 frames 841-842



(c) Sequence 22 frame 24-25

Figure 1: Cases to analyze the influence of dynamic objects on odometry estimation on KITTI odometry dataset [1]. The image corresponding to PC_1, the image corresponding to PC_2, the point clouds visulaization of PC_1 (gree color), PC_2 (red color), $PC_{gt,trans}$ (purple color), and $PC_{ours,trans}$ (blue color) are visualized. We zoom in on some static details of the point clouds to see the registration effect as there are many dynamic objects in the scenes. The highly dynamic scenes in (b) and (c) have no ground truth.

Figs. 3 and 4 excluding the results that have been listed in our submitted paper. The results demonstrate that our method outperforms the LOAM without mapping and even outperforms full LOAM on most sequences of the KITTI odometry dataset [1].

7. Video Demo

We present a video demo, demo.mp4, on sequence 07 of the KITTI odometry dataset [1] with the ground truth. In this video, the trajectory results and the effect of the embedding mask are prseneted.

References

[1] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.*, 32(11):1231–1237, 2013. 2, 3, 4, 5

[2] Guangming Wang, Xinrui Wu, Zhe Liu, and Hesheng Wang. Hierarchical attention learning of scene flow in 3d point clouds. *arXiv preprint arXiv:2010.05762*, 2020. 2, 3

[3] Wenxuan Wu, Zhiyuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: A coarse-to-fine network for supervised and self-supervised scene flow estimation on 3d point clouds. *arXiv preprint arXiv:1911.12408*, 2019. 2

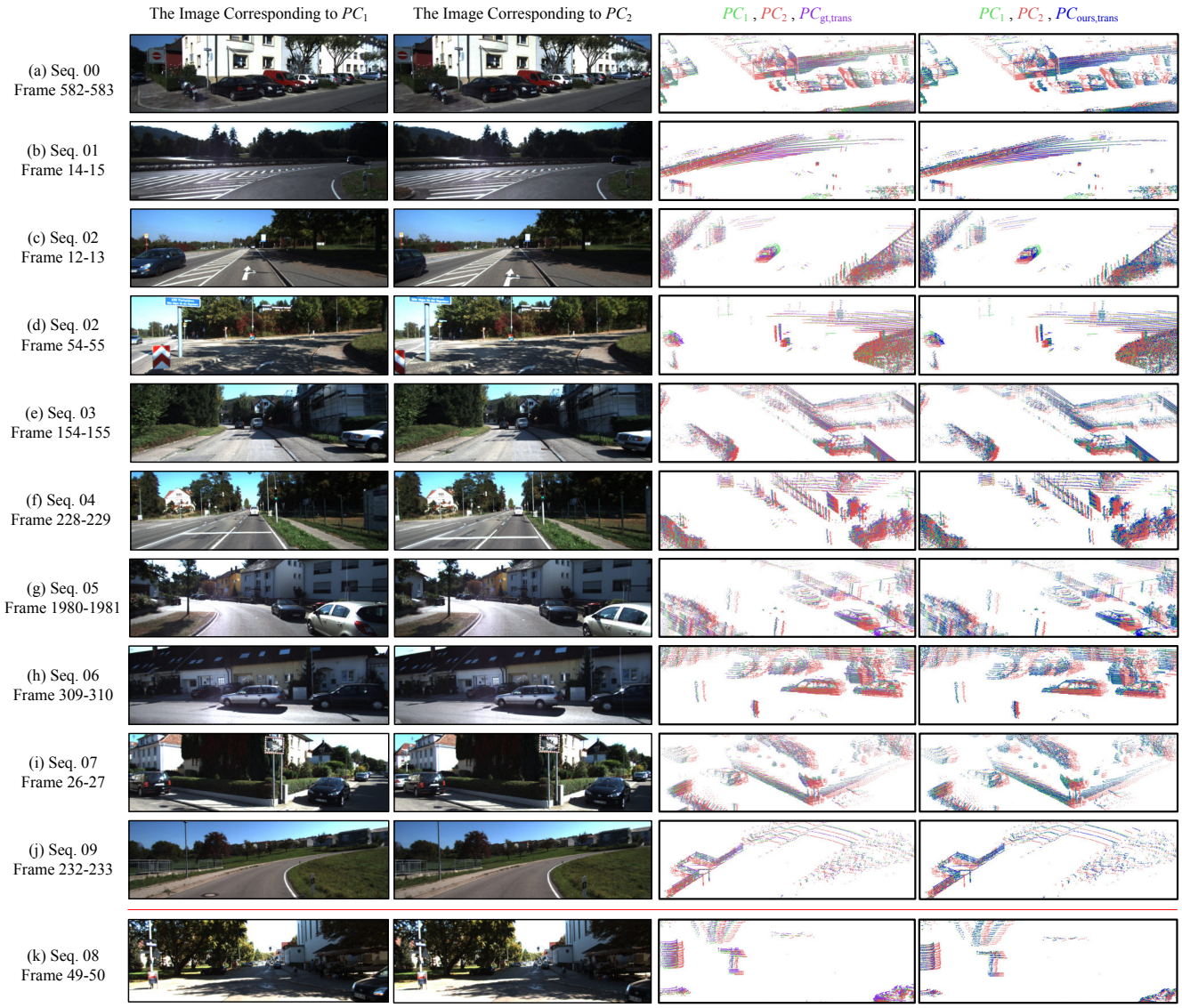
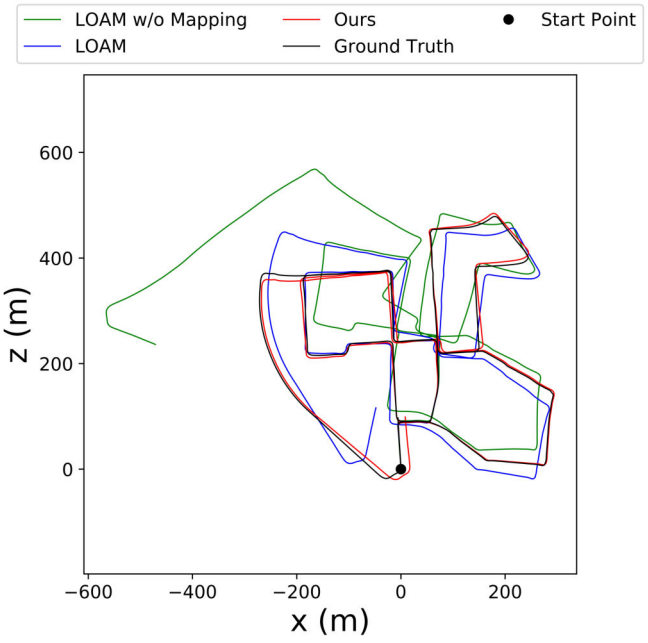
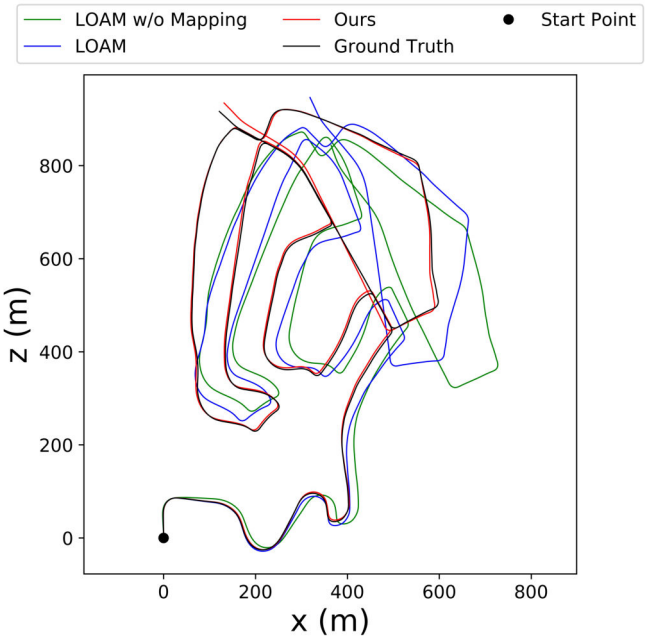


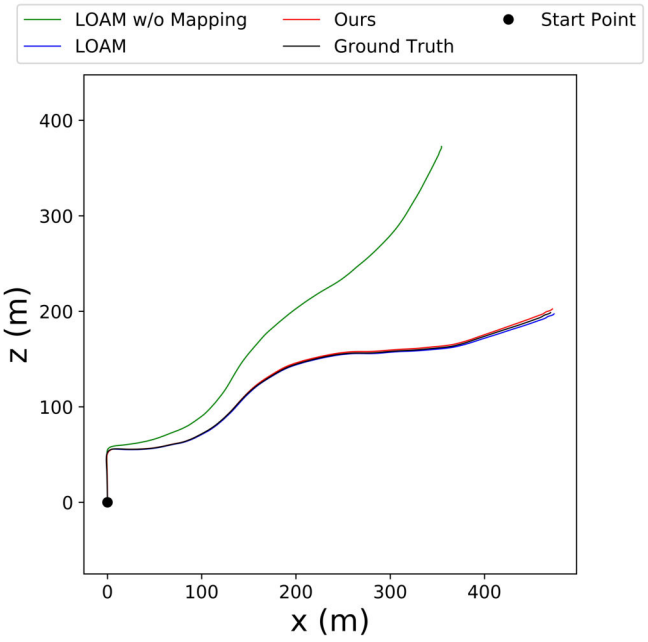
Figure 2: Some sucessful cases of odometry estimation on KITTI odometry dataset [1] with ground truth. The image corresponding to PC_1, the image corresponding to PC_2, the point clouds visulaization of PC_1 (gree color), PC_2 (red color), $PC_{gt,trans}$ (purple color), and $PC_{ours,trans}$ (blue color) are visualized. The ground truth pose visualized in the last line has error, while ours has a good effect of registration.



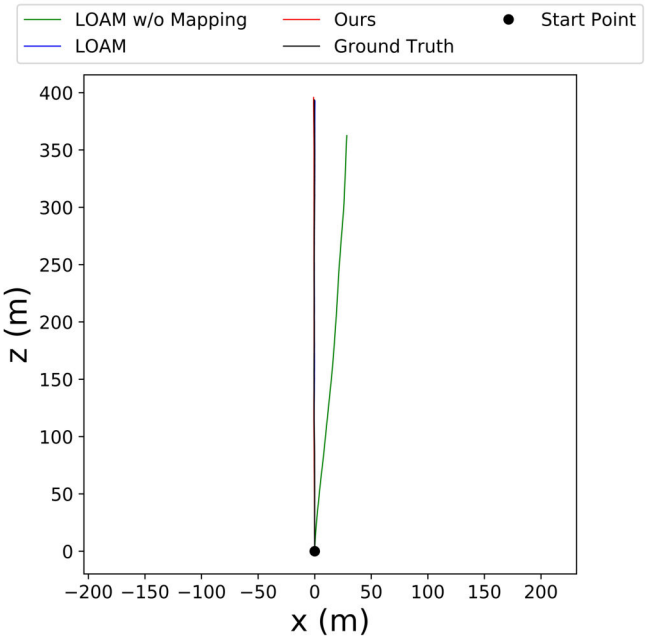
(a) 2D Trajectory Plots of Seq. 00



(b) 2D Trajectory Plots of Seq. 02

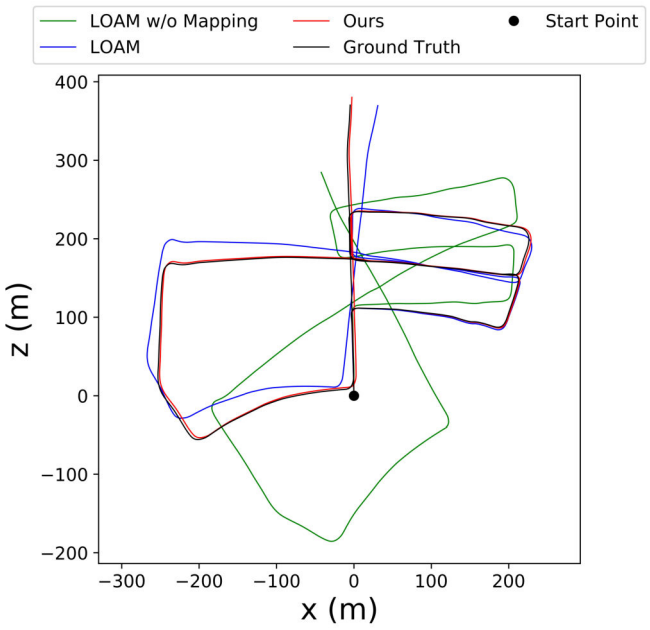


(c) 2D Trajectory Plots of Seq. 03

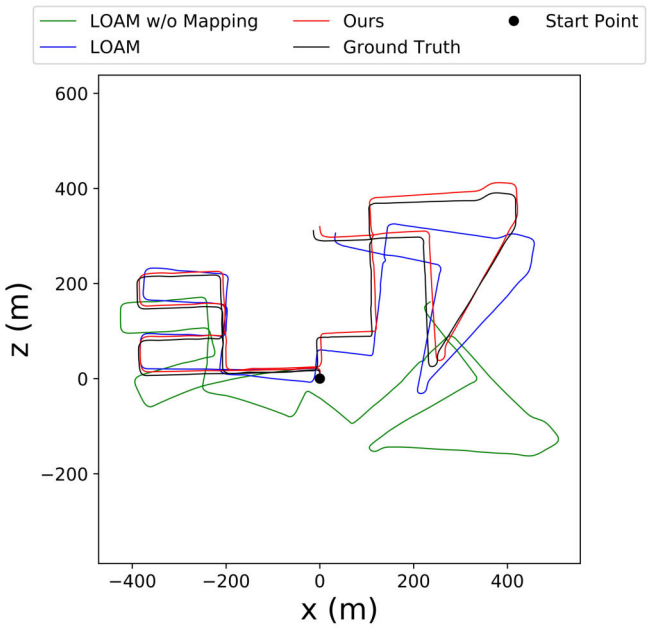


(d) 2D Trajectory Plots of Seq. 04

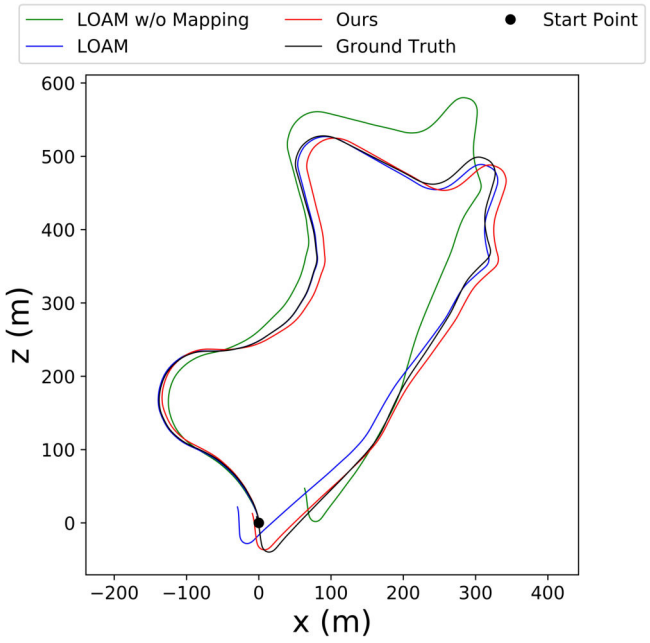
Figure 3: Trajectory results of LOAM and ours on KITTI training sequences 00, 02, 03, and 04 with ground truth.



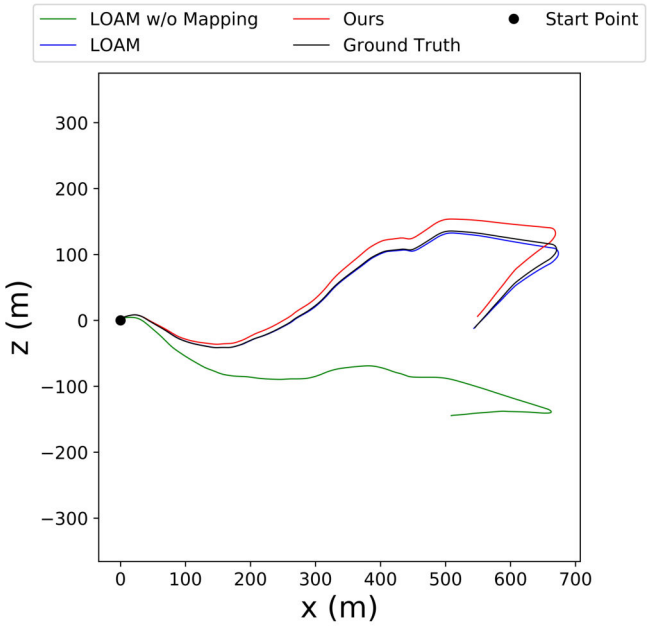
(a) 2D Trajectory Plots of Seq. 05



(b) 2D Trajectory Plots of Seq. 08



(c) 2D Trajectory Plots of Seq. 09



(d) 2D Trajectory Plots of Seq. 10

Figure 4: Trajectory results of LOAM and ours on KITTI training sequence 05, and validation sequences 08, 09, and 10 with ground truth.