Supplementary Material
PICCOLO: Point Cloud-Centric Omnidirectional Localization

Junho Kim, Changwoon Choi, Hojun Jang, and Young Min Kim
Dept. of Electrical and Computer Engineering, Seoul National University, Korea
82magnolia@snu.ac.kr, changwoon.choi00@gmail.com, {j12040208, youngmin.kim}@snu.ac.kr

A. Additional Implementation Details

In this section, we provide several additional details in PICCOLO that are not provided in the original paper. The majority of the description of PICCOLO is provided in Section 3.

A.1. Gradient Step Size Scheduling

To foster convergence, we adaptively decay the gradient step size α (line 7 of Algorithm 1) similarly to learning rate scheduling widely used in neural network training. In our experiments, we decay the step size by a factor of 0.8 if the loss function value does not decrease for 5 consecutive iterations.

B. Hyperparameter Setup

We report the hyperparameter setups of PICCOLO, where the configurations vary only depending on the providence of gravity direction. As shown in Algorithm 1, we first sample $N_r \times N_t$ initial poses, out of which we select K_1 poses through loss value-based filtering, which are consecutively reduced into K_2 poses using color histogram intersection. Then we run gradient descent for K_2 starting points for N_{iter} steps. The quantitative results are reported in Section 4.1.

B.1. Unknown Gravity Direction

For inputs where the gravity direction is unknown, we use $N_r = 32, N_t = 50, N_{iter} = 100, K_1 = 50, K_2 = 6$. Such a setup is used in Table 1, 2, 3, and 5 (excluding 'gravity direction'). This setup shows effective performance in both indoor / outdoor datasets (Stanford2D-3D-S [4], MPO [7]), and generalizes to arbitrary point cloud rotations and flipped images. The capacity of PICCOLO to generalize in such diverse inputs under a fixed hyperparameter configuration alludes to its potential as an off-the-shelf omnidirectional localization algorithm.

B.2. Known Gravity Direction

We use the following setup for inputs where the gravity direction is known: $N_r = 8, N_t = 50, N_{iter} = 100, K_1 = 50, K_2 = 6$. Such a setup is used in Table 4 and 5 ('gravity direction'). Since the gravity direction is known, the search space can be dramatically reduced, and PICCOLO can perform highly accurate localization, as shown in Table 4 and 5.

C. Dataset Details

Here we report minor experimental details about the datasets used in Section 4. For all datasets, we remove panoramas where the ground truth camera position is outside the point cloud's bounding box.

C.1. Stanford2D-3D-S

For GOSMA [6], if the room size exceeds a certain limit, segmentation fault occurs and the algorithm terminates. We exclude those cases when computing the accuracy. We additionally describe the 33 rooms used for creating Table 2. These rooms are chosen using the criterion of Campbell *et al.* [6]: (i) distance to the closest point is greater than 50 cm, (ii) number of labeled pixels should be greater than 2000, (iii) number of 3D points should be greater than 2000.

C.2. MPO

A peculiar aspect of the MPO dataset is that all the ground truth values are the same: they are fixed to $R^* = I, t^* = \mathbf{0}$. Therefore, we apply random rotation and translation to the point cloud to make the dataset more challenging. For rotation, we randomly rotate the point cloud along the z-axis by an angle $\theta \sim \mathcal{U}(0, \pi)$, where $\mathcal{U}(\cdot)$ denotes the uniform distribution. For translation, we randomly apply translations along the x, y, z directions, where $x, y, z, \sim \mathcal{U}(0, 3)$. Note that the units for x, y, and z are in meters.

C.3. OmniScenes

In this section, we report the acquisition process of our newly proposed OmniScenes dataset. OmniScenes dataset

Scenario	Scene Change	# of Images	# of Scenes
Handheld	×	1451	8
Robot	×	1129	8
Handheld	○	698	8
Robot	○	1132	8

Table C.1: Statistical Properties of the OmniScenes dataset.

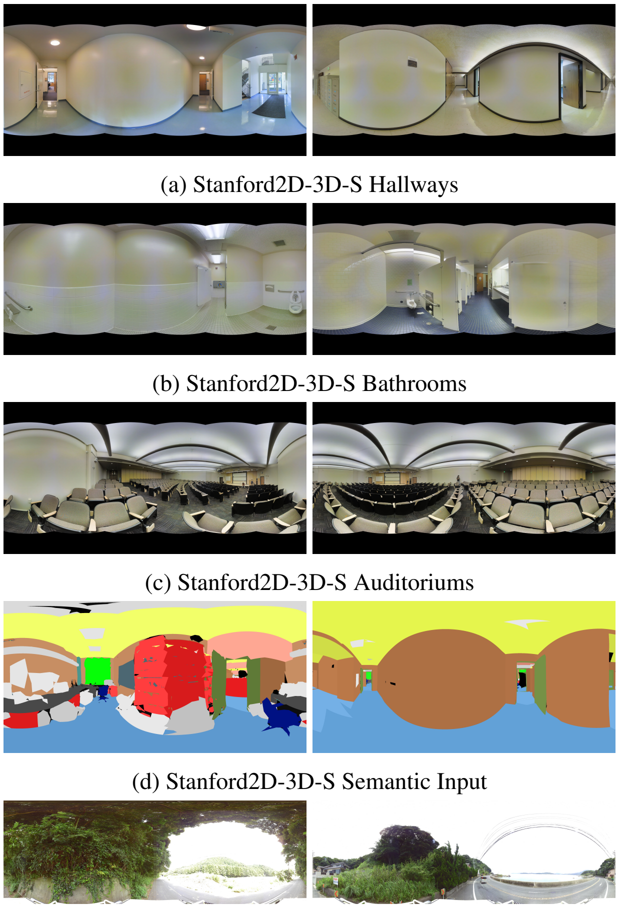
is composed of 3D scans of 8 scenes accompanied by omnidirectional images. 3D scans are collected using the Matterport 3D scanner [1]. The corresponding omnidirectional images are collected with the Ricoh Theta 360° camera [2] under two scenarios: handheld and mobile robot-mounted. For the handheld case, we have the capturer to take 360° videos while walking around the eight scenes. For the mobile robot-mounted case, we use a TurtleBot3 Waffle [3] that is manually controlled. To obtain 6DoF pose annotations for each omnidirectional image, we apply SfM (Structure from Motion), as in Kendall *et al.* [8]. COLMAP [11, 12] and OpenMVG [10] are used to obtain dense SfM reconstructions, which are then manually aligned with the Matterport 3D scans, similar to Valentin *et al.* [13]. The aligned SfM model contains omnidirectional camera poses with respect to the Matterport 3D scans. We further remove invalid pose estimates from SfM through manual inspection. The resulting dataset, OmniScenes, contains a wide variety of pose-annotated omnidirectional images. The statistical properties of our dataset are displayed in Table C.1. Also, we show qualitative samples from OmniScenes in Figure H.4, H.5, H.6, H.7.

D. Difficulties of Using SIFT

The main factor that hinders the use of SIFT [9] on our problem setup is the abundance of scenes with repetitive structure or lacking visual features. For the Stanford2D-3D-S dataset [4], 660 out of 1413 panoramic images are hallways, bathrooms, and auditoriums, which exhibit the aforementioned characteristics, as shown in Figure D.1a, D.1b, D.1c. Further, once the visual inputs are given as semantic labels as in Figure D.1d, it is impossible to use SIFT [9] since the colored labels don't have any distinct features. For the MPO dataset [7], 229 out of 650 panoramic images are coasts and forests, which are also very difficult to establish sparse visual correspondences, as shown in Figure D.1e.

E. Qualitative Comparison with GOSMA [6]

We make a qualitative comparison of PICCOLO and GOSMA in Figure E.1. While GOSMA is successful in small rooms such as offices, it often fails in large areas such as auditoriums. PICCOLO is capable of performing stable omnidirectional localization under diverse challenging environments. Further, recall that semantic labels were given as input, to ensure fair comparison with GOSMA [6]. This indicates that PICCOLO can function seamlessly with any other point-wise information.



(a) Stanford2D-3D-S Hallways

(b) Stanford2D-3D-S Bathrooms

(c) Stanford2D-3D-S Auditoriums

(d) Stanford2D-3D-S Semantic Input

(e) MPO

Figure D.1: Sample data from Stanford2D-3D-S [4] and MPO [7] which lack visual features and exhibit repetitive structure.

# of Points	Initialization	Gradient Descent
10^4	1.77	0.01
10^5	1.90	0.01
10^6	3.74	0.02

Table F.1: Runtime analysis of PICCOLO. All runtime statistics are reported in seconds.

F. Runtime Analysis

In this section, we examine the runtime properties of PICCOLO. Experiments are conducted with a single RTX

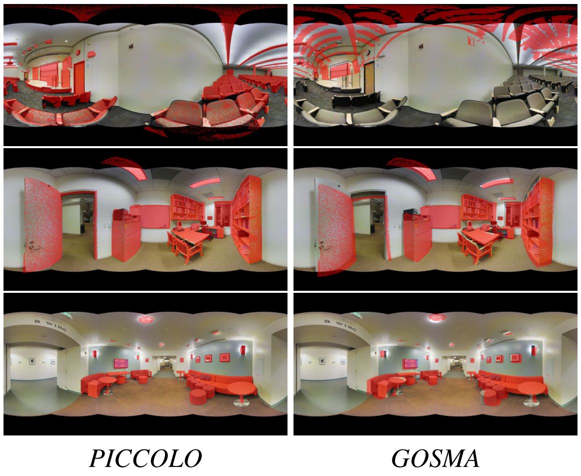


Figure E.1: Qualitative results of PICCOLO and GOSMA [6] in Stanford2D-3D-S dataset [4]. Projected point cloud coordinates are overlayed in red.

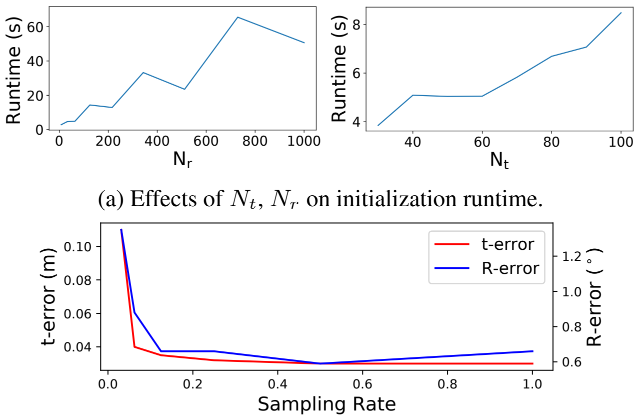
2080 GPU and an Intel Core i7-9700 3.00GHz CPU. We report the amount of time it takes for initialization and gradient descent in Table F.1, with a varying number of points in the input point cloud. We use the same configuration used for unknown gravity direction: $N_r = 32, N_t = 50, N_{iter} = 100, K_1 = 50, K_2 = 6$.

Table F.1 shows that initialization (line 2, 3 of Algorithm 1) terminates within a few seconds, and gradient descent (line 7 of Algorithm 1) finishes on the scale of milliseconds. Note that both initialization and gradient descent are easily parallelizable. Thus PICCOLO can directly benefit from the presence of multiple GPUs, similar to GOSMA [6] and GOPAC [5]. Furthermore, while the number of points is increased by a factor of 10, the runtime only shows a modest increase. Sampling loss scales seamlessly to large point clouds, as no costly operations such as visibility computation take place.

G. Additional Ablation Study

In this section, we perform additional ablation study on PICCOLO. We examine the number of starting points on initialization runtime and the effect of point cloud density on pose error. The results are displayed in Figure G.1.

Number of Starting Points PICCOLO is tested with varying numbers of translation and rotation starting points N_t, N_r on offices from Area 2 of the Stanford2D-3D-S [4] dataset. The median initialization runtime is reported with either N_t or N_r modified from the original setup. Figure G.1a indicates that while increasing N_t, N_r leads to enhanced performance as shown in Figure 4, this also incurs longer runtime. An appropriate N_t, N_r that balances the trade-off should be chosen. We use $N_t = 50, N_r = 32$ in all our experiments, which shows competent performance yet maintains fast runtime.



(a) Effects of N_t, N_r on initialization runtime.

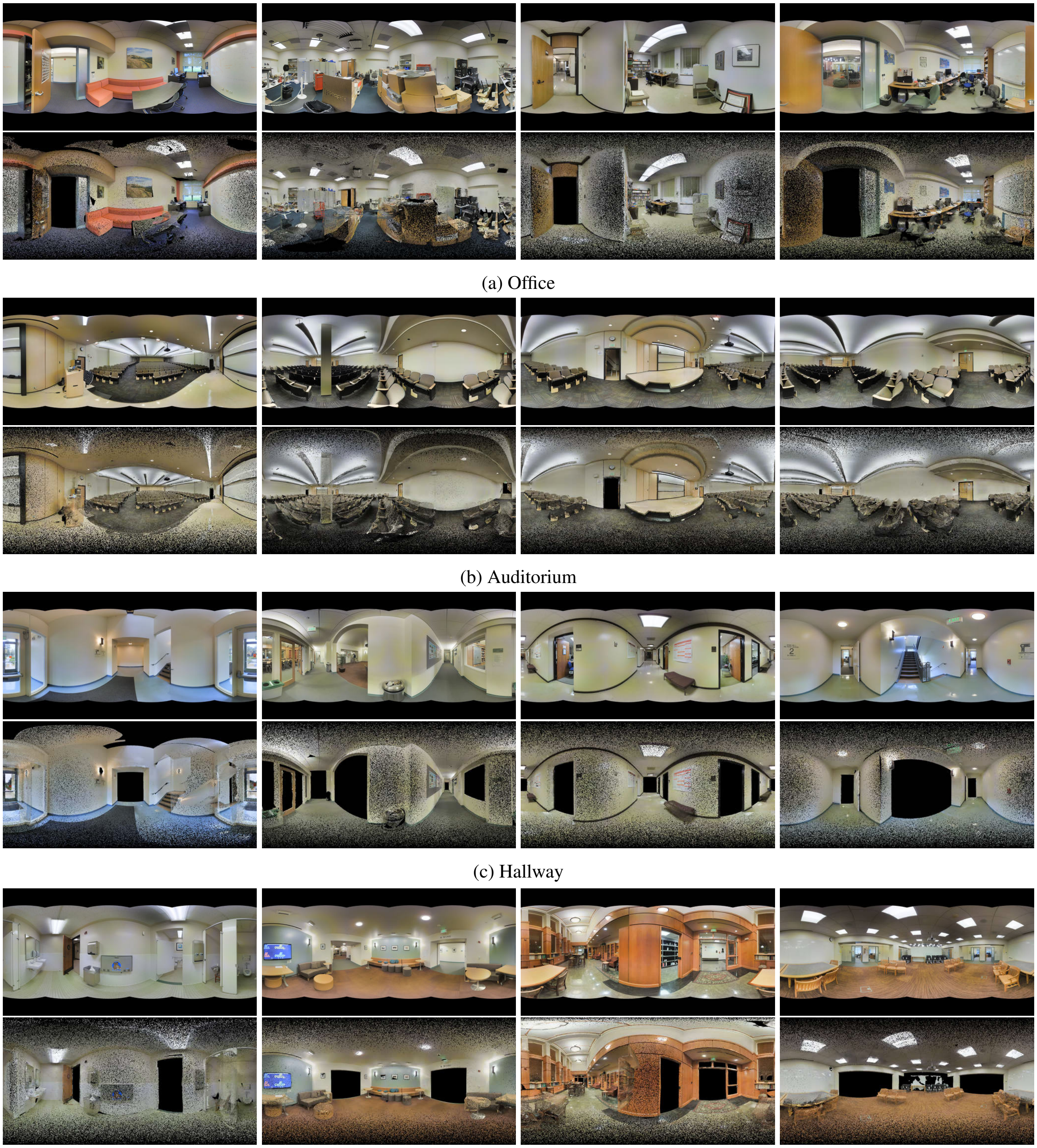
(b) Effect of point cloud sampling rate on localization error.

Figure G.1: Additional ablation study on PICCOLO.

Point Cloud Density PICCOLO is evaluated with varying point cloud sampling rates on the Stanford2D-3D-S [4] dataset. As seen in Figure G.1b, PICCOLO is robust against point cloud density: pose estimation error is very small even when less than 5% of the entire point cloud is used.

H. Additional Qualitative Results

H.1. Stanford2D-3D-S RGB Input



(a) Office

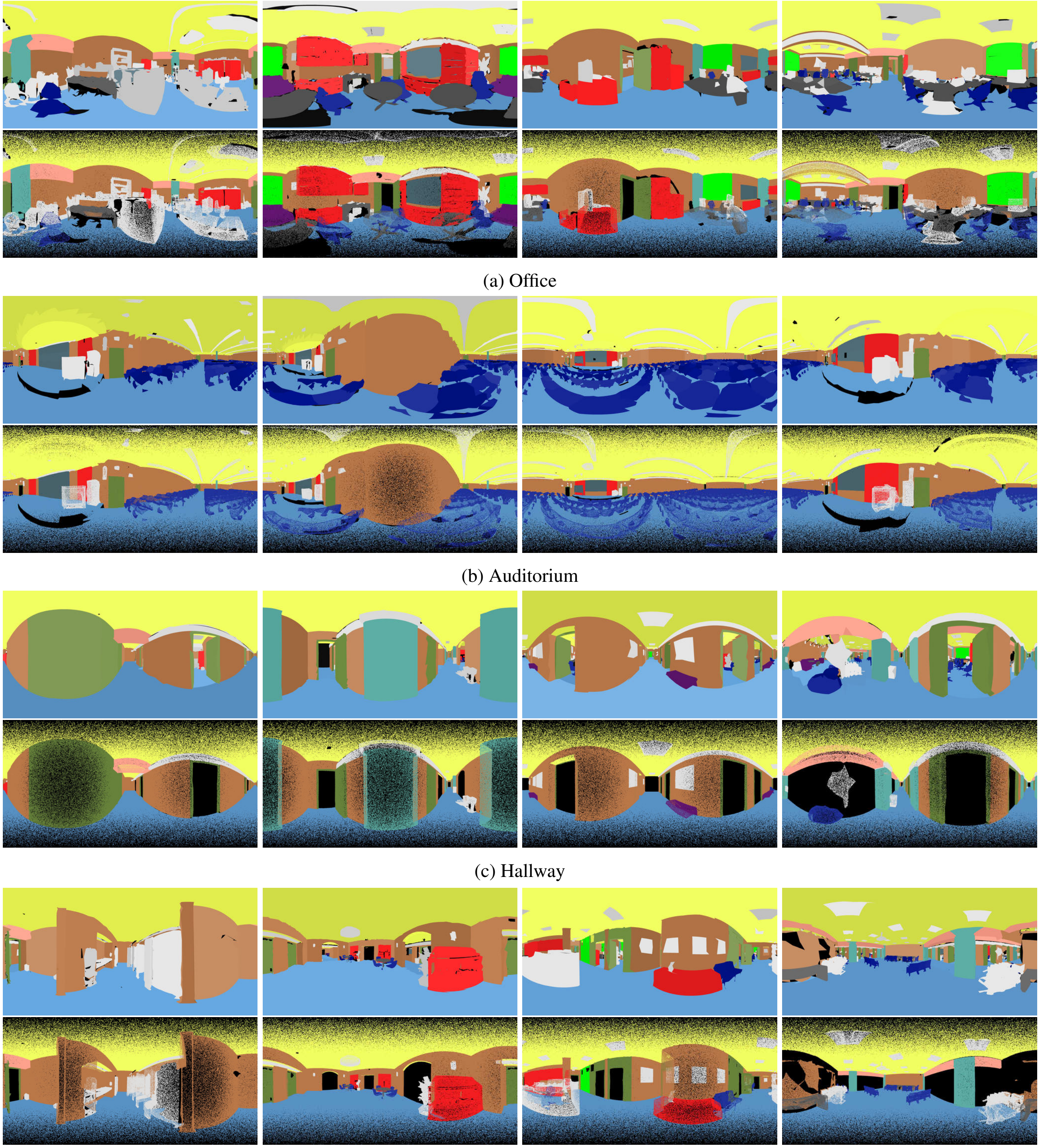
(b) Auditorium

(c) Hallway

(d) WC (e) Lounge (f) Lobby (g) Openspace

Figure H.1: Qualitative results of PICCOLO. We display the input query image (top) and the projected point cloud under the estimated camera pose (bottom).

H.2. Stanford2D-3D-S Semantic Input



(a) Office

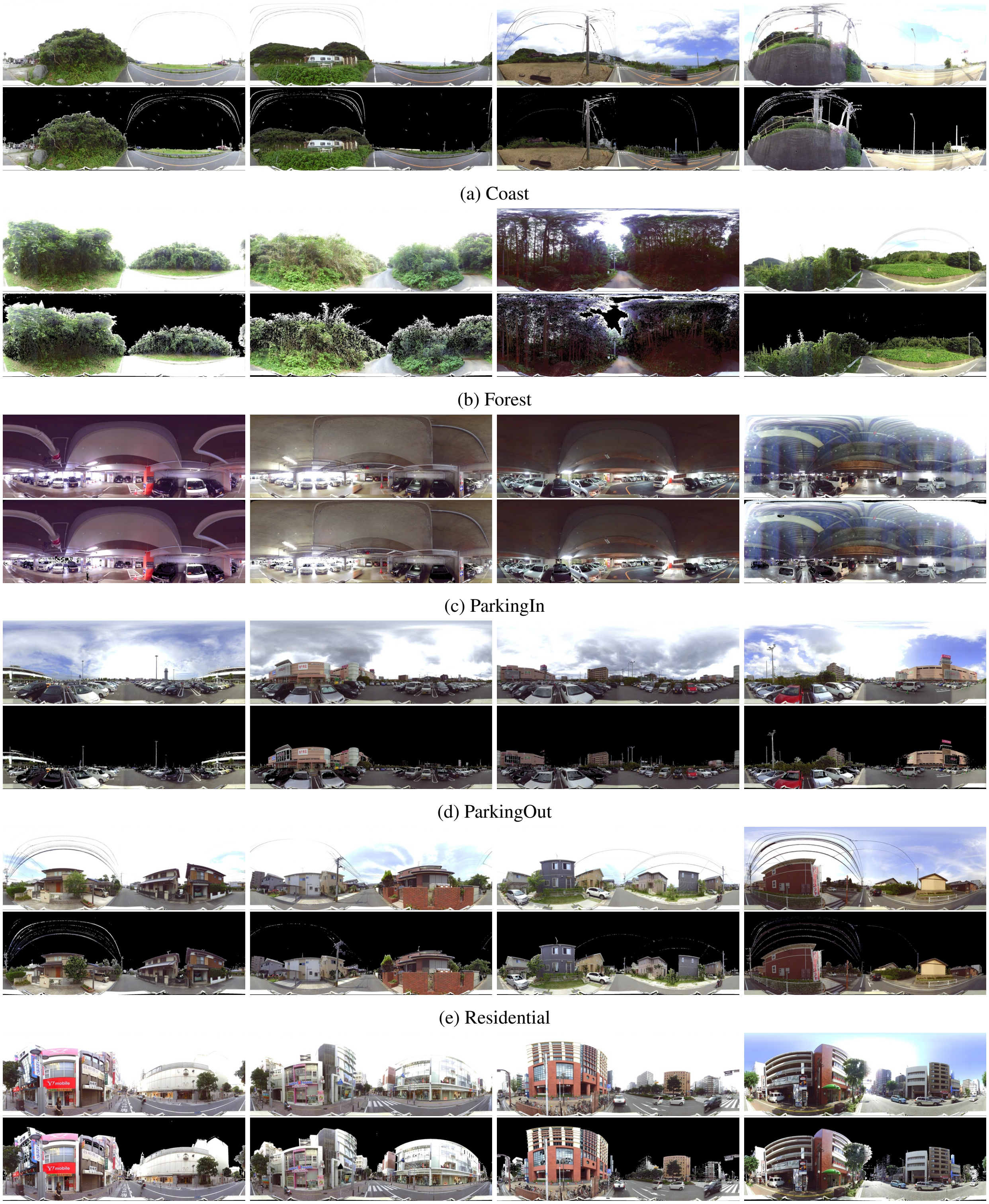
(b) Auditorium

(c) Hallway

(d) WC (e) Lounge (f) Lobby (g) Openspace

Figure H.2: Qualitative results of PICCOLO. We display the input query image (top) and the projected point cloud under the estimated camera pose (bottom).

H.3. MPO



(a) Coast

(b) Forest

(c) ParkingIn

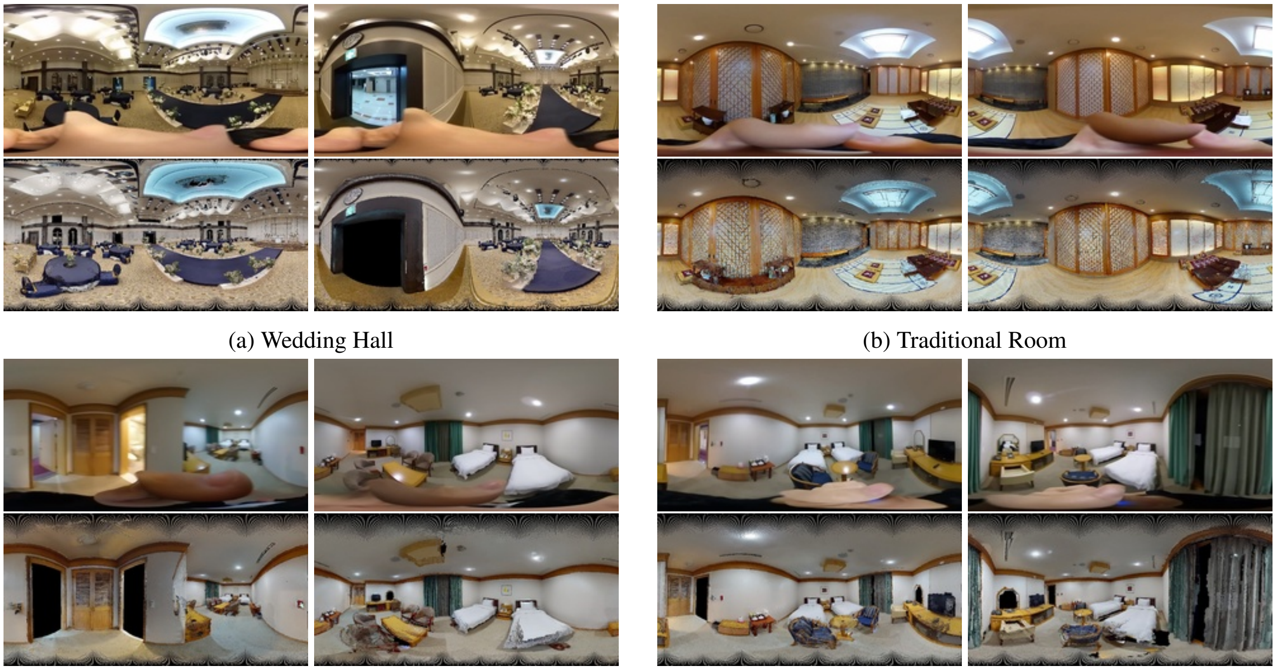
(d) ParkingOut

(e) Residential

(f) Urban

Figure H.3: Qualitative results of PICCOLO. We display the input query image (top) and the projected point cloud under the estimated camera pose (bottom).

H.4. OmniScenes Handheld



(a) Wedding Hall

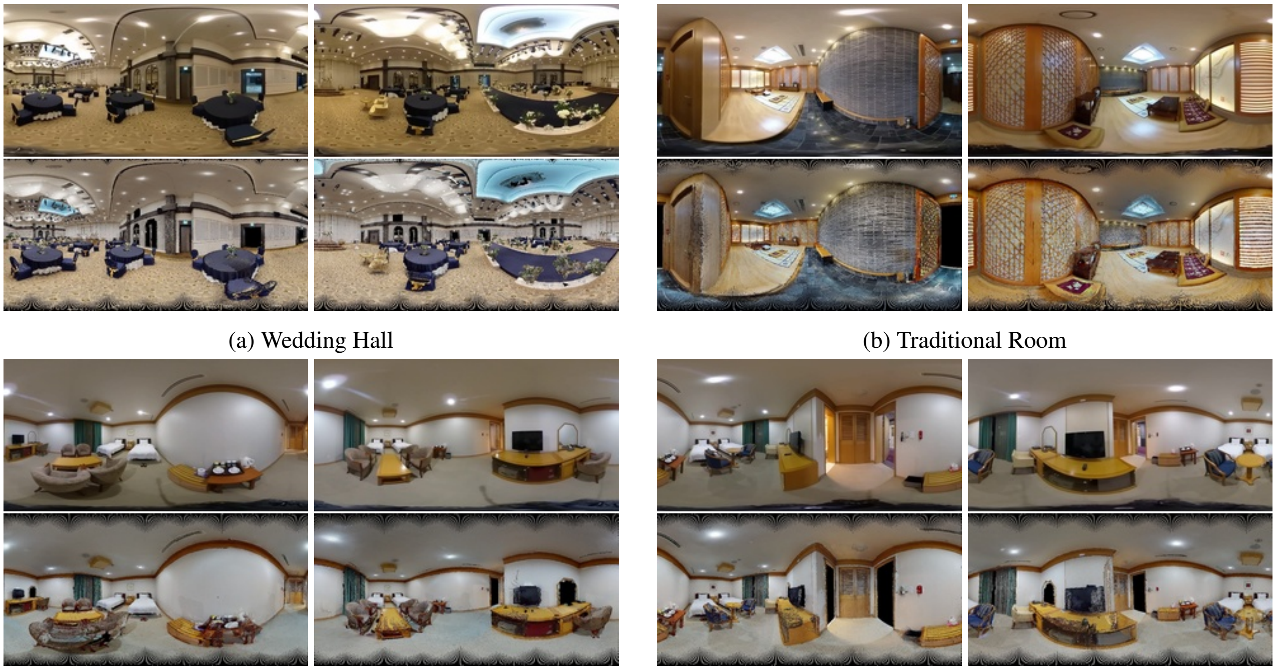
(b) Traditional Room

(c) Hotel Room 1

(d) Hotel Room 2

Figure H.4: Qualitative results of PICCOLO. We display the input query image (top) and the projected point cloud under the estimated camera pose (bottom).

H.5. OmniScenes Robot-Mounted



(a) Wedding Hall

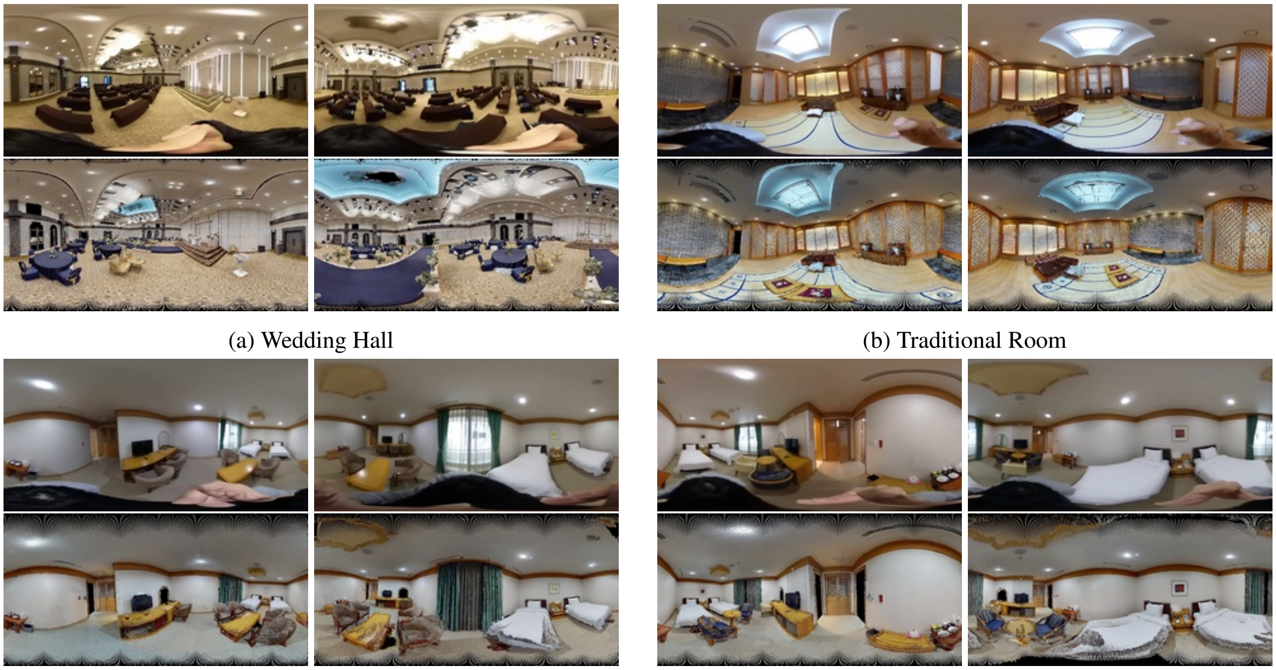
(b) Traditional Room

(c) Hotel Room 1

(d) Hotel Room 2

Figure H.5: Qualitative results of PICCOLO. We display the input query image (top) and the projected point cloud under the estimated camera pose (bottom).

H.6. OmniScenes Handheld with Scene Change


(a) Wedding Hall

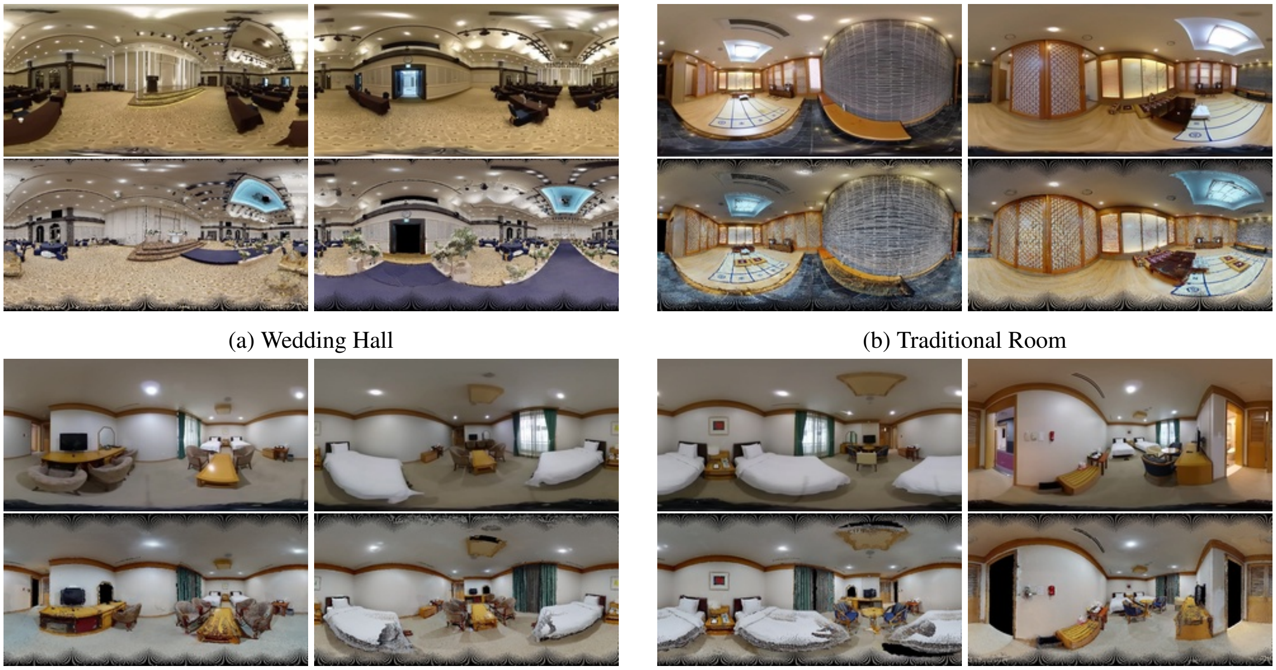
(b) Traditional Room

(c) Hotel Room 1

(d) Hotel Room 2

Figure H.6: Qualitative results of PICCOLO. We display the input query image (top) and the projected point cloud under the estimated camera pose (bottom).

H.7. OmniScenes Robot-Mounted with Scene Change


(a) Wedding Hall

(b) Traditional Room

(c) Hotel Room 1

(d) Hotel Room 2

Figure H.7: Qualitative results of PICCOLO. We display the input query image (top) and the projected point cloud under the estimated camera pose (bottom).

References

[1] Matterport 3d: How long does it take to scan a property? https://support.matterport.com/hc/en-us/articles/229136307-How-long-does-it-take-to-scan-a-property-. Accessed: 2020-02-18. 2

[2] Ricoh theta, experience the world in 360. https://theta360.com/en/. Accessed: 2021-03-16. 2

[3] What is turtlebot? https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/. Accessed: 2021-03-16. 2

[4] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 1, 2, 3

[5] Dylan Campbell, Lars Petersson, Laurent Kneip, and Hongdong Li. Globally-optimal inlier set maximisation for camera pose and correspondence estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page preprint, June 2018. 3

[6] Dylan Campbell, Lars Petersson, Laurent Kneip, Hongdong Li, and Stephen Gould. The alignment of the spheres: Globally-optimal spherical mixture alignment for camera pose estimation. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page to appear, Long Beach, USA, June 2019. IEEE. 1, 2, 3

[7] H. Jung, Y. Oto, O. M. Mozos, Y. Iwashita, and R. Kurazume. Multi-modal panoramic 3d outdoor datasets for place categorization. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4545–4550, 2016. 1, 2

[8] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. 2015. 2

[9] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 2

[10] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. Openmvg: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016. 2

[11] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[12] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[13] J. Valentin, A. Dai, M. Niessner, P. Kohli, P. Torr, S. Izadi, and C. Keskin. Learning to navigate the energy landscape. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 323–332, 2016. 2