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

Abstract

We present PICCOLO, a simple and efficient algorithm for omnidirectional localization. Given a colored point cloud and a 360° panorama image of a scene, our objective is to recover the camera pose at which the panorama image is taken. Our pipeline works in an off-the-shelf manner with a single image given as a query and does not require any training of neural networks or collecting ground-truth poses of images. Instead, we match each point cloud color to the holistic view of the panorama image with gradient-descent optimization to find the camera pose. Our loss function, called sampling loss, is point cloud-centric, evaluated at the projected location of every point in the point cloud. In contrast, conventional photometric loss is image-centric, comparing colors at each pixel location. With a simple change in the compared entities, sampling loss effectively overcomes the severe visual distortion of omnidirectional images, and enjoys the global context of the 360° view to handle challenging scenarios for visual localization. PICCOLO outperforms existing omnidirectional localization algorithms in both accuracy and stability when evaluated in various environments.

1. Introduction

With the recent advancements in 3D sensing technology, 3D maps of the environment are often available for download [1] or can be easily captured with commodity sensors [9]. The 3D map and the accurate location of the user within the map provide crucial information for AR/VR applications or other location-based services. Visual localization is a cheap localization method as it only uses an image input and utilizes the 3D map without additional sensors such as WIFI, GPS, or gyroscopes. However, visual localization is fragile to changes in illumination or local geometric variations resulting from object displacements [37, 43]. Further, with the limited field of view, perspective cameras often fail to regress the camera pose when the observed image lacks visual features (e.g., a plain wall) or the scene exhibits symmetric or repetitive structure [42, 40].

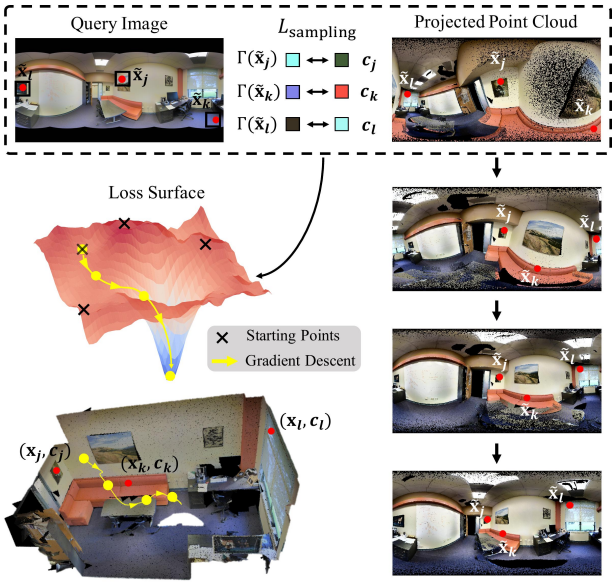


Figure 1: Overview of our approach. PICCOLO minimizes a novel, point cloud-centric loss function called sampling loss. After the initialization phase trims off local minima, PICCOLO minimizes the sampling loss with gradient descent.

Omnidirectional cameras, equipped with a 360° field of view, provide a holistic view of the surrounding environment. Hence these cameras are immune to small scene changes and ambiguous local features [47], which gives them the potential to dramatically improve the performance of visual localization algorithms. However, the large field of view comes with a cost: significant visual distortion caused by the spherical projection equation. This makes it difficult to directly apply conventional visual localization algorithms on omnidirectional cameras [21, 12, 45, 15, 7], as many visual localization algorithms [46, 40, 42, 39] do not account for distortion. Furthermore, learning-based approaches are bound to the settings they are trained on, and cannot generalize to arbitrary scenes.

In this paper, we introduce PICCOLO, a simple yet ef-

Stanford2D-3D-S *MPO* *OmniScenes* *OmniScenes (w. change)*

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

fective omnidirectional localization algorithm. PICCOLO optimizes over *sampling loss*, which samples color values from the query image and compares them with the point cloud color. We only utilize the color information from point clouds, as it is usually available from raw measurements. With a simple formulation, PICCOLO can be adapted to any scene with 3D maps in an off-the-shelf manner. Further, PICCOLO can work seamlessly with any other point-wise information, such as semantic segmentation labels shown in Figure 5. Sampling loss is *point cloud-centric*, as every point is taken into consideration. In contrast, conventional photometric loss widely used in computer vision [13, 11] evaluates the color difference at every pixel location [11, 30], thus is *image-centric*. Our point cloud-centric formulation leads to a significant performance boost in omnidirectional localization, where the image-centric approach suffers from distorted omnidirectional images unless the distortion is explicitly considered with additional processing [44, 15].

The gradient of our proposed sampling loss can be efficiently obtained with differentiable sampling [20]. While differentiable sampling is widely used to minimize discrepancies in the projected space, it is usually part of a learned module [13, 18]. Instead, we utilize the operation in a standalone fashion, making our framework cheap to compute. We further accelerate the loss computation by ignoring the non-differentiable, costly components of projection, such as occlusion handling. These design choices make sampling loss very fast: it only takes 3.5 ms for 10^6 points on a commodity GPU. With the rich information of the global context in point cloud color, our efficient formulation is empirically robust against visual distortions and more importantly, local scene changes. The algorithm quickly converges to the global minimum of the proposed loss function as shown in Figure 1.

Equipped with a light-weight search for decent starting points, PICCOLO achieves stable localization in various datasets. The algorithm is extensively evaluated on indoor/outdoor scenes and scenes with dynamic camera motion, scene changes, and arbitrary point cloud rotation. Several qualitative results of our algorithm are shown in Figure 2 and 5. In addition, we introduce a new dataset called OmniScenes to highlight the practicality of PICCOLO. OmniScenes contains diverse recordings with significant scene changes and motion blur, making it the first dataset targeted for omnidirectional localization where visual localization algorithms frequently malfunction. PICCOLO consistently exhibits performance superior to the previous approaches [6, 44] in all of the tested datasets under a fixed hyperparameter configuration, indicating the practical effectiveness of our algorithm.

2. Related Work

Before we introduce PICCOLO in detail, we clarify our problem setup and how it differs from previous visual localization algorithms [35, 40, 41, 4]. Then we will further describe recent algorithms proposed for omnidirectional localization.

Learning-based Algorithms A large body of recent visual localization literature trains an algorithm on the database of RGB (and possibly depth) images annotated with ground truth poses [41, 4, 27, 39, 40, 23, 42, 32, 17, 33, 34]. While such training facilitates highly accurate camera pose estimation [41, 4, 40, 36], it limits the applicability of these algorithms. To estimate camera pose in new, unseen environments, these algorithms typically require additional pose-labelled samples.

In order to develop an algorithm that could be readily used in an off-the-shelf manner, we make a slight detour from these previous setups: the camera pose must be found *solely* using the point cloud and query image information. One may opt to train these learning-based models [41, 4, 27] with synthesized views from the point cloud as in Zhang

et al. [44]. However, it is costly to obtain such rendered views, and one must devise a way to reduce the domain gap between synthesized images and real query images, which is a non-trivial task.

Feature-based algorithms Another line of work utilizes visual features for localization [35, 36, 28, 19, 38, 8]. Feature-based localization algorithms require each 3D point to be associated with a visual feature, typically SIFT [29], necessitating a structure-from-motion (SfM) point cloud. Provided an efficient search scheme [36, 35, 28], it is relatively straightforward to establish 2D-3D correspondences by matching features extracted from the query image with those in the SfM model.

Our input point cloud is not limited to a structure-from-motion (SfM) point cloud. Due to the developments in RGB-D sensors and Lidar scanners, 3D point clouds of a scene could be obtained in a wide variety of ways other than SfM. These point clouds do not contain associated visual features for feature-based localization. Our setup also does not provide any explicit 2D-3D correspondences, thus disabling the direct usage of PnP algorithms [16, 26]. Further, many point clouds and query images used in our experiments contain repetitive structures or regions that lack features as shown in Figure 5. This hinders the usage of sparse local features such as SIFT [29] in our setup, where we report additional difficulties for using SIFT in the supplementary material. To accommodate these challenges, PICCOLO incorporates information from dense RGB measurements, which are easy to obtain in practice, and are robust against local ambiguities.

Omnidirectional Localization Visual localization on omnidirectional images requires an algorithm specifically designed to account for the unique visual distortion [21, 12, 45, 15]. A number of techniques have been proposed in recent years that tackle visual localization with omnidirectional cameras. These techniques could be divided into two groups, namely algorithms that utilize global optimization techniques and others that leverage deep learning. Campbell *et al*. [5, 6] proposed a family of global optimization-based algorithms for camera pose estimation, GOSMA [6] and GOPAC [5], that could be readily applied for omnidirectional localization in diverse indoor and outdoor environments. While these algorithms have solid optimality guarantees and competitive performance, semantic labels should be fed to these algorithms as additional inputs for reasonable accuracy. On the other hand, deep learning-based omnidirectional localization algorithms such as Zhang *et al*. [44], train neural networks that learn rotationally equivariant features to effectively process omnidirectional images. Although such features enable omnidirectional localization under arbitrary camera rotations, these algorithms cannot generalize to unobserved scenes as they require training on pose-annotated images. We compare the localization performance of PICCOLO with optimization-based localization algorithms GOSMA [6], GOPAC [5], and deep learning-based localization algorithms from Zhang *et al*. [23, 44, 7].

3. Method

PICCOLO is a point cloud-centric omnidirectional localization algorithm, which finds the optimal $SE(3)$ camera pose with respect to the colored point cloud at which the 360° panorama image is taken. PICCOLO solely relies on the point cloud data and the input query image. It does not require a separate training process or explicit 2D-3D correspondences, and therefore could be used in an off-the-shelf manner. We first introduce the formulation of sampling loss, which is the objective function that PICCOLO aims to minimize. Then we will describe our light-weight initialization scheme.

Sampling Loss Given a point cloud $P = \{X, C\}$ and a *single* query image $I \in \mathbb{R}^{H\times W\times 3}$, where $X, C \in \mathbb{R}^{N\times 3}$ are the point cloud coordinates and color values, the objective is to find the optimal rotation $R^* \in SO(3)$ and translation $t^* \in \mathbb{R}^3$ at which the 360° panorama image I is taken. Denote $\Pi(\cdot) : \mathbb{R}^3 \to \mathbb{R}^2$ as the projection function that maps a point $\mathbf{x} = (x_1, x_2, x_3)$ in 3D to a point $\tilde{\mathbf{x}} \in [0, H) \times [0, W)$ in the 360° panorama image's coordinate frame. This could be explicitly written as follows,

$$\Pi(\mathbf{x}) = \left(\frac{H}{\pi}\text{atan}\left(\frac{x_3}{\sqrt{x_1^2 + x_2^2}}\right), \frac{W}{2\pi}\text{atan}\left(\frac{x_2}{x_1}\right)\right). \quad (1)$$

Furthermore, let $\Gamma(\cdot; I)$ indicate the sampling function that maps 2D coordinates $\tilde{\mathbf{x}} \in [0, W) \times [0, H)$ to pixel values $\mathbf{c} \in \mathbb{R}^3$ sampled from the query image I under a designated sampling kernel. Suppose $\Gamma(\cdot; I), \Pi(\cdot)$ could be 'vectorized', i.e., if the input $\tilde{X}$ consists of N points in $\mathbb{R}^2$, $\Gamma(\tilde{X}; I) \in \mathbb{R}^{N\times 3}$ are the sampled image values at 2D coordinates $\tilde{X}$, and vice versa for $\Pi(\cdot)$.

Under this setup, $\Pi(X) \in \mathbb{R}^{N\times 2}$ could be regarded as tentative *sampling locations*, and $\Gamma(\Pi(X); I) \in \mathbb{R}^{N\times 3}$ as the *sampled image values*. If the point cloud P is perfectly aligned with the omnidirectional camera's coordinate frame, one could expect the sampled image values $\Gamma(\Pi(X); I)$ to be very close to the point cloud color values C. Sampling loss is derived from this observation, where the objective is to minimize the discrepancy between $\Gamma(\Pi(X); I)$ and C. Given a candidate camera pose R, t, this could be formulated as follows,

$$L_{\text{sampling}}(R, t) = \|\Gamma(\Pi(R(X - t)); I) - C\|_2. \quad (2)$$

Note that $R(X - t)$ is the transformed point cloud under R, t. Gradients with respect to R, t could be obtained by

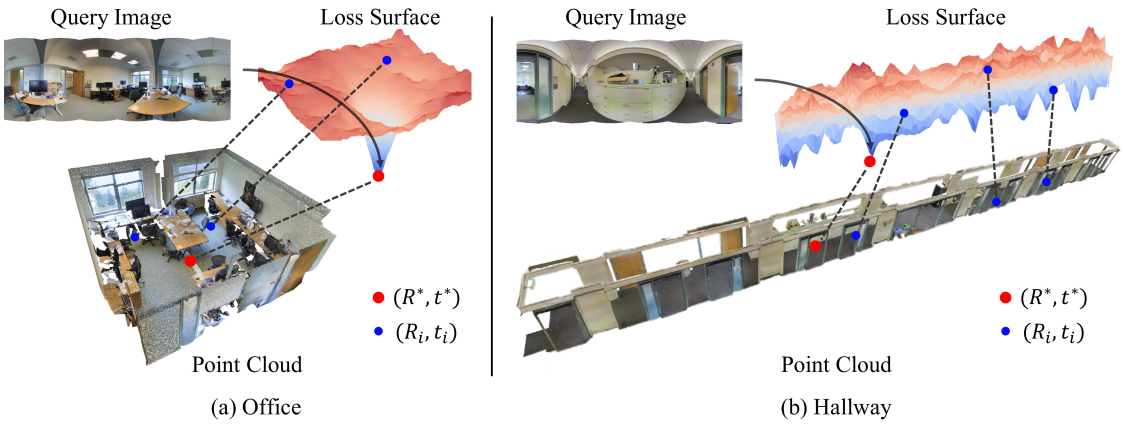


Figure 3: Visualization of loss surfaces obtained from scenes in the Stanford2D-3D-S dataset [3]. The loss surfaces show the minimum loss values of the given (x, y) position in the 3D space. The red dots indicate the ground truth camera positions, and the blue dots link the values on the loss surface and the corresponding camera positions within the input point cloud space. Loss surfaces of small scenes are typically smooth with clear global minimum (left), but those of large scenes contain numerous local minima (right).

differentiating through the sampling function $\Gamma(\cdot; I)$ using the technique from Jaderberg *et al*. [20]. Once the gradients are known, any off-the-shelf gradient based optimization algorithm such as stochastic gradient descent [24] or Adam [25] could be applied to minimize Equation 2, as shown in Figure 1.

Unlike photometric loss which stems from an *image-centric* viewpoint, sampling loss aims at assigning an adequate sampled color value to each point in the point cloud, thus providing a *point cloud-centric* viewpoint. Specifically, photometric loss also compares the colors of the point cloud with the query image, but in the image space, namely,

$$L_{\text{photometric}}(R, t) = \|\Psi(\{R(X - t), C\}) - I\|_2, \quad (3)$$

where $\Psi(\cdot) : \{\mathbb{R}^{N\times 3}, \mathbb{R}^{N\times 3}\} \rightarrow \mathbb{R}^{H\times W\times 3}$ is a rendering function that receives the point cloud to produce a synthesized image. The rendering function is necessary to apply photometric loss in our setup, as only a single image is given, unlike existing applications [11, 30] where multiple images are provided. As photometric loss is evaluated in the image space, it suffers from the visual distortion of omnidirectional cameras. To illustrate, in Figure 2, one can observe that points near the pole (ceilings, floors) are 'stretched', while they correspond to small areas in reality. Since photometric loss makes direct image comparisons, it is severely affected by such artifacts and requires additional processing to account for the distortion [44, 15].

Sampling loss has numerous advantages over photometric loss. First, as seen from Equation 2, it fairly incorporates all points in the point cloud agnostic of whether it is closer to the pole, thus making it more suitable for 6-DoF pose estimation of omnidirectional cameras. Second, sampling loss is cheap to compute, while still allowing for easy gradient computation [20]. Each sampling operation consists of simple image indexing, and we ignore the non-differentiable, costly components of projection, such as occlusion handling. The core part of PICCOLO consists of simple gradient descent on the sampling loss, which is very fast: 3×10^8 points can be processed per second. Nonetheless, sampling loss can effectively handle the holistic view of the 360° image and is robust to visual distortion or minor scene changes.

Initialization Algorithm While sampling loss has various amenable properties, it is non-convex as visualized in Figure 3. Depending on the initial position, optimization using gradient descent can stop at a local minimum, which can be a serious issue for large spaces. To this end, we introduce a lightweight initialization algorithm, which outputs feasible starting points that are likely to yield global convergence.

We uniformly sample the space of possible camera positions, and filter them through a two-step selection process as presented in Algorithm 1. During the first step, we compute sampling loss values across $N_t \times N_r$ candidate camera poses and obtain the top K_1 smallest starting points (line 2). Specifically, N_t translations are chosen from the uniform grid on the point cloud bounding box, for which N_r rotations, uniformly sampled from $SO(3)$, are selected. Since sampling loss is very efficient, we can quickly compute the loss for all of the starting points.

Among the K_1 starting points, the second filtering process further selects K_2 ($K_2 \leqslant K_1$) of them using color histogram intersections (line 3). Top K_2 candidate poses with the highest color distribution overlap with the query image are chosen. Finally, the resulting K_2 starting points are in-

Method	Information	Learning	t-error (m)	R-error (°)
PoseNet [23]	RGB	◯	2.41	28
SphereNet [7]	RGB	◯	2.29	26.7
Zhang *et al.* [44]	RGB	◯	1.64	9.15
PICCOLO	RGB	×	**0.03**	**0.66**
GOSMA [6]	Semantic	×	1.27	51.44
PICCOLO	Semantic	×	**0.01**	**0.28**

Table 1: Quantitative results of omnidirectional localization evaluated on all areas of the Stanford2D-3D-S dataset [3].

dividually optimized for a fixed number of iterations with respect to the sampling loss in $SE(3)$ (line 7). At termination, the optimized camera pose with the smallest sampling loss value is chosen (line 9).

Algorithm 1 Overview of PICCOLO

Inputs: Point cloud P = $\{X, C\}$, query image I
Output: Camera pose $\hat{R}, \hat{t}$.

1: $T \leftarrow [(R_i, t_i) | i \in [1 \,..\, N_t N_r]]$ ▷ Starting points
2: $T \leftarrow \texttt{getTopK}(\texttt{lossValue}(T, P, I), K_1)$
3: $T \leftarrow \texttt{getTopK}(\texttt{histIntersect}(T, P, I), K_2)$
4: $V \leftarrow [\,]$
5: **for all** $(R_i, t_i) \in T$ **do**
6: **for** iter $\in [1 \,..\, N_{\text{iter}}]$ **do**
7: $(R_i, t_i) \leftarrow (R_i, t_i) - \alpha \nabla L_{\text{sampling}}(R_i, t_i)$
8: $V.\texttt{append}(L_{\text{sampling}}(R_i, t_i))$
9: $(\hat{R}, \hat{t}) \leftarrow \arg\min_{R,t} V$

4. Experimental Results

4.1. Performance Analysis

Implementation Details PICCOLO is mainly implemented using PyTorch [31], and is accelerated with a single RTX 2080 GPU. Once the starting point is selected as described in Section 3, we find the camera pose using Adam [25] with step size $\alpha = 0.1$ in all experiments. PICCOLO is straightforward to implement, with the core part of the algorithm taking less than 10 lines of PyTorch code. For results in which accuracy is reported, a prediction is considered correct if the translation error is below 0.1 m and the rotation error is below 5.0°. All translation and rotation errors reported are median values, following the convention of [6, 5]. The full hyperparameter setup and additional qualitative results are available in the supplementary material.

Stanford2D-3D-S We assess the localization performance of PICCOLO against existing methods using the Stanford2D-3D-S dataset [3], as shown in Table 1 and 2. It is an indoor dataset composed of 1413 panoramic images

	PICCOLO	GOSMA	GOSMA$^{-\Lambda}$	GOPAC
t-error (m)	$\mathbf{0.01}^{0.07}_{0.00}$	$0.08^{0.15}_{0.05}$	$0.14^{0.23}_{0.09}$	$0.15^{0.27}_{0.10}$
R-error (°)	$\mathbf{0.21}^{0.56}_{0.11}$	$1.13^{2.18}_{0.91}$	$2.38^{4.61}_{1.25}$	$3.78^{5.10}_{2.47}$

Table 2: Localization results of PICCOLO, GOSMA, GOSMA without class labels (GOSMA$^{-\Lambda}$), and GOPAC for a subset of Area 3 from Stanford2D-3D-S [3]. $Q_2{}^{Q_3}_{Q_1}$ are quartile values of each metric. Results other than PICCOLO are excerpted from [6].

subdivided into six different areas, and many scenes exhibit repetitive structure and lack visual features, as in Figure 5. All areas are used for comparison except for GOPAC [5], where we use a subset from Area 3 consisting of small rooms, as the algorithm's long runtime hinders large-scale evaluation.

PICCOLO outperforms all existing baselines by a large margin, showing an order-of-magnitude performance gain from its competitors. GOSMA [6] and GOPAC [5] are optimization-based methods that do not utilize color measurements. Instead, they require semantic labels for decent performance. For fair comparisons with these algorithms, we make PICCOLO observe color-coded semantic labels as input, as shown in Figure 5, and report the numbers in Table 1 (PICCOLO Semantic) and 2. Semantic labels lack visual features, thus finding camera pose in this setup is closer to solving a blind-PnP problem [10]. However, PICCOLO operates seamlessly and outperforms GOSMA and GOPAC without the aid of rich visual information such as RGB inputs, consistently succeeding around 1 cm error. Although GOSMA and GOPAC are powerful algorithms that guarantee global optimality, they often fail in large scenes such as hallways, where the qualitative results are shown in the supplementary material.

PICCOLO also shows superior performance against deep learning methods [44, 7, 23]. Nevertheless, it should be noted that there is a subtle distinction in the search spaces of these methods. The translation domain for deep learning-based methods is the entire Stanford2D-3D-S dataset, while it is confined to a particular area for PICCOLO, similar to GOSMA [6]. However, deep learning-based methods are given very strong prior information to cope with the large search space; they are trained on synthetic pose-annotated images, which are generated within 30 cm proximity of the test images. This means the training images are very close to the ground truth. Nevertheless, it must be acknowledged that deep learning methods are capable of regressing the pose at wider scales, about 5 times the maximum search scale (1000 m^2) attainable with PICCOLO (Table 3).

	Area (m^2)	t-error (m)	R-error (°)	Acc.
Coast	458.0	0.79	2.18	0.40
Forest	361.2	0.02	0.92	0.67
ParkingIn	92.9	2.77	96.50	0.13
ParkingOut	1381.2	1.74	9.77	0.07
Residential	412.8	0.83	2.53	0.46
Urban	1156.4	0.03	0.85	0.85
All	646.3	0.80	2.10	0.45

Table 3: Localization error and accuracy of PICCOLO on Multi-Modal Panoramic 3D Outdoor (MPO) dataset [22].

Scenario	Scene Change	t-error (m)	R-error (°)	Acc.
Handheld	×	0.02	0.25	0.71
Robot	×	0.02	0.18	0.77
Handheld	◯	0.77	15.39	0.43
Robot	◯	0.05	0.59	0.55

Table 4: Localization error and accuracy of PICCOLO on the OmniScenes dataset.

MPO Multi-Modal Panoramic 3D Outdoor (MPO) [22] dataset is an outdoor dataset which spans a large area ($1000m^2$) with many scenes containing repetition or lacking visual features. As shown in Table 3, PICCOLO performs competently with the same hyperparameter setting as Stanford2D-3D-S [3], despite the large area of the dataset. This validates our claim that PICCOLO could readily function as an off-the-shelf omnidirectional localization algorithm for both indoor/outdoor environments.

Practicality Assessment with OmniScenes Omnidirectional localization is expected to provide stable visual localization under scene changes or dynamic motion, and therefore promises practical applications in VR/AR or robotics. We introduce a new dataset called OmniScenes collected to evaluate the performance on scenes with the aforementioned challenges. We collect dense 3D scans of eight areas including wedding halls and hotel rooms using the Matterport3D Scanner [1]. Corresponding 360° panoramic images are acquired with the Ricoh Theta 360° camera [2] under two scenarios, handheld and mobile robot mounted. Handheld scenarios are typically more challenging as unconstrained motion could take place and the capturer partially occludes scene details. The images are taken at different times of day and include significant changes in furniture configurations and motion blurs. Further details about the dataset are deferred to the supplementary material.

The evaluation results on the OmniScenes dataset are shown in Table 4. Unlike previous experiments, we assume that the gravity direction is known, as this is often available in practice. PICCOLO exhibits competent error rates when there are no scene changes, agnostic of whether the input 360° panorama is recorded in a handheld or robot-mounted manner. As shown in Figure 5, PICCOLO can estimate camera pose even under severe handheld motion, thanks to the full incorporation of points from sampling loss.

Even though there is no functionality in PICCOLO that accounts for scene changes, there is a considerable amount of success cases given the accuracy in Table 4 and qualitative results shown in Figure 5. As long as the global context provides enough amount of evidence from color samples, omnidirectional localization can succeed. Nonetheless, there is a clear performance gap, and enhancing the robustness of PICCOLO against various scene changes is left as future work.

Loss Function	Information	t-error (m)	R-error (°)
Sampling	Original	**0.03**	**0.66**
Photometric	Original	1.41	42.29
Sampling	Gravity Direction	**0.01**	**0.34**
Photometric	Gravity Direction	0.93	33.41
Sampling	Flipped	**0.03**	**0.69**
Photometric	Flipped	1.42	42.79
Sampling	Rand. Rot.	**0.23**	**2.21**
Photometric	Rand. Rot.	1.48	43.33

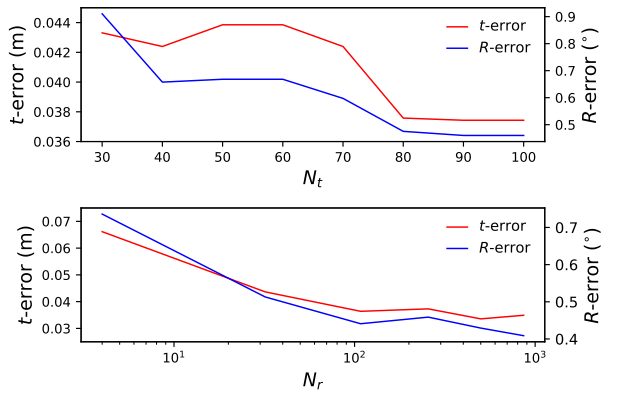
Table 5: Ablation study on sampling loss and gravity direction. 'Flipped' denotes flipped query images and 'Rand. Rot.' denotes randomly rotated point cloud inputs.

4.2. Ablation Study

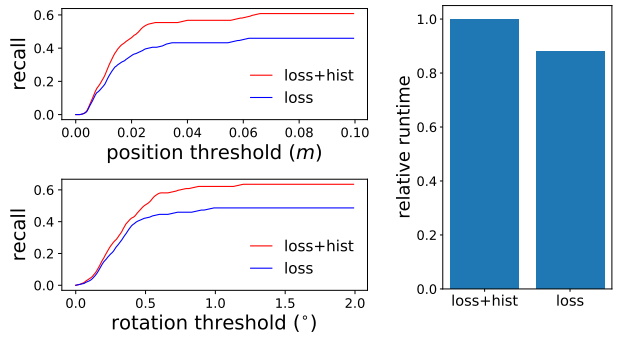
In this section we ablate various components of PICCOLO. Experiments are conducted using all areas of the Stanford2D-3D-S dataset [3], unless specified otherwise.

Sampling Loss We compare PICCOLO with a variant that uses photometric loss from Equation 3 in place of the sampling loss, to ablate the effect of sampling loss in our algorithm. The rendering function is implemented as a simple projection of the 3D point cloud, similar to projections shown in Figure 5. We use the warping function to obtain gradients with respect to R, t, as in previous works [11, 30, 13]. All other hyperparameter setups and the initialization algorithm are the same as PICCOLO.

The design choice of using sampling loss shows a large performance gain over photometric loss, as shown in Table 5. As sampling loss fairly incorporates all points in point cloud, it is free from visual distortion and thus more suitable than photometric loss for 6-DoF omnidirectional localization.



(a) Effects of N_t, N_r on localization error.



(b) Comparison of two initialization schemes.

Figure 4: Ablation study on the initialization pipeline.

Gravity Direction If the gravity direction is known, the number of initial positions is significantly reduced and PICCOLO can perform highly accurate localization as shown in Table 5. Knowing the gravity direction is a reasonable assumption as many panoramic images or 3D scan datasets [3, 22] contain the information. In practice, one can easily infer the gravity direction of omnidirectional cameras using integrated gyroscopes, and that of 3D maps with RANSAC [14]-based plane fitting. Nonetheless, PICCOLO stably performs without knowing the gravity direction as shown in Table 1 and 3.

In case PICCOLO might be biased towards the gravity-aligned conventional data, we evaluate PICCOLO in flipped input images and arbitrarily rotated point clouds. Under the same hyperparameter setup as Section 4.1, PICCOLO demonstrates consistent performance, as shown in Table 5. Such results imply that PICCOLO is amenable for novel scenes, and could be directly applied to a wide variety of non-standard inputs without training.

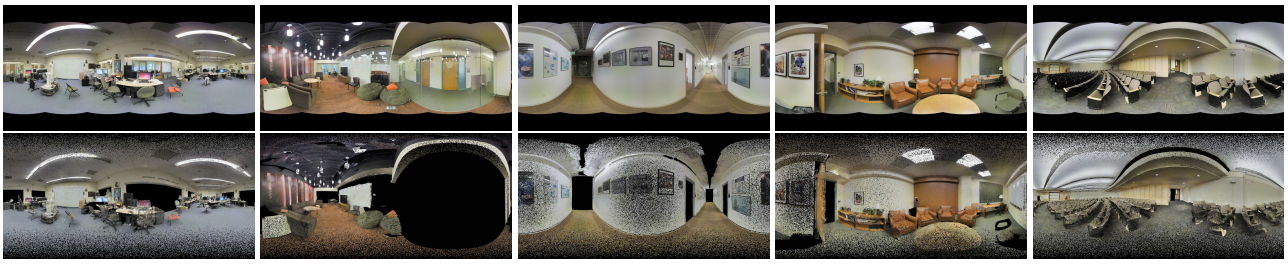
Initialization Pipeline We finally ablate various components of the initialization pipeline presented in Section 3. The first main parameters to be examined are the number of initial points N_t, N_r sampled from the range of possible transformations. We evaluate the effect of different values of N_t, N_r on auditoriums from Area 2 of the Stanford2D-3D-S dataset [3]. As shown in Figure 4a, larger N_t, N_r tend to improve the error values, but result in computational overhead. An adequate set of N_t, N_r should be chosen considering the trade-off. We use $N_t = 50, N_r = 32$ for all our experiments. This means we have about 1600 initial points to test, but the initialization finishes within a few seconds, thanks to the efficiency of sampling loss. For runtime-critical applications, one may cache the projected point cloud coordinates at each candidate starting pose once for each scene and use it afterward. This would significantly reduce the time spent on initialization.

We further examine the efficacy of our two-stage initialization scheme. Recall the two-stage initialization in Section 3 first selects K_1 candidate locations using *loss values* followed by filtration to K_2 candidates using *color histograms*. We compare the performance of PICCOLO selecting K_2 initial poses from $N_t \times N_r$ candidates using (i) loss only, and (ii) the two-stage method presented in Section 3. All rooms in Area 3 of the Stanford2D-3D-S dataset are selected for evaluation with $N_t = 50, N_r = 32, K_1 = 50, K_2 = 6$. We display the results in Figure 4b. Our two-stage initialization enables a significant performance boost with only a small increase in runtime.

5. Conclusion

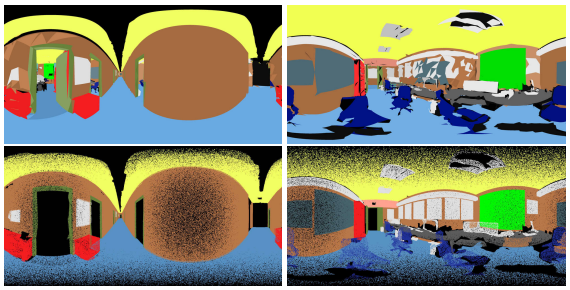
In this paper, we present PICCOLO, a simple, efficient algorithm for omnidirectional localization. We introduce sampling loss, which enforces each point in the 3D point cloud to correctly *sample* from the query image. Sampling loss is clearly beneficial for omnidirectional localization, as it is a *point cloud-centric* formulation, free from spherical distortion, and computationally efficient. In experiments conducted on various indoor and outdoor environments, PICCOLO outperforms existing algorithms by a significant margin. Furthermore, when evaluated on our newly proposed dataset, OmniScenes, PICCOLO shows competent performance even amidst diverse camera motion and scene changes. We expect PICCOLO to be applied in a wide variety of virtual reality / robotics applications where omnidirectional cameras are present.

Acknowledgments This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1C1C1008195), the National Convergence Research of Scientific Challenges through the National Research Foundation of Korea (NRF) funded by Ministry of Science and ICT (NRF2020M3F7A1094300), and the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2021.

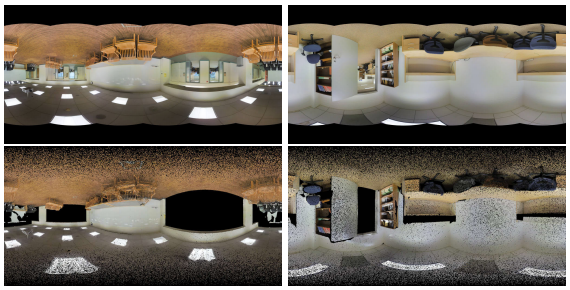


(a) Stanford2D-3D-S Indoor Localization

(b) MPO Outdoor Localization



(c) Stanford2D-3D-S Semantic Input



(d) Stanford2D-3D-S Flipped Input

(e) OmniScenes Handheld

(f) OmniScenes Robot-Mounted

(g) OmniScenes Handheld with Scene Change

(h) OmniScenes Robot-Mounted with Scene Change

Figure 5: Additional qualitative results of PICCOLO with various input settings. 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. 1, 6

[2] Ricoh theta, experience the world in 360. https://theta360.com/en/. Accessed: 2021-03-16. 6

[3] 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. 4, 5, 6, 7

[4] Eric Brachmann, Alexander Krull, Sebastian Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother. Dsac — differentiable ransac for camera localization. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2492–2500, 2017. 2

[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, 5

[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. 2, 3, 5

[7] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 1, 3, 5

[8] Gabriela Csurka and Martin Humenberger. From handcrafted to deep local invariant features. *CoRR*, abs/1807.10254, 2018. 3

[9] Angela Dai, Matthias Nießner, Michael Zollöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration. *ACM Transactions on Graphics 2017 (TOG)*, 2017. 1

[10] Philip David, Daniel DeMenthon, Ramani Duraiswami, and Hanan Samet. Softposit: Simultaneous pose and correspondence determination. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Computer Vision — ECCV 2002*, pages 698–714, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. 5

[11] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 834–849, Cham, 2014. Springer International Publishing. 2, 4, 6

[12] Hannes Fassold. Adapting computer vision algorithms for omnidirectional video, 2019. 1, 3

[13] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks, 2015. 2, 6

[14] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 7

[15] P. Frossard and R. Khasanova. Graph-based classification of omnidirectional images. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 860–869, 2017. 1, 2, 3, 4

[16] J. A. Hesch and S. I. Roumeliotis. A direct least-squares (dls) method for pnp. In *2011 International Conference on Computer Vision*, pages 383–390, 2011. 3

[17] Martin Humenberger, Yohann Cabon, Nicolas Guérin, Julien Morat, Jérôme Revaud, Philippe Rerole, Noé Pion, César Roberto de Souza, Vincent Leroy, and Gabriela Csurka. Robust image retrieval-based visual localization using kapture. *CoRR*, abs/2007.13867, 2020. 2

[18] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks, 2016. 2

[19] A. Irschara, C. Zach, J. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2599–2606, 2009. 3

[20] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2017–2025. Curran Associates, Inc., 2015. 2, 4

[21] M. Jayasuriya, R. Ranasinghe, and G. Dissanayake. Active perception for outdoor localisation with an omnidirectional camera. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4567–4574, 2020. 1, 3

[22] 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. 6, 7

[23] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. 2015. 2, 3, 5

[24] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *Ann. Math. Statist.*, 23(3):462–466, 09 1952. 4

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 4, 5

[26] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *Int. J. Comput. Vision*, 81(2):155–166, Feb. 2009. 3

[27] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *CVPR*, 2020. 2

[28] Yunpeng Li, Noah Snavely, and Daniel P. Huttenlocher. Location recognition using prioritized feature matching. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 791–804, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 3

[29] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 3

[30] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *2011 International Conference on Computer Vision*, pages 2320–2327, 2011. 2, 4, 6

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[32] N. Pion, M. Humenberger, G. Csurka, Y. Cabon, and T. Sattler. Benchmarking image retrieval for visual localization. In *2020 International Conference on 3D Vision (3DV)*, pages 483–494, Los Alamitos, CA, USA, nov 2020. IEEE Computer Society. 2

[33] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. *CoRR*, abs/1812.03506, 2018. 2

[34] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. *CoRR*, abs/1911.11763, 2019. 2

[35] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 752–765, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 2, 3

[36] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(9):1744–1756, 2017. 2, 3

[37] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla. Benchmarking 6dof outdoor visual localization in changing conditions. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8601–8610, 2018. 1

[38] Johannes L. Schonberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[39] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013. 1, 2

[40] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. In *CVPR 2018 - IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, United States, June 2018. 1, 2

[41] 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

[42] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1, 2

[43] Johanna Wald, Torsten Sattler, Stuart Golodetz, Tommasso Cavallari, and Frederico Tombari. Beyond controlled environments: 3d camera re-localization in changing indoor scenes. In *ECCV*, 2020. 1

[44] Chao Zhang, Ignas Budvytis, Stephan Liwicki, and Roberto Cipolla. Rotation equivariant orientation estimation for omnidirectional localization. In *ACCV*, 2020. 2, 3, 4, 5

[45] Qiang Zhao, Chen Zhu, Feng Dai, Yike Ma, Guoqing Jin, and Yongdong Zhang. Distortion-aware cnns for spherical images. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 1198–1204. International Joint Conferences on Artificial Intelligence Organization, 7 2018. 1, 3

[46] Yao Zhou, Guowei Wan, Shenhua Hou, L. Yu, Gang Wang, Xiaofei Rui, and Shiyu Song. Da4ad: End-to-end deep attention-based visual localization for autonomous driving. In *ECCV*, Aug 2020. 1

[47] Zichao Zhang, H. Rebecq, C. Forster, and D. Scaramuzza. Benefit of large field-of-view cameras for visual odometry. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 801–808, 2016. 1