Calibrating Panoramic Depth Estimation for Practical Localization and Mapping

Junho Kim[1], Eun Sun Lee[1], and Young Min Kim[1, 2]

[1] Dept. of Electrical and Computer Engineering, Seoul National University

[2] Interdisciplinary Program in Artificial Intelligence and INMC, Seoul National University

{82magnolia, eunsunlee, youngmin.kim}@snu.ac.kr

Abstract

The absolute depth values of surrounding environments provide crucial cues for various assistive technologies, such as localization, navigation, and 3D structure estimation. We propose that accurate depth estimated from panoramic images can serve as a powerful and light-weight input for a wide range of downstream tasks requiring 3D information. While panoramic images can easily capture the surrounding context from commodity devices, the estimated depth shares the limitations of conventional image-based depth estimation; the performance deteriorates under large domain shifts and the absolute values are still ambiguous to infer from 2D observations. By taking advantage of the holistic view, we mitigate such effects in a self-supervised way and fine-tune the network with geometric consistency during the test phase. Specifically, we construct a 3D point cloud from the current depth prediction and project the point cloud at various viewpoints or apply stretches on the current input image to generate synthetic panoramas. Then we minimize the discrepancy of the 3D structure estimated from synthetic images without collecting additional data. We empirically evaluate our method in robot navigation and map-free localization where our method shows large performance enhancements. Our calibration method can therefore widen the applicability under various external conditions, serving as a key component for practical panorama-based machine vision systems.

1. Introduction

Acquiring depth maps of the surrounding environment is a crucial step for AR/VR and robotics applications, as the depth maps serve as building blocks for mapping and localization. While dense LiDAR or RGB-D scanning [1, 2, 3, 4, 5] has been widely used for depth acquisition, the methods are often computationally expensive or require costly hardware. Panoramic depth estima-

(a) Navigated Area in Perspective & Panoramic Perception

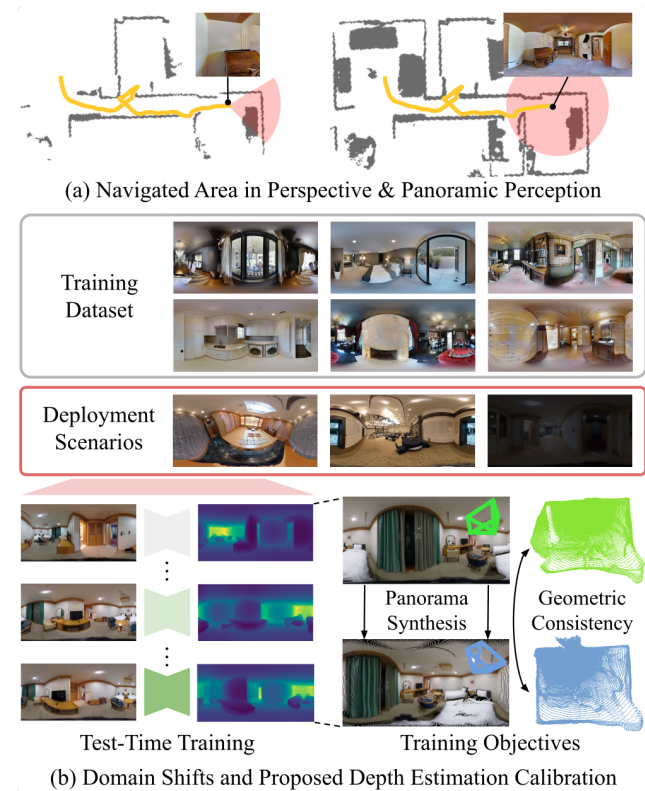(b) Domain Shifts and Proposed Depth Estimation Calibration

Figure 1: Motivation and overview of our approach. Panoramic perception enables efficient navigation due to the large field of view (top). Nevertheless, the performance drops due to the gaps between the training dataset with upright cameras in medium-sized rooms and the deployment scenarios with limited data and various domain shifts. The proposed solution suggests test-time training using geometric consistency to mitigate the gap (bottom).

tion [6, 7, 8, 9, 10, 11, 12, 13], on the other hand, enables quick and cost-effective depth computation. It outputs a dense depth map from a single neural network inference given only 360° camera input, which is becoming more widely accessible [14, 15]. Further, the large field of view

of panoramic depth maps can model the comprehensive 3D context from a single image capture. The holistic view provides ample visual cues for robust localization, and allows efficient 3D mapping. An illustrative example is shown in Figure 1a, where a robot navigation agent equipped with panorama view observes larger areas and builds more comprehensive grid map than the agent with perspective view when deployed for the same trajectory.

While existing panoramic depth estimation methods can estimate highly accurate depth maps in trained environments [11, 8, 7, 6], their performances often deteriorate when deployed in unseen environments with large domain gaps. For example, as shown in Figure 1b, depth estimation networks trained on upright panorama images in medium-sized rooms perform poorly in images containing large camera rotation or captured in large rooms. Such scenarios are highly common in AR/VR or robotics applications, yet it is infeasible to collect large amounts of densely annotated ground-truth data for panorama images or perform data augmentations to realistically and thoroughly cover all the possible adversaries. Further, while numerous unsupervised domain adaptation methods have been proposed for depth estimation [16, 17, 18, 19], most of them mainly consider sim-to-real gap minimization and require the labelled training dataset for adaptation which is infeasible for memory-limited applications.

In this paper, we propose a quick and effective calibration method for panoramic depth estimation in challenging environments with large domain shifts. Given a pre-trained depth estimation network, our method applies test-time adaptation [20, 21, 22] on the network solely using objective functions derived from test data. Conceptually, we are treating depth estimation networks as *sensors* that output depth maps from images, which then makes the process similar to 'calibration' in depth or LiDAR sensing literature for accurate measurements. Our resulting scheme is flexibly applicable in either online or offline manner adaptation. As shown in Figure 1c, the light-weight training *calibrates* the network towards making more accurate predictions in the new environment.

Our calibration scheme consists of two key components that effectively utilize the holistic spatial context uniquely provided by panoramas. First of all, our method operates using training objectives that impose geometric consistencies from novel view synthesis and panorama stretching. To elaborate, as shown in Figure 2, we leverage the full-surround 3D structure available from panoramic depth estimation and generate synthetic panoramas. The training objectives then minimize the geometric discrepancy between depth estimations from the synthesized panoramas and the original view. Second, we propose light-weight data augmentation to cope with offline scenarios where only a limited amount of test-time training data is available. Specif-

ically, we augment the test data by applying arbitrary pose shifts or synthetic stretches, similar to the techniques used for the training objectives.

Since our calibration method aims at adapting the network during the test phase using geometric consistencies, it is compute and memory efficient while being able to handle a wide variety of domain shifts. Our method does not require the computational demands of additional network pre-training [21, 23], or memory to store the original training dataset during adaptation [16, 18, 17, 24]. Nevertheless, our method shows large amounts of performance enhancements when tested in challenging domain shifts such as low lighting or room-scale change. Further, due to the light-weight formulation, our method could easily be applied to numerous downstream tasks in localization and mapping. We experimentally verify that our calibration scheme effectively improves performance in two exemplary tasks, namely map-free localization and robot navigation. To summarize, our key contributions are as follows: (i) a novel test-time adaptation method for calibrating panoramic depth estimation, (ii) a data augmentation technique to handle low-resource adaptation scenarios, and (iii) an effective application of our calibration method on downstream mapping and localization tasks.

2. Related Work

Monocular Depth Estimation Following the pioneering work of Eigen et al. [25], many existing works focus on developing neural network models that output depth maps from image input [26, 27, 28, 29, 30, 31]. Recent approaches such as MiDAS [26] or DPT [27] can make highly accurate depth predictions from images due to extensive training on large depth-annotated datasets [32, 33, 34]. As a result, there have been numerous applications in localization and mapping that leverage monocular depth estimation. For example, map-free visual localization [35] localizes the camera position using maps built from monocular depth estimation, which is highly efficient compared to building a 3D map by Structure-from-Motion. Another example is in robot navigation methods that directly estimate occupancy grid maps from input images [36, 37, 38], which could be implicitly regarded as monocular depth estimation.

Compared to perspective images, monocular depth estimation using panorama images has been relatively under-studied with the limited amount of available data. While recent works [11, 7, 9, 6, 12, 13] have demonstrated accurate depth estimation in trained environments, their performance are known to deteriorate when tested in new datasets with varying lighting or depth distributions [10]. Such performance discrepancies are more noticeable for panoramic images since there are fewer depth-annotated images available compared to perspective images. One possible remedy is to re-project the panoramic image to multiple perspective

images to create individual depth estimations and stitch the results together, as proposed by Rey-Area et al. [39] and Peng et al. [40]. Although this may yield more robust depth estimation utilizing abundant previous frameworks for perspective images, the process involves fusing depth maps resulting from multiple neural network inferences, which is computationally expensive. Our method takes a different direction of quickly calibrating the network to the new environment for robust panoramic depth estimation. We demonstrate the effectiveness of our method on the aforementioned applications of depth estimation, namely map-free localization and robot navigation, to verify its practicality on downstream tasks.

Domain Adaptation for Depth Estimation A vast majority of domain adaptation methods target classification tasks [20, 21, 22, 41, 42, 43, 44], and aim to minimize loss functions defined over the predicted class probabilities. Existing methods could be categorized into those that only use the test data or those that require the original training dataset for adaptation. For the former, pseudo-labelling methods [43], masking-based methods [45, 44], batch normalization updating methods [46], and fully test-time training methods [20, 47] impose self-supervised learning objectives to adapt to the test domain data. On the other hand, methods belonging to the latter, namely unsupervised domain adaptation methods [42, 48, 49] and test time training methods with auxiliary task networks [21, 22, 23], utilize the original training dataset to impose consistencies between the source and target domain predictions.

For depth estimation, most adaptation methods [16, 50, 24, 19, 18] focus on the sim-to-real domain gap and apply techniques from pseudo-labelling or unsupervised domain adaptation. Methods such as DESC [17] and 3D-PL [50] adapt using pseudo labels generated from additional semantic priors or style transferred predictions [51, 52, 53, 54]. In contrast, unsupervised domain adaptation methods [24, 16, 17] additionally utilize the depth-annotated training data and apply style transfer networks to learn a common feature representation between the source and target domain. Compared to existing domain estimation methods for depth estimation, our calibration method can effectively handle a wider range of domain shifts and can perform light-weight online adaptation as no additional training data is required. We extensively evaluate our method against existing domain adaptation techniques in Section 5, where our method outperforms the tested baselines in various depth estimation scenarios.

3. Method

Given a panoramic depth estimation network $F_\Theta(\cdot)$ trained on the source domain $\mathcal{S}$, the object of our calibration scheme is to adapt the network to a new, unseen target

(a) Stretch Loss L_S

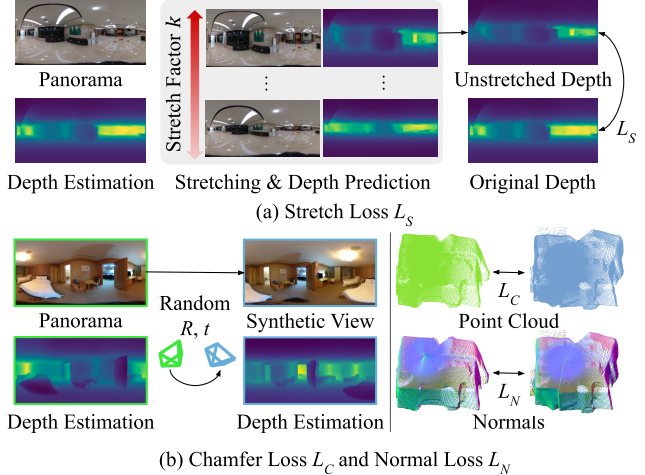(b) Chamfer Loss L_C and Normal Loss L_N

Figure 2: Description of the proposed test-time training objectives.

domain $\mathcal{T}$ during the test phase. Our method could perform adaptation both in an online and offline manner: in the online case, the network is simultaneously optimized and evaluated whereas in the offline case the network is first optimized using samples from the target domain and then evaluated with another set of target domain samples. As shown in Figure 2, our method leverages training objectives that impose geometric consistencies between the synthesized views generated from the full-surround depth predictions (Section 3.1). To further cope with offline adaptation scenarios where only a small number of images are available for training, we propose to apply data augmentation based on panorama synthesis (Section 3.2).

3.1. Test-Time Training Objectives

Given a panorama image $I \in \mathbb{R}^{H \times W \times 3}$, the depth estimation network outputs a depth map $\hat{D} = F_\Theta(I) \in \mathbb{R}^{H \times W \times 1}$. The test-time adaptation enforces consistencies between depth estimations of additional input images synthesized from the current predictions, and eventually achieves stable prediction under various environment setups. The test-time training objective is given as follows,

$$\mathcal{L} = \mathcal{L}_S + \mathcal{L}_C + \mathcal{L}_N, \tag{1}$$

where $\mathcal{L}_S$ is the stretch loss, $\mathcal{L}_C$ is the Chamfer loss, and $\mathcal{L}_N$ is the normal loss.

Stretch Loss Stretch loss aims to tackle the depth distribution shifts that commonly occur in panoramic depth estimation by imposing consistencies between depth predictions made at different panorama stretches. Panoramic depth estimation models make large errors when confronted with images captured in scenes with drastic depth distribution changes [10]. The key intuition for stretch loss is to hallucinate the depth estimation network as if it is making

predictions in a room with similar depth distributions as the trained source domain, through panoramic stretching shown in Figure 2a.

The panorama stretching operation [55, 56] warps the input panorama I to a panorama captured from the same 3D scene but stretched along the x, y axes by a factor of k. For a panorama image I this could be expressed as follows,

$$\mathcal{S}_{\text{img}}^k(I)[u, v] = I[u, \frac{H}{\pi} \arctan(\frac{1}{k} \tan(\frac{\pi v}{H}))], \quad (2)$$

where $I[u, v]$ is the color value at coordinate (u, v) and $\mathcal{S}_{\text{img}}^k(\cdot)$ is the k-times stretching function for images. A similar operation could be defined for depth maps, namely

$$\mathcal{S}_{\text{dpt}}^k(D)[u, v] = \kappa(v)D[i, \frac{H}{\pi} \arctan(\frac{1}{k} \tan(\frac{\pi v}{H}))], \quad (3)$$

where $\kappa(v) = \sqrt{k^2 \sin^2(\pi v/H) + \cos^2(\pi v/H)}$ is the correction term to account for the depth value changes due to stretching.

Stretch loss enforces depth predictions made at large scenes to follow predictions made at *contracted* scenes (using $k < 1$) and for those at small scenes to follow predictions made at *enlarged* scenes (using $k > 1$). The distinction between large and small scenes is made by thresholding on the average depth value using thresholds δ_1, δ_2. Formally, this could be expressed as follows,

$$\mathcal{L}_{\text{S}} = \begin{cases} \sum_{k \in \mathcal{K}_s} \|\hat{D} - \mathcal{S}_{\text{dpt}}^{1/k}(F_\Theta(\mathcal{S}_{\text{img}}^k(I)))\|_2 & \text{if avg}(\hat{D}) < \delta_1 \\ \sum_{k \in \mathcal{K}_l} \|\hat{D} - \mathcal{S}_{\text{dpt}}^{1/k}(F_\Theta(\mathcal{S}_{\text{img}}^k(I)))\|_2 & \text{if avg}(\hat{D}) > \delta_2 \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

where avg$(\hat{D})$ is the pixel-wise average for the depth map $\hat{D} = F_\Theta(I)$, and $\mathcal{K}_l = \{\sigma, \sigma^2\}, \mathcal{K}_s = \{1/\sigma, 1/\sigma^2\}$ are the stretch factors used for contracting and enlarging panoramas. In our implementation we set $\delta_1 = 1, \delta_2 = 2.5, \sigma = 0.8$.

Chamfer and Normal Loss While stretch loss guides depth predictions to have coherent scale, chamfer and normal loss encourages depth predictions to be geometrically consistent at a finer level. The loss functions operate by generating synthetic views at small random pose perturbations from the original viewpoint, and minimizing discrepancies between depth predictions made at synthetic views and the original view.

First, the Chamfer loss minimizes the Chamfer distance between depth predictions made at different poses. Given a panoramic depth map D, let $\mathcal{B}(D) : \mathbb{R}^{H \times W \times 1} \to \mathbb{R}^{HW \times 3}$ denote the back-projection function that maps each pixel's depth values $D[u, v]$ to a point in 3D space $D[u, v] * S[u, v]$ where $S[u, v] \in \mathbb{R}^3$ is a point on the unit sphere corresponding to the panorama image coordinate (u, v). Further, let $\mathcal{W}(I, D; R, t)$ denote the warping function that outputs an

image rendered at an arbitrary pose R, t, as shown in Figure 2b. Then, the Chamfer loss is given as follows,

$$\mathcal{L}_{\text{C}} = \sum_{\mathbf{x} \in \mathcal{B}(\hat{D})} \min_{\mathbf{y} \in \mathcal{B}(D_{\text{warp}})} \|\tilde{R}\mathbf{x} + \tilde{t} - \mathbf{y}\|_2^2, \quad (5)$$

where $D_{\text{warp}} = F_\Theta(\mathcal{W}(I, \hat{D}; \tilde{R}, \tilde{t}))$ is the depth prediction made from the warped image at a randomly chosen pose $\tilde{R}, \tilde{t}$ near the origin. We choose $\tilde{R}$ to be a random rotation around the z-axis and $\tilde{t}$ as a random translation sampled from $[-0.5, 0.5]^3$.

Second, the normal loss imbues an additional layer of geometric consistency by aligning normal vectors of the depth maps. Let $\mathcal{N}(\mathbf{x}) : \mathbb{R}^3 \to \mathbb{R}^3$ denote the normal estimation function that uses ball queries around the point $\mathbf{x}$ to compute the normal vector [57, 58]. The normal loss is then given as follows,

$$\mathcal{L}_{\text{N}} = \sum_{\mathbf{x} \in \mathcal{B}(\hat{D})} (\tilde{R}\mathcal{N}(\mathbf{x}) \cdot (\tilde{R}\mathbf{x} + \tilde{t} - \operatorname*{argmin}_{\mathbf{y} \in \mathcal{B}(D_{\text{warp}})} \|\tilde{R}\mathbf{x} + \tilde{t} - \mathbf{y}\|_2))^2$$
$$(6)$$

Conceptually, the normal loss minimizes the distance between planes spanning from points in the original depth map $\hat{D}$ against the nearest points in the warped image's depth map D_{warp}. Note this is similar in spirit to loss functions used in point-to-plane ICP [59]. We further verify the effectiveness of each loss function in Section 5.

3.2. Data Augmentation

We propose a data augmentation scheme that increases the number of test-time training data for offline adaptation scenarios where only a small number of target domain data is available in a real-world deployment. For example, a robot agent may need to quickly adapt to the new environment after observing a few samples, or AR/VR applications may want to quickly build an accurate 3D map of the environment using a small set of images.

Key to our augmentation scheme is the panorama synthesis from stretching and novel view generation. Given a single panorama image I and its associated depth prediction $\hat{D} = F_\Theta(I)$, the augmentation scheme $\mathcal{A}(I, \hat{D})$ for generating a new panorama is given as follows,

$$\mathcal{A}(I, \hat{D}) = \begin{cases} \mathcal{W}(I, \hat{D}; \tilde{R}, \tilde{t}) & \text{if avg}(\hat{D}) \in [\delta_1, \delta_2] \\ \mathcal{S}_{\text{img}}^k(I) & \text{otherwise,} \end{cases} \quad (7)$$

where $\tilde{R}, \tilde{t}$ are random poses sampled near the origin and k is randomly sampled from $\mathcal{U}(\sigma^2, \sigma)$ if avg$(\hat{D}) > \delta_2$ and $\mathcal{U}(1/\sigma, 1/\sigma^2)$ if avg$(\hat{D}) < \delta_1$. The values for $\delta_1, \delta_2, \sigma$ are identical to those used in Section 3.1. Conceptually, our augmentation scheme generates novel views at random poses if the average depth values are within a range $[\delta_1, \delta_2]$
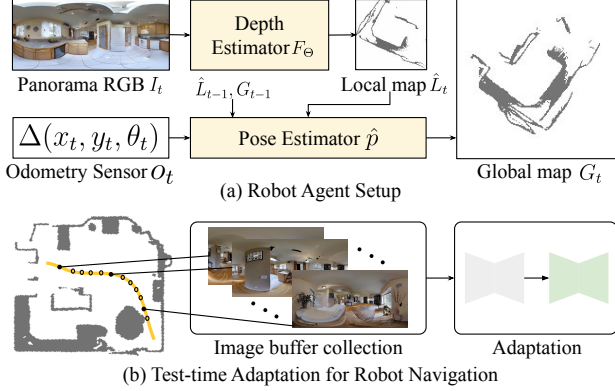
Figure 3: Robot agent with panoramic perception (top) and application of panoramic depth calibration on robot navigation task (bottom).

and applies stretching otherwise, where the scene size determines the stretch factor. Despite the simple formulation, our augmentation scheme enables test-time adaptation only using a small number of image data (at the extreme case, even with a *single* training sample), where we further demonstrate its effectiveness by illustrating its applications in Section 4.

4. Applications

In this section, we show applications of our panoramic depth calibration on two downstream tasks, robot navigation, and map-free localization.

4.1. Robot Navigation

Navigation Agent Setup We assume a navigation agent equipped with a panorama camera and noisy odometry sensor, similar to the setup of recently proposed navigation agents [36, 37, 38]. As shown in Figure 3a, for each time step t the navigation agent first creates a local 2D occupancy grid map $\hat{L}_t$ based on the depth estimation results from the panorama, namely $F_\Theta(I_t)$. Then, the pose estimation network observes the previous and current local map $(\hat{L}_{t-1}, \hat{L}_t)$ along with the noisy odometry sensor reading o_t to produce a pose estimate $\hat{p}_t = C_\Phi(\hat{L}_{t-1}, \hat{L}_t, o_t)$. The pose estimate is further used to stitch the local map L_t against the previous global map G_{t-1} to form an updated global map G_t. Finally, the policy network takes the global grid map and the current image observation as input to output an action policy, namely $a_t = P_\Psi(G_t, I_t)$, where the possible actions are to move forward by 0.25m or turn left or right by $10°$.

Depth Calibration for Robot Navigation We begin each navigation episode by applying our test-time training to calibrate the panoramic depth estimates from a small number

of visual observations collected. As shown in Figure 3b, the agent caches the first N_{fwd} panoramic views seen after it makes a forward action. Then, using the data augmentation from Section 3.2 N_{aug} times for each cached image, the agent performs test-time training with the $N_{\text{fwd}} \times N_{\text{aug}}$ set of images. Once the calibration is completed, the agent uses the updated depth estimation network to create the global map and compute policy for the subsequent steps remaining in the episode. Note that the calibration process for navigation terminates very quickly, with the total number of training steps for each episode being smaller than 300 steps. Nevertheless, the quick calibration results in significant performance improvements for various downstream navigation tasks, which is further verified in Section 5.

4.2. Map-Free Visual Localization

Localization Process Overview First introduced by Arnold et al. [35], map-free visual localization aims at finding the camera pose with respect to a 3D scene where the conventional Structure-from-Motion (SfM) mapping process is omitted, hence the name 'map-free'. Instead, the 3D scene is represented using a 3D point cloud obtained from monocular depth estimation, which in turn greatly reduces the computational burden required for obtaining SfM maps.

We adapt the original map-free localization framework designed for perspective cameras to panoramas, and validate our calibration scheme on the task. As shown in Figure 4, given a single reference image I_{ref} and its associated depth prediction $\hat{D}_{\text{ref}} = F_\Theta(I_{\text{ref}})$, map-free localization initiates with generating a 3D map from the depth map, namely $\mathcal{B}(\hat{D}_{\text{ref}})$. Then we generate synthetic panoramas for the $N_t \times N_r$ poses $\{(R_i, t_i)\}$ and extract global/local feature descriptors, where N_t translations and N_r rotations uniformly sampled from the bounding box of $\mathcal{B}(\hat{D}_{\text{ref}})$.

During localization, global and local descriptors are first similarly extracted for the query image I_q. Then, the top-K poses from the pool of $N_t \times N_r$ poses are chosen whose euclidean distances of the global descriptors are closest to that of the query image $\mathbf{f}_q$. The selected poses are further ranked with local feature matching using SuperGlue [60], where the candidate pose with the largest number of matches is refined for the final prediction. Here, for each local feature match between the query image and synthetic view we retrieve the corresponding 3D point from the point cloud $\mathcal{B}(\hat{D}_{\text{ref}})$ and apply PnP-RANSAC, as shown in Figure 4.

Depth Calibration for Map-Free Localization For each 3D scene, we assume only a small handful of images (between $1 \sim 5$) are available for adaptation, to reflect AV/VR application scenarios where the user wants to quickly localize in a new environment. Depth calibration is then applied to fine-tune the depth estimator, where we increase the number of training samples using data augmentation similar to
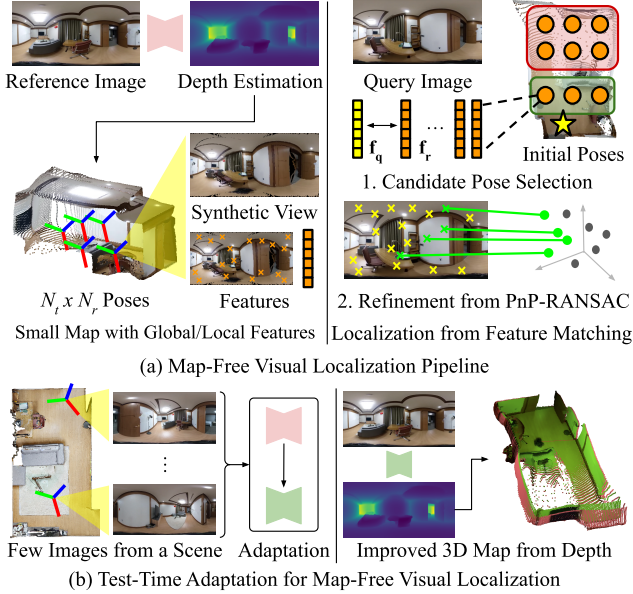
Reference Image Depth Estimation Query Image

f_q f_r Initial Poses

1. Candidate Pose Selection

N_t x N_r Poses Synthetic View Features

Small Map with Global/Local Features

2. Refinement from PnP-RANSAC

Localization from Feature Matching

(a) Map-Free Visual Localization Pipeline

Few Images from a Scene Adaptation Improved 3D Map from Depth

(b) Test-Time Adaptation for Map-Free Visual Localization

Figure 4: Description of map-free localization task (top) and its test-time adaptation pipeline (bottom).

(a) Low Lighting (b) White Balance (c) Gamma

(d) Speckle Noise (e) Gaussian Noise (f) Salt&Pepper Noise

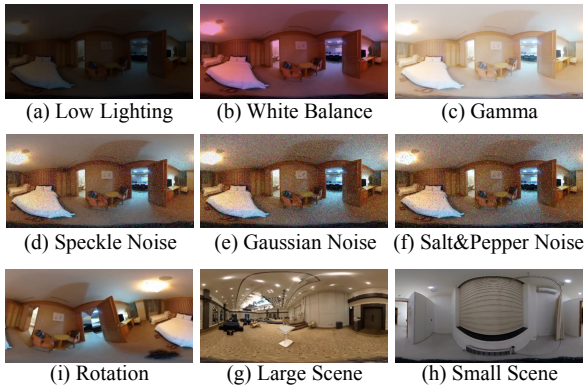(i) Rotation (g) Large Scene (h) Small Scene

Figure 5: Visualization of domain changes.

robot navigation. After calibration, the modified network is applied to create a 3D map from an arbitrary reference image captured from the same environment, which could then be used for localizing new query images.

5. Experimental Results

We first evaluate how our calibration scheme enhances depth prediction (Section 5.1). We then validate its effect on the aforementioned applications, namely robot navigation (Section 5.2) and map-free visual localization (Section 5.3).

Implementation Details We implement our method using PyTorch [61], and use the pre-trained UNet from Albanis et al. [10] as the original network for adaptation. The network is trained using the depth-annotated panorama images from the Matterport3D dataset [62]. For test-time

training, we optimize the loss function from Equation 1 using Adam [63] for 1 epoch, with a learning rate of 10^{-4} and batch size of 4. In all our experiments, we use the RTX 2080Ti GPU for acceleration. Additional details about the implementation is deferred to the supplementary material.

Datasets Unlike the common practice of panoramic depth estimation [11, 7, 8] where the train/test splits are created from the same dataset, we consider entirely different datasets from the training dataset for evaluation. Specifically, we use the Stanford 2D-3D-S dataset [64] and OmniScenes [65] dataset for the depth estimation and map-free localization experiments, and the Gibson [66] dataset equipped with the Habitat simulator [67] for robot navigation experiments. Both Stanford 2D-3D-S and OmniScenes datasets contain a diverse set of 3D scenes, with 1413 panoramas captured from 272 rooms for Stanford 2D-3D-S dataset and 7614 panoramas captured from 18 rooms for OmniScenes. The Gibson dataset contains 14 scenes for the validation split, which is used on top of the Habitat simulator [67] to evaluate various robot navigation tasks.

Baselines As our task has not been studied in previous works, we adapt existing test-time adaptation and unsupervised domain adaptation methods to panoramic depth estimation and implement six baselines.

The four test-time adaptation baselines only use the test data for adaptation. Tent [20] only updates the batch normalization layer during adaptation, where we implement a variant that minimizes the loss function from Equation 1. Flip consistency-based approach (FL) inspired by Li et al [68] enforces the depth predictions between the original and flipped image to be similar. Mask consistency-based approach (MA) inspired by Mate [44] enforces depth consistency against a randomly masked panorama image. Pseudo Labeling (PS) [69] imposes losses against the pseudo ground-truth depth map by averaging predictions made from multiple rotated panoramas.

The two unsupervised domain adaptation methods additionally use the labeled source domain dataset for adaptation, where we use AdaIN [51] to perform style transfer between the source and target domain images. Vanilla T^2Net minimizes the discrepancy between the depth predictions of the source domain image transferred to the target domain and the ground truth. CrDoCo [16] additionally makes the target domain predictions to follow the predictions of the target-to-source transferred images. We provide detailed expositions of each baseline on the supplementary material.

5.1. Depth Estimation

Online Adaptation As shown in Figure 5 we evaluate our method on 10 target domains: generic dataset change, 3 global lighting changes (image gamma, white balance, average intensity), 3 image noises (gaussian, speckle,
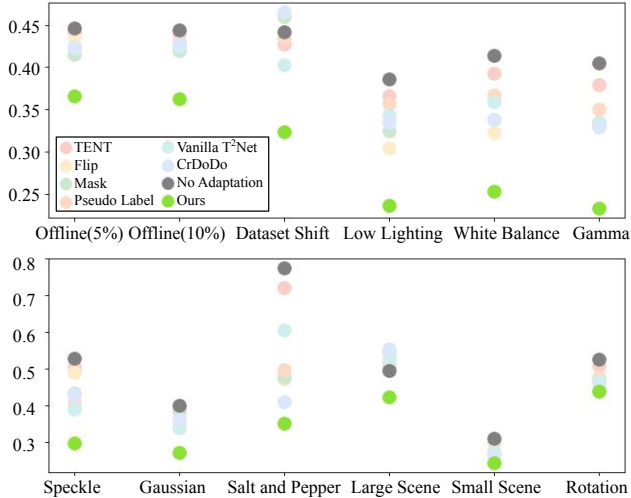
Figure 6: Plot of online adaptation result. The result of our method compared to the baselines with various domain changes (top) and image noises (bottom).

Input Image   No Adaptation   Ours

(a) Depth Maps Generated amidst Gaussian Noise

Ground-Truth   No Adaptation   Ours

Flip   Mask   Pseudo Label

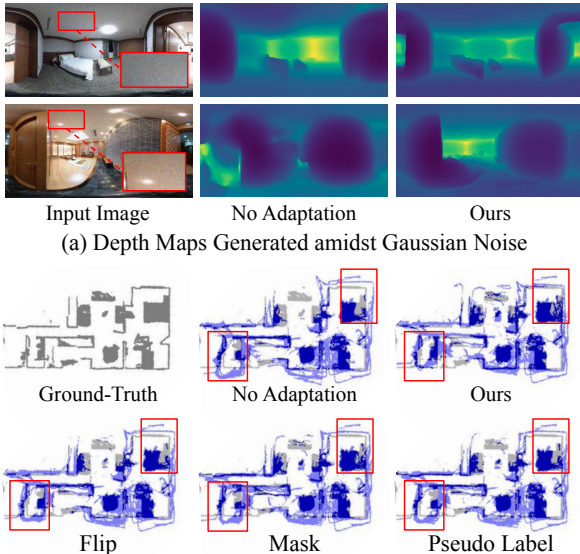(b) Grid Maps Generated from Fixed Trajectory Robot Navigation

Figure 7: Qualitative result of depth maps (top) and grid maps from navigation task (bottom).

salt & pepper), 3 geometric changes (scene scale change to large/small scenes, camera rotation). For scene scale change we use the large rooms manually selected from the evaluation datasets, and for other domains we use the scikit-image library [70] to generate the image-level changes. We provide additional implementation details about the domain setups in the supplementary material.

Figure 6 summarizes the mean absolute error (MAE) of various adaptation methods aggregated from the Stanford2D-3D-S [64] and OmniScenes [65] datasets. We report the full evaluation results in the supplementary mate-

Method	MAE	Abs. Rel.	RMSE	Sq. Rel.
No Adatation	0.4343	0.1949	0.6025	0.1428
Ours w/o Stretch Loss	0.3650	0.1500	0.5450	0.1001
Ours w/o Chamfer Loss	0.3260	0.1468	0.4819	0.1115
Ours w/o Normal Loss	0.3373	0.1512	0.5022	0.0953
Ours w/o Augmentation	0.3972	0.1790	0.5566	0.1247
Ours	**0.3192**	**0.1432**	**0.4683**	**0.0906**

Table 1: Ablation study of key components of our calibration scheme. 'Abs. Rel.' and 'Sq. Rel.' denote the absolute and squared relative error from Eigen et al. [25].

rial. Our method outperforms the baselines across all tested domain shifts, with more than 10cm decrease in MAE in most shifts. The loss functions presented in Section 3.1 thus enables effective depth network calibration. For large scene adaptation, the tested baselines fail to make sufficient performance improvements, whereas our method can largely reduce the error via stretch loss. In addition, note that our method can perform adaptation even in photometric domain shifts such as speckle noise or white balance change, despite the geometry-centric formulation. The multi-view consistency imposed by normal and Chamfer loss ensures the network to make more robust depth predictions amidst these adversaries. A few exemplary depth visualizations are shown in Figure 7, where our online calibration results in depth maps with more accurate depth scales and detail preservation. We report the full results against other depth estimation metrics in the supplementary material.

Offline Adaptation We additionally experiment with offline adaptation scenarios, where the depth network is first trained on a small set of images and then tested on a held-out set. To cope with the data scarcity during training, we apply data augmentation for all the tested methods with $N_{\text{aug}} = 10$. For evaluation, we apply our calibration method separately for each room in the Stanford2D-3D-S [64] and OmniScenes [65] datasets, where the panoramas captured for each room are split for training and testing. Figure 6 shows the adaptation results, where the results are reported after using 5% or 10% of panoramas at each room for training. In both evaluations, our method incurs large amounts of performance enhancement while outperforming all the tested baselines.

Ablation Study To further validate the effectiveness of the various components in our calibration scheme, we perform an ablation study on the offline adaptation setup. We use the OmniScenes [65] dataset for evaluation and use 10% of panoramas in each room for training and the rest for testing. As shown in Table 1, omitting any one of the loss functions leads to suboptimal performance. In addition, the data augmentation scheme incurs a large amount of performance boost, which indicates that despite its simplicity, data augmentation plays a crucial role in data-scarce offline adapta-

| | Exploration | | Point Goal | |
Method	Exp. Ratio	Coll. Rate	Success Rate	Coll. Rate
No Adaptation	0.8835	0.2793	0.4107	0.3864
Flip Consistency	0.9027	**0.2305**	0.5000	0.3516
Mask Consistency	0.8758	0.2677	**0.4643**	0.3460
Pseudo Labeling	0.8701	0.8701	0.3393	0.4058
Ours	**0.9288**	0.2352	**0.4643**	**0.3135**

(a) Exploration and Point Goal Navigation

| | Localization | | Mapping | |
Method	t-Error (m)	R-Error (°)	Cmf. Dist.	MAE
No Adaptation	0.1450	10.2731	0.1959	0.4963
Flip Consistency	0.1329	10.1872	0.1687	0.4743
Mask Consistency	0.1336	**10.1356**	0.1585	0.4637
Pseudo Labeling	0.1347	10.3247	0.1735	0.4711
Ours	**0.1177**	10.2044	**0.1558**	**0.4453**

(b) Localization and Mapping under Fixed Trajectory

Table 2: Robot navigation evaluation against existing methods.

Method	t-error (m)	R-error (°)	Accuracy (0.1m, 5°)	Accuracy (0.2m, 10°)
No Adatation	0.16	0.91	0.32	0.62
Flip Consistency	0.14	0.79	0.36	0.67
Mask Consistency	0.12	0.88	0.41	0.77
CrDoCo	0.11	1.00	0.46	0.78
Ours	**0.09**	**0.87**	**0.52**	**0.86**

Table 3: Map-Free visual localization compared against the baselines. Note that the translation and rotation error thresholds for calculating accuracy is denoted as $(d \, \mathrm{m}, \theta°)$.

tion scenarios.

5.2. Robot Navigation

We consider three tasks for evaluating robot navigation using panoramic depth estimation, following prior works [36, 71, 72]: point goal navigation, exploration, and simultaneous localization and mapping (SLAM) from a fixed robot trajectory. First, point goal navigation aims to navigate the robot agent towards a goal specified from the agent's starting location, e.g. "move to the location 5m forward and 10m right from the origin". Second, the objective of exploration is to explore the given 3D scene as much as possible under a fixed number of action steps. Finally, the SLAM task evaluates the accuracy of the occupancy grid map and pose estimates under a fixed robot trajectory. We use 4 random starting points at each of 14 scenes in the Gibson [66] dataset totaling 56 episodes per task, and set the maximum number of action steps to 500.

Table 2 compares the robot navigation tasks against three baselines (Flip consistency, Mask consistency, and Pseudo Labeling) where our method outperforms the baselines in most metrics. For exploration, our calibration scheme results in largest exploration areas and rates while attaining a small collision rate, which is the total collision divided by the total number of action steps. A similar trend is present

for point goal navigation, where our agent attains the highest success rate with the smallest number of collisions. Note that the success rate is computed as the ratio of navigation episodes where the robot reached within 0.2m of the designated point goal. Finally for fixed-trajectory SLAM, our method exhibits higher localization and mapping accuracy than its competitors. The translation error for localiztion drops largely after adaptation, while the rotation error is similar across all the baselines which is due to the $360 \deg$ field-of-view that makes rotation estimation fairly accurate even prior to localization. On the mapping size, our method attains the smallest 2D Chamfer distance and image error metrics (MAE) measured between the estimated global map and the ground-truth. In addition, as shown in Figure 7 the grid maps resulting from our method best aligns with the ground truth when compared against the maps from the baselines. Thus, the training objectives along with the lightweight augmentation enables quick and effective adaptation for various navigation tasks.

5.3. Map-Free Visual Localization

Similar to the offline evaluation explained in Section 5.1, for each room in the OmniScenes [65] dataset we select 5% of the panorama images for test-time training and the rest for evaluating localization. Then, we treat each evaluation image as the reference image I_{ref} from Section 4.2 and generating a 3D map via depth estimation. To finally evaluate localization we query 10 images that are captured within 2m of each reference image, where we use the dataset's 6DoF pose annotations to determine the criterion.

Table 3 shows the localization performance compared against three baselines (Flip consistency, Mask consistency, and CrDoCo [16]). Following prior works in visual localization [73, 74, 75], we report the median translation and rotation errors along with accuracy where a prediction is considered correct if its translation and rotation error is below a designated threshold. Our method outperforms the baselines in both tested datasets, with almost a 20% increase in accuracy. The geometry correction of our method as shown in Figure 7 leads to more accurate PnP-RANSAC solutions, which in turn results in enhanced localization performance.

6. Conclusion

We propose a simple yet effective calibration scheme for panoramic depth estimation. Domain shifts between training and deployment is a critical problem in panoramic depth estimation as a slight change in the camera pose or lighting can incur large performance drops, while such adversaries are common in practical application scenarios. We introduce three training objectives along with an augmentation scheme to mitigate the domain shifts, where the key idea is to impose geometric consistency via panorama synthesis from random pose perturbations and stretching. Fur-

ther, our experiments show that the light-weight formulation can largely improve performance on downstream applications in mapping and localization. Backed by the plethora of recent advancements in panoramic depth estimation, we project our calibration scheme to function as a key ingredient for practical full-surround depth sensing.

References

[1] "Intel realsense technology." `https://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html`. Accessed: 2023-02-28.

[2] "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: 2023-02-28.

[3] "Faro: 3d measurement, imaging, and realization solutions." `https://www.faro.com/`. Accessed: 2022-10-24.

[4] "Kinect for windows." `https://learn.microsoft.com/en-us/windows/apps/design/devices/kinect-for-windows`. Accessed: 2023-02-28.

[5] "Envision the future: Velodyne lidar." `https://velodynelidar.com/`. Accessed: 2023-02-28.

[6] C. Zhuang, Z. Lu, Y. Wang, J. Xiao, and Y. Wang, "Acdnet: Adaptively combined dilated convolution for monocular panorama depth estimation," in *AAAI Conference on Artificial Intelligence*, 2021.

[7] H. Jiang, Z. Sheng, S. Zhu, Z. Dong, and R. Huang, "Unifuse: Unidirectional fusion for 360° panorama depth estimation," *IEEE Robotics and Automation Letters*, vol. 6, pp. 1519–1526, 2021.

[8] F.-E. Wang, Y.-H. Yeh, M. Sun, W.-C. Chiu, and Y.-H. Tsai, "Bifuse: Monocular 360 depth estimation via bi-projection fusion," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 459–468, 2020.

[9] B. Berenguel-Baeta, J. Bermudez-Cameo, and J. J. Guerrero, "Fredsnet: Joint monocular depth and semantic segmentation with fast fourier convolutions," *ArXiv*, vol. abs/2210.01595, 2022.

[10] G. Albanis, N. Zioulis, P. Drakoulis, V. Gkitsas, V. Sterzentsenko, F. Álvarez, D. Zarpalas, and P. Daras, "Pano3d: A holistic benchmark and a solid baseline for 360° depth estimation," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3722–3732, 2021.

[11] Y. yang Li, Y. Guo, Z. Yan, X. Huang, Y. Duan, and L. Ren, "Omnifusion: 360 monocular depth estimation via geometry-aware fusion," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2791–2800, 2022.

[12] N. Zioulis, A. Karakottas, D. Zarpalas, and P. Daras, "Omnidepth: Dense depth estimation for indoors spherical panoramas," *ArXiv*, vol. abs/1807.09620, 2018.

[13] N. Zioulis, A. Karakottas, D. Zarpalas, F. Álvarez, and P. Daras, "Spherical view synthesis for self-supervised 360° depth estimation," *2019 International Conference on 3D Vision (3DV)*, pp. 690–699, 2019.

[14] "Ricoh theta, experience the world in 360." `https://theta360.com/en/`. Accessed: 2023-03-03.

[15] "Insta360: Action cameras." `https://www.insta360.com/`. Accessed: 2023-03-08.

[16] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang, "Crdoco: Pixel-level domain transfer with cross-domain consistency," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1791–1800, 2019.

[17] A. Lopez-Rodriguez and K. Mikolajczyk, "Desc: Domain adaptation for depth estimation via semantic consistency," *International Journal of Computer Vision*, pp. 1–20, 2022.

[18] H. Akada, S. Bhat, I. Alhashim, and P. Wonka, "Self-supervised learning of domain invariant features for depth estimation," *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 997–1007, 2021.

[19] S.-Y. Lo, W. Wang, J. Thomas, J. Zheng, V. M. Patel, and C.-H. Kuo, "Learning feature decomposition for domain adaptive monocular depth estimation," *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8376–8382, 2022.

[20] D. Wang, E. Shelhamer, S. Liu, B. A. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," in *International Conference on Learning Representations*, 2021.

[21] Y. Sun, X. Wang, Z. Liu, J. Miller, A. A. Efros, and M. Hardt, "Test-time training with self-supervision for generalization under distribution shifts," in *International Conference on Machine Learning*, 2019.

[22] Y. Liu, P. Kothari, B. van Delft, B. Bellot-Gurlet, T. Mordan, and A. Alahi, "Ttt++: When does self-supervised test-time training fail or thrive?," in *Neural Information Processing Systems*, 2021.

[23] N. Hansen, R. Jangir, Y. Sun, G. Alenyà, P. Abbeel, A. A. Efros, L. Pinto, and X. Wang, "Self-supervised policy adaptation during deployment," in *International Conference on Learning Representations (ICLR)*, 2021.

[24] C. Zheng, T.-J. Cham, and J. Cai, "T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks," in *European Conference on Computer Vision*, 2018.

[25] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *NIPS*, 2014.

[26] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 3, pp. 1623–1637, 2020.

[27] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," *ArXiv preprint*, 2021.

[28] M. Fonder, D. Ernst, and M. V. Droogenbroeck, "M4depth: Monocular depth estimation for autonomous vehicles in unseen environments," 2021.

[29] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6602–6611, 2016.

[30] A. Masoumian, H. A. Rashwan, S. Abdulwahab, J. Cristiano, and D. Puig, "Gcndepth: Self-supervised monocular depth estimation based on graph convolutional network," *Neurocomputing*, vol. 517, pp. 81–92, 2021.

[31] C. Godard, O. M. Aodha, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3827–3837, 2018.

[32] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, pp. 1231 – 1237, 2013.

[33] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision*, 2012.

[34] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2041–2050, 2018.

[35] E. Arnold, J. Wynn, S. Vicente, G. Garcia-Hernando, Á. Monszpart, V. Prisacariu, D. Turmukhambetov, and E. Brachmann, "Map-free visual relocalization: Metric pose relative to a single image," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*, pp. 690–708, Springer, 2022.

[36] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural slam," *arXiv preprint arXiv:2004.05155*, 2020.

[37] S. K. Ramakrishnan, Z. Al-Halah, and K. Grauman, "Occupancy anticipation for efficient exploration and navigation," *ArXiv*, vol. abs/2008.09285, 2020.

[38] D. S. Chaplot, D. Gandhi, A. K. Gupta, and R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," *ArXiv*, vol. abs/2007.00643, 2020.

[39] M. Rey-Area, M. Yuan, and C. Richardt, "360monodepth: High-resolution 360deg monocular depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3762–3772, 2022.

[40] C.-H. Peng and J. Zhang, "High-resolution depth estimation for 360° panoramas through perspective and panoramic depth images registration," *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3115–3124, 2022.

[41] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation," in *International Conference on Machine Learning*, 2020.

[42] V. Prabhu, S. Khare, D. Kartik, and J. Hoffman, "Sentry: Selective entropy optimization via committee consistency for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8558–8567, 2021.

[43] D.-H. Lee, "Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks," *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013.

[44] M. J. Mirza, I. Shin, W. Lin, A. Schriebl, K. Sun, J. Choe, H. Possegger, M. Koziński, I.-S. Kweon, K.-J. Yoon, and H. Bischof, "Mate: Masked autoencoders are online 3d test-time learners," *ArXiv*, vol. abs/2211.11432, 2022.

[45] Y. Gandelsman, Y. Sun, X. Chen, and A. A. Efros, "Test-time training with masked autoencoders," in *Advances in Neural Information Processing Systems* (A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, eds.), 2022.

[46] S. Schneider, E. Rusak, L. Eck, O. Bringmann, W. Brendel, and M. Bethge, "Improving robustness against common corruptions by covariate shift adaptation," *ArXiv*, vol. abs/2006.16971, 2020.

[47] C. K. Mummadi, R. Hutmacher, K. Rambach, E. Levinkov, T. Brox, and J. H. Metzen, "Test-time adaptation to distribution shifts by confidence maximization and input transformation," 2021.

[48] S. Roy, A. Siarohin, E. Sangineto, S. R. Bulò, N. Sebe, and E. Ricci, "Unsupervised domain adaptation using feature-whitening and consensus loss," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9463–9472, 2019.

[49] Y. Ganin and V. S. Lempitsky, "Unsupervised domain adaptation by backpropagation," *ArXiv*, vol. abs/1409.7495, 2014.

[50] Y.-T. Yen, C.-N. Lu, W.-C. Chiu, and Y.-H. Tsai, "3d-pl: Domain adaptive depth estimation with 3d-aware pseudo-labeling," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, pp. 710–728, Springer, 2022.

[51] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *ICCV*, 2017.

[52] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[53] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[54] F. Luan, S. Paris, E. Shechtman, and K. Bala, "Deep photo style transfer," *arXiv preprint arXiv:1703.07511*, 2017.

[55] H.-W. Ting, C. Sun, and H.-T. Chen, "Self-supervised 360° room layout estimation," *ArXiv*, vol. abs/2203.16057, 2022.

[56] C. Sun, C.-W. Hsiao, M. Sun, and H.-T. Chen, "Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1047–1056, 2019.

[57] N. J. Mitra, A. T. Nguyen, and L. J. Guibas, "Estimating surface normals in noisy point cloud data," in *SCG '03*, 2003.

[58] F. Tombari, S. Salti, and L. di Stefano, "Unique signatures of histograms for local surface description," in *European Conference on Computer Vision*, 2010.

[59] K.-L. Low, "Linear least-squares optimization for point-to-plane icp surface registration," 2004.

[60] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *CVPR*, 2020.

[61] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[62] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *International Conference on 3D Vision (3DV)*, 2017.

[63] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[64] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, "Joint 2d-3d-semantic data for indoor scene understanding," *arXiv preprint arXiv:1702.01105*, 2017.

[65] J. Kim, C. Choi, H. Jang, and Y. M. Kim, "Piccolo: point cloud-centric omnidirectional localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3313–3323, 2021.

[66] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: Real-world perception for embodied agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9068–9079, 2018.

[67] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A Platform for Embodied AI Research," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[68] B. Li, M. Hu, S. Wang, L. Wang, and X. Gong, "Self-supervised visual-lidar odometry with flip consistency," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3844–3852, 2021.

[69] C. Saltori, E. Krivosheev, S. Lathuilière, N. Sebe, F. Galasso, G. Fiameni, E. Ricci, and F. Poiesi, "Gipso: Geometrically informed propagation for online adaptation in 3d lidar segmentation," in *European Conference on Computer Vision*, 2022.

[70] "Scikit-image: Image processing in python." `https://scikit-image.org/`. Accessed: 2023-02-28.

[71] E. S. Lee, J. Kim, and Y. M. Kim, "Self-supervised domain adaptation for visual navigation with global map consistency," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1707–1716, January 2022.

[72] E. S. Lee, J. Kim, S. Park, and Y. M. Kim, "Moda: Map style transfer for self-supervised domain adaptation of embodied agents," in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, (Berlin, Heidelberg), p. 338–354, Springer-Verlag, 2022.

[73] J. Kim, C. Choi, H. Jang, and Y. M. Kim, "Piccolo: Point cloud-centric omnidirectional localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3313–3323, October 2021.

[74] X. Li, S. Wang, Y. Zhao, J. Verbeek, and J. Kannala, "Hierarchical scene coordinate classification and regression for visual localization," in *CVPR*, 2020.

[75] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," 2015.