

NormalLoc: Visual Localization on Textureless 3D Models using Surface Normals

Jiro Abe Gaku Nakano Kazumine Ogura

Visual Intelligence Research Laboratories, NEC Corporation, Kanagawa, Japan

{j-abe, g-nakano, k-oguraay}@nec.com

Abstract

We propose *NormalLoc*, a novel visual localization method for estimating the 6-DoF pose of a camera using textureless 3D models. Existing methods often rely on color or texture information, limiting their applicability in scenarios where such information is unavailable. *NormalLoc* addresses this limitation by using rendered normal images generated from surface normals of 3D models to establish a training scheme for both global descriptor computation and matching. This approach enables robust visual localization even when geometric details are limited. Experimental results demonstrate that *NormalLoc* achieves state-of-the-art performance for visual localization on textureless 3D models, especially in scenarios with limited geometric detail.

1. Introduction

Visual localization, the task of estimating the 6-Degrees of Freedom (DoF) pose of a camera from a given query image, is a fundamental technique in various fields, including augmented reality (AR) [20, 30], autonomous driving [12], and drone navigation [26]. For accurate pose estimation, many approaches typically rely on structure-based visual localization pipelines. These pipelines rely on a 3D model of the target scene acquired prior to localization. This 3D model can be a Structure-from-Motion (SfM) point cloud [1, 39] reconstructed from database images (i.e., images of the target scene), or a laser-scanned colored point cloud [46]. In the latter case, the point-colors are often provided by images captured by cameras accompanying the laser scanner, and these images can also serve as database images for visual localization. These pipelines generally involve two main stages: (1) global descriptor computation to retrieve database images visually similar to the query, and (2) matching to establish 2D-2D correspondences between the query and retrieved images. These 2D-2D correspondences then enable the establishment of 2D-3D correspondences to

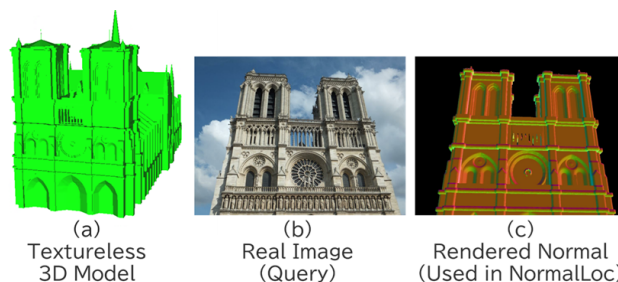


Figure 1. Given a (a) textureless 3D model, *NormalLoc* estimates the 6-DoF pose of a (b) real image by generating (c) rendered normals as database images.

the 3D model, allowing for camera pose estimation using algorithms such as PnP solvers [9, 32].

Recent studies [10, 23, 34] show an increasing focus on visual localization techniques that employ textureless 3D models, as they are often easier and more cost-effective to acquire. For instance, pre-existing CAD designs eliminate the need for field data collection for database creation, and colorless point clouds can be obtained more affordably from standalone LiDAR systems compared to camera-equipped ones. According to the CADLoc benchmark, MeshLoc [33], a method that applies global descriptor computation and matching to synthetic images rendered from a textureless 3D model, can achieve accurate 6-DoF pose estimation if the 3D model contains high geometric detail. However, the use of 3D models with limited geometric detail, as illustrated in Fig. 1(a), remains a challenging task. A potential explanation for the performance degradation of MeshLoc on such low-geometric-detail models is that it directly utilizes pre-trained networks for both global descriptor computation and matching, and these networks are trained on real image domains. These networks may perform poorly when applied to synthetic images rendered from 3D models with low geometric detail because rendered images do not resemble real images.

In this paper, we propose NormalLoc, a more robust visual localization method for textureless 3D models with low geometric detail. NormalLoc achieves robustness by establishing a training scheme for global descriptor computation and matching that leverages both real images and a synthetic rendered image representation, which we refer to as *rendered normals* (see Fig. 1(c)). Rendered normals are generated from the surface normal vectors of the 3D model and can be applied not only to CAD models but also to meshes, point clouds, and SfM datasets such as MegaDepth [25], which are commonly used to train matching networks in the real image domain [28, 45]. This enables training a network capable of handling both real and rendered normal images using large-scale SfM datasets for global descriptor computation and matching, which can then be applied to various types of 3D models. Experimental results demonstrate that NormalLoc achieves state-of-the-art performance, particularly for 3D models with low geometric detail.

2. Related Work

2.1. Visual Localization on 3D Models With Color Information

Visual localization is most often achieved through using a database of real images of the target scene. Such methods often employ image retrieval strategies, referred to as Visual Place Recognition (VPR) [2, 3, 14, 17], to obtain a coarse camera location. For more precise 6-DoF camera pose estimation, 3D structure-based methods are utilized. These approaches establish 2D-3D correspondences between the query image and a 3D model, often via initial 2D-2D matching to database images, followed by PnP solvers. Common 3D models include SfM point clouds [1, 39, 42], textured meshes [33, 47, 54], or Digital Elevation Models (DEMs) [6].

2.2. Visual Localization on 3D Models Without Color Information

Many approaches have been proposed to perform visual localization using 3D models without color information to leverage more readily available 3D models, such as colorless LiDAR point clouds and textureless CAD models, compared to colored point clouds and textured meshes. A common approach is to use a point cloud as the 3D model and perform 2D-3D matching between the query image and the point cloud [10, 19, 23, 24, 35, 51, 53, 55]. Although direct matching between a 2D image and a 3D point cloud is challenging due to the different scene representations, recent advances in deep learning have enabled simultaneously learning of 2D and 3D descriptors [10, 35]. However, direct 2D-3D matching is inherently challenging because the perspective distortions and occlusions present in the 2D query

image are not directly represented in the 3D point cloud. In addition, it is difficult to prepare training datasets for learning correspondences between real images and CAD models with varying geometric detail.

Another approach for visual localization with textureless models is to use synthetically rendered images generated from the 3D model as database images [33, 52]. These methods first convert the 3D model into a set of rendered images using specific shaders [33] or depth maps [52]. These studies demonstrate that deep learning models pre-trained on image datasets can be applied to visual localization in textureless scenarios. However, [52] focuses only on the matching stage, i.e., establishing 2D-2D correspondences between the query image and the already retrieved image. MeshLoc [33] uses both global descriptor computation and matching, but the performance degrades when applied to 3D models with low geometric detail [34].

3. Proposed Method

The goal of visual localization is to estimate the 6-DoF pose of a camera, given a query image and its known intrinsic parameters. The world coordinates of the estimated pose are defined with respect to a given 3D model. Unlike standard visual localization setups, which rely on real images or textured 3D meshes, we do not assume any additional color information for the 3D model.

Fig. 2 gives an overview of the proposed *NormalLoc* pipeline. The pipeline consists of an offline step to prepare a database of images, and an online step to estimate the 6-DoF pose of the query image based on the prepared database.

The proposed pipeline is inspired by standard visual localization pipelines [39] and MeshLoc [33], which include global descriptor computation and matching. A key feature of NormalLoc is its use of *rendered normals* (Fig. 1(c)) to construct the database instead of real images. This approach is motivated by the versatility of normal vectors, which can be computed for various 3D representations, including CAD models, meshes, LiDAR point clouds, 3D Gaussian Splattings (3DGS) [15, 18], Neural Radiance Fields (NeRF) [31, 49], and especially depth maps derived from SfM/MVS. This versatility enables applying global descriptor computation and matching, trained on large datasets (e.g., SfM/MVS [25]), to visual localization on diverse 3D models.

In the following subsections, we describe the offline step (Sec. 3.1), the online step (Sec. 3.2), and the network for global descriptor computation and matching (Sec. 3.3).

3.1. Offline Step

The offline step prepares a database consisting of rendered normals and computes global descriptors for each rendered normal.

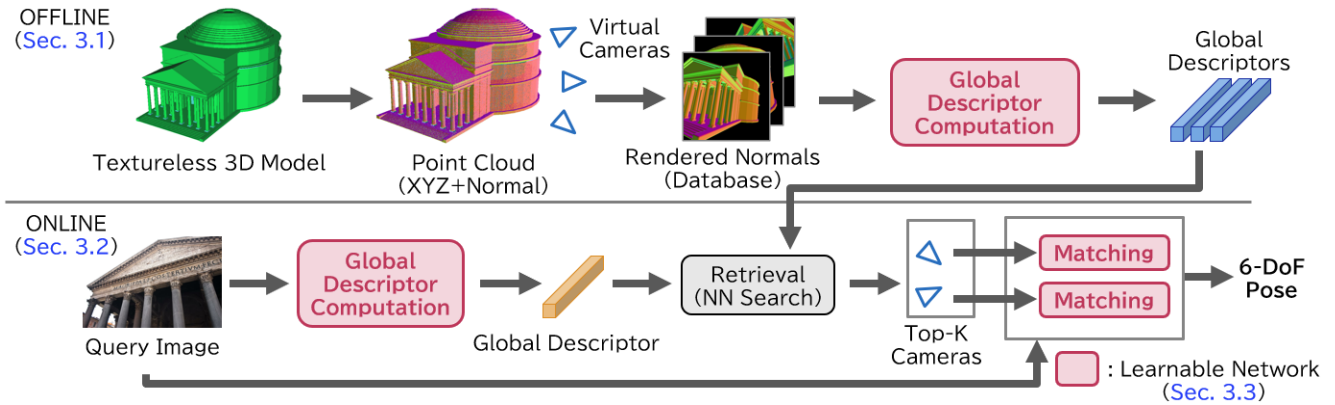


Figure 2. The pipeline of NormalLoc.

First, we convert the 3D model into a point cloud, for example, by uniformly sampling points from the surfaces of the 3D model. This process allows us to apply NormalLoc to various types of 3D models, as a point cloud can be easily obtained from most 3D models. We then compute normal vectors for each point, typically using a k -Nearest Neighbors (k -NN) search and built-in functions from Open3D [56].

Second, we place the virtual cameras in the 3D space to render normal images. We follow the default configuration for each evaluation dataset in our experiments, while generic camera placement strategies, such as on sphere surfaces [34] or grid [4, 38], have also been discussed in the literature. For each virtual camera, we then obtain a depth map and a rendered normal. The depth map is used in the online step to lift 2D-2D correspondences to 2D-3D. We use the Pulsar renderer [22], which can render the point cloud densely by representing each point as a 3D sphere. This renderer allows us to obtain a depth map and a *normal map*, which is a 3-channel image where each pixel directly stores the normal vector (n_x, n_y, n_z) instead of (R, G, B) values. However, normal maps have two undesirable properties as database images: (i) the direction of the normal vectors depends on the definition of the world coordinate system, even if the 3D model itself does not change, and (ii) the sign of the normal vectors is ambiguous. Therefore, we normalize the direction of the normal vectors stored in the normal map by applying the rotation matrix of the virtual camera’s extrinsic matrix to remove the dependence on the world coordinate system. We then choose the sign of the normal vectors so that they point towards the virtual camera. The result of this normalization is the rendered normal.

Then, finally, we compute a global descriptor for each rendered normal. This descriptor should be similar to the global descriptor computed from a real image if the rendered normal and the real image represent a similar part of the target scene. We describe the details of the global de-

scriptor in Sec. 3.3.

As a result of the offline step, we obtain a set of rendered normals, each of which is associated with a global descriptor, the extrinsic parameters of virtual camera, the intrinsic parameters of the virtual camera, and the depth map.

3.2. Online Step

The online step estimates the 6-DoF pose of the query image based on the prepared database. As shown in Fig. 2, the online step mainly consists of global descriptor computation for the query image, image retrieval based on global descriptors, and matching.

First, a global descriptor is computed for the query image using the network described in Sec. 3.3. This descriptor is then used to retrieve top- K most similar rendered normals from the database, as measured by L_2 distance.

Second, for each of the top- K retrieved cameras, 2D-2D correspondences are established between the query image and the rendered normal using the matching method described in Sec. 3.3. These 2D-2D correspondences are lifted to 2D-3D correspondences using the depth map associated with each database image. After performing the matching for all K retrieved cameras, all 2D-3D correspondences are concatenated to form a single set. PnP solver with LO-RANSAC [7], provided by PoseLib [21], is then applied to this combined set to obtain a robust 6-DoF pose.

The online step thus yields the 6-DoF pose of the query image. The details of the network for global descriptor computation and matching are described in Sec. 3.3.

3.3. Network for Global Descriptor Computation and Matching

3.3.1. Network Architecture

Fig. 3 illustrates the network architecture used for global descriptor computation and matching.

Our network architecture is primarily inspired by LoFTR [45], an end-to-end approach that learns 2D-2D

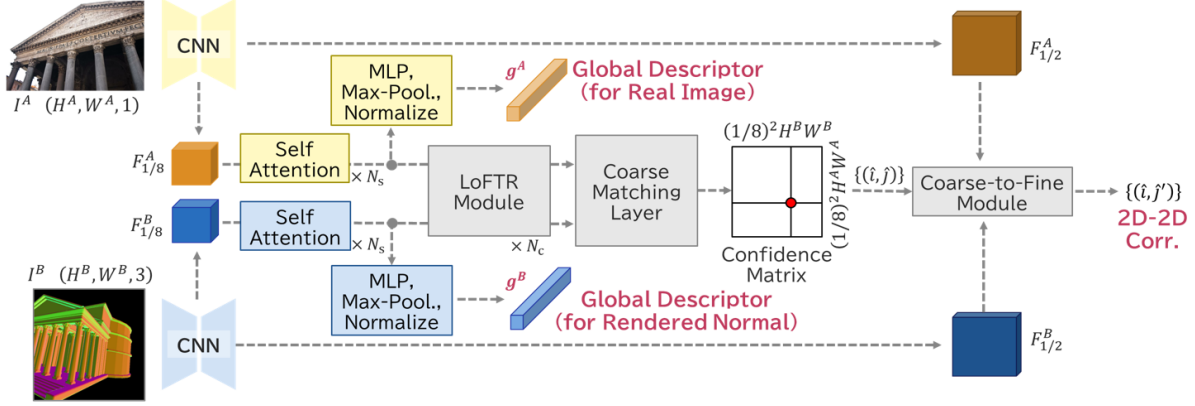


Figure 3. Network architecture used for the global descriptor computation and the matching.

correspondences without explicit keypoint detection, which also facilitates cross-modality matching [48]. Our network is designed to jointly learn global descriptors and 2D-2D correspondences for a pair of a real image (I^A) and a rendered normal (I^B).

The network employs a standard convolutional architecture, FPN [27], as its CNN backbone for multi-scale feature extraction, yielding 1/8 scale features ($F_{1/8}^A, F_{1/8}^B$) and 1/2 scale features ($F_{1/2}^A, F_{1/2}^B$) from the real image and rendered normal, respectively. These backbones do not share any weights. To generate global descriptors, self-attention layers are applied to each 1/8 scale feature. The outputs of these self-attention layers are then processed through a multi-layer perceptron (MLP), max-pooling, and L_2 normalization to obtain the high-dimensional global descriptors (g^A, g^B). Concurrently, the network estimates 2D-2D correspondences between I^A and I^B using the LoFTR module, following the design introduced in the original LoFTR [45]. Consistent with recent work [28], we replace absolute positional encoding with rotary embeddings [44] in each self-attention layer.

3.3.2. Training on SfM/MVS Dataset

We train the network on MegaDepth [25], which is a large SfM dataset commonly used for training matching networks on real images [40, 45]. These datasets provide dense depths, from which we can obtain rendered normals by converting them into point clouds and applying k -NN normal computation. This allows us to adapt common training procedures for real image matching by substituting the real image input with the corresponding rendered normal I^B .

To train the global descriptor computation, we augment the original LoFTR loss with a global loss \mathcal{L}_g . Our total loss is formulated as $\mathcal{L} := \mathcal{L}_g + \mathcal{L}_c + \mathcal{L}_f$, where \mathcal{L}_c is the negative log-likelihood loss over the confidence matrix and \mathcal{L}_f is the L_2 loss for the final 2D-2D correspondences.

As depicted in Fig. 4, we explore two strategies for in-

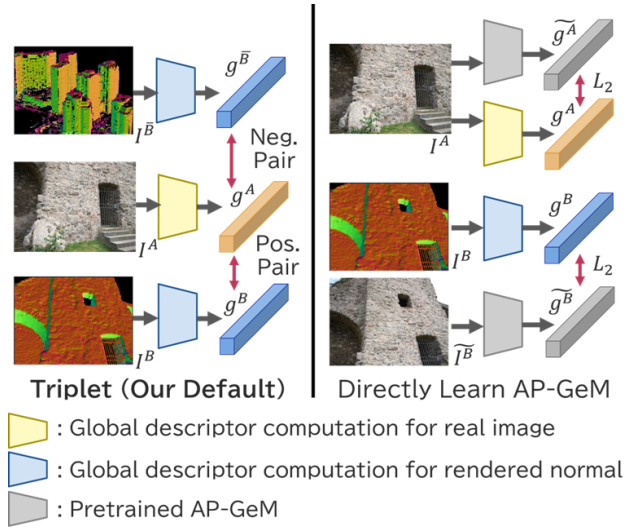


Figure 4. Two strategies for learning global descriptors.

corporating \mathcal{L}_g : a triplet loss [43] or direct learning (distillation) of the AP-GeM descriptor [36]. The distillation strategy, which shares similarities with approaches such as MeshVPR [5] (a VPR method on textured meshes), is motivated by the larger number of scenes available in the Landmarks-clean dataset (586 scenes) compared to MegaDepth (196 scenes). Our experiments in Sec. 4.4 demonstrate that the triplet loss achieves superior performance despite this potential disadvantage, and is therefore adopted as the default for NormalLoc.

Specifically, the two strategies are formulated as follows. For the triplet loss strategy, given a rendered normal $I^{\bar{B}}$ sampled from another scene of I^B , the global loss is defined as $\mathcal{L}_g := [d(g^A, g^B) - d(g^A, g^{\bar{B}}) + \alpha]_+$, where $g^{\bar{B}}$ denotes the global descriptor derived from $I^{\bar{B}}$, $d(x, y) := \|x - y\|_2$, $[x]_+ := \max(x, 0)$, and α is a margin value. On the other hand, for the direct learning strategy, given a

real image \tilde{I}^B related to I^B , the global loss is defined as $\mathcal{L}_g := d(g^A, \tilde{g}^A) + d(g^B, \tilde{g}^B)$, where \tilde{g}^A and \tilde{g}^B denote the global descriptors obtained by applying the pre-trained AP-GeM to I^A and \tilde{I}^B , respectively.

4. Experiments

We evaluate NormalLoc for visual localization on textureless CAD models (Sec. 4.2) and city-scale mesh models (Sec. 4.3). We also analyze the impact of the global loss induction strategy (Sec. 4.4).

4.1. Implementation Details

The input real images and rendered normals are resized such that their longer side is 640 pixels during both training and inference. The network consists of $N_s = 2$ self-attention layers, $N_c = 3$ LoFTR modules, and uses a 2,048-dimensional global descriptor. The margin α in the triplet loss is set to 0.1.

We train our network on the MegaDepth dataset [25], which comprises internet images from 196 outdoor scenes. To enhance generalization across various 3D model qualities, we generate three rendered normals for each image with different k values ($k = 8, 64, 512$) from the corresponding dense depth map. One of these is randomly selected as the input I^B during training. Training proceeds for 30 epochs (approx. 1.5 days) on 8 Quadro RTX5000 GPUs with a batch size of 8. For each epoch, 96 pairs per scene with overlap ratios from 0.1 to 0.7 are sampled. The initial learning rate is 2.0×10^{-4} , warmed up to 2.0×10^{-3} over the first 4 epochs, and then halved every 4 epochs.

For inference, rendered normals are computed with $k = 64$ nearest neighbor points. The number of retrieved cameras K is set to 20 for CAD models and 50 for mesh models.

4.2. Evaluation on CADLoc

We evaluate NormalLoc on the CADLoc benchmark [34], which contains 17 textured and 9 textureless CAD models representing five building-scale scenes, along with corresponding real query images. For this evaluation, we focused only on the 9 textureless models (Fig. 5). These models exhibit varying levels of geometric detail.

For the CADLoc evaluation, CAD models are converted into point clouds by uniformly sampling 100 million points and downsampling by an interval of 0.01 (CADLoc normalizes scene length to 1 unit, so 0.01 is a relative interval). Following the CADLoc setup, virtual cameras are placed on three concentric spheres (radii 0.6, 1.2, 1.8) centered at the 3D model’s centroid. Camera elevation is limited to ± 15 degrees. A total of 432 camera positions are sampled at 10-degree intervals for azimuth and elevation, with each camera aimed at the model’s centroid and configured to produce 640x640 pixel square images with a 500-pixel focal length.

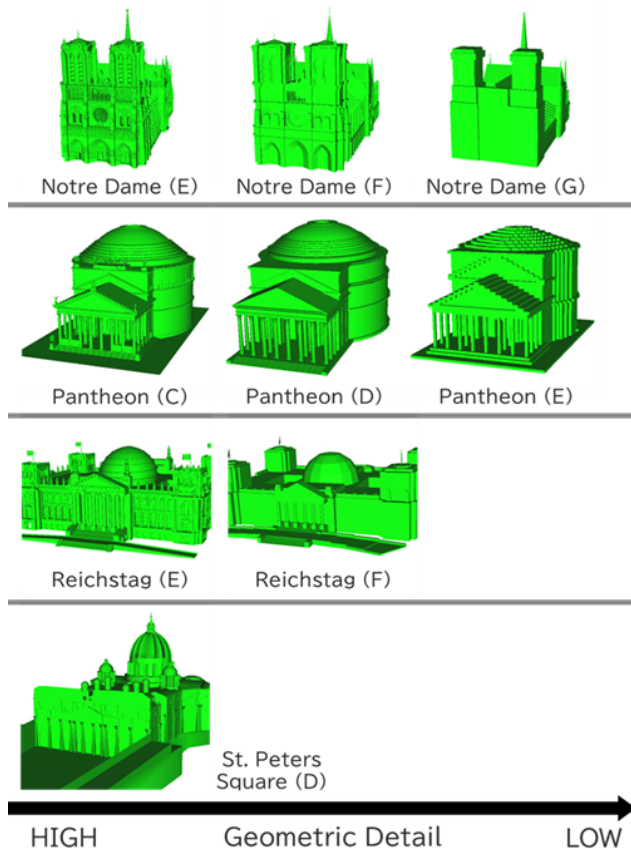


Figure 5. All textureless CAD models in CADLoc [34].

We evaluate performance using the success ratio for mean Dense Correspondence Reprojection Error (DCRE) [34, 50], computed at three thresholds (10%, 20%, 30%) with the official CADLoc script. DCRE, for each pixel in the query image, measures the pixel error between ground truth (GT) and estimated poses. Specifically, for each pixel, its 3D position is calculated using the depth map derived from the CAD model at the GT pose. This 3D point is then projected onto the image plane using both the GT and estimated poses to compute the pixel error. The mean DCRE for each query image is normalized by the query image diagonal and is referred to as “Mean DCRE - GA” in the original CADLoc. The DCRE computation is independent of database images.

Table 1 compares NormalLoc’s success ratios on Notre Dame CAD models against MeshLoc [33], the CADLoc benchmark baseline. NormalLoc consistently outperforms MeshLoc across all Notre Dame models. The performance gap is particularly significant for Notre Dame (F), which has less geometric detail than Notre Dame (E). However, for models with minimal surface geometry such as Notre Dame (G), NormalLoc’s accuracy decreases, similarly to

Table 1. Comparison of success ratios under three thresholds (10% / 20% / 30%) for mean DCRE evaluated on the Notre Dame CAD models.

	Notre Dame (E)	Notre Dame (F)	Notre Dame (G)
MeshLoc - LoFTR	70.9 / 75.7 / 82.0	5.3 / 12.2 / 21.2	1.6 / 10.1 / 20.1
MeshLoc - Patch2Pix+SG	66.1 / 74.6 / 80.4	6.9 / 18.0 / 24.3	1.6 / 5.8 / 22.2
MeshLoc - SuperGlue	70.9 / 78.3 / 83.1	23.8 / 36.5 / 48.1	4.8 / 14.3 / 30.7
NormalLoc	78.8 / 82.5 / 86.2	79.9 / 82.5 / 87.3	5.8 / 21.2 / 39.2

MeshLoc.

Fig. 6 shows the success ratios under the 30% threshold for mean DCRE for all textureless CAD models in CADLoc. NormalLoc consistently achieves superior performance across all models, with a particularly significant advantage for those with low geometric detail.

Fig. 7 shows examples of matching between query images and rendered normals using pre-trained LoFTR [45] and our trained network. In these visualizations, 2D-2D correspondences are color-coded by confidence (red for highest, blue for lowest). Our network is effective at matching even when unrelated objects appear in the query image, or when the query image captures only a portion of the CAD model.

4.3. Evaluation on Aachen Day-Night v1.1

We evaluate NormalLoc’s performance using the Aachen Day-Night v1.1 benchmark [41], a large-scale outdoor visual localization dataset with query images acquired in the city of Aachen, Germany. The dataset comprises 6,697 daytime reference images, 824 daytime query images, and 191 nighttime query images. Consistent with MeshLoc [33], we exclusively utilize the more challenging nighttime queries. To evaluate localization on textureless 3D models, we employ the meshes provided by MeshLoc [33]. These meshes were generated by applying Screened Poisson Surface Reconstruction (SPSR) [16] to a point cloud derived from SfM/MVS of the reference images. MeshLoc provides four meshes, designated AC15, AC14, AC13, and AC13-C, which have different levels of surface quality due to the control of the maximum resolution of the Octree in the SPSR procedure. Notably, AC13-C is a compressed version of AC13 that preserves color information as surface textures rather than vertex properties. Fig. 8 shows these four meshes represented as rendered normals. The AC13-C model offers the least geometric detail but features significantly smaller storage requirements (0.27GB) compared to the AC15 model (2.36GB), which possesses the most geometric detail.

As illustrated in Fig. 9, our evaluation utilizes not only rendered normals as database representations but also four additional image types provided by MeshLoc [33]: real images, texture images rendered directly from meshes, Ambient Occlusion (AO) images computed using MeshLab [8],

and tricolor images simulated with three colored directional light sources. All rendered images, including the rendered normals, were generated using the same camera extrinsics and intrinsics as the Aachen Day-Night v1.1 reference images. To generate the rendered normals, we convert the meshes into point clouds by uniformly sampling 1 billion points and downsampling them at 3 cm intervals. Normal computation for the highly detailed AC15 model typically takes about 13 minutes.

To benchmark NormalLoc against pipelines that directly apply global descriptor computation and matching pre-trained on real image datasets, we established a baseline. This baseline differs from NormalLoc only in its global descriptor computation and matching components. Specifically, the baseline employs pre-trained AP-GeM [36] global descriptor and pre-trained LoFTR [45] matcher. Additionally, we compare NormalLoc’s performance with that reported for MeshLoc [33] using tricolor images. MeshLoc operates in a *hybrid scenario*, where real images are utilized for global descriptor computation and rendered images are used only for matching. This hybrid setup explores the applicability of pre-trained image matchers to rendered images.

Table 2 presents the success ratios for the Aachen Day-Night v1.1 dataset, evaluated under three thresholds for absolute camera position and pose error: (0.25m, 2°), (0.5m, 5°), and (5m, 10°). The table is vertically divided into three scenarios: the *textured scenario*, where color information (e.g., real or texture images) is used for both global descriptor computation and matching; the *hybrid scenario*, which employs color information only for global descriptor computation; and the *textureless scenario*, where no color information from the 3D model is used for either global descriptor computation or matching.

As shown in Table 2, NormalLoc achieves the highest performance in the textureless scenario. Furthermore, NormalLoc achieves higher performance than MeshLoc in the hybrid scenario on the AC13-C model, which has the least geometric detail. However, a performance gap persists between NormalLoc and the baseline utilizing texture images.

4.4. Impact of Global Descriptor Loss

As introduced in Sec. 3.3.2, we consider two strategies for incorporating the global loss: a triplet loss or directly

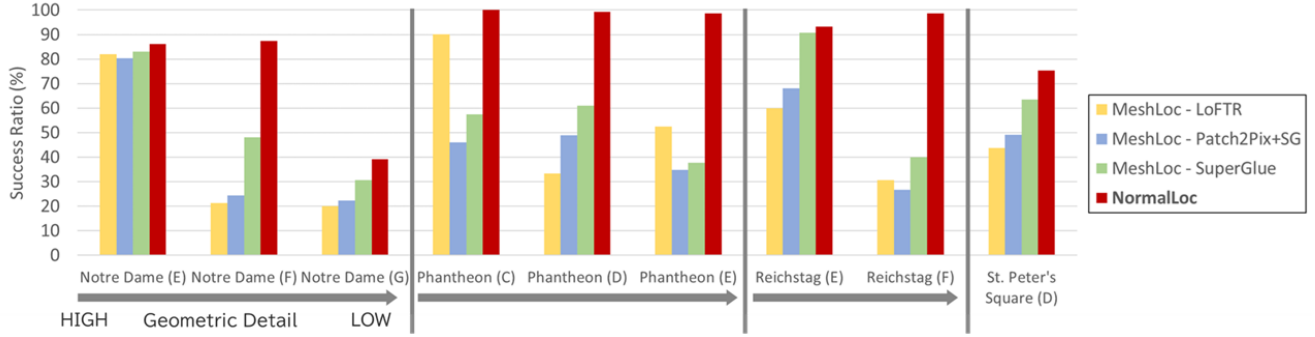


Figure 6. Comparison for all textureless CAD models in CADLoc. Performance is measured by the success ratio under the 30% threshold for mean DCRE.

Table 2. Performance evaluation for the Aachen Day-Night v1.1 dataset. The performance is evaluated as the success ratios under three thresholds for the absolute error of the estimated camera position and pose: $(0.25m, 2^\circ)$, $(0.5m, 5^\circ)$, and $(5m, 10^\circ)$. The table is divided vertically into three scenarios: the *textured scenario*, which uses color information of the 3D models for both global descriptor computation and matching; the *hybrid scenario*, which uses color information only for global descriptor computation; and the *textureless scenario*, which uses no color information for either global descriptor computation or matching.

	AC15	AC14	AC13	AC13-C
<i>Textured Scenario</i>				
Baseline (real image)	72.8 / 88.0 / 95.3	73.8 / 88.0 / 95.3	73.3 / 88.0 / 95.3	72.8 / 88.0 / 95.8
Baseline (texture)	61.8 / 77.5 / 84.3	62.3 / 77.5 / 82.7	55.5 / 72.3 / 79.6	63.4 / 81.2 / 88.5
<i>Hybrid Scenario</i>				
MeshLoc - Patch2Pix+SG [33]	40.3 / 66.0 / 80.1	39.3 / 68.6 / 80.6	23.0 / 55.0 / 78.5	9.4 / 25.1 / 57.6
MeshLoc - SuperGlue [33]	37.2 / 60.7 / 77.5	33.0 / 65.4 / 79.1	22.0 / 50.8 / 74.3	7.3 / 23.0 / 53.9
<i>Textureless Scenario</i>				
Baseline (AO)	3.7 / 10.5 / 24.6	5.8 / 15.2 / 32.5	0.0 / 0.5 / 8.4	0.0 / 0.0 / 2.6
Baseline (tricolor)	8.4 / 20.9 / 37.2	8.4 / 22.5 / 41.9	2.6 / 9.9 / 29.3	0.5 / 4.2 / 19.9
NormalLoc	47.6 / 65.4 / 78.0	44.5 / 63.9 / 74.9	43.5 / 60.2 / 75.9	36.1 / 57.1 / 76.4

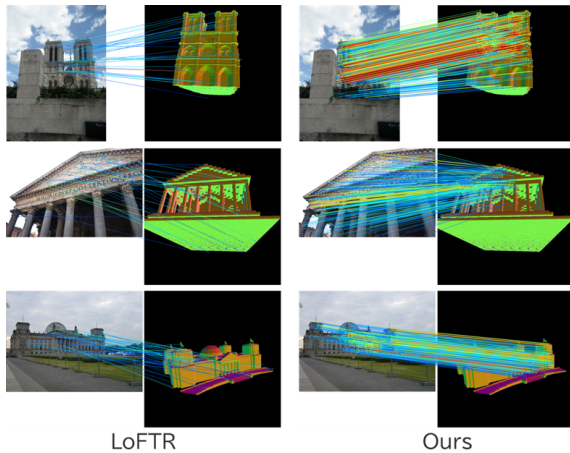


Figure 7. Examples of matching between real image and rendered normal.

learning pre-trained AP-GeM global descriptors. Fig. 10 presents an evaluation of these different global loss strate-

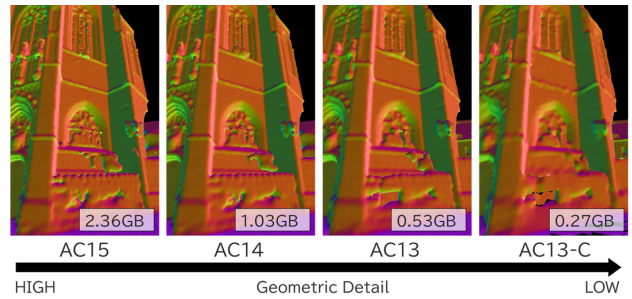


Figure 8. Examples of rendered normals for the four meshes provided by MeshLoc.

gies on the Aachen Day-Night v1.1 dataset (AC13 model, $(5m, 10^\circ)$ threshold). We vary K (the number of retrieved images) among 5, 10, and 50. To assess the impact of global descriptor learning, we compare the triplet loss and direct learning strategies against a baseline where the network is trained only for matching (i.e., without a global loss) and applies pre-trained AP-GeM to rendered normals during

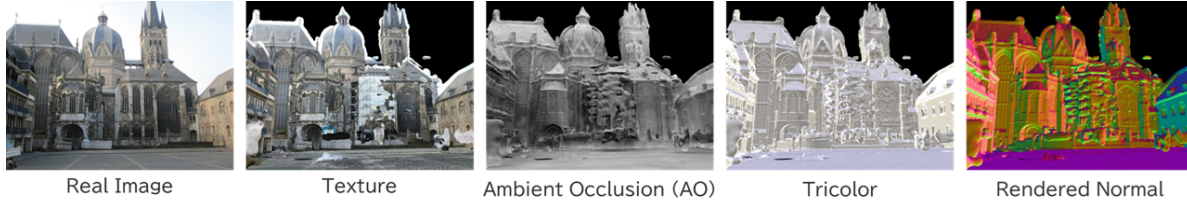


Figure 9. Examples of database images used for evaluation on the Aachen Day-Night v1.1 dataset.

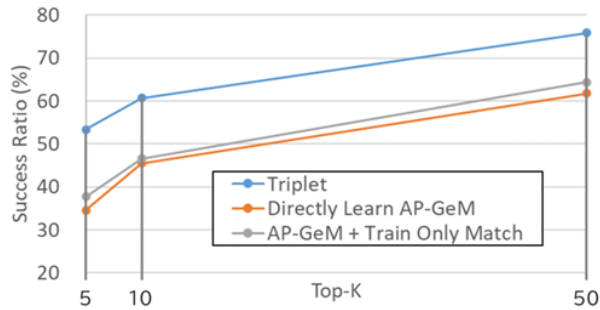


Figure 10. Comparison of global descriptor losses on Aachen Day-Night v1.1. The performance is evaluated as the success ratio under the $(5m, 10^\circ)$ threshold for the absolute error of the estimated camera position and pose on the AC13 model with various K values, where K is the number of retrieved images.

Table 3. Performance evaluation on the DUC1 scene in the In-Loc [46] dataset. Performance is evaluated as the success ratio under the both of the 10° absolute pose error and three thresholds for absolute camera position (0.25m, 0.5m, 1.0m).

Method	0.25m / 0.5m / 1.0m ($@10^\circ$)
KAPTURE [13]+R2D2 [37]	41.4 / 60.1 / 73.7
HLoc [39]+LoFTR [45]	47.5 / 72.2 / 84.8
NormalLoc	6.1 / 12.6 / 16.7

inference. The results indicate that the triplet loss strategy achieves higher performance than the direct learning strategy. Furthermore, the direct learning strategy performs slightly worse than training only the matching component.

4.5. Limitations

Localization on Indoor Environments. NormalLoc’s performance on indoor environments, specifically the In-Loc [46] dataset, is evaluated in Table 3. InLoc is a large-scale indoor dataset featuring 3D models acquired by terrestrial laser scanners and provided as point clouds. We apply the same settings as used in the Aachen Day-Night v1.1 evaluation (Sec. 4.3). Although NormalLoc can successfully localize query images capturing general views of halls or stairs, even when acquired at different times than the 3D model, it achieves a low success ratio of only about

15%. This reduced performance is likely attributable to the shorter distances to scene geometry typical of indoor environments, similar to challenges encountered in textured scenarios. Specifically, NormalLoc often struggles with queries that depict geometrically simple subjects such as plain walls or floors, or those captured in close proximity to objects. In these cases, distinguishing the normal map rendered at the ground truth pose from those rendered at incorrect positions becomes highly challenging. Such difficult queries are more prevalent in indoor settings. For future work, exploring wider FoV queries or integrating NormalLoc with matching methods that incorporate line correspondences in addition to 2D-2D point correspondences [11, 29] could improve performance on indoor environments, as discussed for textured scenarios.

5. Conclusion

In this paper, we have presented NormalLoc, a novel visual localization method designed for textureless 3D models with low geometric detail. NormalLoc leverages rendered normal images, generated from 3D model surface normals, to establish a robust training scheme for global descriptor computation and matching. The versatility of rendered normals bridges the gap between diverse 3D data representations such as CAD models and SfM point clouds, and thereby enables a unified training framework. Our experimental results on the CADLoc and Aachen Day-Night v1.1 datasets demonstrate that NormalLoc achieves state-of-the-art performance, particularly in scenarios with limited geometric detail. We note that the design of NormalLoc, with its initial conversion of 3D models to point clouds, opens avenues for further exploration of broader applications using different types of 3D models beyond CAD models and meshes, such as mobile or airborne LiDAR point clouds.

References

- [1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 1, 2
- [2] Amar Ali-Bey, Brahim Chaib-draa, and Philippe Giguere. Boq: A place is worth a bag of learnable queries. In *CVPR*, pages 17794–17803, 2024. 2
- [3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, pages 5297–5307, 2016. 2
- [4] Mathieu Aubry, Bryan C Russell, and Josef Sivic. Painting-to-3d model alignment via discriminative visual elements. *ACM TOG*, 33(2):1–14, 2014. 3
- [5] Gabriele Berton, Lorenz Junglas, Riccardo Zaccone, Thomas Pollok, Barbara Caputo, and Carlo Masone. Meshvpr: Citywide visual place recognition using 3d meshes. In *ECCV*, pages 321–339. Springer, 2024. 4
- [6] Jan Brejcha, Michal Lukáč, Yannick Hold-Geoffroy, Oliver Wang, and Martin Čadík. Landscapear: Large scale outdoor augmented reality by matching photographs with terrain models using learned descriptors. In *ECCV*, pages 295–312. Springer, 2020. 2
- [7] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized ransac. In *Joint pattern recognition symposium*, pages 236–243. Springer, 2003. 3
- [8] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, Guido Ranzuglia, et al. Meshlab: an open-source mesh processing tool. In *Eurographics Italian chapter conference*, pages 129–136. Salerno, 2008. 6
- [9] Yaqing Ding, Jian Yang, Viktor Larsson, Carl Olsson, and Kalle Åström. Revisiting the p3p problem. In *CVPR*, pages 4872–4880, 2023. 1
- [10] Mengdan Feng, Sixing Hu, Marcelo H Ang, and Gim Hee Lee. 2d3d-matchnet: Learning to match keypoints across 2d image and 3d point cloud. In *International Conference on Robotics and Automation (ICRA)*, pages 4790–4796. IEEE, 2019. 1, 2
- [11] Shuang Gao, Jixiang Wan, Yishan Ping, Xudong Zhang, Shuzhou Dong, Yuchen Yang, Haikuan Ning, Jijunnan Li, and Yandong Guo. Pose refinement with joint optimization of visual points and lines. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2888–2894. IEEE, 2022. 8
- [12] Lionel Heng, Benjamin Choi, Zhaopeng Cui, Marcel Gepfert, Sixing Hu, Benson Kuan, Peidong Liu, Rang Nguyen, Ye Chuan Yeo, Andreas Geiger, et al. Project autovision: Localization and 3d scene perception for an autonomous vehicle with a multi-camera system. In *International Conference on Robotics and Automation (ICRA)*, pages 4695–4702. IEEE, 2019. 1
- [13] Martin Humenberger, Johann Cabon, Nicolas Guerin, Julien Morat, Vincent Leroy, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, and Gabriela Csurka. Robust image retrieval-based visual localization using kapture. *arXiv preprint arXiv:2007.13867*, 2020. 8
- [14] Sergio Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition. In *CVPR*, pages 17658–17668, 2024. 2
- [15] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. In *CVPR*, pages 5322–5332, 2024. 2
- [16] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM TOG*, 32(3):1–13, 2013. 6
- [17] Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg. Anyloc: Towards universal visual place recognition. *IEEE Robotics and Automation Letters*, 9(2):1286–1293, 2023. 2
- [18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42(4):139–1, 2023. 2
- [19] Minjung Kim, Junseo Koo, and Gunhee Kim. Ep2p-loc: End-to-end 3d point to 2d pixel localization for large-scale visual localization. In *ICCV*, pages 21527–21537, 2023. 2
- [20] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007. 1
- [21] Viktor Larsson and contributors. PoseLib - Minimal Solvers for Camera Pose Estimation, 2020. 3
- [22] Christoph Lassner and Michael Zollhofer. Pulsar: Efficient sphere-based neural rendering. In *CVPR*, pages 1440–1449, 2021. 3
- [23] Jiaxin Li and Gim Hee Lee. Deepi2p: Image-to-point cloud registration via deep classification. In *CVPR*, pages 15960–15969, 2021. 1, 2
- [24] Minhao Li, Zheng Qin, Zhirui Gao, Renjiao Yi, Chenyang Zhu, Yulan Guo, and Kai Xu. 2d3d-matr: 2d-3d matching transformer for detection-free registration between images and point clouds. In *ICCV*, pages 14128–14138, 2023. 2
- [25] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, pages 2041–2050, 2018. 2, 4, 5
- [26] Hyon Lim, Sudipta N Sinha, Michael F Cohen, and Matthew Uyttendaele. Real-time image-based 6-dof localization in large-scale environments. In *CVPR*, pages 1043–1050. IEEE, 2012. 1
- [27] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 4
- [28] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *ICCV*, pages 17627–17638, 2023. 2, 4
- [29] Yuya Matsumoto, Gaku Nakano, and Kazumine Ogura. Indoor visual localization using point and line correspondences in dense colored point cloud. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3616–3625, 2024. 8

- [30] Sven Middelberg, Torsten Sattler, Ole Untzelmann, and Leif Kobbelt. Scalable 6-dof localization on mobile devices. In *ECCV*, pages 268–283. Springer, 2014. 1
- [31] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2
- [32] Gaku Nakano. A versatile approach for solving pnp, pnpf, and pnpfr problems. In *ECCV*, pages 338–352. Springer, 2016. 1
- [33] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. Meshloc: Mesh-based visual localization. In *ECCV*, pages 589–609. Springer, 2022. 1, 2, 5, 6, 7
- [34] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. Visual localization using imperfect 3d models from the internet. In *CVPR*, pages 13175–13186, 2023. 1, 2, 3, 5
- [35] Quang-Hieu Pham, Mikaela Angelina Uy, Binh-Son Hua, Duc Thanh Nguyen, Gemma Roig, and Sai-Kit Yeung. Lcd: Learned cross-domain descriptors for 2d-3d matching. In *AAAI*, pages 11856–11864, 2020. 2
- [36] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *ICCV*, pages 5107–5116, 2019. 4, 6
- [37] Jerome Revaud, Philippe Weinzaepfel, César Roberto de Souza, and Martin Humenberger. R2D2: repeatable and reliable detector and descriptor. In *NeurIPS*, 2019. 8
- [38] Bryan C. Russell, Josef Sivic, Jean Ponce, and H el ene Dessales. Automatic alignment of paintings and photographs depicting a 3d scene. In *ICCV Workshops*, pages 545–552, 2011. 3
- [39] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, pages 12716–12725, 2019. 1, 2, 8
- [40] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, pages 4938–4947, 2020. 4
- [41] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *CVPR*, pages 8601–8610, 2018. 6
- [42] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. 2
- [43] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015. 4
- [44] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 4
- [45] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, pages 8922–8931, 2021. 2, 3, 4, 6, 8
- [46] 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*, pages 7199–7209, 2018. 1, 8
- [47] Gabriele Trivigno, Carlo Masone, Barbara Caputo, and Torsten Sattler. The unreasonable effectiveness of pre-trained features for camera pose refinement. In *CVPR*, pages 12786–12798, 2024. 2
- [48]  nder Tuzcuođlu, Aybora K ksal, Buđra Sofu, Sinan Kalkan, and A Aydin Alatan. Xoftr: Cross-modal feature matching transformer. In *CVPR*, pages 4275–4286, 2024. 4
- [49] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *CVPR*, pages 5481–5490. IEEE, 2022. 2
- [50] Anirudh Viswanathan, Bernardo R Pires, and Daniel Huber. Vision based robot localization by ground to satellite matching in gps-denied situations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 192–198. IEEE, 2014. 5
- [51] Bing Wang, Changhao Chen, Zhaopeng Cui, Jie Qin, Chris Xiaoxuan Lu, Zhengdi Yu, Peijun Zhao, Zhen Dong, Fan Zhu, Niki Trigoni, et al. P2-net: Joint description and detection of local features for pixel and point matching. In *ICCV*, pages 16004–16013, 2021. 2
- [52] Haiping Wang, Yuan Liu, Bing Wang, Yujing Sun, Zhen Dong, Wenping Wang, and Bisheng Yang. Freereg: Image-to-point cloud registration leveraging pretrained diffusion models and monocular depth estimators. *arXiv preprint arXiv:2310.03420*, 2023. 2
- [53] Shuzhe Wang, Juho Kannala, and Daniel Barath. Dgc-gnn: Leveraging geometry and color cues for visual descriptor-free 2d-3d matching. In *CVPR*, pages 20881–20891, 2024. 2
- [54] Yuan Xiong, Jingru Wang, and Zhong Zhou. Virtualloc: large-scale visual localization using virtual images. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(3):1–19, 2023. 2
- [55] Qunjie Zhou, S ergio Agostinho, Aljoša O sep, and Laura Leal-Taix e. Is geometry enough for matching in visual localization? In *ECCV*, pages 407–425. Springer, 2022. 2
- [56] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 3