BDLoc: Global Localization from 2.5D Building Map

Hai Li Tianxing Fan Hongjia Zhai Zhaopeng Cui Hujun Bao Guofeng Zhang *

State Key Lab of CAD&CG, Zhejiang University†
{garyli, tianxingfan, zhj1999, zhpcui, baohujun, zhangguofeng}@zju.edu.cn

ABSTRACT

Robust and accurate global 6DoF localization is essential for many applications, *i.e.*, augmented reality and autonomous driving. Most existing 6DoF visual localization approaches need to build a dense texture model in advance, which is computationally extensive and almost infeasible in the global range. In this work, we propose BDLoc, a hierarchical global localization framework via the 2.5D building map, which is able to estimate the accurate pose of the query street-view image without using detailed dense 3D model and texture information. Specifically speaking, we first extract the 3D building information from the street-view image and surrounding 2.5D building map, and then solve a coarse relative pose by local to global registration. In order to improve the feature extraction, we propose a novel SPG-Net which is able to capture both local and global features. Finally, an iterative semantic alignment is applied to obtain a finner result with the differentiable rendering and the cross-view semantic constraint. Except for a coarse longitude and latitude from GPS, BDLoc doesn't need any additional information like altitude and orientation that are necessary for many previous works. We also create a large dataset to explore the performance of the 2.5D map-based localization task. Extensive experiments demonstrate the superior performance of our method.

Index Terms: Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Mixed / augmented reality; Computing methodologies—Artificial intelligence—Computer vision—Computer vision problems

1 INTRODUCTION

Estimating the precise 6DoF camera pose of a street-view image in the world coordinate is essential for the large-scale positioning services, such as sharing experience in multi-user augmented reality and motion planning in multi-robot control systems.

Most state-of-the-art 6DoF visual localization approaches employ the feature matching based strategy by finding correspondences between 2D keypoints extracted from the street-view image and the 3D dense point cloud [33, 34, 40]. All these methods require the pre-reconstructed dense texture models, which are not easy to obtain in the global range. On the contrary, the worldwide 2.5D building map has been integrated into some mapping service like Google Map, OpenStreetMap, *etc.*, for many years. Thus, in this work, we propose a 2.5D building map-based visual localization framework that eliminates the prerequisite of the pre-built high precision 3D model.

As shown in Fig. 1, the 2.5D building map, abbreviated as 2.5D map, is a special 3D map constructed by top-view building contours with heights measured by remote sensors. This process can be executed either manually or automatically from satellite maps and digital elevation models. Compared to the dense 3D map that includes a great number of faces with complex textures, the 2.5D map is composed of several texture-less faces close to the building periphery, saving massive storage memories and transmission bandwidth. However, the simplicity brings a considerable loss of details, which makes it unlikely to apply in the current visual localization pipelines.

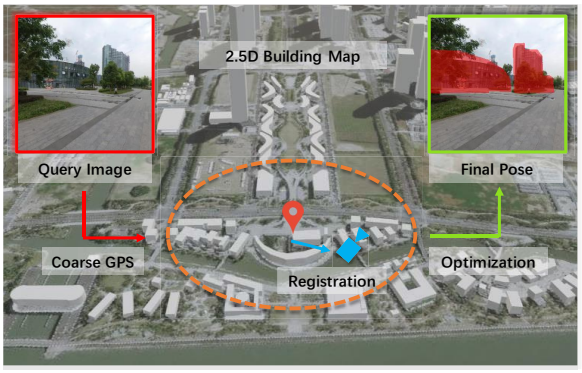


Figure 1: The 2.5D map is built upon the satellite map in the world coordinate and illustrates the rough shape of buildings. Given a query street-view image with a coarse GPS signal, the proposed BDLoc framework aims to find its global position and orientation based on 2.5D building map. The red mask in upper-right image is the reprojection of the 2.5D buildins via final pose.

In a pioneering work [4], Arth *et al.* proposed an automatic 2.5D map-based localization method which exploits semantic cues and line segment correspondences between the image and the reprojection of the 2.5D model to estimate the position and orientation by generating translation hypotheses that maximize the probability of overlap. The following work proposed by Argmagan *et al.* [3] shares a similar idea and first leverages deep learning techniques for detecting primitives and predicting moving directions. Although the above approaches achieve promising results in their experiment settings, there exist two significant limitations. Firstly, these methods require fixed height and approximate orientation, which limits the view freedom and degenerates the problem to a 3DoF pose refinement. Secondly, since the 2.5D map has no texture information and doesn't accurately match the shape with actual buildings, the cross-view geometric correspondences are unstable.

In this paper, we propose a multi-phase hierarchical 6DoF global localization framework named BDLoc. Unlike previous approaches, our method does not need prior knowledge of altitude and orientation but only a coarse longitude and latitude from GPS and get rid of the cross-view geometric 3D-2D correspondences. Specifically speaking, we first extract the building information from the street-view image and get the surrounding 2.5D building maps according to the coarse GPS information. Then we compute the coarse relative pose between 3D points from the estimated street-view depth map and the point cloud sampled from 2.5D building map. Finally, an

*Corresponding author: Guofeng Zhang
†Hai Li, Tianxing Fan, Hongjia Zhai, Hujun Bao and Guofeng Zhang are also affiliated with ZJU-SenseTime Joint Lab of 3D Vision.

978-1-6654-0158-6/21/$31.00 ©2021 IEEE
DOI 10.1109/ISMAR52148.2021.00022

optimization is applied to refine the estimated pose. However, it is nontrivial to design such a system.

At first, although we can obtain a local 2.5D building map from the coarse GPS information, it is still challenging to estimate the relative pose between the street-view image and the local 2.5D map. The recent success of 3D registration of colour-less point cloud [5, 7] shows the potential of deep networks to extract the structure feature. However, the distinctiveness of 2.5D building is relatively weak. In order to solve this problem, we exploit the advanced semantic segmentation [50] and depth estimation from monocular [20] or binocular [6] images which enable to extract 3D building information from the street-view image and thus drastically bridges the 2.5D building map and street-view image. To improve the feature extraction and matching, we propose SPG-Net, a hybrid architecture of sparse voxel convolution and spatial graph convolution to capture both local and global 3D features accompanied by an effective view-dependent graph construction strategy. In this way, we can register the street-view image with the 2.5D map roughly.

Secondly, it is hard to conduct the pose optimization with the 2.5D map and street-view image because the shared information is extremely weak. To solve this problem, we adopt the recently developed differentiable rendering techniques, which is able to bridge 3D and 2D by rendering the 3D model to the 2D image in a differentiable way and let the gradient of residuals backpropagate to all input parameters such as camera poses. We exploit the high-level semantic information for optimization and define the loss as the offset between the silhouette of building masks and the projected ones. In such way, we can optimize the camera poses without requirement of the correspondences between 2D and 3D features. With an accurate global 6DoF pose estimation from the street-view image and 2.5D building map, our method can be easily applied to the mobile augmented reality applications and generate plausible AR effects.

To summarize, our major contributions are as follows:

- We proposed BDLoc, a systematic approach for 2.5D map based global localization. To the best of our knowledge, this is the first complete 2.5D building map-based 6DoF visual localization framework.
- We propose a hybrid architecture (SPG-Net) for point clouds feature extraction. SPG-Net can model the geometric relationship between different buildings, which significantly improves the registration accuracy.
- We introduce a novel pose optimization method with the 2.5D map and street-view image via the semantic information and differentiable rendering.
- We create a large dataset to explore the performance of the 2.5D map-based localization task. The proposed method outperforms the exiting methods by a large margin.

2 Related Work

2.1 Visual Localization based on 2.5D Map

Retrieval based and 2D-3D feature matching based methods are commonly used to estimate the location of a given query image. NetVLAD [2] is the representative work of retrieval based method which utilizes a trainable VLAD layer to aggregate patch level feature. Some retrieval based methods were proposed to handle the weakly supervised information [11] and matching strategy for patch level feature [14]. Irschara *et al.* [16] propose the first 2D-3D matching method for camera localization. Liu *et al.* [21] propose an efficient camera localization approach in a large-scale 3D Map which takes the co-visibility into consideration to handle the ambiguity of local features. However, these works only depend on appearance features and thus require a large image database and a textured dense 3D model respectively, which are not suitable for 2.5D map.

2.5D map-based localization is a sub-class of geolocalization and has been studied over decades. Meierhold *et al.* [26] establish line correspondences between image and 2.5D model. But the quality of 3D line features is not good enough for localization so that user interaction is introduced in their method. Matei *et al.* [25] manually annotate the building outlines for further matching. Chu *et al.* [10] compute a descriptor from vertical building outlines in perspective input images, which is then matched with a 2.5D map. But in order to facilitate the detection of vertical edges and vanishing points. They partially use manual annotation of the input images. The above methods put forward the basic insight that the building silhouettes and planes contribute to localization but cannot run automatically.

Taneja *et al.* [37] propose another pipeline to achieve automation. Semantic segmentation is applied to extract buildings in images, and then the semantic images are iteratively updated by matching with the projected 2.5D building map through pose estimation and segmentation refinement. Similiarly, Arth *et al.* [4] use both semantic cues and line segment correspondences between the real image and projected 2.5D map for localization. And Armagan *et al.* [3] achieve the idea by CNN. Later, Liu *et al.* [22] integrate 2.5D map as a constraint in non-linear optimization by associating reconstructed points to corresponding facade. However, these line matching methods overemphasize the line segments on the contour of building masks and these features in 2.5D maps cannot always be accurate. Unlike their methods, our optimization process mainly focuses on the silhouette area, which is more robust. Besides, high-quality initialization is required by those methods.

2.2 Large-scale Point Cloud Learning

Point-based methods for point cloud learning have made progress since PointNet [28], the milestone of this field, has been proposed. PointNet uses pooling operation to aggregate point-wise feature to obtain global feature. However, the pooling operation can not extract the local feature effectively. Some approaches [19, 29, 39, 44, 46] were proposed to capture point cloud feature from local to global like the way in image-based CNN. Those methods are for general point cloud classification and segmentation tasks. For the large scale point cloud recognition task, PointNetVLAD [41] is the first work. Furthermore, some approaches [17, 24, 48] were proposed to improve the performance of large-scale point cloud recognition. PCAN [48] assigns different weights for the different point-wise feature before the aggregation operation and shows the attention map is also important for point cloud learning. LPD-Net [24] extracts the local information of the point cloud by using a dynamic and static graph to search neighbourhoods.

Besides, there are some learning-based methods [1, 18, 42] for point cloud registration that use a neural network to extract the 3D descriptor. FCGF [8] is a fully convolutional network which aims to extract local geometric feature for each point which can be used for point cloud registration. Huang *et al.* [15] propose a fast feature-metric approach which does not need search correspondences of point cloud. Compared to the traditional pairwise registration, Gojcic *et al.* [13] present an end-to-end multiview point cloud registration method. They first register pairwise point cloud and then conduct the globally consistent refinement.

3 Method

As shown in Fig. 2, the proposed BDLoc framework consists of three phases: lifting phase, registration phase and optimization phase. We take the GPS-tagged image as input in our setting. The lifting phase segments the building information from the current image and lifts it to 3D through depth prediction. Then, the registration phase retrieves the surrounding 2.5D map according to the associated GPS signals, and register the lifted 3D building points to the 2.5D map through point-wise feature matching. In this way, we can obtain a coarse pose of the street-view image. At last, the optimization

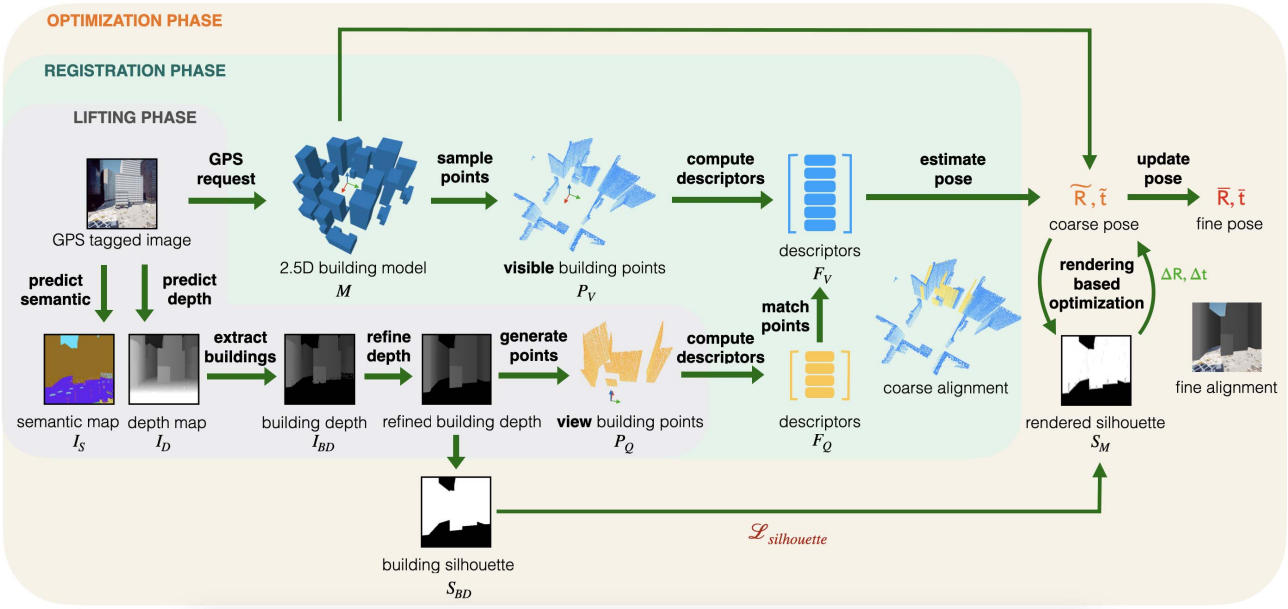


Figure 2: Overview of proposed BDLoc framework. The lifting phase (grey part) predict the semantic map and the depth map from the current image and extract the building segments for further refinement, then back project to 3D and get view building points. The registration phase (green part) first download the surrounding 2.5D building map using GPS signal and sample viewed depth points from the current position. These two point sets are fed into SPG-Net for feature extraction, followed by a coarse pose estimation via matching based alignment. Finally, the optimization phase (yellow part) iteratively optimized the pose supervised by occupancy distance of building silhouette and rendered silhouette.

phase is taken to optimize the silhouette distance of buildings in the street-view image and projected 2.5D map and refine the pose iteratively. Below we will elaborate on each phase in detail.

3.1 Lifting Phase

The primary purpose of the lifting phase is to unify the representation between the image and the map. As shown in Fig. 1, the 2.5D map consists of texture-less building models, and only contains simplified structure information. Thus, the first step is to extract the building geometry cues from the query street-view image. This process includes visual semantic segmentation [38, 50] and depth estimation [6, 20], which are two essential aspects of visual reasoning that have been extensively researched for years and shown a promising generalization capability. In this work, we take a simple U-Net architecture [30] which predicts semantic segmentation I_S and depth map I_D for a query image I_Q. Notice that these networks can be replaced by any state-of-the-art method for better performance. With pixel-wise semantic labels, we can extract corresponding building depth I_{BD} from the I_D. To form a compact and clean geometry structure, we simply filter the predicted depth [32] and lifting them to 3D point representation P_Q via depth back-projection.

3.2 Registration Phase

The buildings in the 2.5D map are usually represented as prism meshes, which consist of a few simple non-textured faces. Due to the lack of details, the ambiguity rises as the model scale increases.

To narrow the search range, we take the GPS signal as an initial position and download the surrounding building models centred at requested longitude and latitude. We also initialize a random orientation and altitude w.r.t. the requested position to establish a local coordinate system. Once the local registration is established, We can directly obtain the view pose in world coordinate via known transformation.

Matching from the image to 2.5D map via context features is not robust owing to the simplicity of the building model. Inspired by the recent success of point registration approaches, we formulate the task as a local to global 3D registration problem instead. The goal is to find the point correspondences from P_Q generated in the lifting phase to the surrounding map M. To make the representation compatible with depth points P_Q, we rendering the depth map panoramically at origin and back-project them to a full point clouds P_V. This process also reduces the points number and registration difficulty accordingly.

The essential step of 3D registration is to compute the point-wise feature for P_Q and P_V. Recently, FCGF [8] adopt a fully convolutional network to extract local geometric features for each point. The network is stacked by multiple sparse 3D convolution layers based on Minkowski Engine [9]. The operation of each layer is defined as the following,

$$x'_u = \sum_{i \in N^3(u,K,C)} W_i x_{u+i} \quad u \in C', \tag{1}$$

where $N^3(u,K,C)$ is a set of locations u where the kernel K convolution should conduct based on coordinates C, W and x denote the learnable weights and feature respectively. C' is the output coordinates w.r.t. x'. Thanks to the fully convolutional design, the extracted local feature is coordinate-agnostic. However, compared to scene-scanned point clouds, the points sampled from the 2.5D map are excessively uneven (too dense near the building surface and thoroughly empty for the other space, as shown in Fig. 4). Moreover, the rough building shapes cannot maintain the uniqueness of local geometric feature, which further degenerates the feature distinctiveness. Thus, a vanilla FCGF can not perform well in this scenarios.

SPG-Net: To further improve the feature extraction for this task, we propose Sparse-Point-Graph architecture, named SPG-Net,

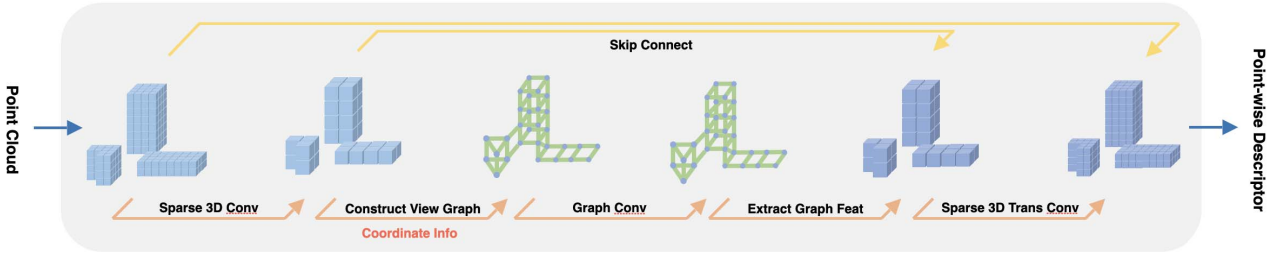
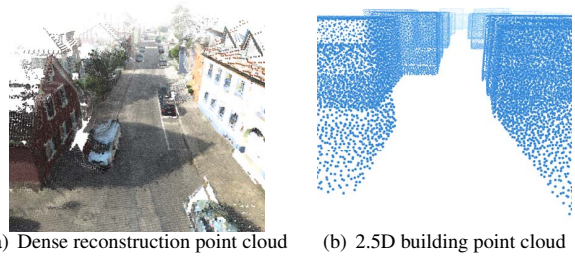


Figure 3: Structure of SPG-Net. The local geometric features are extracted by several voxel-based sparse 3D convolution layers, the global relation features are then aggregated by spatial graph convolutions through a view-dependent graph. Finally, these features are spread to each input points by transpose convolution combined with local features via skip connection.



(a) Dense reconstruction point cloud (b) 2.5D building point cloud

Figure 4: Compared with the dense reconstruction point cloud which contains rich scenes (a), the points sampled in 2.5D building maps are locally dense but globally sparse (b).

which is a hybrid point-wise feature extraction network that encodes both local geometry and global relations. As shown in Fig. 3, our SPG-Net adopts a two-step U-shape structure to compute the per-pixel feature. The sparse voxel-based convolution layers are leveraged to encode local geometric feature, and the points are becoming sparse with the depth of network increase. After that, we apply the graph convolutions based on the undirected view-graph constructed on the output coordinates of sparse 3D convolution. The graph convolution aggregates and updates feature spatially based on a message-passing paradigm [12]. The formulation is shown as the following,

$$x'_n = W_0 x_n + \frac{1}{|A(n)|} \sum_{j \in A(n)} W_1 x_j. \quad (2)$$

The updated feature x'_n for node n is the summation of feature from itself and the mean feature from the adjacent nodes $A(n)$, W_0 and W_1 denote the learnable weights. The graph convolution effectively encodes the global relation as a complementary feature and let information passes through connecting edges. It is worth noting that the output features from sparse 3D convolutions don't include coordinate information. Therefore we additionally append the 3D coordinate to geometric features before feed into the graph convolution. Finally, the mixed features are spread to each input point through sparse 3D transpose convolution layers combined with the local features via skip connection.

View-dependent Graph: Constructing a graph for global relation extraction is another critical problem. The intuitive idea is to find the nearest neighbour in Euclidean distance or embedding space [43]. However, existing methods are prone to connect locally due to the uneven point distribution in our case. To this end, we propose a view-dependent graph construction strategy. Our view-dependent graph neglects the depth interval by projecting the points to a cylinder surface centred at the origin of local coordinates. The edges are generated from projected nodes using the k nearest neighbours algorithm based on the surface distance. This linking approach is built on a reasonable assumption that the relationship between buildings would not change much within a limited position offset. In our case, the noise range of the GPS signal is stable. Thus the local message passing could mitigate the difference caused by position effectively.

Loss Function: We adopt the margin-based contrastive loss shown in Equation 3, where f_i is the first feature of positive pairs P_+, the f_+ and f_- are the corresponding positive and negative match respectively. The margin value for positive and negative pairs are m_+ and m_-.

$$\mathscr{L}_{constrastive} = \sum_{f_i \in P_+} [D(f_i, f_+) - m_+]^2 + [m_- - D(f_i, f_-)]^2. \quad (3)$$

We randomly choose one of the k nearest points for the positive match in a specific distance range and select the negative outside this range accordingly. This loss guarantees the uniqueness of the learned feature.

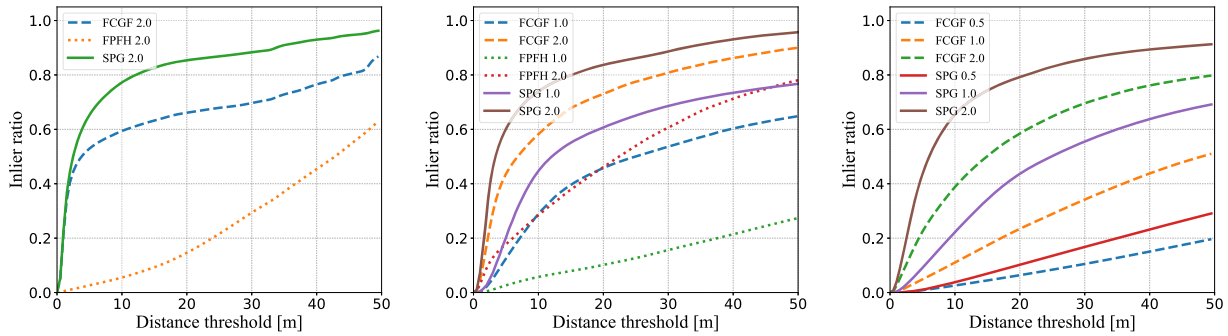
We use a shared weight SPG-Net to compute the per point feature F_Q and F_V for both P_Q and P_V. A coarse relative rotation $\widetilde{R}$ and translation $\widetilde{t}$ from local coordinate to image coordinate is then solved by the feature-matching based registration method [49].

3.3 Optimization Phase

The $\widetilde{R}$ and $\widetilde{t}$ solved from 3D point registration are not optimal as P_Q and P_V may not be strictly aligned. Therefore, we propose an refinement method which leverages the silhouettes of buildings. As shown in Fig. 2, we extract the building silhouette S_{BD} from I_{BD} as a reference and render the 2.5D building map to get the image S_M using $\widetilde{R}$ and $\widetilde{t}$.

To minimize the distance from S_M to S_{BD}, we proposed a masked optimizer based on SoftRas [23]. The mask is used to block the gradient from an invalid area which could interfere with the optimization direction. In our case, the boundary between the sky and the roof is important for judging the optimization effect, while the bottom of the building in the 2.5D map may not align well with the actual buildings. To this end, we also extract the binary mask B_M of building and sky from I_S and constrain the weight during backpropagation. The main formulation is presented in Equation 4, where σ controls the sharpness of the probability distribution, δ indicates whether pixel is inside the projected triangle or not, i and j are the indices of the pixel in the image and triangle on mesh, respectively, and $d(i,j)$ is the closest distance from the pixel to triangle edges.

$$I^i_{silhouette} = 1 - \prod_j (1 - sigmoid(\delta^i_j \cdot \frac{d^2(i,j)}{\sigma})). \quad (4)$$



(a) Synthetic Data (b) KITTI360 Data (c) Real Data

Figure 5: Feature match recall with respect to different distance thresholds.

The masked pixel-wise mean square loss is adopted for pose refinement (Equation 5).

$$\mathscr{L}_{silhouette} = \|B_M(\cdot S_{BD} - S_M)\|_2. \tag{5}$$

In each iteration, the coarse pose is updated constrained by $\mathscr{L}_{silhouette}$ to get a finer pose $\bar{R}$ and $\bar{t}$.

4 Implementation Details

We use the 10-layer U-Net architecture for depth and semantic prediction separately. The depth is represented in log space, and we mask out the infinite and empty part for better convergence. We use three labels for semantic prediction: sky, building and others. Before feeding into SPG-Net, we first downsample the points according to the specific voxel size and then apply a random rotation and translation to augment the point distribution. During training, we select one of the 3 nearest points in 3 meters as the positive match and the point farther than 12 meters as the negative. Our SPG-Net consists of 2 sparse 3D convolution layers followed by 2 graph convolution layers and then upsampled with 2 sparse transposed 3D convolution layers. We use the Adam optimizer with a learning rate of 1×10^{-4} for network optimization and a learning rate of 0.1 for silhouette optimization.

5 Experiments

In this section, we verify the effectiveness of our method through experiments on multiple datasets.

5.1 Dataset

The dataset with good alignment between the 2.5D map and street-view image is rare, and previous works [3, 4] only use dozens of images which are far from enough. As a result, in this work, we create a large dataset that contains more than 17,000 images and corresponding 2.5D buildings. For each dataset, we randomly take 4/5 for training and the remaining 1/5 for testing. Our whole dataset includes three types of data:

Synthetic Data: We use a pre-built urban model downloaded from the Internet and manually generate the 2.5D model in Unreal Engine. We first sample 10,000 views randomly and then render the corresponding image, semantic map, depth map simultaneously through the AirSim simulator [36]. We use a semantic filter to remove the view that can barely see the building and sky. This procedure guarantees the effectivenessof the training data.

KITTI360 Data: We adopt the recent published KITTI360 dataset [45] which contains 2D stereo image, semantic map and 3D model. The bounding box provided in KITTI360 uses a similar representation as 2.5D buildings. Therefore we can easily transfer it to our format by modifying the roof shape and contour. Since KITTI360 does not include ground-truth depth, we adopt a pre-trained model from [6] to predict dense disparity and triangulate depth points using the given baseline. We compare the predict depth and projected depth from 2.5D map, and select about 2,000 views with small depth errors and enough building and sky information for evaluation.

Real Data: Additionally, we create a large scale city park dataset (Fig. 1) which contains a large number of images captured by a handheld camera. The corresponding 2.5D map is downloaded from OpenStreetMap. We use the SfM technique [35] to reconstruct the whole area and compute the per image pose with respect to the SfM coordinate. We then align the reconstructed point cloud to the 2.5D map using ICP-based registration given manually annotated point pairs. We use the multi-view stereo method [47] to recover the depth map for each view, and leverage the method proposed in [38] to generate the semantic map. To mitigate the segmentation error caused by inductive bias form pre-trained model, we compute the overlapping area of projected building mask from 2.5D map and predict building labels. The data whose IOU is larger than 0.9 will be selected. Finally, about 5,000 views are selected with enough building and sky information.

5.2 Feature Match Evaluation

To prove the feature extraction ability of the proposed SPG-Net described in Sect. 3.2, we evaluate the performance of feature recall. The metric we used is shown as the following,

$$R_d = \frac{1}{M}\sum^{M}(\mathsf{T}x_i - y_j) < \tau, \tag{6}$$

where M is the total number of fragment pairs (x and y), and T is the ground truth transform matrix. This metric represents the ratio of matching points within a certain distance threshold τ.

We compare our SPG-Net with the traditional descriptor FPFH [31] and the sparse voxel-based method FCGF [8] in Synthetic and KITTI360 data. We additionally compare the performance in different voxel-size in KITTI360 data and Real data. The results are reported in Fig. 5. The vertical axis is the inlier ratio in various distance thresholds, and the horizontal axis is the corresponding distance. From (a) and (b), we can find that our SPG-Net outperforms FCGF nearly 20 percent on a 10-meter matching distance while FPFH performs poorly in such point distribution. Since SPG-Net and FCGF utilize the similar sparse voxel convolution for local geometry feature extraction, we conclude that the performance improvement mainly comes from the proposed view-dependent graph convolution strategy. From (b) and (c), we can find that the inlier ratio boosts as the voxel size increases. We conclude that the large voxel size makes the network more conducive to learning the global relationship features, which is more critical in such extreme distribution.

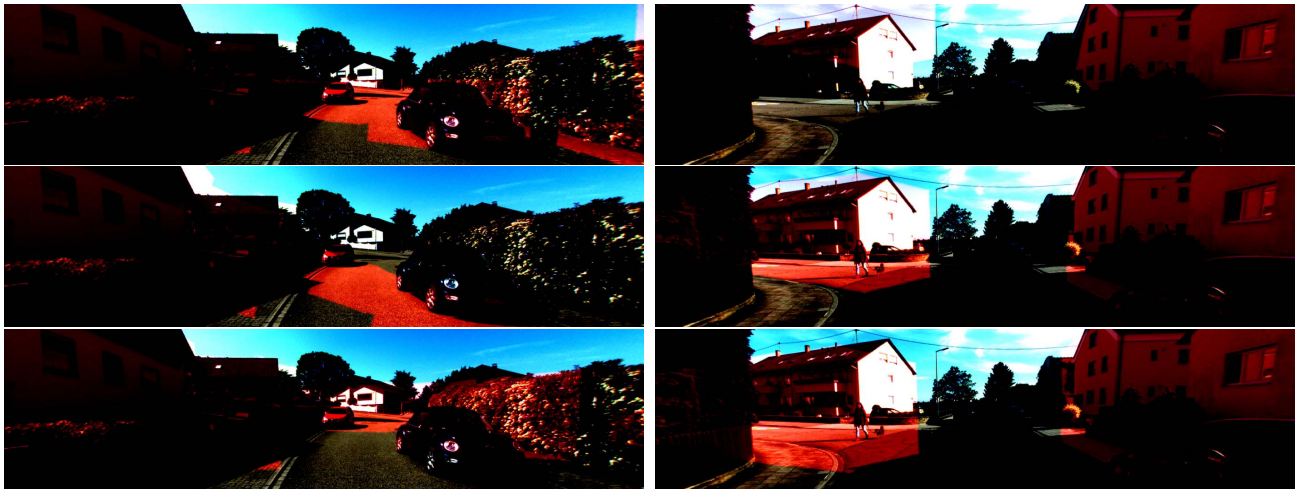


Figure 6: Qualitative results of KITTI360 Data, from top to bottom are the AR results of the initial pose, optimized pose after 10 iteration and finial pose.

Table 1: Quantitative results of pose refinement in constrained setting.

Method	RRE [deg]	RTE [m]
2.5D Align [3]	2.48	3.43
Mask-SoftRas (Ours)	**1.12**	**2.97**

Table 2: Evaluation of lifting phase.

Data Type	log RMSE	mean IOU
Synthetic	0.1418	0.9669
KITTI360	0.7951	0.9146
Real	0.9387	-

5.3 Optimization Evaluation

We then conduct an experiment to evaluate the accuracy of our pose refinement method, abbreviated as Mask-SoftRas. We adopt the same constrained setting in 2.5D Align network [3]: pre-fix the height and provide an initial orientation ($<$ 15 deg) and position ($<$ 10 m). We try our best to re-implement the 2.5D Align network as our main comparison method. Our data contains more complex building shapes than their data, which are hard to annotate edges and facades manually. Therefore, we feed the 2.5D Align network with the semantic segmentation from our network directly and predict the moving direction from classification. We then compare the optimization result of ours and 2.5D Align using Relative Translation Error (Equation 7) and Relative Rotation Error (Equation 8) to measure localization accuracy of our framework as follows,

$$RTE = |t - t^*|, \tag{7}$$

$$RRE = \arccos((trace(R^T R^*) - 1)/2), \tag{8}$$

where R^* and t^* are the ground-truth rotation and translation, while R and t are the estimated ones.

The quantitative results on Synthetic dataset are shown in Table 1. Even without training, our Mask-SoftRas method still outperforms 2.5D Align, which means that with good initialization condition, Mask-SoftRas can generalize well in similar tasks.

5.4 Localization Evaluation

Finally, we evaluate the localization accuracy of the complete framework. We randomly select the number within 10 meters to simulate GPS signals and get the corresponding 2.5D map. The map coordinate system is then established on a random orientation.

For Synthetic data, the semantic segmentation and depth estimation networks are supervised by the ground truth. For KITTI360 data, we directly use the estimated depth from [6] and the given baseline, and train the semantic segmentation network. For Real data, we use the predicted semantic from [38] and train the depth estimation network. During registration, we down-sample P_V and P_Q to voxel-size of 2 meters and only use the points within 200 meters for feature extraction and matching.

The qualitative results of registration and iterative optimization during optimization phase are shown in Fig. 6, Fig. 7 and Fig. 8. We apply the Mask-SoftRas optimization for over 200 iterations. From the visualization, we can see that the projected buildings (red mask) are gradually aligned to the outline of the actual buildings.

We report the mean IOU of building segmentation as well as log RMSE of predicted depth and projected depth of 2.5D map from the lifting phase in Table 2. The localization results also reported in Table 3. We use inlier ratio in a specific distance and orientation threshold as our metric and compare the result of BDLoc from the registration phase and the optimization phase with the 2.5D Align in different combinations of distance and orientation thresholds. From the results, we can see that precise semantic segmentation and depth prediction will improve the accuracy of localization, however the actual scene will also affect the final result. The buildings in KITTI360 data are relatively close and dense which is more sensitive to feature matching while the buildings in Real data are far and sparse. Therefore, although the depth error is larger than KITTI360 data, our method achieves better performance in Real data. The localization result also prove the effectiveness of our optimization strategy. Due to the reconstruction noise and inaccurate 2.5D building map, the performance on the real-world data is not as good as that on the synthetic data. Even though, our method still outperforms others by a large margin.

6 Application for Augmented Reality

Accurate global localization is very important for augmented reality applications. Many AR applications need two stages, *i.e.*, offline

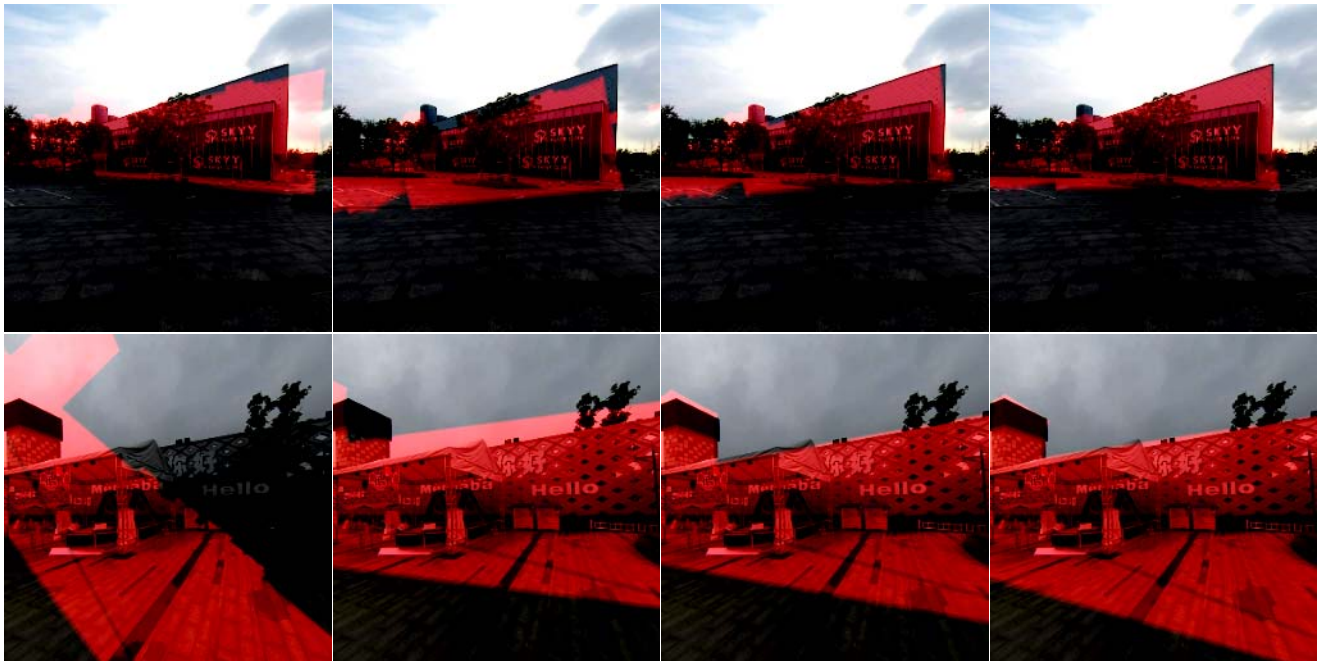


Figure 7: Qualitative results of Real Data, from left to right are the AR results of the initial pose, optimized pose after 10 iterations, pose after 50 iterations and finial pose.

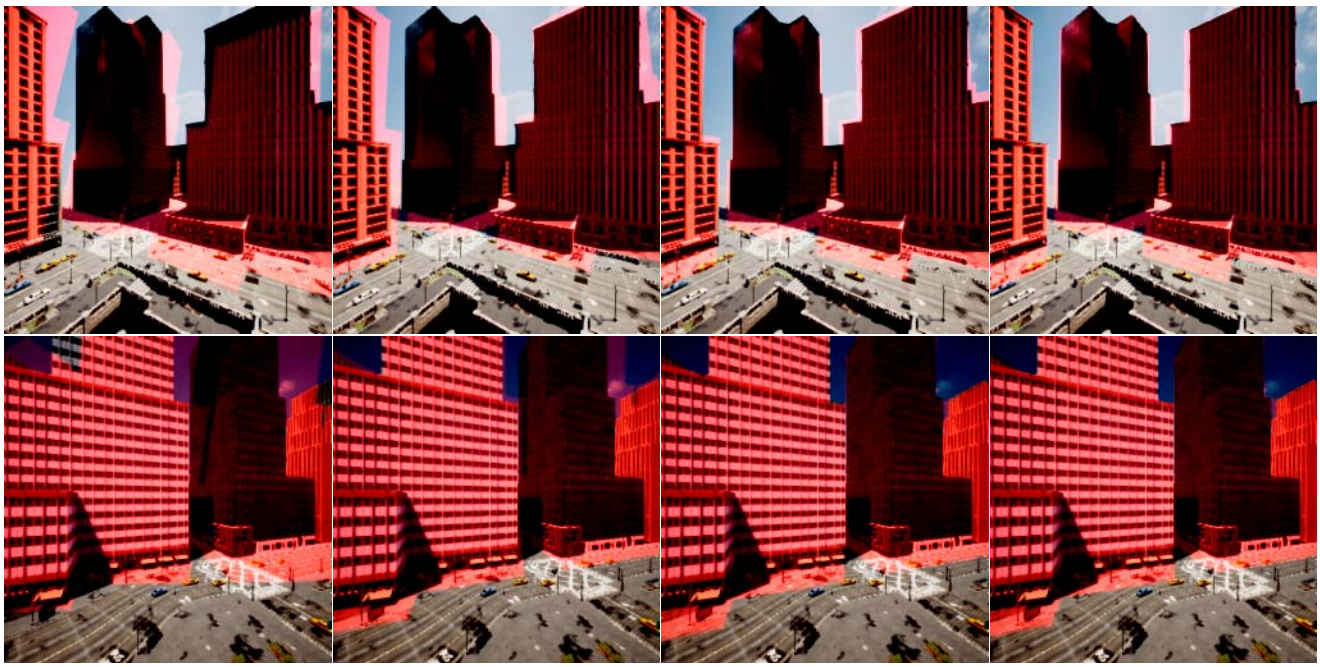


Figure 8: Qualitative results of Synthetic Data, from left to right are the AR results of the initial pose, optimized pose after 10 iterations, pose after 50 iterations and finial pose.

Table 3: Quantative results of localization.

	Synthetic	KITTI360	Real
distance [m]	2.5 / 5.0 / 7.5	2.5 / 5.0 / 7.5	2.5 / 5.0 / 7.5
orient. [deg]	5 / 10 / 15	5 / 10 / 15	5 / 10 / 15
2.5D-Align [3]	1.69 / 13.93 / 39.43	1.70 / 14.52 / 43.16	1.60 / 10.44 / 33.36
Ours(Registration Phase)	16.94 / 50.50 / 89.65	2.55 / 26.80 / 52.34	5.27 / 29.20 / 57.93
Ours(Optimization Phase)	**63.65 / 89.44 / 97.15**	**3.76 / 30.12 / 56.48**	**5.56 / 31.44 / 61.43**

Figure 9: Augmented reality results with different perspectives.

Figure 10: Augmented reality results of SLAM with BDLoc based relocalization.

stage and online stage. In the offline stage, the virtual objects need to be placed in the desired positions in the world coordinate system. Then in the online stage, we can run SLAM and use a global localization method to estimate the global pose of one frame and compute the transformation between the world coordinate system and the coordinate system of SLAM. Then the estimated 6DoF poses by SLAM can be aligned into the world coordinate and the virtual objects can be accurately composited into the live video stream.

We first show an example that we directly use the estimated 6DoF poses by our method for augmented reality without SLAM, as shown in Fig. 9. We took two pairs of images with different perspectives from synthetic data and real data respectively and use our BDLoc method to estimate the corresponding pose for each image. Although there are a little jittering, the ISMAR 2021 logo faithfully appears in the desired position for most frames.

Fig. 10 shows another AR example by combining our localization method and ORB-SLAM2 [27]. We run the ORB-SLAM2 on KITTI360 data and activate the global localization simultaneously. When enough building detected in the current view, our BDLoc start to estimate the relative transformation from local to global and register the trajectory into global coordinate. Thus, we can show the virtual objects placed in global coordinate at offline stage. The complete video is provided in the supplementary materials.

7 Limitation and Future Work

Currently, our method still has some limitations. First, it is still a challenge to handle the buildings with complex shapes for our method due to the geometry information loss in the 2.5D map. Second, there are very few datasets that can be used for 2.5D based visual localization task, and therefore our network is only trained on the specific dataset. When adopted to totally different scenes, the inaccuracy of depth and semantic prediction may cause large localization errors. But compared to the existing appearance based localization strategies, our method does not need to build high-precision maps frequently which greatly improves the reusability of maps and robustness to appearance change. Besides, in practical applications, our method can be used in a certain scale of scenes with only part of the scene data for training. To further improve the robustness to various scenarios, we consider to use sequence images for 2.5D-based localization and improve localization accuracy through spatio-temporal consistency constraints.

8 Conclusion

We propose a global localization framework based on the 2.5D building map. Our framework solves the 6DoF position and orientation only using a coarse GPS signal as an initialization. With extensive experiments, we show that our BDLoc achieves better performance compared to other methods in each stage, and leads to satisfied results in synthetic and real data. We also show that our method can be directly applied to AR applications with accurate 6DoF pose estimates.

Acknowledgments

The authors would like to thank Zhuang Zhang and Hanqing Jiang for their help in generating the 3D point cloud and depth maps of the real data. This work was partially supported by NSF of China (Nos. 61822310 and 61932003).

REFERENCES

[1] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey. PointNetLK: Robust & efficient point cloud registration using pointnet. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7163–7172, 2019.

[2] R. Arandjelovic, P. Gronát, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6):1437–1451, 2018.

[3] A. Armagan, M. Hirzer, P. M. Roth, and V. Lepetit. Learning to align semantic segmentation and 2.5D maps for geolocalization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4590–4597, 2017.

[4] C. Arth, C. Pirchheim, J. Ventura, D. Schmalstieg, and V. Lepetit. Instant outdoor localization and SLAM initialization from 2.5D maps. *IEEE Trans. Vis. Comput. Graph.*, 21(11):1309–1318, 2015.

[5] X. Bai, Z. Luo, L. Zhou, H. Fu, L. Quan, and C. Tai. D3Feat: Joint learning of dense detection and description of 3D local features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6358–6366, 2020.

[6] X. Cheng, Y. Zhong, M. Harandi, Y. Dai, X. Chang, H. Li, T. Drummond, and Z. Ge. Hierarchical neural architecture search for deep stereo matching. In *Proceedings of Annual Conference on Neural Information Processing Systems*, 2020.

[7] C. Choy, W. Dong, and V. Koltun. Deep global registration. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2511–2520, 2020.

[8] C. Choy, J. Park, and V. Koltun. Fully convolutional geometric features. In *Proceedings of IEEE International Conference on Computer Vision*, pp. 8957–8965, 2019.

[9] C. B. Choy, J. Gwak, and S. Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3075–3084, 2019.

[10] H. Chu, A. C. Gallagher, and T. Chen. GPS refinement and camera orientation estimation from a single image and a 2D map. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 171–178, 2014.

[11] Y. Ge, H. Wang, F. Zhu, R. Zhao, and H. Li. Self-supervising fine-grained region similarities for large-scale image localization. In *Proceedings of European Conference on Computer Vision*, vol. 12349, pp. 369–386, 2020.

[12] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, vol. 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272, 2017.

[13] Z. Gojcic, C. Zhou, J. D. Wegner, L. J. Guibas, and T. Birdal. Learning multiview 3D point cloud registration. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1756–1766, 2020.

[14] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer. Patch-NetVLAD: Multi-scale fusion of locally-global descriptors for place recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[15] X. Huang, G. Mei, and J. Zhang. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11363–11371, 2020.

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

[17] J. Komorowski. Minkloc3D: Point cloud based large-scale place recognition. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pp. 1790–1799, 2021.

[18] H. M. Le, T. Do, T. Hoang, and N. Cheung. SDRSAC: Semidefinite-based randomized approach for robust point cloud registration without correspondences. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 124–133, 2019.

[19] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. PointCNN: Convolution on X-transformed points. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 828–838, 2018.

[20] Z. Li and N. Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2041–2050, 2018.

[21] L. Liu, H. Li, and Y. Dai. Efficient global 2D-3D matching for camera localization in a large-scale 3D map. In *Proceedings of IEEE International Conference on Computer Vision*, pp. 2391–2400, 2017.

[22] R. Liu, J. Zhang, S. Chen, and C. Arth. Towards SLAM-based outdoor localization using poor GPS and 2.5D building models. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality*, pp. 1–7, 2019.

[23] S. Liu, W. Chen, T. Li, and H. Li. Soft rasterizer: A differentiable renderer for image-based 3D reasoning. In *Proceedings of IEEE International Conference on Computer Vision*, pp. 7707–7716, 2019.

[24] Z. Liu, S. Zhou, C. Suo, P. Yin, W. Chen, H. Wang, H. Li, and Y. Liu. LPD-Net: 3D point cloud learning for large-scale place recognition and environment analysis. In *Proceedings of IEEE International Conference on Computer Vision*, pp. 2831–2840, 2019.

[25] B. C. Matei, N. V. Valk, Z. Zhu, H. Cheng, and H. S. Sawhney. Image to LIDAR matching for geotagging in urban environments. In *Proceedings of IEEE Workshop on Applications of Computer Vision*, pp. 413–420, 2013.

[26] N. Meierhold, A. Bienert, and A. Schmich. Line-based referencing between images and laser scanner data for image-based point cloud interpretation in a cad-environment. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37, 01 2008.

[27] R. Mur-Artal and J. D. Tardós. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Trans. Robotics*, 33(5):1255–1262, 2017.

[28] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 77–85, 2017.

[29] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of Annual Conference on Neural Information Processing Systems*, pp. 5099–5108, 2017.

[30] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proceedings of Medical Image Computing and Computer-Assisted Intervention*, vol. 9351, pp. 234–241, 2015.

[31] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3D registration. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 3212–3217, 2009.

[32] K. Sanford. Smoothing kinect depth frames in real-time. *Digital image. CodeProject*, 24, 2012.

[33] P. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12716–12725, 2019.

[34] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(9):1744–1756, 2017.

[35] J. L. Schönberger and J.-M. Frahm. Structure-from-Motion Revisited. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[36] S. Shah, D. Dey, C. Lovett, and A. Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.

[37] A. Taneja, L. Ballan, and M. Pollefeys. Registration of spherical panoramic images with cadastral 3D models. In *Proceedings of International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission,*, pp. 479–486, 2012.

[38] A. Tao, K. Sapra, and B. Catanzaro. Hierarchical multi-scale attention for semantic segmentation. *arXiv preprint arXiv:2005.10821*, 2020.

[39] H. Thomas, C. R. Qi, J. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas. KPConv: Flexible and deformable convolution for point clouds. In *Proceedings of IEEE International Conference on Computer*

Vision, pp. 6410–6419, 2019.

[40] C. Toft, E. Stenborg, L. Hammarstrand, L. Brynte, M. Pollefeys, T. Sattler, and F. Kahl. Semantic match consistency for long-term visual localization. In *Proceedings of European Conference on Computer Vision*, vol. 11206, pp. 391–408.

[41] M. A. Uy and G. H. Lee. PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4470–4479, 2018.

[42] J. Vongkulbhisal, B. I. Ugalde, F. D. la Torre, and J. P. Costeira. Inverse composition discriminative optimization for point cloud registration. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2993–3001, 2018.

[43] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.*, 38(5):146:1–146:12, 2019.

[44] W. Wu, Z. Qi, and F. Li. PointConv: Deep convolutional networks on 3D point clouds. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9621–9630, 2019.

[45] J. Xie, M. Kiefel, M.-T. Sun, and A. Geiger. Semantic instance annotation of street scenes by 3D to 2D label transfer. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[46] Q. Xu, X. Sun, C. Wu, P. Wang, and U. Neumann. Grid-GCN for fast and scalable point cloud learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5660–5669, 2020.

[47] Q. Xu and W. Tao. Multi-scale geometric consistency guided multi-view stereo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5483–5492, 2019.

[48] W. Zhang and C. Xiao. PCAN: 3D attention map learning using contextual information for point cloud based retrieval. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12436–12445, 2019.

[49] Q. Zhou, J. Park, and V. Koltun. Fast global registration. In *Proceedings of European Conference on Computer Vision*, vol. 9906, pp. 766–782, 2016.

[50] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. D. Newsam, A. Tao, and B. Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition,*, pp. 8856–8865, 2019.