CROSSLOC3D: Aerial-Ground Cross-Source 3D Place Recognition

Tianrui Guan[1] Aswath Muthuselvam[1] Montana Hoover[1] Xijun Wang [1]
Jing Liang [1] Adarsh Jagan Sathyamoorthy [1] Damon Conover [2] Dinesh Manocha[1]

[1]University of Maryland, College Park [2]DEVCOM Army Research Laboratory

Abstract

We present CROSSLOC3D*, a novel 3D place recognition method that solves a large-scale point matching problem in a cross-source setting. Cross-source point cloud data corresponds to point sets captured by depth sensors with different accuracies or from different distances and perspectives. We address the challenges in terms of developing 3D place recognition methods that account for the representation gap between points captured by different sources. Our method handles cross-source data by utilizing multi-grained features and selecting convolution kernel sizes that correspond to most prominent features. Inspired by the diffusion models, our method uses a novel iterative refinement process that gradually shifts the embedding spaces from different sources to a single canonical space for better metric learning. In addition, we present* CS-CAMPUS3D*, the first 3D aerial-ground cross-source dataset consisting of point cloud data from both aerial and ground LiDAR scans. The point clouds in* CS-CAMPUS3D *have representation gaps and other features like different views, point densities, and noise patterns. We show that our* CROSSLOC3D *algorithm can achieve an improvement of 4.74% - 15.37% in terms of the top 1 average recall on our* CS-CAMPUS3D *benchmark and achieves performance comparable to state-of-the-art 3D place recognition method on the Oxford RobotCar. The code and* CS-CAMPUS3D *benchmark will be available at github.com/rayguan97/crossloc3d .*

1. Introduction

Place recognition is an important problem in computer vision, with a wide range of applications including SLAM [21], autonomous driving [6], and robot navigation [25], especially in a GPS-denied region. Given a point cloud query, the place recognition task will predict an embedding such that the query will be close to the most structurally similar point clouds in the embedding space. The goal of place recognition is to compress information from a database with a location tag and find the closest data point from the database given a query, which is essentially an information retrieval task on a global scale.

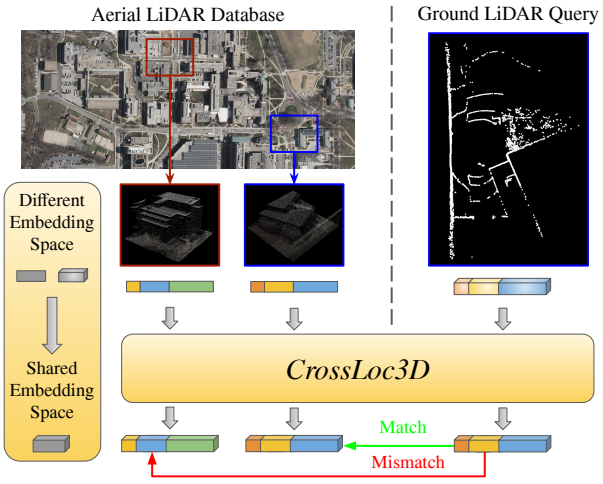

Figure 1: **CROSSLOC3D:** Our method processes cross-source point clouds into a better, shared embedding, which achieves a better retrieval outcome in a cross-source setting.

In this paper, we deal with the problem of 3D cross-source place recognition, as illustrated in Fig. 1. The goal of 3D cross-source place recognition in the context of aerial views and ground views is extremely challenging and not well-studied in the field [13]. There are two aspects of cross-source data that result in additional challenges: cross-view and data consistency. First, the perspective differences between aerial and ground datasets would cause partial overlap of the scans, leading to a lack of point correspondences. Second, there may be no data consistency between sources, which will cause representation gaps. The challenge of using multiple sources of data is primarily related to the different nature and fidelity of sensors. It is harder to match the 3D point clouds from different sources [13] than it is to perform similar matches between the 2D images captured from the same locations, or even point clouds captured only by ground or aerial sensors. When different kinds of LiDAR sensors or the same kind of LiDAR at different distances, capture scans at the same location, the data is significantly different. 3D cross-source data is point

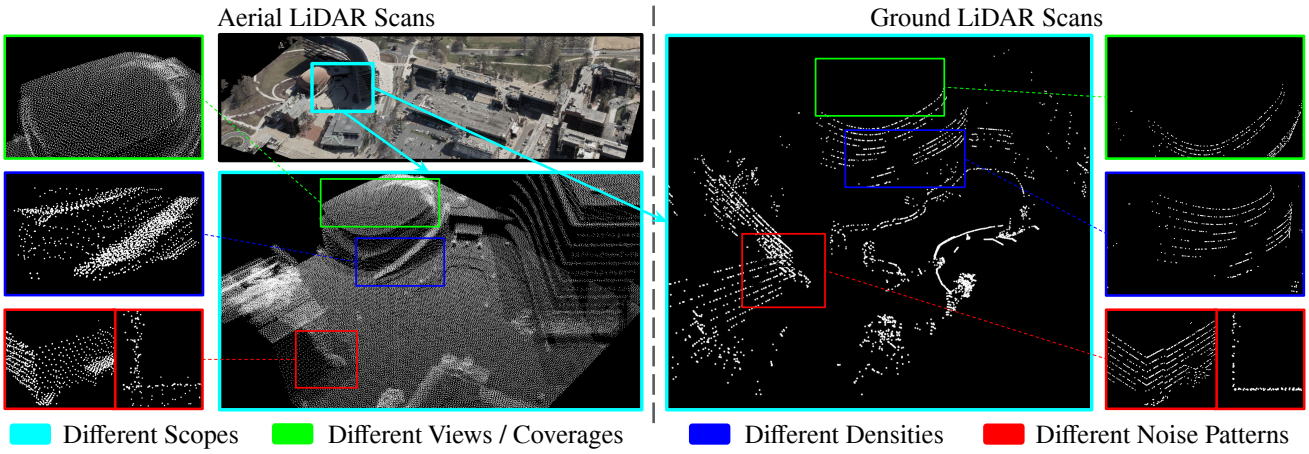

Figure 2: **Representation gap between aerial and ground sources:** We use the bounding box with the same color to focus on the same region and highlight the differences between aerial (left) and ground (right) LiDAR scans. **Scopes (cyan):** The aerial scans cover a large region, while ground scans cover only a local area. **Coverages (green):** The aerial scans cover the top of the buildings, while ground scans cover more details on the ground. **Densities (blue):** The distribution and density of the points are different because of various scan patterns, effective ranges, and fidelity of LiDARs. **Noise Patterns (red):** The aerial scans have larger noises, as we can see from a bird-eye view and top-down view of a corner of the building.

cloud data captured by different depth sensors and differ significantly in terms of scope, coverage, point density, and noise distribution pattern, as shown in Fig. 2. We need a method that can better close the representation gap between the two sources by converting features in different embedding spaces to a common embedding space.

While the problems of localization and point-set registrations are well-studied, there is relatively less work on cross-source localization. Ge et al. [33] propose a localization method using point cloud data from both the air and the ground sensors. However, this method relies on semantic information for accurate 2D template matching; in addition, their data is private. To the best of our knowledge, there are no existing works that only utilize raw point information obtained from both ground and aerial LiDAR sensors and there are no known open datasets for such applications.

Main Results: In this paper, we propose a novel 3D place recognition method that works well on both single-source and cross-source point cloud data. We account for the representation gap by using multi-grained features and selecting appropriate convolution kernel sizes. Our approach is inspired by the diffusion model [11, 23] and cold diffusion [4] and we propose a novel iterative process to refine multi-grained features from coarse to fine. We also propose a novel benchmark dataset that consists of point cloud data from both ground and aerial views. The novel aspects of our approach include:

1. We propose CROSSLOC3D, a novel place recognition method that utilizes multi-grained voxelization and multi-scale sparse convolution with a feature selection module to actively choose useful features and close the representation gap between different data sources.
2. We propose an iterative refinement process that shifts the feature distributions of various input sources toward a canonical latent space. We show that starting the refinement process from the coarsest features, which are most similar across different sources, toward the finer features, leads to better recall compared to doing it in the reverse order or simply concatenating features from different resolutions.
3. We present the first public 3D Aerial-Ground Cross-Source benchmark in a campus environment, CS-CAMPUS3D, which consists of both aerial and ground LiDAR scans. We collect ground LiDAR data on mobile robots and process the aerial data from the state government into a suitable format to cross-reference the ground data. The dataset and code will be publicly released and made available for benchmark purposes.

We have evaluated CROSSLOC3D and other state-of-the-art 3D place recognition methods on the CS-Campus3D dataset and observe an improvement of 4.74% - 15.37% in terms of the top 1 average recall. Additionally, we observe that CROSSLOC3D achieves close to 99% in terms of top 1% average recall on Oxford RobotCar [20] and performs comparably, within a margin of 0.31%, to the SOTA methods on the traditional single-source 3D place recognition task.

2. Related Work

2.1. 2D and 3D Place Recognition

Place recognition is finding a match for a query position among a database of locations based on its traits and features. Image-based place recognition [36] has been

well-studied using Vector of Locally Aggregated Descriptor (VLAD) approaches [16, 3, 2] and has been extended into 3D domains. As the first point-based method for large-scale place recognition, PointNetVLAD [27] uses PointNet [22] as a feature extractor for the NetVLAD [2] and has been evaluated on the Oxford RobotCar [20] dataset. Since the creation of this benchmark, many other methods have been developed with improved performance [7, 17, 30, 8, 15].

2.2. Multi-View Localization

Multi-view localization utilizes different perspectives as inputs for localization and has been widely studied, especially in the context of image-based geo-localization [38, 26, 34, 37, 9, 31]. Gawel et al. [9] use semantically segmented images from both the air and the ground to build a semantic graph for localization on a global map. Xia et al. [31] approach such cross-view localization tasks from the probabilistic perspective on the satellite image instead of as an image retrieval objective. [24] proposes a method to combine local LiDAR sensors with an occupancy map obtained from overhead satellite imagery for place recognition and pose estimation. In addition, Ge et al. [33] convert both aerial and ground LiDAR into 2D maps and utilize semantic and geometric information for 2D template matching.

2.3. 3D Cross-Source Matching

3D cross-source matching has been studied under the context of point-level registration. Huang et al. [13] present a 3D cross-source registration dataset consisting of indoor point clouds captured from either Kinect or LiDAR or calculated from 3D reconstruction algorithms from 2D images [28] as different sources. This benchmark poses many challenges due to noise, partial overlap, density difference, and scale difference. To deal with the lack of correspondences caused by noise and density differences, Huang et al. [12] propose a semi-supervised approach to improve the robustness of the registration. In the presence of partial overlap and scale difference of the two matching point clouds, Huang et al. [14] propose a coarse-to-fine algorithm to gradually finalize the candidate of the match. In our context, the challenges are similar in terms of the points representation gap, but our problem introduces more difficulties due to the discrepancy caused by perspective differences in the data captured from aerial and ground viewpoints. In addition, the existing cross-source benchmark [13] focuses on point-level registration of indoor objects, while our benchmark focuses on outdoor large-scale place recognition.

3. Problem Definition

Let query $\mathcal{Q}_g$ be a set of 3D points in a single frame captured by a LiDAR sensor L_g from the ground represented in its relative coordinate. The range radius of L_g is r.

Let $\mathcal{M}_a$ be a large-scale 3D point cloud map consisting of multiple frames captured by a LiDAR sensor L_a from the aerial viewpoint. The global point cloud map $\mathcal{M}_a$ is divided into a collection of M overlapping submaps $\mathcal{D}_a = \{m_1, m_2, ...m_M\}$, where the center of the submaps is uniformly spaced at a distance of d and with an area of coverage (AOC) of A. The parameters d and A are chosen so that $|m_i| \ll |\mathcal{M}_a|$ and $A \approx \pi r^2$.

Definition. *Given a query $\mathcal{Q}_g$ and a database $\mathcal{D}_a$, the goal is to retrieve the submap $m_i \in \mathcal{D}_a$ whose coverage includes the location L at which $\mathcal{Q}_g$ is captured in map $\mathcal{M}_a$.*

Approach. Our network takes a set of points $\mathcal{Q}_g$ as input and learns a function $f(.)$ that maps $\mathcal{Q}_g$ to a feature vector v_g. For $m_i, m_j \in \mathcal{D}_a$, we expect $f(.)$ to satisfy $d(f(\mathcal{Q}_g), f(m_i)) < d(f(\mathcal{Q}_g), f(m_j))$ if $\mathcal{Q}_g$ is structurally similar to m_i but dissimilar to m_j. The 3D place recognition task can be defined as finding the submap $m_* \in \mathcal{D}_a$ such that $d(f(\mathcal{Q}_g), f(m_*)) < d(f(\mathcal{Q}_g), f(m_i)), \forall i \neq *$.

Unique Setting. Compared to previous 3D place recognition tasks in which all data are collected from the same platform, our data are mixed with two different sources from both aerial scans and ground scans, which leads to cross-localization problems. This setting poses a unique challenge due to the differences in cross-source data as indicated in Fig. 2. Under this setting, most existing 3D place recognition methods [27, 19, 39, 32] lead to poor performance and low recall, as shown in Table. 3.

4. Method

4.1. Overview

Our network, CROSSLOC3D, takes a point cloud as input and outputs a global descriptor such that two sets of structurally similar points have closer global descriptors. Our proposed architecture consists of three stages: 1) multi-grained feature extraction with feature selection, 2) an iterative refinement model accounting for distribution shift between different sources, and 3) a NetVLAD [2], as shown in Fig. 3. The network needs to approximate a function with inputs from both ground and aerial sources, which do not have any exact point correspondence or similar noise distributions. During training, we use lazy triplet margin loss [27], which is commonly used for place recognition.

4.2. Multi-grained Features with Selection

We process the point cloud patches into a uniform size and extract their multi-scale features at different voxelization sizes. The purpose of the module is to accommodate data from different sources by obtaining features in different granularities and selecting their respective dominant features. Let $p \in \{\{\mathcal{Q}_g\}, \mathcal{D}_a\} \subseteq \mathbb{R}^{n\times 3}$ be a patch from either ground or aerial sources and down-sampled to size n.

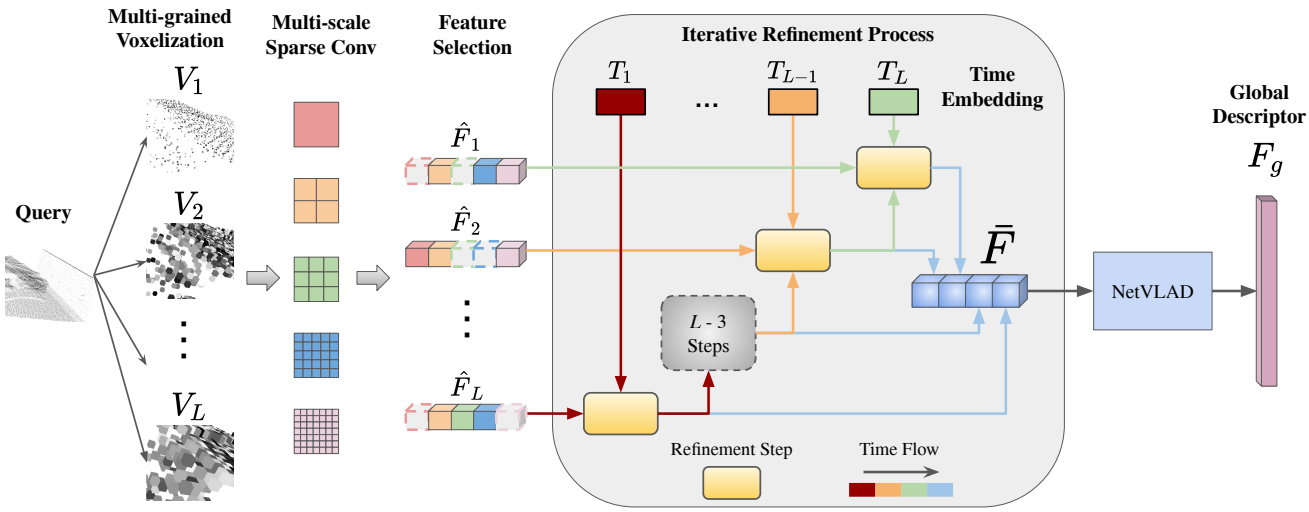

Figure 3: **Architecture of our network CROSSLOC3D:** Our network consists of three parts: multi-grained feature selection, iterative refinement, and NetVLAD. First, we run the points through multi-grained voxelizations and several streams of sparse convolution operations with different kernel sizes. We use feature selection to choose the best field of view and keep the corresponding features $\hat{F}_i$ for each voxelization V_i. Second, we perform an iterative refinement process to obtain a set of local features $\bar{F}$. Finally, the NetVLAD [2] module will produce the global descriptor F_g by aggregating local features $\bar{F}$.

Multi-grained Features: First, we perform L voxelization on p with different resolutions $\{r_1, r_2, ..., r_L | \ \forall\ 0 \leq i < j < L, r_i < r_j\}$ and obtain a list of voxel sets $V_1, V_2, ..., V_L$. A series of sparse convolution operations, batch normalization, and activation is performed on each voxel set V_i. After that, we use sparse convolution with 5 different field of viwes $1, 3, 5, 7, 9$ to obtain a multi-scale feature $F_i \in \mathbb{R}^{5 \times N_i \times D_v}$ from V_i, where N_i is the number of voxels in V_i and D_v is the feature dimension.

Feature Selection: Given the multi-scale feature F_i, where $i = 1, 2, ..., L$ for each voxel grain, we may have some redundant information from different quantization sizes and scopes of the kernels. In addition, when the input points are from one source, a specific quantization size coupled with a specific field of view could work better when the input is from another source. As a result, we add another feature selection module that can choose the best field of view for each feature F_i based on a confidence score. We can obtain the feature after selection:

$$\hat{F}_i = F_i[\ topk(\ softmax(\ g(F_i),\ dim = 0),\ k = k_s)\ ], \quad (1)$$

where $g(.)$ is a learnable function by a few layers of convolutions, k_s is the number of features to select ($0 < k_s \leq 5$), and $\hat{F}_i \in \mathbb{R}^{k_s \times N_i \times D_r}$ for some feature size D_r.

Using a large voxel size can lose information, but it provides some flexibility in matching when one source has a different noise pattern or density from another source. With feature selection, we want to keep the best information obtained from multi-grained quantizations and multi-scale kernels.

Note that during inference, the database can be computed ahead of time, and only the query point needs to be passed through the network. The increase in run-time can be negligible given the amount of improvement from the proposed CROSSLOC3D, as shown in Table. 4.

4.3. Iterative Refinement

Inspired by diffusion model [11, 23, 4], we hope to close the representation gap between different sources, by gradually learning a distribution shift towards a canonical space at each refinement step in the training process. Let $\hat{F} = permute([\hat{F}_1, \hat{F}_2, ..., \hat{F}_L]))$, where $\hat{F}[i] \in \mathbb{R}^{k_s \times \hat{N}_i \times D_r}$ and $permute(.)$ gives control over the order of refinement process. Note that only when the original order is maintained, $N_i = \hat{N}_i$ for all i. For simplicity of the notations, we assume the ascending order of $\hat{F}$ and $N_i = \hat{N}_i$.

Let $r_i(.) : A \rightarrow A$ be a refinement function at step i, where L is the total number of steps for the refinement and $A \subseteq \mathbb{R}^{k_s \times * \times D_r}$. Let $\tilde{F}_1 = r(\hat{F}[1])$ be the first step of the refinement, then for $1 < i \leq L$, $\tilde{F}_i$ is defined as follows:

$$\tilde{F}_i = r(concate(\big[\tilde{F}_{i-1}, \hat{F}[i]\big],\ dim = 1)), \quad (2)$$

where $\tilde{F}_i \in \mathbb{R}^{k_s \times \sum_{j=1}^{i} N_j \times D_r}$.

The final output of the refinement at step i is defined as:

$$\bar{F}_i = concate(\big[\tilde{F}_i, \hat{F}[i+1], ..., \hat{F}[L]\big],\ dim = 1) \quad (3)$$

After L steps of iterative refinement, we can obtain $\bar{F}_i \in \mathbb{R}^{k_s \times N \times D_r}$, where $N = \sum_{j=1}^{L} N_i$. We concatenate all $\bar{F}_i$ along the feature dimension and reshape it to get $\bar{F} \in \mathbb{R}^{N \times D_g}$, where $D_g = k_s \times D_r \times L$.

Refinement Function: $r(.)$ function is implemented with a series of External Attention (EA) [10] blocks, where each block is considered a substep of the refinement. Instead of using features to generate query, key, and value tensors, the

EA block only uses a single stream for query input and uses internal memory units to generate attention maps, which leads to strong performance with only linear time complexity. Please refer to EANet [10] for more details.

Time Embedding: At step i, there are multiple substeps within $r_i(.)$. For each substep j, we add a learnable sinusoidal positional embedding to the iteration. Let $t = iL+j$, then the positional embedding is defined as follows:

$$PE(t) = concate(\left[t, sin(2\pi wt), cos(2\pi wt)\right]) , \quad (4)$$

where w is a learnable parameter. $PE(t)$ is directly added to the input features after a few linear and activation functions.

The diffusion models [11, 23] use a deep neural network to learn a small distribution shift between two consecutive steps. Although the fundamental theory of the diffusion process is based on the assumption of Gaussian noise in the forward process, Bansal et al. [4] show that the process can be learned even when the degradation is completely deterministic. We want to simulate a deterministic function that shifts different feature distributions caused by different sources to a common canonical space. Ideally, at the later stage of the refinement, the features from different sources should be in a better embedding space for similarity measurement. We provide more analysis in Sec. 6.4.

4.4. Vector of Locally Aggregated Descriptors

NetVLAD [2] (Network Vector of Locally Aggregated Descriptors) is a deep neural network architecture that is commonly used in image-based or point-based place recognition tasks. Due to its permutation invariant property, the NetVLAD layer takes in a set of local feature descriptors and outputs a fixed-length global descriptor that encodes information about the entire input. It learns K cluster centers and uses the sum of residuals as global feature descriptors based on cluster assignment.

Let $\bar{F} = \{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^{N\times D_g}$ be a set of local feature descriptors from the iterative refinement step. Let $C = \{c_1, c_2, \ldots, c_K\} \in \mathbb{R}^{K\times D_g}$ be a set of cluster centroids of the same dimension D_g. The global feature $F_g \in \mathbb{R}^{D_g\times K}$ is computed as follows:

$$F_g(j,k) = \sum_{i=1}^{N} a_k(x_i)(x_i(j) - c_k(j)) , \quad (5)$$

where $a_k(x_i) = \frac{e^{-\alpha||x_i - c_k||^2}}{\sum_{k'} e^{-\alpha||x_i - c_{k'}||^2}}$ is the soft assignment function and α is a temperature parameter that controls the softness of the assignment. Note that the NetVLAD layer can be trained end-to-end with similarity loss between the global descriptors of different inputs.

4.5. Triplet Margin Loss for Metric Learning

We train the embedding function $f(.)$ with a lazy triplet margin loss [27]. Let $\mathcal{T} = \{p_A, \{p_{pos}\}, \{p_{neg}\}\}$ be a training triplet, where p_A, $\{p_{pos}\}$ and $\{p_{neg}\}$ denote the anchor point cloud, a set of positive point clouds that are structurally similar to p_a, and a set of negative point clouds that are dissimilar to p_a, respectively. For more efficient training, we use a hard example mining technique [29] to get hard positives and negatives in the triplet. The lazy triplet loss is formulated as follows:

$$\mathcal{L}_{\text{lazy}}(\mathcal{T}, \alpha) = \max(0, \delta_{pos} - \delta_{neg} + \alpha), \quad (6)$$

where α is a constant margin parameter, $\delta_{pos} = D(f(p_A), f(p_{pos}))$ and $\delta_{neg} = D(f(p_A), f(p_{neg}))$ for some distance function D.

Based on the lazy triplet margin loss, the similarity of the point cloud is based on the distance of the output embedding defined by distance function D. During inference, we calculate the distance between the database embedding and query embedding to retrieve the predicted neighbors based on the ranking of the distance.

5. CS-Campus3D Dataset

In this section, we present a novel CS-CAMPUS3D benchmark for cross-source 3D place recognition task. The CS-Campus3D benchmark consists of both aerial and ground LiDAR patches, which are tagged with UTM coordinates at their respective centroids. The coordinates of the patches are shifted and scaled to $[-1, 1]$, and the points in each patch are randomly down-sampled to size $n = 4096$. In Table. 1, we compare our dataset with a traditional 3D place recognition benchmark dataset, Oxford RobotCar.

Aerial LiDAR Scan: The aerial data is obtained from the state government and is publicly available [1]. The dataset is collected by an airborne LiDAR sensor mounted on an airplane. The points are in UTM coordinates and point spacing is roughly $0.35\ m$. The vertical and horizontal accuracies are $0.3\ m$ and $0.1\ m$, respectively.

The state-wide aerial LiDAR scan is trimmed down to a $1628 \times 3377\ m$ region that covers the same area as the ground scan does, and ground surfaces are removed. To construct the database, we uniformly sample $100 \times 100\ m$ patches several times every $19\ m$, moving north and east, and obtained 27520 patches from aerial data. The patch size is chosen according to the effective range of our LiDAR.

Ground LiDAR Scan: The ground LiDAR scan is collected on a Boston Dynamic Spot and a Clearpath Husky equipped with a Velodyne VLP 16 LiDAR with an effective range of around $100\ m$ and an accuracy of $\pm 3\ cm$. We tele-operate the robot in the region with corresponding aerial coverage and save the points from each frame. We run 27 different routes with partially overlapping and non-overlapping locations and obtain a total of 7705 patches from the ground data.

Ground Truth Correspondence: We use a U-blox NEO-M9N GPS module to get the latitude and longitude coordinates of the current LiDAR frame. We convert the geo-

Dataset	Oxford RobotCar [20]	CS-CAMPUS3D (Ours)
Scenarios	Urban, Suburban	Campus
Platform	Nissan LEAF (Car)	Spot, Husky (Mobile robots)
Road	Car Lane	Car Lane, Alley, Side Walk, etc.
Perception Sensors	RGB Camera, LiDAR	LiDAR
Geographical Coverage	$\sim 10\ km$ (Driving distance)	$1628 \times 3377\ m$ (Area)
Num. of Ground Submaps (training/testing)	21711/3030	6167/1538
Num. of Aerial Submaps (training/testing)	N/A	27520/0
Patch Size	$\sim 10 - 20\ m$	$\sim 100\ m$
Positive Pair Distance	$25\ m$	$100\ m$

Table 1: **Our proposed benchmark CS-CAMPUS3D:** We give a side-by-side comparison between the most popular 3D place recognition benchmark Oxford RobotCar and our proposed dataset CS-CAMPUS3D. We highlight that our benchmark covers more areas due to the flexibility of mobile robots and contains LiDAR data from both ground and aerial sources. We use those two datasets for evaluation.

graphic coordinates to UTM coordinates to match with the LiDAR data. The accuracy of the GPS is around $1.5\ m$.

Evaluations: In Oxford RobotCar [20], the patch size is between $10 - 20\ m$ and the successful retrieval distance is within $25\ m$. Following that ratio, we set a successful retrieval distance to be $100\ m$, which means the query point cloud is successfully localized if the retrieved neighbor is within $100\ m$. We have created two evaluation settings:

1. We construct the database set with all aerial LiDAR scans and a subset of ground LiDAR scans for training; we use the rest of the ground data as queries for testing.
2. For a more challenging setting, we construct the database set with only aerial LiDAR scans and only use ground LiDAR scans as queries. The ground scans will have to find a match only based on aerial scans.

We use the same evaluation metrics as [27, 17], explained in Sec. 6.1. The CS-CAMPUS3D database, benchmark, and all processing scripts will be publicly released.

6. Experiments and Evaluations

6.1. Dataset and Evaluation Metric

We use the Oxford RobotCar [20] dataset and CS-CAMPUS3D (ours) for evaluation. Oxford RobotCar is collected on a vehicle with LiDAR sensors while driving on urban and suburban roads in various weather conditions. It has been the benchmark dataset for most of the 3D place recognition methods because the evaluation is justified on this benchmark. Since Oxford RobotCar has a wide coverage of the city as well as repetitive routes under different conditions at different times, the point cloud retrieval task must rely on the similarity between the structures instead of exact point correspondence. The proposed benchmark, CS-CAMPUS3D, also satisfies this condition as the patches are disjoint and collected from different sources. For more details about the datasets, please refer to Table. 1 and Sec. 5.

Methods	Reference	AR@1% ↑	AR@1 ↑
PointNetVLAD [27]	CVPR 2018	80.3	-
PCAN [35]	CVPR 2019	83.8	-
LPD-Net [19]	ICCV 2019	94.9	86.3
DH3D [7]	ECCV 2020	85.3	74.2
PPT-Net [15]	ICCV 2021	98.1	93.5
SOE-Net [30]	CVPR 2021	96.4	-
Minkloc3D [17]	WACV 2021	97.9	93.0
Minkloc3Dv2 [18]	ICPR 2022	**98.9**	**96.3**
Minkloc3D-S [39]	RAL 2021	93.1	82.0
SVT-Net [8]	AAAI 2022	97.8	93.7
TransLoc3D [32]	preprint	98.5	<u>95.0</u>
CROSSLOC3D **(Ours)**	-	<u>98.59</u>	94.36

Table 2: **State-of-the-art comparisons on Oxford RobotCar:** We compare CROSSLOC3D with other state-of-the-art methods on single-source 3D place recognition benchmark. All numbers are obtained from the original paper. We underscore the second best method.

Evaluation Metrics: Like most of the place recognition benchmarks [27, 17], we used *AverageRecall@N* and *AverageRecall@1%* as evaluation metrics. *Recall@N* is defined as the percentage of true neighbors that are correctly retrieved based on the top N predicted neighbors from the network. *Recall@1%* is equivalent to *Recall@N* when N is equal to 1% of the database size.

6.2. Implementation Details

CROSSLOC3D: The input point size n is 4096. We set $L = 3$, where the voxelization size is $[0.05, 0.12, 0.4]$. The sparse convolution is implemented based on MinkowskiEngine [5]. The feature section k_s is 3. The order of the refinement is reversed (coarse $\rightarrow$ fine). During refinement, we set the number of sub-steps to 3 for all L steps. The initial dimension of the time embedding is 8. The feature dimensions of the refinement and NetVLAD are 512 and 256, respectively. We use Euclidean distance for the distance function D during training and inference.

Training Parameters: Models are trained on a single NVIDIA RTX A5000 for 200 epochs. We use the Adam optimizer with a learning rate of 0.005 and $(\beta_1, \beta_2) = (0.9, 0.999)$. We use an expansion batch sampler where the initial batch size is 64 and can grow up to 128.

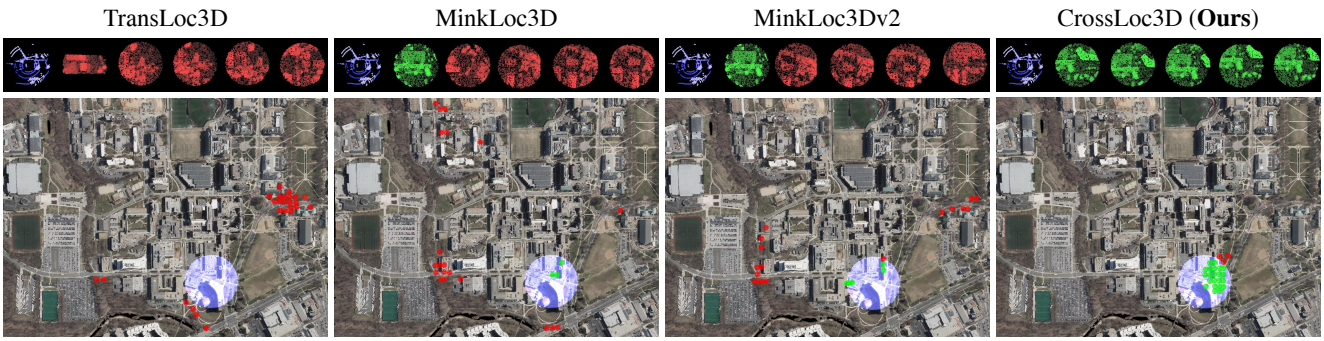

Figure 4: **Qualitative comparisons between CROSSLOC3D and other SOTA methods: Top:** Input ground query (blue) and top 5 retrievals (ranked from left to right) from database of aerial data (true neighbor – green, false neighbor – red). **Bottom:** Distributions of ground query and top 25 retrievals.

Evaluation Set: (Database / Query)	Methods	AR@1% ↑	AR@1 ↑
Aerial + Ground / Ground-Only	PointNetVLAD [27]	41.46	35.57
	LPD-Net [19]	56.49	45.94
	Minkloc3D [17]	<u>79.10</u>	<u>69.38</u>
	Minkloc3Dv2 [18]	76.68	67.06
	Minkloc3D-S [39]	59.33	44.39
	TransLoc3D [32]	69.04	58.16
	CROSSLOC3D **(Ours)**	**83.04**	**74.12**
Aerial-Only / Ground-Only	PointNetVLAD [27]	43.53	19.07
	LPD-Net [19]	40.70	11.99
	Minkloc3D [17]	<u>85.22</u>	<u>55.36</u>
	Minkloc3Dv2 [18]	83.48	52.46
	Minkloc3D-S [39]	71.88	32.17
	TransLoc3D [32]	80.64	42.97
	CROSSLOC3D **(Ours)**	**85.74**	**70.73**

Table 3: **State-of-the-art comparisons on the proposed CS-CAMPUS3D test set:** We compare CROSSLOC3D with other state-of-the-art methods on the cross-source 3D place recognition benchmark in two different settings. We underscore the second best method.

6.3. Comparisons

SOTA Comparisons: In Table. 2 and Table. 3, we show the performance of the proposed CROSSLOC3D and other SOTA methods. On two evaluation sets of CS-CAMPUS3D, CROSSLOC3D achieves an improvement of $0.52 - 3.94\%$ and $4.74 - 15.37\%$ in terms of *AR@1%* and *AR@1*, respectively. We see a trend of improvement in *AR@1%* when the database only contains aerial data because the total database size decreases. In addition, CROSSLOC3D has comparable performance to the state-of-the-art method within a margin of 0.31% in terms of 1% average recall. In Fig. 4, we can see that all top 5 retrievals and most of the top 25 retrievals by CROSSLOC3D are within the range of true neighbors, while the predictions of other SOTA methods are distributed away from the ground query.

In Fig. 5, we plot the top 25 predicted neighbors and the corresponding average recall for two evaluation settings in CS-CAMPUS3D. There is a clear drop in performance when the database is entirely based on aerial data since the ground-to-aerial match is more difficult. Our method consistently performs better on top 25 candidates retrieval for *aerial + ground* case, and takes lead on top 15 candidates for the *aerial-only* case, compared to other SOTA methods.

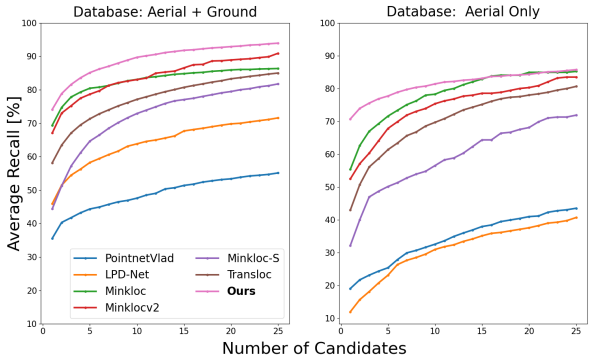

Figure 5: ***AR@N* curves on CS-CAMPUS3D:** Our method performs consistently better than other SOTA methods for top 25 neighbors.

Method	AR@1 ↑	Parameters ↓	Running Time per Query ↓
PointNetVLAD [27]	19.07	19.78M	25 ms
LPD-Net [19]	11.99	19.81M	35 ms
Minkloc3D [17]	55.36	**1.1M**	**21 ms**
TransLoc3D [32]	42.97	10.97M	46 ms
TransLoc3D†	62.57	27.85M	81 ms
CROSSLOC3D **(Ours)**	**70.73**	15.35M	26 ms

Table 4: **Performance v.s. Efficiency on CS-CAMPUS3D:** Even our model have a relatively large numbers of parameters, we did not compromise much in terms of run-time. † denotes that we manually increase the number of parameters based on the original implementation.

Complexity Comparisons: In Table. 4, we highlight the model parameter sizes and run-times of different SOTA methods given their performances on CS-CAMPUS3D when the database only consists of aerial data.

Num. of Voxel Grains (L)	Quantization Size	Feature Selection	AR@1% ↑	AR@1 ↑
1	0.01	✗	71.37	58.64
		✓	72.38	60.89
	0.08	✗	68.93	54.10
		✓	68.77	55.57
	0.2	✗	58.98	43.16
		✓	58.32	42.11
	0.4	✗	54.11	37.27
		✓	51.26	35.99
2	0.01 & 0.08	✗	74.76	63.85
		✓	76.22	65.75
	0.01 & 0.4	✗	71.14	59.68
		✓	72.60	60.36
3	0.05 & 0.12 & 0.4	✗	74.69	64.33
		✓	**77.47**	**66.14**
4	0.05 & 0.12 & 0.2 & 0.4	✗	70.97	61.07
		✓	73.22	63.14

Table 5: **Ablation studies on the voxel grains and feature selection:** We achieve the best performance when $L = 3$, with 2.78% increase on *AR@1%* with feature selection.

Quantization Size	Refinement Process	AR@1% ↑	AR@1 ↑
0.01 & 0.08	✗	71.03	60.43
	Fine → Coarse	72.54	61.78
	Coarse → Fine	**75.28**	**65.26**
0.05 & 0.12 & 0.4	✗	70.41	60.47
	Fine → Coarse	74.96	65.56
	Coarse → Fine	**77.47**	**66.14**
0.05 & 0.12 & 0.2 & 0.4	✗	71.29	59.46
	Fine → Coarse	73.22	63.14
	Coarse → Fine	**77.38**	**67.15**

Table 6: **Ablation studies on the refinement process:** We use simple concatenation of multi-grain features after passing through EA blocks [10] individually as a baseline. Refinement starting from coarse to fine leads to $4.25 - 7.06\%$ performance improvement on *AR@1%*.

6.4. Ablation Study

Voxel grains and feature selection: In Table. 5, we show the effect of number of voxel grains L with different quantization size and the feature selection. When we only choose one voxel grain, the fine resolution achieves better performance than the coarse one, and the feature selection module does not lead to performance improvement because all information is useful when there are no multi-grained voxelizations to provide potentially repetitive information from different perspectives. As L increases, the effect of feature selection is more significant. The performance start to drop when $L > 3$. One interesting observation is that, as L increases, the model does not require a fine quantization size to achieve good performance, which reduces the computation burden caused by a large L.

Refinement process: In Table. 6, we evaluate the necessity of the refinement process and different orders of refinement.

Quantization Size	Sub-steps at Refinement	AR@1% ↑	AR@1 ↑
0.01 & 0.08	1	**73.99**	61.81
	2	72.54	61.78
	3	72.58	**63.41**
0.05 & 0.12 & 0.4	1	73.13	63.0
	2	**74.96**	**65.56**
	3	74.26	63.16
0.05 & 0.12 & 0.2 & 0.4	1	73.70	61.51
	2	73.22	63.14
	3	**75.61**	**64.45**

Table 7: **Ablation studies on the refinement sub-steps:** We see an improvement of $1.45 - 2.39\%$ on *AR@1%* from adjusting the number of sub-steps.

Quantization Size	Time Embedding	AR@1% ↑	AR@1 ↑
0.01 & 0.08	✗	**73.34**	61.04
	✓	72.54	**61.78**
0.05 & 0.12 & 0.4	✗	71.37	58.42
	✓	**74.96**	**65.56**
0.05 & 0.12 & 0.2 & 0.4	✗	72.62	59.71
	✓	**73.22**	**63.14**

Table 8: **Ablation studies on time embedding:** We notice a trend of improvement with time embedding.

We try starting the refinement process from the finest and coarsest levels and discover that refinement starting from a coarse grain has the most performance improvement, which validates the intuition provided in Sec. 4.3. The coarse representations between different sources are most similar and therefore an easy starting point for metric learning. In the end, the fine-level features can be in a better canonical space and lead to better retrieval results.

Number of sub-steps during refinement: In Table. 7, we run different numbers of sub-steps. Based on the result, we can observe that more refinement sub-steps have better results for a larger number of voxel grains L.

Time Embedding: In Table. 8, we show the effect of time embedding for the refinement. For quantization size $[0.01, 0.08]$, we see a 0.8% decrease in terms of *AR@1%* , which could be explained by the fewer refinement steps L for time embedding to be effective.

7. Conclusions, Limitations, and Future Work

In this paper, we discuss and define the notion of cross-source data, and present a novel cross-source benchmark, CS-CAMPUS3D, that poses a new challenge for localization tasks and the vision community in general. We present CROSSLOC3D, a novel cross-source 3D place recognition method that uses multi-grained features with an iterative refinement process to close the gap between data sources. We show superior performance on the proposed cross-source

CS-CAMPUS3D, and demonstrate similar performance on the well-established 3D place recognition benchmark Oxford RobotCar, compared to the SOTA methods.

Currently, the task of cross-source 3D place recognition still has room for improvement on the proposed CS-CAMPUS3D. In addition, the point registration problem is another challenging topic to pursue. The tasks related to localization and registration using cross-source data open up a series of new challenges based on our proposed benchmark.

Acknowledgement. This research was supported by Army Cooperative Agreement No. W911NF2120076 and ARO grant W911NF2110026.

References

[1] The state of maryland lidar. https://imap.maryland.gov/pages/lidar-download. 5

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. pages 5297–5307, 06 2016. 3, 4, 5

[3] Relja Arandjelovic and Andrew Zisserman. All about vlad. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013. 3

[4] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S. Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise, 2022. 2, 4, 5

[5] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 6

[6] Anh-Dzung Doan, Yasir Latif, Tat-Jun Chin, Yu Liu, Thanh-Toan Do, and Ian D. Reid. Scalable place recognition under appearance change for autonomous driving. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9318–9327, 2019. 1

[7] Juan Du, Rui Wang, and Daniel Cremers. Dh3d: Deep hierarchical 3d descriptors for robust large-scale 6dof relocalization. In *European Conference on Computer Vision (ECCV)*, 2020. 3, 6

[8] Zhaoxin Fan, Zhenbo Song, Hongyan Liu, Zhiwu Lu, Jun He, and Xiaoyong Du. Svt-net: Super light-weight sparse voxel transformer for large scale place recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(1):551–560, Jun. 2022. 3, 6

[9] Abel Gawel, Carlo Del Don, Roland Siegwart, Juan Nieto, and Cesar Cadena. X-view: Graph-based semantic multi-view localization. *IEEE Robotics and Automation Letters*, 3(3):1687–1694, 2018. 3

[10] Meng-Hao Guo, Zheng-Ning Liu, Tai-Jiang Mu, and Shimin Hu. Beyond self-attention: External attention using two linear layers for visual tasks. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2021. 4, 5, 8

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. 2, 4, 5

[12] Xiaoshui Huang, Guofeng Mei, and Jian Zhang. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[13] Xiaoshui Huang, Guofeng Mei, Jian Zhang, and Rana Abbas. A comprehensive survey on point cloud registration. *ArXiv*, abs/2103.02690, 2021. 1, 3

[14] Xiaoshui Huang, Jian Zhang, Qiang Wu, Lixin Fan, and Chun Yuan. A coarse-to-fine algorithm for registration in 3d street-view cross-source point clouds. In *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–6, 2016. 3

[15] Le Hui, Hang Yang, Mingmei Cheng, Jin Xie, and Jian Yang. Pyramid point cloud transformer for large-scale place recognition. In *ICCV*, 2021. 3, 6

[16] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3304–3311, 2010. 3

[17] Jacek Komorowski. Minkloc3d: Point cloud based large-scale place recognition. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1789–1798, 2021. 3, 6, 7

[18] J. Komorowski. Improving point cloud based place recognition with ranking-based loss and large batch training. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 3699–3705, Los Alamitos, CA, USA, aug 2022. IEEE Computer Society. 6, 7

[19] Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Peng Yin, Wen Chen, Hesheng Wang, Haoang Li, and Yun-Hui Liu. Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 3, 6, 7

[20] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017. 2, 3, 6

[21] Raúl Mur-Artal and Juan D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 1

[22] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016. 3

[23] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 2, 4, 5

[24] Tim Y. Tang, Daniele De Martini, and Paul Newman. Get to the Point: Learning Lidar Place Recognition and Metric Localisation Using Overhead Imagery. In *Proceedings of Robotics: Science and Systems*, Virtual, July 2021. 3

[25] Tim Yuqing Tang, D. Martini, and Paul Newman. Get to the point: Learning lidar place recognition and metric localisation using overhead imagery. *Robotics: Science and Systems XVII*, 2021. 1

[26] Aysim Toker, Qunjie Zhou, Maxim Maximov, and Laura Leal-Taix'e. Coming down to earth: Satellite-to-street view synthesis for geo-localization. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6484–6493, 2021. 3

[27] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 5, 6, 7

[28] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision - 3DV 2013*, pages 127–134, 2013. 3

[29] C. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl. Sampling matters in deep embedding learning. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2859–2867, Los Alamitos, CA, USA, oct 2017. IEEE Computer Society. 5

[30] Y. Xia, Y. Xu, S. Li, R. Wang, J. Du, D. Cremers, and U. Stilla. Soe-net: A self-attention and orientation encoding network for point cloud based place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3, 6

[31] Zimin Xia, Olaf Booij, Marco Manfredi, and Julian FP Kooij. Visual cross-view metric localization with dense uncertainty estimates. In *European Conference on Computer Vision*, pages 90–106. Springer, 2022. 3

[32] Tian-Xing Xu, Yuan-Chen Guo, Zhiqiang Li, Ge Yu, Yu-Kun Lai, and Song-Hai Zhang. Transloc3d : Point cloud based large-scale place recognition using adaptive receptive fields, 2021. 3, 6, 7

[33] Ge Xuming, Fan Yuting, Zhu Qing, Wang Bin, Xu Bo, Hu Han, and Chen Min. Semantic maps for cross-view relocalization of terrestrial to uav point clouds. *International Journal of Applied Earth Observation and Geoinformation*, 114:103081, 2022. 2, 3

[34] Hongji Yang, Xiufan Lu, and Yingying Zhu. Cross-view geo-localization with layer-to-layer transformer. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29009–29020. Curran Associates, Inc., 2021. 3

[35] Wenxiao Zhang and Chunxia Xiao. Pcan: 3d attention map learning using contextual information for point cloud based retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12436–12445, 2019. 6

[36] Xiwu Zhang, Lei Wang, and Yan Su. Visual place recognition: A survey from deep learning perspective. *Pattern Recognition*, 113:107760, 2021. 2

[37] Sijie Zhu, Mubarak Shah, and Chen Chen. Transgeo: Transformer is all you need for cross-view image geo-localization. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1152–1161, 2022. 3

[38] Sijie Zhu, Taojiannan Yang, and Chen Chen. Vigor: Cross-view image geo-localization beyond one-to-one retrieval. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5316–5325, 2020. 3

[39] Kamil Zywanowski, Adam Banaszczyk, Michał R. Nowicki, and Jacek Komorowski. Minkloc3d-si: 3d lidar place recognition with sparse convolutions, spherical coordinates, and intensity. *IEEE Robotics and Automation Letters*, PP:1–1, 2021. 3, 6, 7