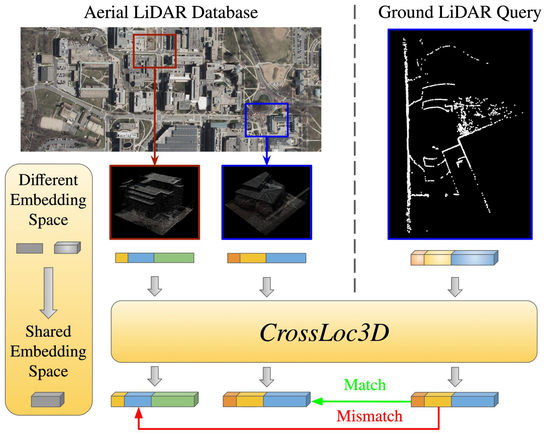
CrossLoc3D:Aerial-Ground Cross-Source 3D Place Recognition

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

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

ICCV23 PARIS

Introduction



CrossLoc3D is a place recognition architecture that works on **multi-source** point cloud data. It 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.

CS-CAMPUS3D Dataset

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	$25\ m$	$100\ m$



CS-CAMPUS3D dataset has **point cloud** from **two different data sources**.

Aerial Data: 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.

Ground LiDAR Scan: The ground LiDAR scan is collected with effective range of around 100 m and an accuracy of ±3 cm.

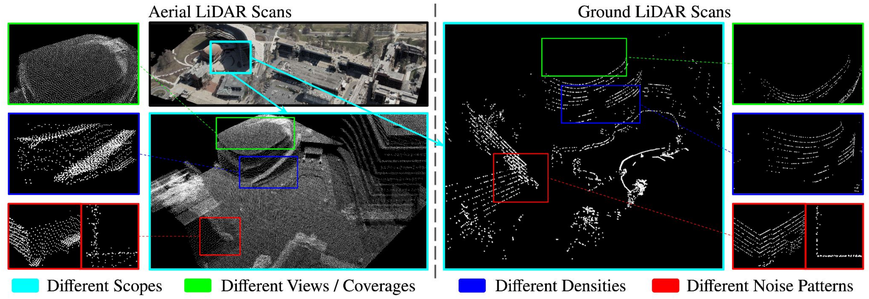
Correspondence:
The geographic coordinates are converted to UTM coordinates to match with the LiDAR data. And the accuracy of the GPS is around 1.5m.

Training set
- Aerial data - corresponds to the region in training split of the ground data.
- Ground data - training split in ground database

Testing set
- Aerial data - whole database is included.
- Ground data - query points are taken from a disjoint set, ground patch not seen in ground data train set.

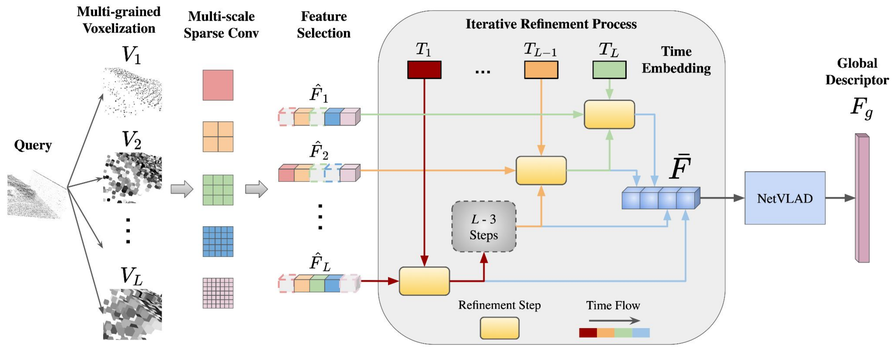
Ground LiDAR scan routes:
1. For the ground LiDAR scan collection.The total length of the collection is roughly 247 minutes.
2. We remove the LiDAR scan in the location where the covariance of the GPS location exceeds 10 m due to poor satellite signal.

Representation gap in 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 are different because of various scan patterns, effective ranges, and fidelity of LiDARs.
Noise Patterns (red): The aerial scans have larger noises, as shown in the of a corner of the building.
We need a method that can better close the representation gap between the two sources by converting the point cloud features in different embedding spaces to a common embedding space.

Architecture of CrossLoc3D



1.Multi-grained Features with Selection
a. **Voxelization** on the pointcloud with different resolutions is performed with L levels. Followed by **sparse convolution** with 5 different field of views to obtain a multi-scale feature. We add **Feature Selection** module, few layers of convolution, that can choose the best field of view for each feature based on a confidence score. With feature selection, we want to keep the best information obtained from multi-grained quantizations and multi-scale kernels.

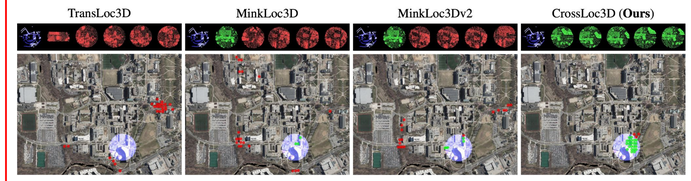
2.Iterative Refinement
a. We close the **representation gap** between different sources, by gradually learning a distribution shift towards a **canonical space** at each refinement step in the training process. function is implemented with a series of External Attention (EA) blocks, where each block is considered a **substep** of the refinement. In the EA block, the query is correlated with a pre-trained memory unit containing the **context of the entire** Ground point cloud dataset.

3.Vector of Locally Aggregated Descriptors
a. 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.

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

Inference time: CrossLoc3D has a inference time of 26ms. Even our model have a relatively large numbers of parameters, we did not compromise much in terms of run-time.

Evaluation on Datasets



Top:
- *Blue*–Input ground query. Top 5 retrievals (ranked from left to right) from aerial database.
- *Green*–true neighbor.
- *Red*–false neighbor.

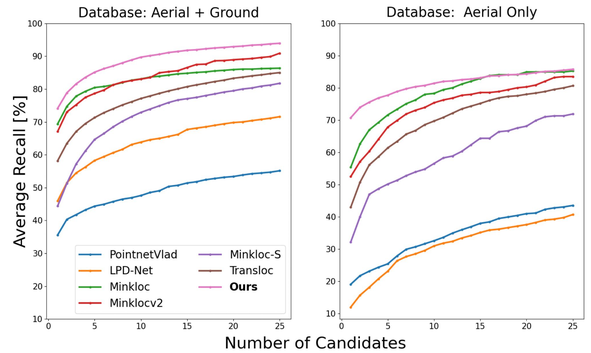
Bottom:
- Distributions of ground query and top 25 retrievals.

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	95.0
CrossLoc3D **(Ours)**	-	98.59	94.36

State-of-the-art comparisons on Oxford Robot- Car

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]	79.10	69.38
	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]	85.22	55.36
	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**

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



AR@N curves on CS-CAMPUS3D: Our method performs consistently better than other SOTA methods for top 25 neighbors for *Aerial+ground* case. and takes lead on top 15 candidates for the *aerial-only* case, compared to other SOTA methods.
There is a drop in performance when the database is entirely based on aerial data since the ground-to-aerial match is more difficult.

For more details