## PointNetVLAD: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition Supplementary Material

Mikaela Angelina Uy Gim Hee Lee Department of Computer Science, National University of Singapore {mikacuy,gimhee.lee}@comp.nus.edu.sg

## **1. Benchmark Datasets**

We provide additional information on the benchmark datasets that are used to train and evaluate the network, which are based on the Oxford RobotCar dataset [1] and three in-house datasets. Figure 1 (top row) shows a sample reference map for each of the four regions. Figure 1 (middle row) shows sample submaps from the different regions. Figure 2 illustrates data splitting into disjoint reference maps, which was done by randomly selecting  $150m \times 150m$  regions.



Figure 1. Top row shows a sample reference map from (a) Oxford, (b) U.S., (c) R.A and (d) B.D.. Middle row shows a sample submap from each of the regions representing the local area marked by the red box on the reference map. Bottom row shows the corresponding preprocessed submaps of the local areas from the middle row.



Figure 2. Data splitting: Blue points represent submaps in the training reference map and red points represent submaps in the testing reference map. The data split was done by randomly selecting regions in the full reference map.

## 2. Implementation Details

We use a batch size of 3 tuples in each training iteration. Each tuple is generated by selecting an anchor point cloud  $P_a$  from the set of submaps in the training reference map followed by an on-line random selection of  $P_{pos}$  and  $\{P_{neg}\}$  for each anchor. Each training tuple contains 18 negative point clouds, i.e.  $|\{P_{neg}\}| = 18$ . Hard negative mining is used for faster convergence by selecting the hardest/closest negatives from 2000 randomly sampled negatives to construct  $\{P_{neg}\}$  for each anchor  $P_a$  in an iteration. The hard negatives are obtained by selecting the 18 closest submaps from the cached global descriptors f of all submaps in the training reference map, and the cache is updated every 1000 training iterations. We also found network training to be more stable when we take the best/closest of 2 randomly sampled positives to  $P_a$  in each iteration.

## References

W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 2017.