Pose Correction for Highly Accurate Visual Localization in Large-scale Indoor Spaces

Supplementary material

In this supplementary material, we describe more details of each method explained in *Section 3.4 Extended pose correction* in the main paper. Finally, we show results of experiments performed on a large-scale outdoor dataset, the RobotCar Seasons dataset [8].

A. Extended Pose Correction

A.1. Divided matching

Divided matching finds feature matches in left, right, top, and bottom halves side of an image, and the original image (*e.g.* Figure 1(a)). The identified matches are concatenated for use in the PnP algorithm [5] inside the RANSAC loop [3] to correct the pose from X^- to X^+.

A.2. Inter-pose matching

Inter-pose matching utilizes a multiple local feature map, $\mathbb{F}_i$, to find feature matches using SuperGlue [7]. To use $\mathbb{F}_i$ from a different view, we first find connected positions using the Scangraph. For example, let us assume p_i is a retrieved image's position, and p_j and p_k are the connected positions in the Scangraph (*c.f.* Figure 1(b)). We project each set of local features in $\mathbb{F}_i$, $\mathbb{F}_j$, and $\mathbb{F}_k$ onto the image plane of X^- to generate synthetic local feature images (*e.g.* $\mathbf{I}'_i$, $\mathbf{I}'_j$, and $\mathbf{I}'_k$). This enables generation of synthetic local feature images from different views. Finally, we find feature matches using each synthetic local feature image with the query image. The identified matches are concatenated and the PnP algorithm in the RANSAC loop follows to correct the pose from X^- to X^+.

In indoor spaces, the distance to the scene geometry tends to be short. Thus, concave structures or clutters that cause occlusion often have a stronger effect than outdoor spaces. In these places, even the best scanned position in the database, p_i, used for pose estimation may not be sufficient to cover an arbitrary query's view. Inter-pose matching may help in finding more correct feature matches and enhancing localization accuracy.

A.3. Filtering process

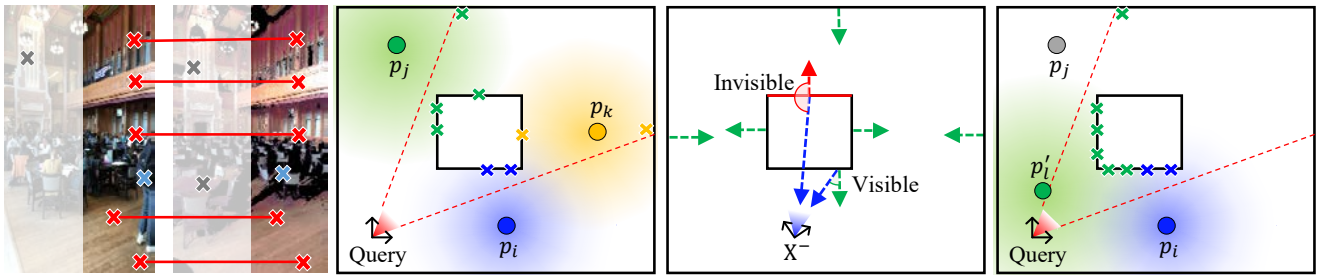
Ideally, it is best to select only the visible features when generating synthetic local feature images. For this, we proposed two filtering approaches.

One is point normal filtering that removes invisible local features in the inference time. We utilize surface normal directions to select visible features. We first generate a local feature map ($\hat{\mathbb{F}}_i$) that contains the normal direction of local features. When projecting the local features onto synthetic local feature images, we select visible features using these $\hat{\mathbb{F}}_i$ by computing cosine distance between the direction vector and the corresponding normal, as shown in Figure 1(c). This cosine distance allows filtering of the occluded local features that face the opposite directions.

The other is the virtual local feature (VLF) map that extends the database with more virtual positions. A virtual position is set for every edge in the Scangraph. For example, if p_i and p_j are connected in the Scangraph, a virtual position, p'_l is obtained by scoring samples from circular grid, where p_i and p_j form diameter of the circle. A candidate that can observe the most even and numerous local features extracted from the two adjacent positions shows the highest score and is selected as a virtual position p'_l. Since the possible candidates of a virtual position between the two adjacent locations are infinite, the highest score in the samples can guarantee only a suboptimal position depending on the grid size. Figure 2 shows the obtained virtual positions.

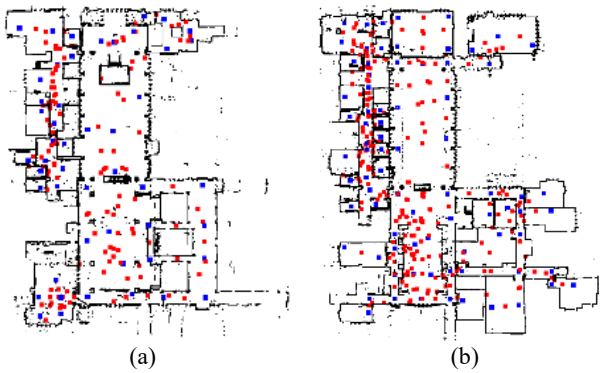
At the virtual position p'_l, we perform the hidden point removal (HPR) algorithm [4] to find visible local features from the entire local feature map $\mathbb{F}$, and create a corresponding VLF map $\mathbb{F}'_l$. While point clouds corresponding to local features in the database are sparse, the HPR algorithm requires dense point clouds for accurate filtering. To make dense point clouds and yield an accurate filtered output, we merge the point clouds of local features and the original point clouds scanned by sensor before the HPR algorithm. After the HPR algorithm has been conducted, we acquire local features visible from p'_l and make $\mathbb{F}'_l$.

In the pose correction step, we create synthetic local feature images by projecting $\mathbb{F}_i$ and $\mathbb{F}'_l$ similar to inter-pose matching, as shown in Figure 1(d). Note that virtual positions make the database denser, which is beneficial for



(a) Divided matching (b) Inter-pose matching (c) Point normal filtering (d) Virtual local feature map

Figure 1. (a) An example illustrating divided matching focusing on the local features in the right side of the images. (b) This example illustrates a retrieved image's position p_i, and positions p_j and p_k connected to p_i in the Scangraph that are used for inter-pose matching from top view. Each colored region and cross represent the area covered by the scanned position and local feature's corresponding point in 3D space, respectively. Each set of local features in the same color is used for finding feature matches with query's local features using SuperGlue. (c) Local features that have point normal directions opposite to the X^- are considered invisible features and removed before projection onto the image plane X^-. (d) Local features (green crosses) in a virtual local feature map $\mathbb{F}'_l$ at the virtual position p'_l are projected on the image plane to generate a synthetic local feature image. Both sets of local features (green and blue crosses) are used in the same way as inter-pose matching.



(a) (b)

Figure 2. Locations of virtual positions. From 277 distinct scanned positions (blue dots), 638 virtual positions (red dots) are generated in InLoc dataset. (a) Top view of DUC1. (b) Top view of DUC2.

finding a position that shares a similar view to an arbitrary query's view. In addition, the VLF map removes local features that are invisible from p'_l ahead of the inference time (*i.e.* database construction time), which reduces the chances of invisible local features to be projected on $\mathbf{I}'$ during the inference time.

B. Outdoor Dataset

We evaluate our method with a large-scale outdoor dataset, the RobotCar Seasons dataset [8], in which the view-difference problem is not significant because database and query images are captured along vehicle trajectory. We used a coarse-to-fine method [6] for comparison, which uses NetVLAD [1] for global retrieval and SuperPoint [2] for local features. The basic pose correction module is then applied to on the output of the method to determine whether the pose correction enhances the accuracy. Note that the PV step is excluded in the outdoor experiments, similar to the method for comparison [6].

Error [*m* / *deg*]	day all			night all		
	0.25 / 2	0.5 / 5	5.0 / 10	0.25 / 2	0.5 / 5	5.0 / 10
NVLD+SP [6]	53.0	79.3	95.0	5.9	17.1	29.4
NVLD+SP+PC	53.8	79.3	95.1	6.9	19.8	29.5

Table 1. Evaluation results for the RobotCar Seasons dataset. NVLD, SP, and PC denote NetVLAD, SuperPoint, and pose correction, respectively.

As presented in Table 1, the performance gain using the pose correction was not significant compared to the experiments in large-scale indoor datasets. This is because the view-difference problem is not significant for the trajectory-based outdoor dataset compared to the indoor datasets. In other words, the pose correction module can help in enhancing localization accuracy especially when the sparsity of image positions inheres in the database, *i.e.* large-scale indoor spaces.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016. 2

[2] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *CVPR Workshops*, 2018. 2

[3] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24(6):381–395, 1981. 1

[4] Sagi Katz, Ayellet Tal, and Ronen Basri. Direct visibility of point sets. *TOG*, 26(3):24–es, 2007. 1

[5] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *CVPR*, 2011. 1

[6] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From Coarse to Fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 2

[7] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 1

[8] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6DOF outdoor visual localization in changing conditions. In *CVPR*, 2018. 1, 2