SANet: Scene Agnostic Network for Camera Localization Supplementary Material

Luwei Yang[1,*] Ziqian Bai[1,*] Chengzhou Tang[1] Honghua Li[2] Yasutaka Furukawa[1] Ping Tan[1]
[1] Simon Fraser University [2] Alibaba A.I Labs
{luweiy, ziqianb, cta73, furukawa, pingtan}@sfu.ca, honghua.lhh@alibaba-inc.com

To make our submission self-contained, the supplementary document provides additional details in: 1) The feature extractor used for constructing feature pyramids; 2) Query-Scene Registration; 3) Iterative Scene Coordinate Prediction; 4) Training; 5) Query Pose Estimation via RANSC+PnP[1]. In the End, we provide additional analysis and coordinate map examples.

1. Pipeline Details

Feature Extractor We use DRN38[5] as our backbone feature extractor for constructing both scene feature pyramid $\{\mathbf{F}_s^l\}$ and query image feature pyramid $\{\mathbf{E}^l\}$. Figure 2 lists the backbone network structure with the output feature map for each level l.

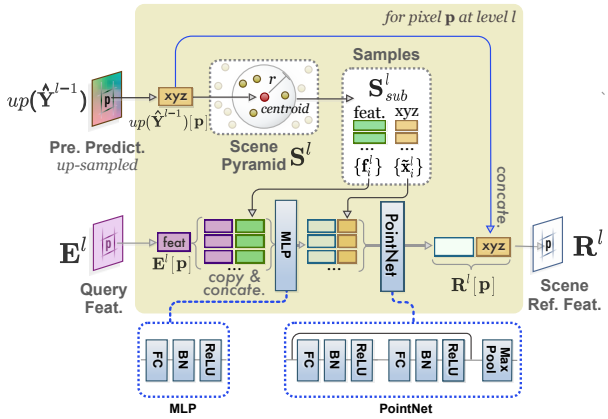

Figure 1: Comprehensive view of QSR module, the detailed components and operations are marked with blue rectangles and line.

Query-Scene Registration Figure 1 shows the comprehensive view of Query-Scene Registration module. Comparing with the Fig.3 in main text (Sec 4.2), we detail two components and an operation in the following. The `MLP` consists of a `Fully-Connected (FC)` layer followed by

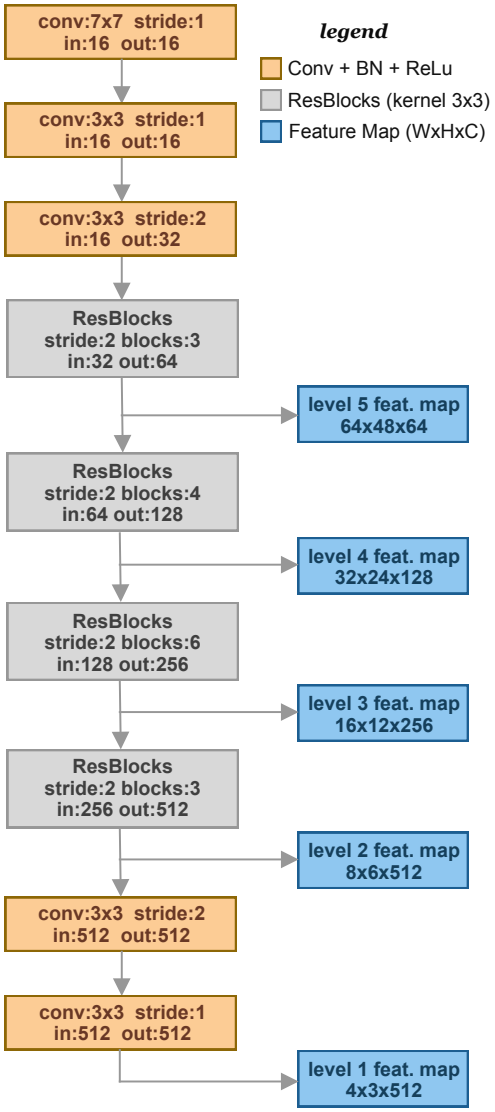

Figure 2: Feature Extractor (DRN38[5]) structure

*These authors contributed equally to this work.

`Batch Normalization (BN)` and `ReLU` activation, while the `PointNet` has two `FC-BN-ReLU` blocks with one skip connection, and a `MaxPooling` in the end.

Recall that in QSR module, for one specific pixel $\mathbf{p}$ defined in finer levels $l > 1$, we sample points $\mathbf{S}^l_{sub}$ from scene pyramid $\mathbf{S}^l$ that are close to the previous up-sampled coordinate $up(\hat{\mathbf{Y}}^{l-1})[\mathbf{p}]$. During the sampling, we normalize the 3D scene space represented by $\mathbf{S}^l_{sub}$ through subtracting $up(\hat{\mathbf{Y}}^{l-1})[\mathbf{p}]$ from `xyz` coordinates $\{\tilde{\mathbf{x}}^l_i\}$ to reduce the input space for better generalization. Later, we additionally append $up(\hat{\mathbf{Y}}^{l-1})[\mathbf{p}]$ into $\mathbf{R}^l[\mathbf{p}]$ to provide enough information for recovering the normalized 3D scene space represented by $\mathbf{S}^l_{sub}$.

Iterative Scene Coordinate Prediction Recall that after getting the scene reference feature $\mathbf{R}^l$ from the QSR module, we further fuse the image contextual information then compute the prediction at level l. To reduce the network size and avoid over-smoothed results, we do not fuse the image context at the final level $l = 5$. Instead, we directly decode $\mathbf{R}^5$ by a 1×1 `Conv` to get the prediction.

Training For pixel $\mathbf{p}$ at level l, if its ground-truth coordinate does not locate inside the sphere when sampling $\mathbf{S}^l_{sub}$, we discard the gradient of that pixel.

Query Pose Estimation We follow DSAC++[1] and provide details in finding the best hypothesis h^* and estimating the final pose Θ_q of query frame from h^*. Recall that we build the hypothesis pose set $\mathbf{H} = \{h_j | j = 1, ..., k\}$ by solving the PnP problem[2] for each randomly sampled 4-point tuple from predicted scene coordinate map $\hat{\mathbf{Y}}$. We then find the best hypothesis that is most coherent with the predicted scene coordinate. In particular, each hypothesis is scored with counted inliers. Given a pose hypothesis h, the scoring function is defined as:

$$\xi(h) = \sum_{\mathbf{p} \in \hat{\mathbf{Y}}} sig(\beta(\gamma - \pi(h, \hat{\mathbf{Y}}[\mathbf{p}]))), \tag{1}$$

where the hyper-parameter β controls the softness of the sigmoid function $sig(\cdot)$, and the variable γ is a manually defined parameter indicating the inlier threshold. The function $\pi(h, \hat{\mathbf{Y}}[\mathbf{p}])$ defines the reprojection error at pixel $\mathbf{p}$ under the hypothesis pose h as follows,

$$\pi(h, \hat{\mathbf{Y}}[\mathbf{p}]) = \left\| K h^{-1} \hat{\mathbf{Y}}[\mathbf{p}] - \mathbf{x}_\mathbf{p} \right\|. \tag{2}$$

To obtain the final camera pose Θ_q , the hypothesis with the highest score is selected first,

$$h^* = \arg\max_h \xi(h).$$

Next, initialized by h^*, the final camera pose Θ_q is refined iteratively by first selecting the inliers from all coordinates of which the reprojection error is lower than the inlier threshold γ, then camera pose is optimized by involving all newly selected inliers.

For experiments, we sample 128 hypotheses from predicted coordinate map. We set hyper-parameter $\beta = 4.0$, the inlier threshold $\gamma = 0.5$ for outdoor scene and $\gamma = 0.75$ for indoor scene respectively. In terms of refinement, we iterate the process (i.e. select inliers and optimize pose) until convergence or 100 times.

2. Additional Analysis

Additional Coordinate Map Prediction Examples Figure 4 shows 6 additional examples of scene coordinate map comparison on *7Scenes*[3] dataset.

Fine-tuning on Outdoor Dataset We investigate the effect of fine-tuning in this section. The fine-tuning is necessary due to the distribution gap between indoor and outdoor images. As shown in Table 1, without fine-tuning (F.T.), the performance degrades gently on 4 outdoor scenes. In general, if sufficient outdoor data is given, a common indoor/outdoor model could be trained.

Table 1: Pose median error w/ or w/o fine-tuning (F.T)

	K. College	O. Hospital	S. Facade	S.M Church
w/o F.T.	0.76°0.39m	0.47°0.34m	0.56°0.15m	0.84°0.23m
w/ F.T.	0.54°0.32m	0.53°0.32m	0.47°0.10m	0.57°0.16m

Reliance on Retrieval Quality and Number of Images in the Scene Pyramid
To quantify the influence of retrieval **quality**, we simulate different levels of retrieval quality by keeping the top k retrieval results and replace the rest of the 10 retrieved images with least scored ones. To test the effect of the **number** of images in the scene pyramid, we use the top k retrieved images to construct the scene pyramid. The performances are measured by the ratio of pose errors that are less than (5°, 5cm) on *7Scenes*. As shown in Figure 3, even with the worst retrieval quality or only 1 image in the scene pyramid, our method could still localize more than 50% queries.

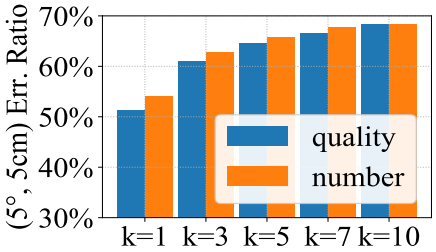

Figure 3: Pose accuracy w.r.t Retrieval **quality** and **Number** of images in the scene pyramid

Time w.r.t Database Scale We also investigate how the running time of our pipeline scales w.r.t number of scene

images. As shown in Table 2, the time of indexing all scene images and image retrieval increases linearly, while the pose estimation takes a constant time as we only retrieve a fixed number (i.e. 10) of scene images.

Table 2: Time of each step w.r.t number of scene images.

Steps	**500 imgs.**	**1000**	**2000**	**5000**	**7000**
Index. VLAD feat. (all scene imgs.)	23s	50s	128s	263s	427s
Retrieval (per query)	16ms	27ms	61.8ms	123ms	171ms
Estimate Pose (per query)	0.37s	0.37s	0.37s	0.37s	0.37s
Total (per query)	0.39s	0.40s	0.43s	0.49s	0.54s

References

[1] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 4654–4662, 2018. 1, 2

[2] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 25(8):930–943, 2003. 2

[3] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 2930–2937, 2013. 2, 4

[4] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor visual localization with dense matching and view synthesis. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

[5] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

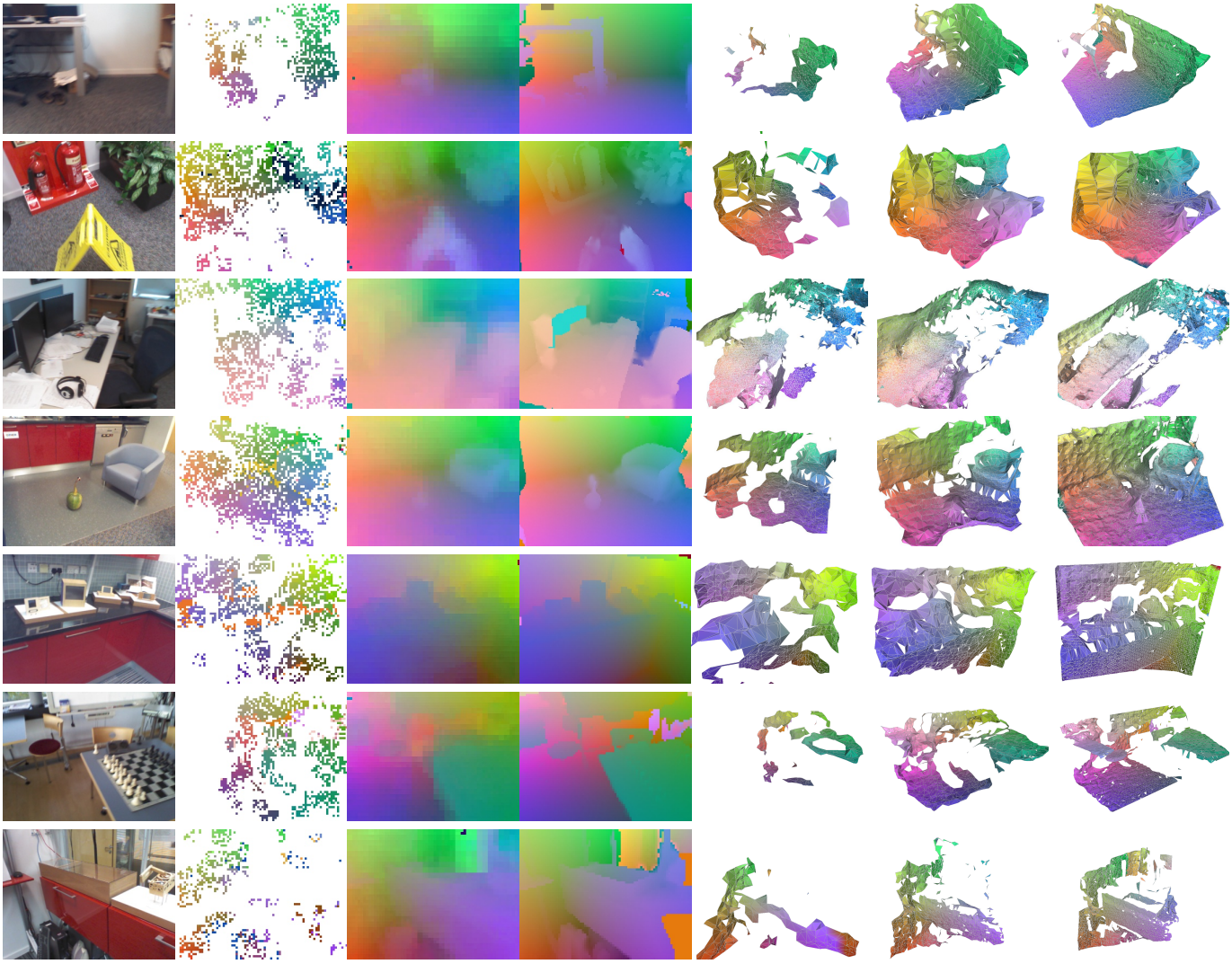

Figure 4: Additional examples of scene coordinate map comparison with InLoc [4] and the ground truth (G.T.) on *7Scenes* [3], the scene coordinate positions `xyz` are encoded in `rgb` channels for visualization. The last three columns show the geometry (Geo.) comparison by reconstructing the mesh from the scene coordinate map.