Privacy Preserving Partial Localization Supplementary Material

Marcel Geppert[1] Viktor Larsson[2] Johannes L. Schönberger[3] Marc Pollefeys[1,3]

[1]ETH Zurich [2]Lund University [3]Microsoft

1. Overview

In the supplementary material we present the following;

- Two variants of the solver for pose estimation with known gravity direction or unknown scale (Section 2)
- Explanations on how to fuse the pose from non-orthogonal maps (Section 3)
- Explanations on the map coordinate system's impact on the 1D pose ambiguity (Section 4)
- Additional results from the noise/outlier sensitivity experiment on synthetic data (Section 5)
- Additional results for the 7 Scenes dataset (Section 6)
- Experiments with known vertical direction (Section 7)
- Experiments with unknown scale (Section 8)
- An additional qualitative analysis of the results with disjoint queries (Section 9)
- Additional experiments for trajectory-based queries using higher quality relative poses (Section 10)

2. Variants of the solver

2.0.1 Known Gravity

In the case of known gravity, we can assume that both the query and the map are gravity-aligned. This reduces the remaining rotation to be around the gravity direction,

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} . \tag{1}$$

The localization problem is now asymmetric in the coordinate axes, because only the two that are orthogonal to the gravity (x and y) depend on the rotation angle, *i.e.*

$$(\cos\theta, 0, -\sin\theta)\boldsymbol{X}_q + t_1 = x_m \tag{2}$$
$$(\sin\theta, \cos\theta, 0)\boldsymbol{X}_q + t_2 = y_m \tag{3}$$
$$(0, 0, 1)\boldsymbol{X}_q + t_3 = z_m \tag{4}$$

The localization server storing the z-offsets can thus compute the translation offset t_3 from a single point, ignoring the rotation. For the two other coordinate-servers, the problem reduces to a two-dimensional version of the full-rotation case, where we are estimating a two-dimensional unit vector $\hat{\boldsymbol{r}} = (\cos\theta, \pm\sin\theta)$. The methods from Section 3.3.1 and Section 3.3.2 in the main paper generalize directly to the 2D-case with the only difference of requiring fewer points (2 instead of 3) for the minimal sample and yielding fewer roots in equation (12) in the main paper. We present experiments for this setup in Section 7.

2.0.2 Unknown Scale

So far we have assumed that the query point cloud and the map have a consistent scale. For settings where it is not possible to estimate metric scale on the client side, it is also possible to solve for the relative scale jointly with the camera pose. In this case, we simply get

$$s\boldsymbol{r}^T\boldsymbol{X} + t = x, \quad \boldsymbol{r}^T\boldsymbol{r} = 1 \ , \tag{5}$$

where s is the relative scale between the query and map. Reparameterizing in terms of $\boldsymbol{v} = s\boldsymbol{r}$, we get an unconstrained problem. This allows to solve linearly for $(\boldsymbol{v}, t)$ from four correspondences. The non-minimal problem simplifies greatly as it becomes a linear least squares problem. Experiments with this method are shown in Section 8.

3. Non-Orthogonal maps

For simplicity we only show the case of orthogonal map directions in the paper. However, the method works just as well with non-orthogonal maps. Given the independent unit vectors $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ that determine the map directions, we can start from the original 3D alignment constraint

$$R\boldsymbol{X}_q + \boldsymbol{t} = \boldsymbol{X}_m \Rightarrow \mathbf{e}^T\left(R\boldsymbol{X}_q + \boldsymbol{t}\right) = \mathbf{e}^T\boldsymbol{X}_m \ . \tag{6}$$

This leads to the formulas for the obtained partial poses

$$\hat{\boldsymbol{r}}_k = R^T\mathbf{e}_k, \ \ \hat{\boldsymbol{t}}_k = \mathbf{e}_k^T\boldsymbol{t}, \ \ k = 1, 2, 3 \ . \tag{7}$$

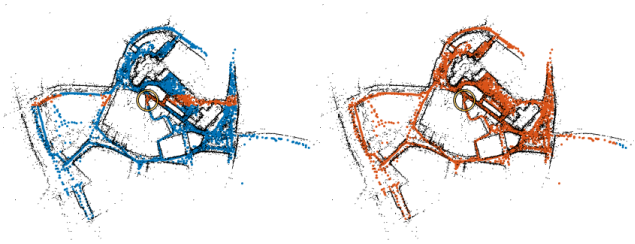

Figure 1. **Examples for pose ambiguity in 1D localization:** Out of all images (blue) we highlight the ones that are localized close to a given 1D offset (red). These images lie close to a 3D plane that is parallel to the map planes. *Left:* a horizontal localization direction (vertical in the figure). *Right:* (3D) Vertical localization (orthogonal to the figure plane). Since the scene is relatively flat, almost all images localize to a similar 1D offset.

To recover the full pose in the original reference frame we get

$$\hat{R} = \begin{bmatrix} \hat{\boldsymbol{r}}_1 & \hat{\boldsymbol{r}}_2 & \hat{\boldsymbol{r}}_3 \end{bmatrix} = R^T \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} = R^T E \quad (8)$$

and finally

$$R = \left(\hat{R} E \right)^T . \quad (9)$$

Similarly,

$$\hat{\boldsymbol{t}} = \begin{bmatrix} \hat{t}_1 \\ \hat{t}_2 \\ \hat{t}_3 \end{bmatrix} = E^T \boldsymbol{t} \Rightarrow \boldsymbol{t} = E \, ^T \hat{\boldsymbol{t}} . \quad (10)$$

This construction works for any set of linearly independent (but necessarily orthogonal) unit vectors $\mathbf{e}_1, \mathbf{e}_2$ and $\mathbf{e}_3$.

4. Pose ambiguity

For a given scene we can define a measure for the 1D pose ambiguity as the number of valid image poses that localize to the same 1D location. This is visualized in Figure 1. For example, assuming a scene that is very stretched out along a single dimension, only very few valid poses will be close to any 1D location along this dimension. In contrast, for a planar scene almost all valid poses will be close to a certain 1D location in the vertical direction as shown in Figure 1 (right). Therefore, the choice of the map coordinate system has an impact on the ambiguity of any 1D localization.

5. Additional Results on Synthetic Data

In this section we report additional metrics/plots from the experiment on synthetic data in the main paper. Please refer to the main paper for details on the experimental setup. To have a metric which includes both the rotation and translation, we also compute the average corner displacement,

Scene	**Method**								
	Point-to-Point [4]			*Point-to-Plane* [3]			Ours		
	τ_1	τ_2	τ_3	τ_1	τ_2	τ_3	τ_1	τ_2	τ_3
Alamo	62.9	92.2	97.8	38.6	84.9	95.0	58.6	88.5	95.1
Gendarmenmarkt	32.9	72.2	86.6	15.0	61.5	83.9	18.2	54.6	76.2
Madrid Metropolis	37.2	79.2	93.1	18.6	73.0	89.4	30.7	70.1	82.8
Roman Forum	46.1	82.1	93.1	28.7	76.8	91.1	34.7	73.5	87.5
Tower of London	36.8	76.0	92.9	19.2	68.6	88.7	25.8	61.7	81.4
τ_1 = (0.05m / 2)	τ_2 = (0.2m / 5)	τ_3 = (0.5m / 10)							

Table 1. **Localization with Known Vertical:** The table shows the recall for different combined error thresholds $_1$; $_2$; $_3$.

i.e. we take the eight corners of the unit cube and transform them with the ground truth pose and then back with the estimated pose. The reported error is then the average displacement of the corners. Figure 2 shows additional metrics for the varying noise experiment. In Figure 3 we show the recall curve for the rotation error for three fixed outlier ratios (50%, 60% and 70% outliers). The figures show that compared to the random-plane method from Speciale et al. [3], the proposed method is more sensitive to noise but more robust to outliers.

6. *7 scenes* Precision/Recall

Figure 4 shows the Precision/Recall plot for the *7 scenes* [2] scenes that are omitted in the paper.

7. Known Gravity Experiments

Many modern handheld devices (e.g. phones) have an Inertial Measurement Unit (IMU) that can provide estimates of the vertical (gravity) direction. As shown in the main paper, this knowledge can be used to simply the localization problem as it constrains the relative rotation between the query and the map. In this section we evaluate the impact on the localization accuracy of adding these measurements. We replicate the setup for the experiment on the 1DSfM dataset [5] in the main paper. Since the dataset only provides image data, we synthesize the gravity measurements using the ground truth reconstruction. Table 1 shows the percentage of successful localizations for different thresholds. Adding the additional prior on the relative rotation improves the localization accuracy. This is partly since there are fewer pairwise constraints on the rotation matrix that we are ignoring in the distributed approach.

8. Unknown Scale Experiments

For each query image we uniformly sample a scaling factor in the interval $[0.1, 10.0]$ and transform the query point cloud accordingly. We compare our results against point-to-point matching also estimating the unknown scale, but do not include localization from random plane maps as we

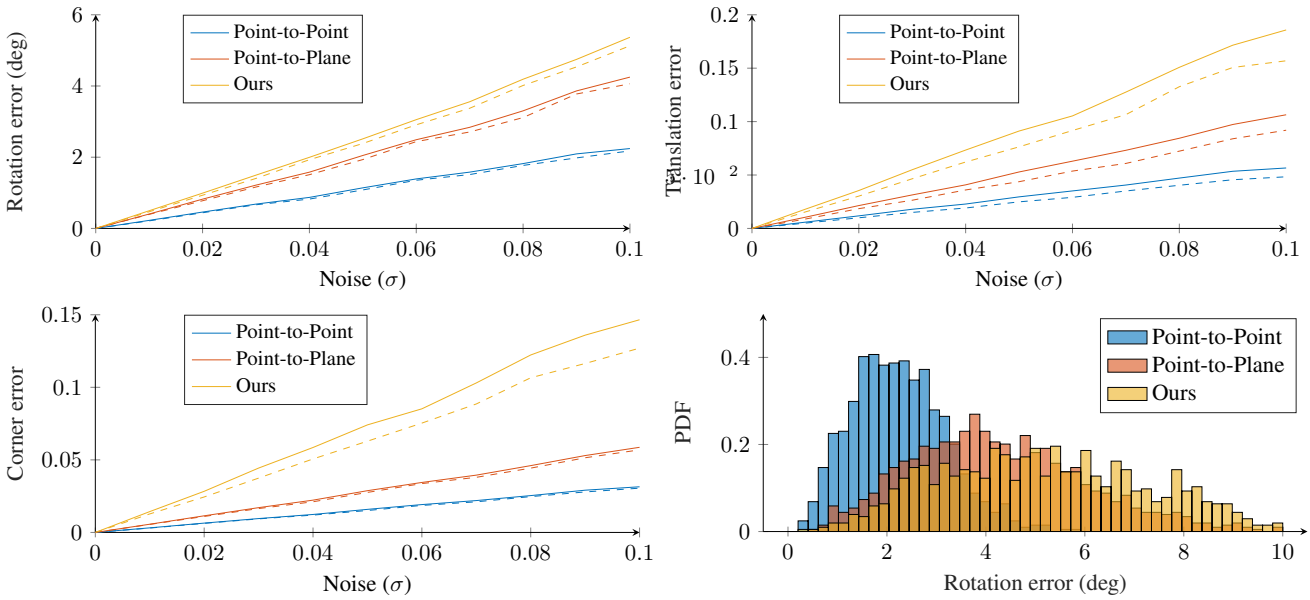

Figure 2. **Evaluation on Synthetic Data.** The plots show the average and median (*dashed*) error for various noise levels. For details on the experimental setup see the main paper. We also report the *corner error* as defined in Section 5. The bottom right figure shows a histogram of the rotation errors for = 0.1.

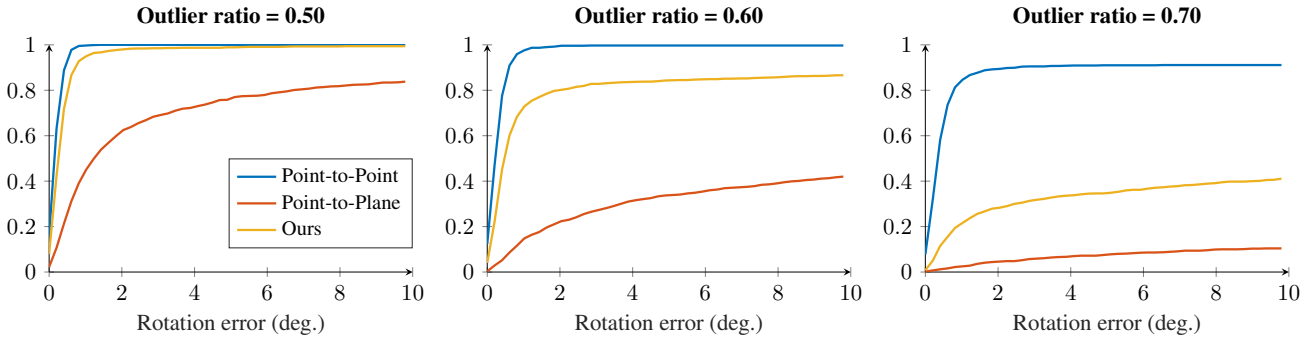

Figure 3. **Evaluation on Synthetic Data with Outliers.** The figures shows the recall curve for the rotation error with fixed outlier ratios (50%, 60%, and 70%). See the main paper for details on the experimental setup.

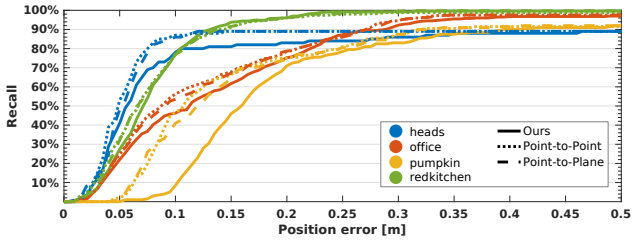

Figure 4. Precision/recall for the remaining *7 scenes* scenes.

do not have a solver available for this case. For consistency, we use the same RANSAC inlier thresholds as for the experiments with known scale. Table 2 shows the recall for the different error thresholds. Additionally estimating the query scale significantly limits the pose accuracy. Note, however, that in most cases the method does not fail completely. Most queries can still be localized within a few meters and the method can therefore still be used for approximate localization. For the tested datasets our method achieved a recall of at least 57% and 73% for error thresholds of $(5m/10^\circ)$ and $(10m/10^\circ)$, respectively. With the additional position refinement, this increased to 79% and 83%. Interestingly, the pose accuracy of the point-to-point method even increases when allowing to change the scale, indicating that this can account for noise in the data. (Note that the estimated scales with this method still lie very close to the ground truth and we therefore do not expect a problem with our experiment data.) Allowing the additional degree of freedom leads to slightly increased errors in the rotation orthogonality, *i.e.*, we observed larger differences between

Scene	Method								
	Point-to-Point [4]			Ours			Ours + Refinement		
	τ_1	τ_2	τ_3	τ_1	τ_2	τ_3	τ_1	τ_2	τ_3
Alamo	40.9	82.9	93.8	3.7	13.1	23.4	5.6	18.5	28.9
Gendarmenmarkt	17.6	57.7	77.1	5.0	22.0	41.8	2.1	22.1	47.8
Madrid Metropolis	15.7	57.7	74.5	3.3	17.5	40.5	1.8	28.8	52.2
Roman Forum	25.4	69.4	84.5	5.5	24.7	42.9	6.1	28.3	50.3
Tower of London	23.2	63.4	80.3	5.9	24.0	47.0	5.7	31.9	55.6
τ_1 = (0.05m / 2) τ_2 = (0.2m / 5) τ_3 = (0.5m / 10)									

Table 2. **Results with unknown query scale:** Estimating the scale together with the pose removes constraints in the partial localization problems. This leads to increased noise in the estimated rotations. We can improve the result by querying just the positions with a scaled and aligned query cloud after merging the partial rotations, denoted as *Ours + Refinement* in the table. However, this could leak additional information to the servers.

the estimated partial rotations and the corresponding rows of the projected rotation matrix. We can improve the accuracy for the medium and large error threshold by estimating just the 1D positions again after assembling the full rotation matrix and aligning the query to the map accordingly. However, this might provide additional information to the server which is not desired. Also, it reduces the recall for very small error thresholds. This is most likely due to adding an additional approximation even for the easy cases that already provide high quality positions after the initial query.

9. Qualitative Analysis of Disjoint Queries

We perform an additional qualitative analysis of the disjoint query experiments performed with the RobotCar [1] dataset. Figure 5 shows the obtained 2D positions with different query distances. The queries were performed by either splitting the query point cloud of a single pose, or by using point clouds corresponding to poses either 4m, 10m, or 50m driven distance apart for the two partial queries. We then used the INS data provided by the dataset to fuse the results. It is clear that the quality of the estimated poses degrades with larger driven distance. While the single pose queries are generally very close to the ground truth and only experience some noise in different parts of the map, queries with 50m distance in between show clear systematic errors in most areas.

10. High Quality Trajectory Experiments

When using multiple query images and assembling their geometric constraints using estimated relative poses, the achieved localization accuracy highly depends on the accuracy of those relative poses. In the experiment presented in the main paper we used the raw INS data that is provided as part of the RobotCar [1] dataset. We observed that these poses are often not accurate enough to show the full potential of our method. We therefore performed two additional experiments to minimize the impact of these inaccuracies. (1) We initialize the trajectories between both query images using the INS data, but then perform a refinement step based on the visual correspondences between the images of the query sequence. (2) We directly use the ground truth image poses to compute the relative pose to eliminate the impact of the trajectory estimation completely and present an upper bound on the localization accuracy. The precision-recall plots are shown in Figure 6. Using the refined poses we can achieve a significantly higher recall with small errors, but this difference decreases with increasing errors. When using ground truth relative poses this effect is even stronger. The lower recall within 20 difference of our method compared to the point-to-point method can in parts be explained with the higher number of required constraints. Our method requires at least 3 correspondences in each query image, whereas point-to-point only requires 3 correspondences in total.

11. Datasets

We used three publicly available datasets in the presented evaluations. The 7 scenes dataset[1] [2] is licensed under a Microsoft Research License Agreement (MSR-LA). The RobotCar dataset[2] [1] is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0). For the 1DSfM dataset[3] [5] the authors do not specify a license.

References

[1] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017. 4

[2] 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 *Computer Vision and Pattern Recognition (CVPR)*, 2013. 2, 4

[3] Pablo Speciale, Johannes L. Schönberger, Sudipta N. Sinha, and Marc Pollefeys. Privacy preserving image queries for camera localization. In *International Conference on Computer Vision (ICCV)*, 2019. 2

[4] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 1991. 2, 4

[5] Kyle Wilson and Noah Snavely. Robust global translations with 1DSFM. In *European Conference on Computer Vision (ECCV)*, 2014. 2, 4

[1] https://www.microsoft.com/en-us/research/project/rgb-d-dataset-7-scenes/

[2] https://robotcar-dataset.robots.ox.ac.uk

[3] http://www.cs.cornell.edu/projects/1dsfm/