Supplementary Material: Privacy Preserving Image Queries for Camera Localization

Pablo Speciale[1,2] Johannes L. Schönberger[2] Sudipta N. Sinha[2] Marc Pollefeys[1,2]

[1] ETH Zürich [2] Microsoft

In this supplementary document, we provide several visualizations of the datasets used in our experiments. We also include additional experimental results and implementation details that could not be included in the main paper.

1. Dataset Visualization

We captured 15 datasets, of which seven were recorded with various mobile phone cameras, and eight were acquired with the Microsoft HoloLens using the Research Mode capability[1]. We also used four public Internet photo collection datasets. In Figure 1, we visualize the 3D point cloud reconstructions of the scenes. Also in Figure 2, we show a few query images from our datasets and the corresponding 2D feature points and 2D feature lines.

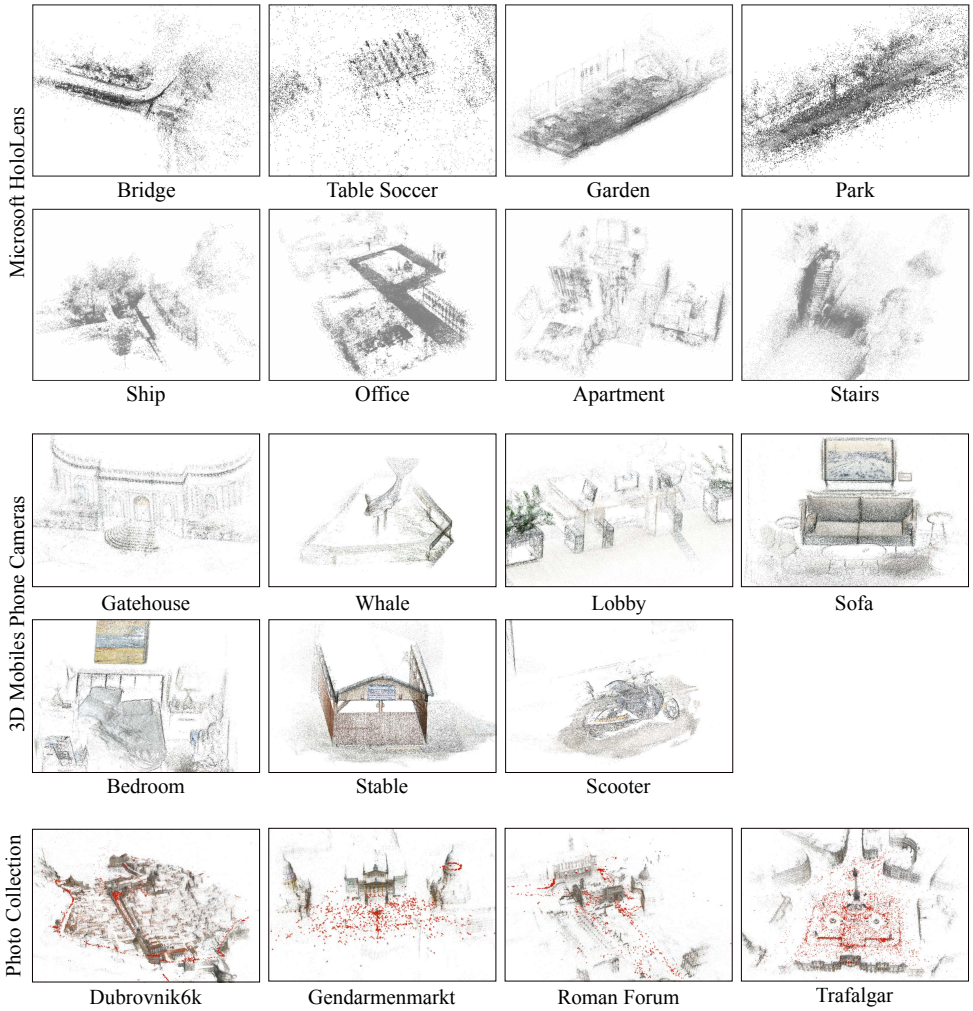

Figure 1: Datasets. Point cloud reconstructions of the 19 scenes used in our evaluations. The four reconstructions shown in the bottom row are from publicly available Internet photo collection datasets. For those scenes, the estimated camera positions are visualized in red.

[1] https://docs.microsoft.com/en-us/windows/mixed-reality/research-mode

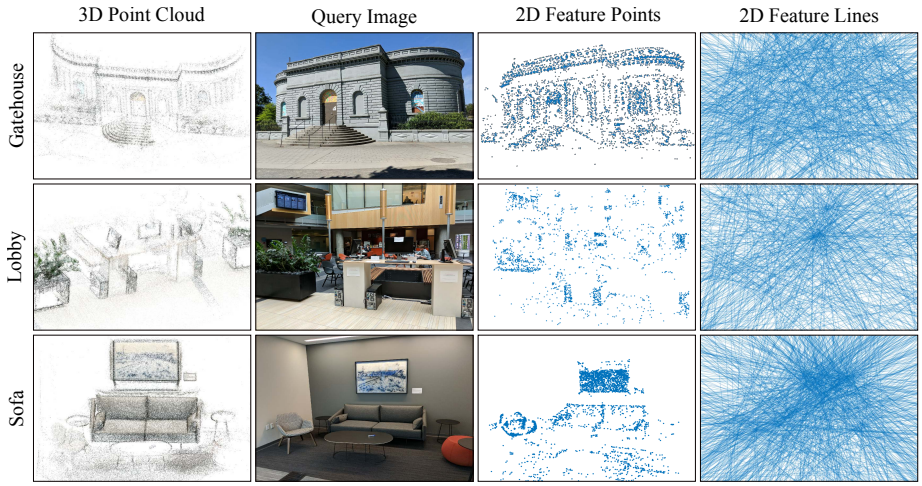

Figure 2: Query Images. Three examples of query images from various datasets. From left to right: a visualization of the pre-computed 3D point cloud of the scene, a query image, the associated 2D feature point locations and the corresponding 2D feature lines in the image.

Figure 3: Image Inversion Attacks. Six examples of query images and inversion attacks on the DUBROVNIK6K dataset. For each example, shown from left to right are the original image and the images reconstructed using all the SIFT descriptors and using only the inlier SIFT descriptors (their positions are revealed during our pose estimation method). People present in the scene are well concealed.

2. Image Feature Inversion attack

Figure 3 shows several examples of query images from the DUBROVNIK6K dataset. In each case, the results of running inversion attacks using a trained CNN model similar to the one proposed by Dosovitskiy and Brox [1] are also shown. Specifically, our CNN model takes the 2D locations of the SIFT features and the 128-dimensional descriptors as input and outputs a color image. These examples demonstrate that when all the 2D feature locations are known, the reconstructed image reveals considerable detail about the scene including the identity and other attributes of the people in the scene. In contrast, when the inversion attack is done on the subset of inlier features, whose 2D positions are revealed during our pose estimation method, only some parts of the background scene are revealed and information about the people who were observed in the query images or other transient objects in the foreground are effectively concealed.

POINT TO POINT (Traditional)			LINE TO POINT (Privacy Preserving)		
***p*3P**	**i:** 31 **r:** 64 **s:** 2.05 **t:** 3.54	***p*2P+u**	**i:** 15 **r:** 64 **s:** 2 **t:** 3.21	***l*6P**	**i:** 54 **r:** 69 **s:** 3.95 **t:** 28.6	***l*4P+u**	**i:** 24 **r:** 70 **s:** 2 **t:** 5.82
m-*p*3P	**i:** 38 **r:** 64 **s:** 1.70 **t:** 3.81	**m-*p*2P+u**	**i:** 11 **r:** 65 **s:** 2 **t:** 3.16	**m-*l*6P**	**i:** 60 **r:** 65 **s:** 3.97 **t:** 36.4	**m-*l*4P+u**	**i:** 22 **r:** 69 **s:** 2 **t:** 6.79
m-P3P+λ	**i:** 23 **r:** 62 **s:** 1 **t:** 1.82	**m-P2P+λ+u**	**i:** 12 **r:** 62 **s:** 1 **t:** 0.97	**m-L4P+λ**	**i:** 26 **r:** 69 **s:** 1.47 **t:** 9.3	**m-L3P+λ+u**	**i:** 19 **r:** 70 **s:** 1.9 **t:** 1.61
m-P3P+λ+s	**i:** 24 **r:** 62 **s:** 1 **t:** 4.19	**m-P2P+λ+u+s**	**i:** 12 **r:** 62 **s:** 1 **t:** 2.37	**m-L3P+λ+s**	**i:** 18 **r:** 68 **s:** 2.00 **t:** 1.1	**m-L2P+λ+u+s**	**i:** 13 **r:** 68 **s:** 2 **t:** 1.17

Notation: **i:** mean number of iterations, **r:** inlier ratio [%], **s:** mean number of solutions, **t:** minimal solver time [ms].

***p*3P**	1.84 / 1.42	***p*2P+u**	1.88 / 1.43	***l*6P**	1.50 (3.61) / 1.06 (2.84)	***l*4P+u**	1.39 (3.12) / 1.10 (2.62)
m-*p*3P	2.06 / 1.51	**m-*p*2P+u**	1.88 / 1.51	**m-*l*6P**	1.54 (3.98) / 1.54 (2.96)	**m-*l*4P+u**	1.46 (3.17) / 2.22 (3.90)
m-P3P+λ	1.71 / 1.42	**m-P2P+λ+u**	1.62 / 1.42	**m-L4P+λ**	1.36 (4.65) / 0.86 (2.80)	**m-L3P+λ+u**	1.54 (4.88) / 1.25 (3.14)
m-P3P+λ+s	1.72 / 1.43	**m-P2P+λ+u+s**	1.63 / 1.41	**m-L3P+λ+s**	1.41 (4.18) / 0.57 (2.82)	**m-L2P+λ+u+s**	1.09 (3.86) / 0.56 (2.71)

Table 1: Additional Results. RANSAC statistics (top) and reprojection errors in pixels (bottom) for *initial/refined* results with traditional (Point to Point) and our proposed (Line to Point) methods. Point to point errors computed with known secret point locations are in brackets.

3. Additional Results

Implementation Details. When running structure from motion on our datasets, we used the default parameters of the COLMAP SfM pipeline. In our localization experiments, we use a residual threshold in RANSAC of 4 pixels in the image to decide whether a 2D–3D correspondence is an inlier. Note that in the traditional scenario, the residual is computed as the standard reprojection error (see Equation (2) in the main paper), whereas in our proposed privacy preserving scenario, the residual is computed as the geometric distance between the projected 3D point against the observed 2D line (see Equation (5) in the main paper). Following standard procedure, we determine the number of RANSAC hypotheses needed so that at least one outlier free minimal set of 2D–3D correspondences is sampled with a confidence of 99%.

Various Pose Estimation Statistics. Table 1 reports the mean count of the RANSAC iterations needed, inlier ratio, number of solutions in the minimal solver and time required to solve a single minimal problem. The results show that, while our method is slower than the baseline approach, the running times are small enough to make our method practical for real-time applications. Especially, the specialized solvers with known structure and gravity have competitive timings. We used the same RANSAC threshold for all the methods, but in practice this threshold could be made smaller for our methods, since the line-to-point error is almost always smaller than the point-to-point reprojection errors. This, and a slightly higher chance of including some outlier matches that accidentally lie along the 2D lines, leads to slightly higher inlier ratios for our method.

Measurement Noise Sensitivity. Figure 4 shows how the pose estimation accuracy varies when increasing levels of noise σ_x is added to the 2D feature locations. We observe a similar trend for our method and the conventional method.

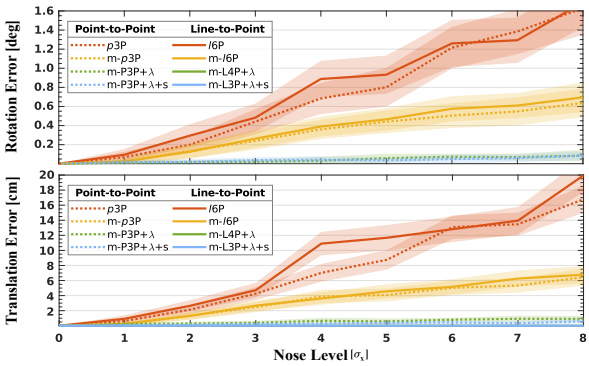

Figure 4: Noise Sensitivity. Rotation and translation errors obtained when Gaussian noise is added to the input 2D feature positions. The values of σ_x on the x-axis are in pixels. Both conventional methods (Point-to-Point) and our methods (Line-to-Point) perform similarly.

References

[1] A. Dosovitskiy and T. Brox. Inverting visual representations with convolutional networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2