Privacy Preserving Partial Localization

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

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

Abstract

Recently proposed privacy preserving solutions for cloud-based localization rely on lifting traditional point-based maps to randomized 3D line clouds. While the lifted representation is effective in concealing private information, there are two fundamental limitations. First, without careful construction of the line clouds, the representation is vulnerable to density-based inversion attacks. Secondly, after successful localization, the precise camera orientation and position is revealed to the server. However, in many scenarios, the pose itself might be sensitive information.

We propose a principled approach overcoming these limitations, based on two observations. First, a full 6 DoF pose is not always necessary, and in combination with egomotion tracking even a one dimensional localization can reduce uncertainty and correct drift. Secondly, by lifting to parallel planes instead of lines, the map only provides partial constraints on the query pose, preventing the server from knowing the exact query location. If the client requires a full 6 DoF pose, it can be obtained by fusing the result from multiple queries, which can be temporally and spatially disjoint. We demonstrate the practical feasibility of this approach and show a small performance drop compared to both the conventional and privacy preserving approaches.

1. Introduction

Over the last years, an increasing number of industrial solutions have emerged for cloud-based localization and mapping in mixed reality and robotics (*e.g.*, Microsoft Azure Spatial Anchors [23], Facebook LiveMaps [1], or Google VPS [35]). The need for cloud-based solutions is primarily motivated by the requirement for scalability as well as to enable shared experiences and crowd-sourced mapping. This trend, however, raises a host of privacy concerns [25, 33, 36, 58], as localization and mapping systems typically rely on camera images as the primary sensory information. The first works on tackling the privacy issues are based on the principle of lifting traditional point-based features to lines to conceal the appearance of images and maps [12, 17, 50, 53, 54].

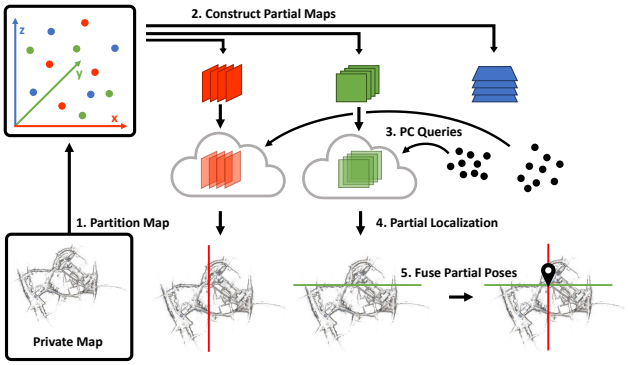

Figure 1. **System architecture:** We lift the original point cloud map to privacy-preserving parallel plane maps to only allow constraints in a single direction. This can be duplicated with multiple maps and localization services to accumulate different constraints locally. The clients send their localization queries to each server, which compute partial camera localization poses. Only the client can assemble the full localization result from the partial answers. Although we need three orthogonal maps for full 6 DoF pose estimation, additional constraints, such as localization in the ground plane, can reduce this requirement.

As already mentioned in the first works by Speciale *et al*. [53, 54], their proposed privacy-preserving representation of maps comes with two fundamental limitations. First, without careful construction, their maps are easily attackable using a density analysis on the line or plane clouds, as recently also studied in more detail by Chelani *et al*. [8]. Sparsifying the line cloud is one mitigation to this problem [8, 53] but comes with significant performance tradeoffs in terms of recall and accuracy of the localization results. Second, after successful server-side localization in privacy preserving line cloud maps, the client reveals their precise camera location in the scene. In many scenarios, the user's location itself is sensitive information that should be protected [2, 4, 16, 55], especially when tracked over time. The method presented in this paper is orthogonal to previous works in the sense that it explicitly ignores protection of the scene appearance, but prevents leaking the user's precise location to the server. To this end, we accept the potential privacy risks of sending a 3D query point cloud that were previously discussed and tackled [34, 53].

An important observation is that, depending on the application, it is often not even necessary to estimate the full 6 degree of freedom (DoF) pose all the time. While applications like augmented reality can only function with the a fully known pose, many common localization tasks are solved on a restricted manifold. As a classical example, navigation for autonomous driving is generally restricted to the ground plane and therefore rarely benefits from also estimating the vertical component. Furthermore, local egomotion tracking is a crucial part of many real-time localization systems, where the tracking-based pose estimate is globally corrected using less frequent full localization queries. In these cases, the current pose is generally already known up to some uncertainty. By only correcting the pose estimate in one dimension at a time, the potentially revealed pose information is significantly reduced. In the scenarios where we do not require full 6 DoF poses, we can further push the ideas of Speciale *et al*. [53, 54] and remove information from our map representations. Our main idea is to remove the constraints along two dimensions in the map, so that each query can only be localized along the remaining dimension. Instead of lifting each point to a randomly oriented line, we add another degree of freedom and lift the points to parallel planes. As such, we can only observe the motion orthogonal to the planes. Estimating the pose against this map using a local point cloud allows us to determine three of the six degrees of freedom, namely one row of the rotation matrix and a single translation component.

Combining multiple partial queries in different directions enables us to obtain a 2D or full 3D pose. To maintain the privacy of the queries, we assume that these cannot be co-registered on the server-side. This can be achieved by distributing the partial maps to different service providers that must not be able to associate corresponding queries (*e.g*., must not communicate or cooperate with each other). A practical example could be a theme park aiming to provide visual localization for AR experiences to their guests using external infrastructure. The park would be responsible for creating the plane maps and distributing them to three independent cloud service providers. Co-registration can also be hindered client-side by submitting temporally or spatially disjoint queries, where only the client knows the relative pose between them (*e.g*., obtained via local egomotion tracking). In some settings, other client-side information allows for recovering the full pose from the 1D localization. For example, a car driving in a GPS-denied urban canyon might know which street it is currently on, but not its exact position. The remaining degree of freedom can be recovered using our method with a single partial query.

In contrast to the random line clouds presented in [53], our maps can be thought of as a set of parallel planes. This representation is inherently safe from density-based attacks as proposed by Chelani *et al*. [8]. With this work, we hope to make another important step to allow for a widespread adoption of cloud localization services in mixed reality and robotics.

In summary, this paper makes the following contributions: **(i)** We present a principled approach with stronger guarantees on the privacy-preservation of the map by preventing known vulnerabilities of the existing approaches. **(ii)** Our work is the first to provide location privacy in the domain of image-based localization. **(iii)** Extensive experiments on real-world datasets show only minor accuracy and recall trade-offs compared to the previous approaches. This underlines the high practical relevance of our work.

2. Related Work

Image-based Localization The state-of-the-art methods in image-based localization have reached an impressive level of maturity. Recent work focuses on improving the robustness to drastic appearance and illumination changes [3, 45, 56], scalability to large spaces [27, 38, 40, 62], and real-time capability on mobile devices [5, 20, 24, 27, 28, 39]. Other recent work studied the problem of finding compressed map representations [7, 14] or enabling cross-device localization and mapping [11]. The existing approaches can be broadly characterized into structure-based methods [20, 38–40] relying on an explicit 3D map representation and retrieval [21, 52] or learning-based methods [6, 24, 48]. Typically, only structure-based approaches provide sufficient accuracy to enable mixed reality and robotics applications [42, 59]. While most current localization systems aim to recover a full 6 DoF pose from a set of correspondences, sequences of individually insufficient constraints have been used to estimate a full pose as well [60].

Image Privacy A serious privacy risk induced by image-based localization systems is due to the reliance on capturing image information to perform the localization task. This becomes especially problematic when the images are shared with other devices or cloud-based localization and mapping services [25, 33, 36, 58]. This is also the case when only abstract feature representations are used in these systems, as model inversion techniques can easily recover the original image content from (sparse) image features [10, 32, 34].

Speciale *et al*. [53, 54] was the first to address the privacy problems in image-based localization by obfuscating the geometry of 2D image or 3D map points in structure-based approaches. Their main objective was to mitigate the vulnerability of the traditional methods to model inversion techniques [32, 34]. Our approach is based on the same principle but overcomes two of its main limitations. Meanwhile, several follow-up works also extended upon their original idea to address the full structure-from-motion [17, 18] and the real-time SLAM problem [50]. In addition, Dusmanu *et al*. [12] showed how a similar idea can also be applied to

improve the privacy of local and global image descriptor representations, which are the backbone of most localization systems. Recently, Shariati *et al*. [49] explored the potential of low-resolution cameras as an alternative to solving the privacy problems associated with recording images.

Despite this tremendous progress, the recent privacy-preserving approaches are still vulnerable to leaking some amount of image information, which has already been pointed out as a limitation in their original work [54]. Recently, Chelani *et al*. [8] studied this problem in more detail. Their work only studies attacks on line clouds but is, in principle, applicable to random plane clouds as well. They conclude that, without careful sparsification of the 3D map representation, the original point-cloud can be easily reconstructed using a density attack on the line cloud representation. Their experiments also show that a sufficient level of sparsification comes with significant trade-offs in terms of accuracy and recall of the localization results. Our approach is safe from density-based attacks on the 3D map, as each partial map is a 1D representation of the scene. In detail, we break their underlying assumption [8] of uniform randomly distributed 3D line or plane orientations. Instead, the planes in each partial map are disjoint sets and all of them have the same orientation. Attacking our approach will be significantly harder, as it requires the inversion of the full 3D map representation from a 1D projection.

Location Privacy A general privacy issue with any location-based service concerns the tracking of user behavior by analyzing their location over time. This is especially concerning in the setting of image-based localization, where a service provider knows the precise 6 DoF pose and not only the user's approximate location (*e.g*., when using GPS). Some existing works based on differential privacy [13] and k-anonymity [37] have been applied to the general problem of location privacy [2, 4, 16, 55]. These approaches, however, come with significant trade-offs in terms of location accuracy, which is generally not acceptable in the mixed reality or robotics setting, where a precise 6 DoF camera pose is required. In our work, we are the first to tackle the problem of location privacy based on idea of geometric lifting proposed by Speciale *et al*. [53, 54] in the context of scene and image appearance hiding, which enables us to recover precise 6 DoF camera poses

Federated Learning Federated learning [22, 26] in the machine learning community is related to our approach, in that one of its goals is to preserve the privacy of the training data. In particular, the underlying principle of federated learning is to distribute training data across different machines in a data center, such that each machine only has access to its part of the data. Similarly, our approach relies on the distribution of 1D maps across different service providers, where each partial map has much stronger privacy guarantees as compared to storing all of them on the same machine. The advantages for user privacy are significant. First, the data is better protected from attackers that gain access to part of the data. Second, by implementing the proposed distribution idea across different service providers, user privacy is also further increased against the service providers.

3. Method

In this section, we first give an overview of the system and describe the architecture in Section 3.1. Next, we describe the detailed steps to create partial maps in Section 3.2 before we explain how to perform localization against a partial map in Section 3.3.

3.1. Overview

The main idea of our method is to remove the geometric constraints along two of the three dimensions in the map. Consequently, a query can only be localized along the remaining dimension. This effectively hides the client pose from the localization service. If a full 6 DoF pose is required by the client, it can be fused from the results of multiple independent queries to maps constructed with different plane normals. In order to only provide the cloud service with the absolutely necessary information, the maps need to be created by the client or a trusted third party before being shared with the localization service. An illustration of our system architecture is shown in Figure 1. We assume that the client is able to create a small, local 3D reconstruction of the scene for use as a query. This can be done either by active depth sensing, or by running a Structure-from-Motion or SLAM algorithm locally on the device.

3.2. Partial Map

A map in our system only contains the original point offsets along its assigned direction (this is illustrated in Figure 2). Geometrically, one can think of the map containing a set of parallel planes, which pass through the original 3D map points. This is similar to the plane maps proposed in [53] except that they are parallel instead of randomly oriented. As we show in Section 3.3, this only allows the server to partially recover the pose of the query while the full pose is not observable.

3.2.1 Map Creation

For the map creation, we start from a standard 3D point cloud reconstruction of the scene. This initial map is privacy sensitive, *i.e*., our proposed map creation process must be carried out either by the client itself or by a trusted third party. For each 3D point $\boldsymbol{X}_m = (x_m, y_m, z_m)$, we simply throw away two of the three dimensions and only store the

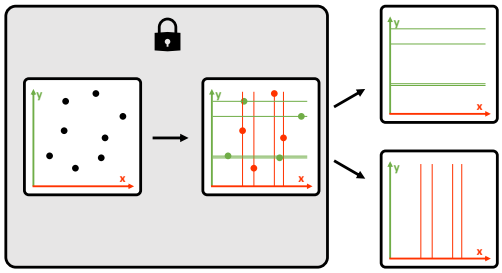

Figure 2. **Disjoint map creation:** To generate multiple, disjoint maps we split the original point cloud and project each point to its assigned coordinate axis. Afterwards, we distribute only the computed plane offsets along the coordinate axes to the servers. The original point map could reveal private information, so the map generation needs to be performed in a trusted environment. We show the 2D case for simplicity, but the same process trivially generalizes to 3D.

offsets along the remaining one, together with the original point descriptors. Note that the choice of coordinate system of the map impacts the pose ambiguity since it defines the localization directions. We visualize this ambiguity in the supplementary material.

To ensure privacy with multiple maps, they need to be carefully created such that they do not reveal additional information if combined. We therefore split the original point cloud into disjoint sets, each being projected onto a different coordinate axis. This prevents a potential attacker with access to all maps from gaining information by intersecting the plane maps later on. We use random sampling to select the points for the different sets so that each partial map still covers the entire scene. This process is shown in Figure 2. Note that due to the disjoint point sets, even when combined the maps are also not vulnerable to the density-based attacks recently proposed by Chelani *et al*. [8].

3.3. 1D Localization

To localize a query point cloud, we assume that each 3D query point is associated with a feature descriptor, which allows to establish tentative correspondences to the map planes using standard descriptor-based matching approaches. In the following derivation, we assume that map and query point cloud have consistent scale, which is a practical assumption for most SLAM systems. We provide variants of the solver for the special cases of unknown scale of the query and known gravity direction at query time in the supplementary material.

In detail, assuming a point cloud map, we can split the constraint of a 3D pose into three separate 1D constraints

$$R\boldsymbol{X}_q + \boldsymbol{t} = \boldsymbol{X}_m \implies \begin{cases} \boldsymbol{r}_1^T \boldsymbol{X}_q + t_1 = x_m \\ \boldsymbol{r}_2^T \boldsymbol{X}_q + t_2 = y_m \\ \boldsymbol{r}_3^T \boldsymbol{X}_q + t_3 = z_m \end{cases} \tag{1}$$

where $\boldsymbol{X}_q$ and $\boldsymbol{X}_m$ are corresponding 3D points in the query and map respectively. The key insight from Eq. (1) for our method is that the estimation problem separates into three constraints for each map coordinate $\boldsymbol{X}_m = (x_m, y_m, z_m)^T$ and we can use only one of the equations for each correspondence. Consequently, given our 1D map and a query point cloud with associated feature descriptors, we are able to recover the query position along the localization direction and a partial orientation, namely the corresponding row of the rotation matrix. The server then establishes tentative correspondences with the map based on descriptor matching. From these correspondences, the server estimates the corresponding partial pose $(\boldsymbol{r}_k^T, t_k)$. Due to imperfect descriptor matching, we use LO-RANSAC [9] as a robust estimator to deal with potential outlier correspondences. Note that the server can only observe $(\boldsymbol{r}_k^T, t_k)$ and thus only partially knows the position of the query. Given $(\boldsymbol{r}_k^T, t_k)$ there exist three degrees of freedom left in $[R\ \boldsymbol{t}]$, two in the position (the remaining translation parameters), and one rotational degree of freedom. Furthermore, our formulation of the partial pose constraint is independent of the plane normal direction. As a consequence, the server can be agnostic to its assigned coordinate axis and only store the plane coordinates as simple scalar distance to the origin.

3.3.1 Minimal Solver for 1D Localization

Each localization server needs to estimate one row of the transformation $[R\ t]$. Denote this row $(\boldsymbol{r}, t)$, where $\boldsymbol{r}$ is a unit vector. These should then satisfy

$$\boldsymbol{r}^T \boldsymbol{X} + t = x \tag{2}$$

for an inlier corresponding point $\boldsymbol{X} \in \mathbb{R}^3$ in the query and $x \in \mathbb{R}$ in the map. We have 3 degrees of freedom, since we can only enforce the unit-norm constraint on one row of the rotation $\boldsymbol{r}$, because each row is estimated independently.

Given three correspondences, we can minimally solve for $(\boldsymbol{r}, t)$, collecting three correspondences as

$$\begin{bmatrix} \boldsymbol{X}_1^T & 1 \\ \boldsymbol{X}_2^T & 1 \\ \boldsymbol{X}_3^T & 1 \end{bmatrix} \begin{pmatrix} \boldsymbol{r} \\ t \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \ . \tag{3}$$

Ignoring the norm-constraint, the solutions to the underdetermined linear system above can be written as

$$\begin{pmatrix} \boldsymbol{r} \\ t \end{pmatrix} = \begin{pmatrix} \boldsymbol{a} \\ \alpha \end{pmatrix} + \lambda \begin{pmatrix} \boldsymbol{n} \\ \nu \end{pmatrix}, \quad \lambda \in \mathbb{R} \ . \tag{4}$$

The norm constraint now yields a quadratic equation in λ,

$$\boldsymbol{r}^T \boldsymbol{r} = \boldsymbol{a}^T \boldsymbol{a} + 2\lambda \boldsymbol{a}^T \boldsymbol{n} + \lambda^2 \boldsymbol{n}^T \boldsymbol{n} = 1 \ . \tag{5}$$

Solving this we can recover two solutions for $(\boldsymbol{r}, t)$.

3.3.2 Least Squares Fitting

We now consider the case where we want to fit a model to a non-minimal (> 3) number of correspondences, *i.e.*

$$\min_{\boldsymbol{r},t} \sum_k \left(\boldsymbol{r}^T \boldsymbol{X}_k + t - x_k\right)^2 \quad \text{s.t.} \quad \boldsymbol{r}^T \boldsymbol{r} = 1 \ . \tag{6}$$

The optimal t w.r.t. $\boldsymbol{r}$ is then $t^\star = \frac{1}{n}\sum_k (x_k - \boldsymbol{r}^T \boldsymbol{X}_k)$. Inserting this into (6), the problem can be rewritten as

$$\min_{\boldsymbol{r}} \|A\boldsymbol{r} - \boldsymbol{b}\|_2^2 \quad \text{s.t.} \quad \boldsymbol{r}^T \boldsymbol{r} = 1 \ , \tag{7}$$

where the rows of A and $\boldsymbol{b}$ are

$$A_i^T = \boldsymbol{X}_i^T - \frac{1}{n}\sum_k \boldsymbol{X}_k^T, \quad b_i = x_i - \frac{1}{n}\sum_k x_k \tag{8}$$

Now if $\boldsymbol{b}$ was zero, the solution could easily be found using SVD. For the in-homogeneous case the problem is more challenging but can be solved using a Lagrangian formulation. See Gander [15] for more details about norm-constrained least squares problems. For (7) the Lagrangian is then $\mathcal{L} = \|A\boldsymbol{r} - \boldsymbol{b}\|_2^2 + \lambda(\boldsymbol{r}^T\boldsymbol{r} - 1)$. Differentiating w.r.t. $\boldsymbol{r}$

$$\nabla_{\boldsymbol{r}} \mathcal{L} = 2A^T A\boldsymbol{r} - 2A^T\boldsymbol{b} + 2\lambda\boldsymbol{r} = 0 \ , \tag{9}$$

allows us to solve for $\boldsymbol{r}^\star(\lambda) = M_\lambda^{-1} A^T \boldsymbol{b}$, where

$$M_\lambda = \left(A^T A + \lambda I\right) \ . \tag{10}$$

Now inserting this into the norm constraint yields

$$\boldsymbol{b}^T A^T M_\lambda^{-T} M_\lambda^{-1} A^T \boldsymbol{b} = 1 \ , \tag{11}$$

which is a rational equation in λ. From this we get a degree 6 polynomial in λ as

$$p(\lambda) = \boldsymbol{b}^T A^T \mathrm{adj}(M_\lambda)^T \mathrm{adj}(M_\lambda) A^T \boldsymbol{b} - \det(M_\lambda)^2. \tag{12}$$

Finding the roots of this polynomial, we can recover λ and, by back-substitution, we can recover the corresponding $(\boldsymbol{r}, t)$. The correct root can then be selected by evaluating the original cost function (6) and taking the minimizer.

3.4. Combining multiple 1D localizations

If 1D localization is not sufficient, the client can reconstruct the 3D pose $[R \ \boldsymbol{t}]$ by stacking the results from three independent queries to maps with different directions, as shown in Eq. (1). We explain how to combine results from non-orthogonal maps in the supplementary material. Since the estimation on each server is carried out independently, the pairwise orthonormality constraints on the rotation matrix rows cannot be enforced. Therefore, the client projects the returned rows onto the closest valid rotation matrix using SVD [19]. In experiments we show that, in practice, this has negligible impact on the accuracy of the final result. Finally, if an accurate and reliable source of odometry is available, we can even combine partial localizations over time by propagating the partial poses to a common coordinate frame. We show the feasibility of this approach in our experiments. Note that if 2D localization is sufficient we can still recover the full 3D orientation from two queries due to constraints on the rotation matrix.

4. Experiments

In this section, we experimentally validate the accuracy and robustness of our proposed solution. When comparing to standard, full pose estimation methods we query the same pose in all three directions independently and combine the results afterwards. The results of our evaluations show little performance loss over its baselines and thus underline the practical relevance of our approach.

4.1. Synthetic Data

We first validate our approach on synthetic data. We generate synthetic maps by uniformly sampling 100 3D-points in the unit cube $[0, 1]^3$. These points are then randomly rotated and translated to create the query point cloud. To each query point, we add zero-mean Gaussian noise. Outlier correspondences are created by randomly resampling a subset of the map points. Finally, we generate three 1D maps, as described in Section 3.2.1, by randomly splitting the map into three equal parts. Figure 3 shows the distribution of the equation residuals after applying the minimal (Section 3.3.1) and non-minimal solver (Section 3.3.2) to 10,000 synthetic instances. Next, we compare applying the proposed localization method with direct point-to-point alignment, as well as the random-plane method from Speciale *et al*. [54]. For each method, we use LO-RANSAC [9] with a fixed number of iterations to estimate the transformation between the map and query. For our approach, we report the result of three independent 1D localizations. Figure 4 shows the average rotation error obtained for varying noise levels added to the query. For this experiment, we set the outlier ratio to zero. We see that the proposed approach is slightly more sensitive to noise in the query compared to the two competing methods. Note that $\sigma = 0.1$ is an extremely high noise level, corresponding to 10% of the point cloud extent. Next, we vary the outlier ratio and fix the noise level to $\sigma = 0.005$. Figure 5 shows the percentage of instances, where the rotation error is below 5 degrees. Note that the method from Speciale *et al*. [54] needs to sample six plane-point correspondences in each iteration, making it more sensitive to outliers. The proposed approach only requires three points (same as for point-to-point alignment), but instead needs to perform three independent RANSAC optimizations, which reduce the overall robustness. More synthetic results can be found in the supplementary.

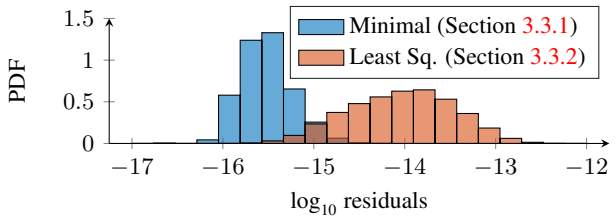

Figure 3. **Numerical stability:** The figure shows the $\log_{10}$ equation residuals for 10,000 synthetic instances.

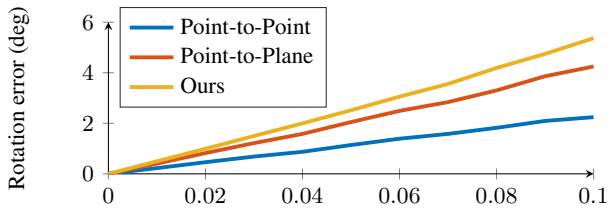

Figure 4. **Noise sensitivity:** The average rotation error (degrees) versus standard deviation. Note that the width of the point cloud is one.

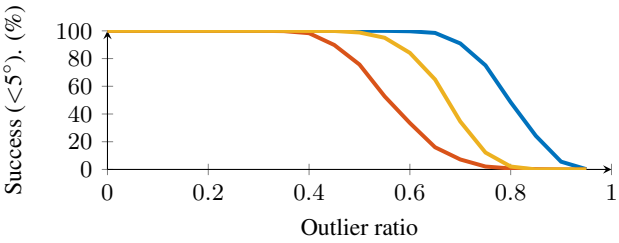

Figure 5. **Outlier robustness:** The percentage of successful instances (less than 5 degree rotation error) versus the outlier ratio.

4.2. Real Data

In this section, we evaluate our proposed solution on real localization datasets. We assume that the client is either equipped with a SLAM system or has depth sensing hardware to obtain the client-side query point clouds, and we evaluate both setups in different scenarios. First, we consider the case of large-scale outdoor localization based on multiple internet image collections datasets [61]. In this scenario, specialized sensors are not commonly used, and active depth sensing in consumer devices tends to fail in outdoor environments. We therefore build the queries from multiple images using a Structure-from-Motion pipeline [44, 47]. Second, we evaluate on the *7 scenes* [51] dataset consisting of 7 different indoor scenes captured with a Microsoft Kinect as an RGB-D sensor.

For both scenarios, we compare our method to two different baselines: traditional point cloud alignment [57] (*Point-to-Point*) and the privacy preserving solution of aligning the query points to a map of randomly oriented planes, as presented by Speciale *et al*. [54] (*Point-to-Plane*). Note that for the Point-to-Plane [54] approach, the plane map is vulnerable to density attacks [8] and, upon successful localization, the client pose is revealed to the server.

We report pose accuracies as the recall within different error thresholds. We use three different combinations of orientation and position error thresholds: (0.05m, 2°), (0.2m, 5°), and (0.5m, 10°).

4.3. Structure-from-Motion Queries

Dataset For the Structure-from-Motion evaluation we rely on the well-known 1DSFM dataset by Wilson and Snavely [61]. The scenes *Alamo* (703 images), *Gendarmenmarkt* (825 images), *Madrid Metropolis* (279 images), *Roman Forum* (1275 images), and *Tower of London* (577 images) cover tourist attractions around the world and are crowd-sourced from the internet. Additionally, we use the city-scale datasets *Aachen* [41, 43, 63] (6697 images) and *Dubrovnik6K* [28] (5856 images) to showcase the applicability of our method for even larger scenes. For each scene, we first generate a complete model using the COLMAP [44, 46] Structure-from-Motion pipeline to obtain pseudo-ground-truth poses. We then manually scale the ground-truth models to approximately metric scale to provide meaningful error measures. Finally, we only consider registered images by COLMAP for our evaluation.

Setup To generate the evaluated multi-image queries, we first select a single image from the model. For the selected image, we find the 3 images with the most commonly observed points and a minimum baseline of 1m to the query image to enable triangulation. We subsequently use the query images to triangulate the query point cloud by optimizing the observed 3D structure only from constraints between the selected set of four images. In addition, we carefully remove the query images and all of their feature observations from the map and optimize the structure without these constraints. Finally, we use raw, pairwise SIFT feature matches [29] without two-view geometric verification to obtain correspondences between query and map. We select each image in the model once as the query image and choose the corresponding image set automatically.

Results For each method, we select the best RANSAC inlier thresholds by maximizing the area under curve (AUC) of the position precision-recall plot between errors of 0m and 5m. This strategy leads to thresholds 0.15m, 0.05m, and 0.25m for *Point-to-Point*, *Point-to-Plane*, and ours, respectively. Table 1 shows the recall for the different error thresholds. Overall our method provides comparable localization accuracy as the randomly-oriented plane maps from Speciale *et al*. [54], while providing better protection against density-based attacks and hiding the pose from the server. Still, our method exhibits higher variance in the accuracy, and recall for τ_3 is slightly below [54]. This is likely due to the higher number of required correspondences.

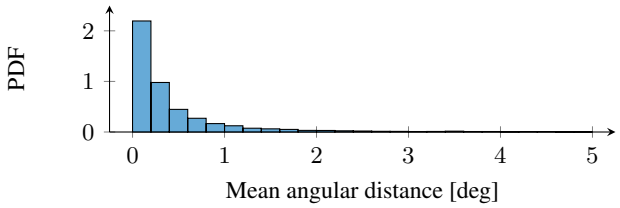

Figure 6. **Partial rotation errors:** We compute the angle between the partial rotation of each separate query and the corresponding row of the rotation matrix after projecting onto the manifold. The histogram is over the mean of the three errors over all images. Although we cannot enforce orthogonality constraints in the queries, the obtained rotation matrix rows are close to a proper rotation.

Scene	Method								
	Point-to-Point [57]			*Point-to-Plane* [54]			Ours		
	τ_1	τ_2	τ_3	τ_1	τ_2	τ_3	τ_1	τ_2	τ_3
Alamo	21.3	60.8	86.1	12.6	54.5	86.1	22.1	62.6	79.5
Gendarmenmarkt	7.7	40.8	72.4	4.6	33.4	61.6	5.6	31.1	56.7
Madrid Metropolis	4.7	32.1	68.2	1.8	23.0	59.9	6.6	36.1	62.8
Roman Forum	11.3	53.0	79.2	7.6	47.5	76.8	12.2	43.8	66.8
Tower of London	5.9	43.7	73.9	3.3	37.1	72.3	8.7	37.1	58.9
Aachen	12.4	76.4	95.5	8.6	67.7	91.6	12.0	57.9	81.5
Dubrovnik	4.8	35.3	60.8	2.7	27.1	54.6	3.9	24.0	45.7
τ_1 = (0.05m / 2°) τ_2 = (0.2m / 5°) τ_3 = (0.5m / 10°)									

Table 1. **Structure-from-Motion results:** Percentage of poses below different combined position and orientation error thresholds for the compared methods. The queries are built from multiple images using a Structure-from-Motion pipeline.

We also evaluate how close the estimated transform is to a proper rotation and translation. For each query we compute the angle between the rows of the rotation matrix before and after projecting with SVD. Figure 6 shows the distribution of the angles for the 1DSfM datasets. Even though the orthogonality constraints are not enforced during estimation, the composed rotation matrices are close to proper rotations, and the projection only yields small corrections.

4.4. 2D Trajectory Queries

Dataset We use the Oxford RobotCar dataset [30, 31] to highlight the special setup that emerges in localization for autonomous driving. We build a map from run *2014-12-09-13-21-02* and use images of run *2014-11-28-12-07-13* as queries. Hereby we ignore query images outside of the mapped area. We use all three wide-angle cameras for mapping, but only query with images of *mono_rear*.

Setup We first find feature matches between both runs using image retrieval and pairwise feature matching. Then we jointly optimize the structure and camera poses of both runs together to account for small errors in the provided calibration and ground truth poses. Afterwards we split the two runs and retriangulate all points separately while keeping the camera poses constant. We generate two plane maps

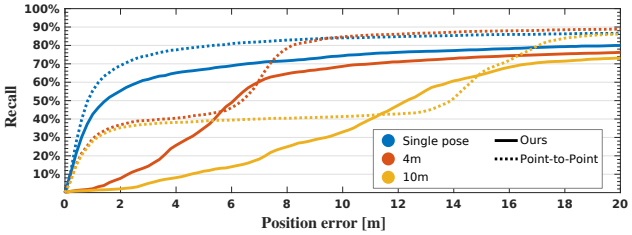

Figure 7. **Disjoint query precision-recall:** 2D position precision-recall plot with poses combined from two queries at the same pose, and with 4m and 10m distance between the queries using the RobotCar dataset. Relative poses between the two query locations are computed using the provided INS data. Point-to-Point alignment combines the constraints of both queries using the same relative pose estimate.

with orthogonal, horizontal plane normals and ignore the vertical component during localization. Note that we still estimate the 3 DoF orientation, although a single yaw angle would likely be sufficient. The two map queries can either be based on a single pose, or on two different locations, with the relative pose between two queries known from the provided INS data. We then combine the two partial poses into a 2D position and 3D orientation. As reference, we also combine all point-to-point constraints of the two images and perform standard point cloud alignment, but drop the point-to-plane method as second baseline.

Results Figure 7 shows the precision vs. recall curves of our method compared to point-to-point alignment from a single pose, or with 4m or 10m distance between the two queries. Compared to point-to-point alignment our method suffers more from noise in the relative poses, likely due to the inability to overfit to only one part of the query. We provide additional evaulation with high quality relative poses in the supplementary material.

4.5. Depth Sensor Queries

Dataset The *7 scenes* dataset [51] consists of high-framerate video sequences, capturing seven different indoor scenes. To reduce redundancy and simplify the experiments, we first downsample the frames, keeping only every tenth image for both the train and test sequences.

Setup We use the training images for building the map and evaluate the localization accuracy on the test images. We keep the train/test split as given by the dataset and use COLMAP to create the initial map. To obtain comparable poses for testing, we then use the ground-truth image positions to align the COLMAP model to the ground-truth coordinate system. As the dataset does not provide camera calibration data, we manually estimate the transformation between the depth and RGB camera and project the Kinect depth measurements into the color image. We build the 3D

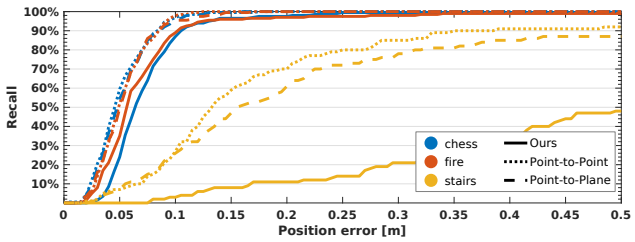

Figure 8. **Depth sensor precision-recall:** Position precision-recall plot for a subset of *7 scenes*. For most scenes accuracy and robustness of our method is comparable with the baselines. Please refer to the supplementary material for the remaining scenes.

query points by unprojecting the detected SIFT keypoints with valid depth measurements. To localize we again find correspondences based on unverified SIFT feature matches between the query and map images. We optimize the RANSAC inlier threshold for each method independently on the *RedKitchen* scene, maximizing the AUC of the position precision-recall plot up to an error of 0.1m (0.04m, 0.01m, and 0.06m for *Point-to-Point*, *Point-to-Plane*, and ours, respectively).

Results We show the percentages of poses within the three combined error thresholds in Table 2. A detailed distribution of the position errors for a subset of the scenes is shown in Figure 8. For the medium and large thresholds, our method's results are comparable to the baselines. For the *stairs* scene, our results are significantly worse than the baselines. This is explained by very few reliably matchable features in the scene and thus only few correspondences are found. The effect is also visible in the baseline poses, but our method suffers significantly due to only having roughly a third of the correspondences available for each dimension. This could be avoided by not dividing the map at the beginning and using all map points for each partial map. However, this would also have implications on the privacy aspect, as we discuss in Section 5. Interestingly, for some scenes our method achieves higher accuracy than the baseline methods. However, this is likely caused by the impact of the different thresholds for these particular scenes.

5. Discussion

With this work, we made another step towards providing strong privacy guarantees for image-based localization in cloud services. The benefits in data protection come at the cost of either obtaining fewer constraints on the pose or requiring multiple, independent services, together with a small reduction in localization accuracy and recall. The reduced accuracy mostly stems from the fact that only a fraction of the correspondences is available to each localization service, and that we cannot enforce the full rotation matrix constraints during the partial localizations. In the

Scene	**Method**								
	Point-to-Point [57]			*Point-to-Plane* [54]			Ours		
	τ_1	τ_2	τ_3	τ_1	τ_2	τ_3	τ_1	τ_2	τ_3
chess	46.5	99.5	100.0	38.0	99.5	100.0	24.0	97.5	99.5
fire	52.0	99.5	100.0	49.5	99.0	100.0	35.0	97.0	99.0
heads	33.0	76.0	89.0	26.0	80.0	89.0	39.0	77.0	82.0
office	18.5	64.8	97.5	20.8	61.3	97.0	16.8	59.0	95.2
pumpkin	2.0	51.5	90.5	1.0	53.0	89.5	0.0	51.0	90.5
redkitchen	23.4	83.6	98.8	20.8	82.8	99.8	26.6	87.0	99.6
stairs	7.0	62.0	89.0	5.0	41.0	80.0	0.0	11.0	48.0
τ_1 = (0.05m / 2°) τ_2 = (0.2m / 5°) τ_3 = (0.5m / 10°)									

Table 2. **Depth sensor results:** Percentage of poses below the error thresholds for the compared methods with the *7 scenes* dataset.

case of multiple services, we require that neither the partial maps nor localization results are shared between those services. Furthermore, an attacker must not gain access to, or intercept all servers. This is important to guarantee the protection of location privacy, as the full localization result can be composed from access to the partial maps or the partial responses. By only storing disjoint subsets of the full map in each 1D part, both trivial intersection and more sophisticated density-based attacks [8] do not apply.

In this work we ignored potential privacy violations from point clouds with feature descriptors as this was already approached in other works. However, the currently available methods can not be combined with our approach, so new methods will be required to protect both the scene appearance and client location. Further, we ignored any issues arising from scaling the system up to scene sizes that require pose priors (*e.g.*, GPS) to select map partitions for the local area. There are many closed-scene scenarios of limited size that do not require globally registered map, *e.g.*, theme parks. Even if selecting a local map partition is required, we believe that hiding the exact client pose within this area can still significantly benefit the client's privacy.

Directions for future research include a more optimal subset selection for the 1D maps considering uncertainty of the 3D points as well as finding an optimal orientation of the coordinate axes to maximize plausible camera poses.

6. Conclusion

We presented a new cloud-based localization approach that provides additional guarantees on the level of privacy preservation of the map representation, which overcomes one of the major limitations of the existing approaches [53, 54]. Furthermore, our approach is the first to provide location privacy in the context of image-based localization in mixed reality and robotics. By reducing the full pose estimation problem to the absolute minimum amount of data required for the problem at hand, we significantly limit the risk of leaking confidential information for both consumers as well as cloud providers.

References

[1] Inside Facebook Reality Labs: Research updates and the future of social connection. https://tech.fb.com/inside-facebook-reality-labs-research-updates-and-the-future-of-social-connection/, 2019. 1

[2] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Conference on Computer & communications security (SIGSAC)*, 2013. 1, 3

[3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[4] Claudio Agostino Ardagna, Marco Cremonini, Ernesto Damiani, S De Capitani Di Vimercati, and Pierangela Samarati. Location privacy protection through obfuscation-based techniques. In *IFIP Annual Conference on Data and Applications Security and Privacy*, 2007. 1, 3

[5] Clemens Arth, Daniel Wagner, Manfred Klopschitz, Arnold Irschara, and Dieter Schmalstieg. Wide area localization on mobile phones. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, 2009. 2

[6] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC-differentiable RANSAC for camera localization. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[7] Song Cao and Noah Snavely. Minimal scene descriptions from structure from motion models. In *Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[8] Kunal Chelani, Fredrik Kahl, and Torsten Sattler. How privacy-preserving are line clouds? recovering scene details from 3d lines. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 3, 4, 6, 8

[9] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized ransac. In *Joint Pattern Recognition Symposium*, 2003. 4, 5

[10] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[11] Mihai Dusmanu, Ondrej Miksik, Johannes Lutz Schönberger, and Marc Pollefeys. Cross-Descriptor Visual Localization and Mapping. In *arXiv*, 2021. 2

[12] Mihai Dusmanu, Johannes Lutz Schönberger, Sudipta Sinha, and Marc Pollefeys. Privacy-Preserving Image Features via Adversarial Affine Subspace Embeddings. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[13] Cynthia Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, 2008. 3

[14] Marcin Dymczyk, Simon Lynen, Michael Bosse, and Roland Siegwart. Keep it brief: Scalable creation of compressed localization maps. In *International Conference on Intelligent Robots and Systems (IROS)*, 2015. 2

[15] Walter Gander. Least squares with a quadratic constraint. *Numerische Mathematik*, 36(3):291–307, 1980. 5

[16] Bugra Gedik and Ling Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 2008. 1, 3

[17] Marcel Geppert, Viktor Larsson, Pablo Speciale, Johannes Lutz Schönberger, and Marc Pollefeys. Privacy Preserving Structure-from-Motion. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 2

[18] Marcel Geppert, Viktor Larsson, Pablo Speciale, Johannes L. Schönberger, and Marc Pollefeys. Privacy preserving localization and mapping from uncalibrated cameras. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[19] Nicholas J Higham. Matrix nearness problems and applications. In *Applications of Matrix Theory*, 1988. 5

[20] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2009. 2

[21] Herve Jegou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Perez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2012. 2

[22] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019. 3

[23] Neena Kamath. Announcing Azure Spatial Anchors for collaborative, cross-platform mixed reality apps. https://azure.microsoft.com/en-us/blog/announcing-azure-spatial-anchors-for-collaborative-cross-platform-mixed-reality-apps/, 2019. 1

[24] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *International Conference on Computer Vision (ICCV)*, 2015. 2

[25] Alex Kipman. Azure Spatial Anchors approach to privacy and ethical design. https://www.linkedin.com/pulse/azure-spatial-anchors-approach-privacy-ethical-design-alex-kipman, 2019. 1, 2

[26] Jakub Konečnỳ, Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015. 3

[27] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3d point clouds. In *European Conference on Computer Vision (ECCV)*, 2012. 2

[28] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. Location recognition using prioritized feature matching. In *European Conference on Computer Vision (ECCV)*, 2010. 2, 6

[29] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 2004. 6

[30] Will Maddern, Geoffrey Pascoe, Matthew Gadd, Dan Barnes, Brian Yeomans, and Paul Newman. Real-time kinematic ground truth for the oxford robotcar dataset. *arXiv preprint arXiv: 2002.10152*, 2020. 7

[31] 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. 7

[32] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[33] Mary Lynne Nielsen. Augmented Reality and its Impact on the Internet, Security, and Privacy. https://beyondstandards.ieee.org/augmented-reality/augmented-reality-and-its-impact-on-the-internet-security-and-privacy/, 2015. 1, 2

[34] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[35] Tilman Reinhardt. Google Visual Positioning Service. https://ai.googleblog.com/2019/02/using-global-localization-to-improve.html, 2019. 1

[36] Franziska Roesner. Who Is Thinking About Security and Privacy for Augmented Reality? https://www.technologyreview.com/s/609143/who-is-thinking-about-security-and-privacy-for-augmented-reality/, 2017. 1, 2

[37] Pierangela Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 2001. 3

[38] Torsten Sattler, Michal Havlena, Filip Radenovic, Konrad Schindler, and Marc Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *International Conference on Computer Vision (ICCV)*, 2015. 2

[39] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *International Conference on Computer Vision (ICCV)*, 2011. 2

[40] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2017. 2

[41] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 6

[42] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are large-scale 3D models really necessary for accurate visual localization? In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[43] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image Retrieval for Image-Based Localization Revisited. In *British Machine Vision Conference (BMVC)*, 2012. 6

[44] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[45] Johannes L Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic visual localization. *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[46] Johannes L. Schönberger, Filip Radenović, Ondrej Chum, and Jan-Michael Frahm. From Single Image Query to Detailed 3D Reconstruction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 6

[47] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 6

[48] Paul Hongsuck Seo, Tobias Weyand, Jack Sim, and Bohyung Han. Cplanet: Enhancing image geolocalization by combinatorial partitioning of maps. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[49] Armon Shariati, Christian Holz, and Sudipta Sinha. Towards privacy-preserving ego-motion estimation using an extremely low-resolution camera. *IEEE Robotics and Automation Letters*, 2020. 3

[50] Mikiya Shibuya, Shinya Sumikura, and Ken Sakurada. Privacy preserving visual SLAM. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 2

[51] 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. 6, 7

[52] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision (ICCV)*, 2003. 2

[53] Pablo Speciale, Johannes L. Schönberger, Sing Bing Kang, Sudipta Sinha, and Marc Pollefeys. Privacy Preserving Image-Based Localization. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 8

[54] 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. 1, 2, 3, 5, 6, 7, 8

[55] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2002. 1, 3

[56] Carl Toft, Will Maddern, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, et al. Long-term visual localization revisited. *Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2020. 2

[57] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 1991. 6, 7, 8

[58] Jan-Erik Vinje. Privacy Manifesto for AR Cloud Solutions. https://medium.com/openarcloud/privacy-manifesto-for-ar-cloud-solutions-9507543f50b6, 2018. 1, 2

[59] Florian Walch, Caner Hazirbas, Laura Leal-Taixé, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization with spatial LSTMs. In *International Conference on Computer Vision (ICCV)*, 2017. 2

[60] Greg Welch and Gary Bishop. Scaat: Incremental tracking with incomplete information. In *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1997. 2

[61] Kyle Wilson and Noah Snavely. Robust global translations with 1DSFM. In *European Conference on Computer Vision (ECCV)*, 2014. 6

[62] Bernhard Zeisl, Torsten Sattler, and Marc Pollefeys. Camera pose voting for large-scale image-based localization. In *International Conference on Computer Vision (ICCV)*, 2015. 2

[63] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference Pose Generation for Visual Localization via Learned Features and View Synthesis. *arXiv*, 2005.05179, 2020. 6