Privacy Preserving Localization via Coordinate Permutations

Linfei Pan[1] Johannes L. Schönberger[2] Viktor Larsson[3] Marc Pollefeys[1,2]
[1]ETH Zurich [2] Microsoft [3]Lund University
[1]{linfei.pan, marc.pollefeys}@inf.ethz.ch, [2]joschonb@microsoft.com, [3]viktor.larsson@math.lth.se

Abstract

Recent methods on privacy-preserving image-based localization use a random line parameterization to protect the privacy of query images and database maps. The lifting of points to lines effectively drops one of the two geometric constraints traditionally used with point-to-point correspondences in structure-based localization. This leads to a significant loss of accuracy for the privacy-preserving methods. In this paper, we overcome this limitation by devising a coordinate permutation scheme that allows for recovering the original point positions during pose estimation. The recovered points provide the full 2D geometric constraints and enable us to close the gap between privacy-preserving and traditional methods in terms of accuracy. Another limitation of random line methods is their vulnerability to density based 3D line cloud inversion attacks. Our method not only provides better accuracy than the original random line based approach but also provides stronger privacy guarantees against these recently proposed attacks. Extensive experiments on standard benchmark datasets demonstrate these improvements consistently across both scenarios of protecting the privacy of query images as well as the database map.

1. Introduction

The interest in commercial solutions for cloud-based localization and mapping in mixed reality and robotics (*e.g.*, Google VPS [34], Microsoft Azure Spatial Anchors [21], or Facebook LiveMaps [1]) has raised a host of privacy concerns [24, 31, 35, 51]. Recently, several approaches have been introduced by the research community to address these concerns [14, 17–19, 25, 45–47]. The common theme underlying these methods is based on the principle of lifting traditional point-based features to randomly oriented lines to conceal the appearance of query images and database maps.

While this family of approaches is successful in enabling privacy preserving image-based localization and mapping, it comes at significant trade-offs in terms of accuracy and recall. These trade-offs are primarily caused by the fact that the privacy preserving line-to-point or point-to-line correspondences provide weaker geometric constraints than the traditional point-to-point correspondences [46]. When concealing the content of images or maps, the number of constraints is effectively halved, as each point-to-line or line-to-point reprojection only contributes a 1D geometric constraint compared to the 2D constraint from a point-to-point reprojection. Especially for the hard to localize images, where only few correspondences can be recovered during feature matching, this limitation makes a significant difference in terms of accuracy and recall.

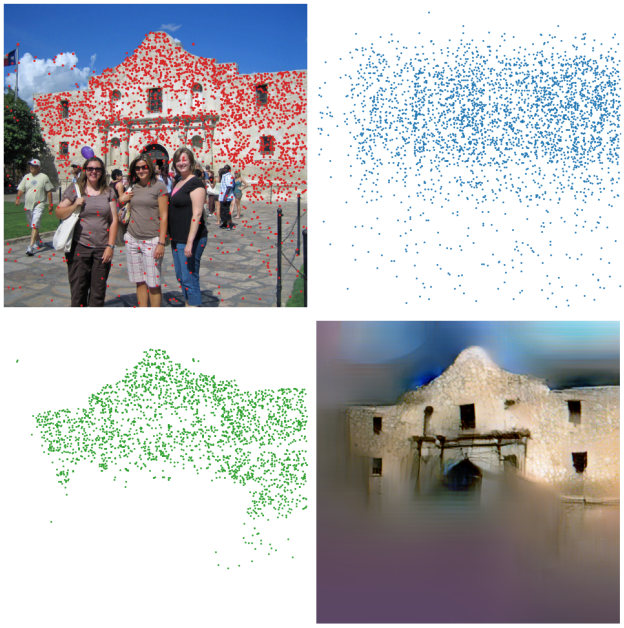


Figure 1: **Privacy-preserving localization.** *Top Left:* The client extracts keypoints and descriptors from the query image. *Top Right:* The client secretly pairs keypoints and permutes their coordinates; hiding their true position from the server. *Bottom Left:* Server performs localization using the permuted points and recovers the original 2D positions for points that exist in both query and map. *Bottom Right:* Feature inversion on the set of recovered points only reveals objects that are already present in the map.

Furthermore, without careful construction of random line maps, density based attacks have been shown to be effective in uncovering information from the privacy preserving representations, which was already mentioned as a limitation by Speciale *et al*. [46]. The main idea behind density based attacks exploits the fact that pairs of randomly oriented lines have a relatively higher chance of intersecting near the true surface as compared to empty or unobserved space. As such, a naive attack could simply gather statistics from pairwise intersections of random lines and assume the true surface in dense regions of the intersected space. More sophisticated attacks based on this idea can yield realistic image renderings from 3D line maps when attacking naively constructed line maps [9].

In this work, we revisit the approach of lifting points to lines in order to improve upon these two main limitations. In contrast to the original approach, we introduce a coordinate permutation scheme by which we effectively construct axis-aligned lines. During the pose estimation routine, the permutation scheme enables us to recover the original points for a large fraction of the inlier correspondences. Therefore, we can use the same 2D geometric constraints as in the traditional setup. This significantly improves the accuracy of the localization results. Moreover, when applying our approach for hiding the 3D map, we drastically increase the combinatorial complexity for density based attacks as well as reduce the chance for line intersections due to constructing axis-aligned lines. Counter-intuitively, our proposed method therefore not only leads to better accuracy but also provides a stronger level of privacy preservation. We experimentally demonstrate these improvements on standard benchmark datasets and by attacking our axis-aligned line maps using the approach by Chelani *et al*. [9]. In summary, we make the following **contributions**:

- Propose a new privacy-preserving lifting approach, applicable to protecting the map and query.
- Attacking the lifted representation has a higher combinatorial complexity compared to random lifting, rendering previous density based attacks ineffective.
- In contrast to previous work, we can recover the original 2D or 3D points, allowing for stronger geometric constraints and more accurate poses.
- Experimentally, we show significant improvements over previous privacy preserving work, closing the gap to traditional point-based non-private methods.

2. Related Work

Image-based Localization Methods in image-based localization have made tremendous progress. Over the years, the methods have been scaled for robust localization in large-scale scenes [27, 38, 40, 54] with compressed map representations [8, 15], while some of the methods can even run in real-time on mobile devices [4, 20, 22, 27, 28, 39]. Some of the recent works focus on improving the robustness to drastic environmental changes [3, 42, 50] or enabling cross-device localization and mapping scenarios [13]. The developed approaches are generally divided into structure-based methods [20, 38–40] using on an explicit geometric map representation or learning-based methods [7, 22, 44] with an implicit map representation.

Privacy Risks Image-based localization is subject to privacy risks due to requiring images as an input [24, 31, 35, 51]. For example, in a cloud-based localization system, users may be concerned about sharing images of private spaces with the service provider. In a practical setting, the clients could only send local image features to the cloud to avoid these risks. However, recent inversion attacks have proven extremely effective in recovering the original images from only sparse local image features [12, 30, 32]. Recently, Chelani *et al*. proposed a method to query a localization service with a database of objects to recover information about scene contents [10].

Privacy Preservation To protect against these risks, several methods have been developed recently. The first work on this topic [46, 47] proposed a solution to preserve the privacy of 2D image or 3D map points in image-based localization. Their main idea was to lift 2D or 3D points to random lines by dropping one of the two geometric constraints during the pose estimation problem. Based on the same idea, a number of works followed up to address the problems of structure-from-motion [18, 19] and real-time SLAM [45], or to directly protect the high-dimensional feature representations instead of the geometry [14].

Underlying all these approaches, there are two fundamental limitations that were already observed by the original work [47]. First, by dropping one geometric constraint per correspondence for pose estimation, the accuracy of these approaches is significantly lowered, especially in the challenging cases with few matches between query and database. Second, without careful sparsification of the 3D map, density based attacks can recover information from 3D line clouds [9]. Sparsification of the map is an effective protection against density based attacks but comes with a further reduction of accuracy as well as recall.

To protect in a more principled manner against these attacks, Geppert *et al*. [17] proposed to decompose the full 3D map into 1D partial maps, which are distributed across different servers. Each server then solves a partial localization problem and sends back the answer to the client, which can assemble a full 6-DoF localization result from the partial answers. Despite being effective against density based at-

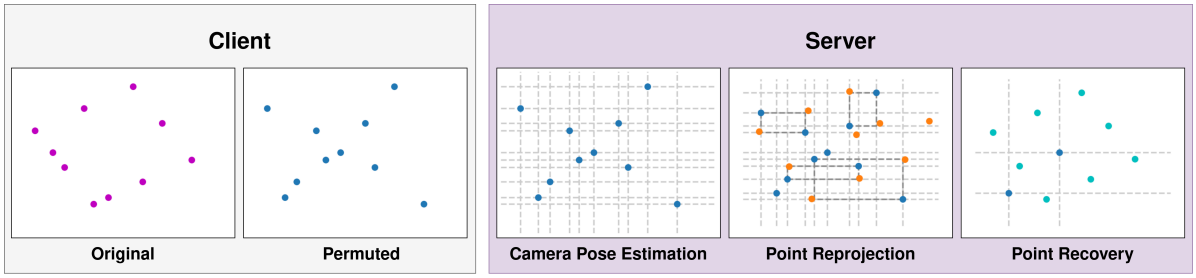


Figure 2: **Method overview for localization with private queries.** The client extracts keypoints in the query image and secretly pairs them to swap one of their coordinates. The server only receives the permuted points, which can be considered as axis-aligned lines. The secretly paired points form rectangles, which can be leveraged during RANSAC to efficiently recover the original point positions for pairs that are both inliers. The scoring of candidate poses and the final non-linear pose refinement jointly leverage the stronger 2D constraints for recovered points and otherwise the weaker 1D constraints.

tacks, the system is complex and it comes with further accuracy and recall trade-offs. In contrast, our method does not require a complex multi-server setup and we improve upon the two main limitations of the random lifting approaches simultaneously. Our method provides both stronger privacy guarantees while also closing the accuracy gap to traditional localization approaches.

3. Method

We now present our method for privacy-preserving localization. We address both the scenario of protecting the client's 2D image points in the localization query (Section 3.2) as well as protecting the service's 3D points in the database map (Section 3.3).

Our approach follows the main idea in Speciale *et al*. [46, 47], where the exact 2D or 3D point positions are hidden by introducing ambiguity in the coordinates. In the original works, this is done by lifting each point to a random line. The line representation still enables localization, using line-to-point instead of point-to-point correspondences, but makes it difficult to identify potentially sensitive image or map content. Similar to the original idea [46], we also obscure the true position of the point. However, instead of replacing it with a random line, we select a subset of the point coordinates and corrupt them. This is done by secretly forming pairs of points and swapping their coordinates at random. Since the list of pairs is not known to an attacker, it is practically intractable to recover the original point positions. As we will later show, for point-pairs that exist in both the query and the map, we can devise a scheme to recover the true positions, allowing us to use the full 2D-3D geometric constraints for accurate pose refinement.

An overview of our pipeline can be found in Figure 2 and we detail the steps of our method in the following sections. For ease of presentation, we first review the traditional setup and then explain the 2D setting that hides the query, but the same principle generalizes to the 3D setting as well.

3.1. Review of Traditional Localization

Traditionally, localization is performed by establishing a set of tentative 2D-3D point correspondences between the query image and the database map. Each correspondence $(\boldsymbol{x}, \boldsymbol{X}) \in \mathbb{R}^2 \times \mathbb{R}^3$ yields two geometric constraints on the unknown camera pose $(R, \boldsymbol{t}) \in \mathrm{SE}(3)$ as

$$\lambda \bar{\boldsymbol{x}} = R\boldsymbol{X} + \boldsymbol{t} \ , \tag{1}$$

where we let $\bar{\boldsymbol{x}}$ denote the homogeneous 2D point in the normalized image plane (after removing the intrinsic calibration), and λ represents the unknown depth. To identify outlier matches and estimate an initial pose, the standard approach is to use hypothesize-and-test frameworks, such as RANSAC [16], in order to generate multiple pose candidates by fitting $(R, \boldsymbol{t})$ to minimal subsets of the points, followed by measuring the consensus among the rest of the matches. The consistency between a candidate pose and a 2D-3D match is computed using the 2D reprojection error

$$d_p = \|\boldsymbol{x} - \pi(R\boldsymbol{X} + t)\| \ , \tag{2}$$

where $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ is the projection mapping. Once a candidate pose, together with the inlier matches, have been established, it is refined by minimizing the reprojection errors for all inliers jointly.

3.2. Localization with Private Queries

The traditional method relies on the availability of the original 2D image points, and is thus not privacy preserving. To preserve the privacy of the image, Speciale *et al*. [47] proposed to lift 2D image points into random 2D lines. Our method also builds upon 2D lines but aligns them with the coordinate axes through a random coordinate permutation scheme. The following paragraphs first establish notation by reviewing (random) line constraints for localization and then explain our permutation scheme in detail.

Random Line Constraints In Speciale *et al*. [47], each 2D line $\boldsymbol{\ell}$ back-projects into a 3D plane $\boldsymbol{\Pi}_i = \boldsymbol{P}^\top \boldsymbol{\ell}_i$ passing

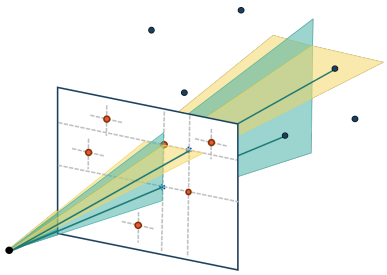


Figure 3: Illustration of geometric constraints induced by coordinate permutation and point recovery mechanism in the scenario of private queries. Red image points are secretly permuted. Black points are 3D map points. Blue points are the recovered, original image points obtained by back-swapping and form a rectangle together with their respective permuted points.

through camera projection center and the line in the image plane. The corresponding 3D point $\boldsymbol{X}$ is expected to lie on this plane, which constrains the pose as

$$\boldsymbol{\ell}^\top (R\boldsymbol{X} + \boldsymbol{t}) = 0 \ . \tag{3}$$

Note that this only yields a single geometric constraint instead of two as in Equation (1). In this case, it is still possible to estimate the pose, though the minimal problem now requires 6 instead of 3 correspondences. The minimal solver for this problem is known as $p6LP$ [33] and was used in [47]. Further, to validate a match, we can only measure the distance between the 2D line and the projected 3D point

$$d_l = \frac{|\boldsymbol{\ell}^\top \bar{\pi}(R\boldsymbol{X} + \boldsymbol{t})|}{\sqrt{l_1^2 + l_2^2}} \ , \tag{4}$$

where $\boldsymbol{\ell} = [l_1, l_2, l_3]^\top$ and $\bar{\pi}$ is the homogeneous projected 2D point. This is a weaker constraint compared to the 2D error measured in Equation (2) and thus explains the observed loss in accuracy by this approach.

Coordinate Permutation Constraints In our case, we have 2D image points, where we know that only one of the two coordinates is correct because of the a priori secret permutation (illustrated in Figures 2 and 3). In the context of the line-based framework from Speciale *et al*. [47], this can be considered as, instead of having a random line, having two axis-aligned lines passing through the given point. The original point will share one of the coordinates, and thus lie on one of these two lines. Mathematically, each point $\boldsymbol{x} = [x_1, x_2]^\top$ gives rise to two axis-aligned lines $\boldsymbol{\ell}_1, \boldsymbol{\ell}_2$, where $\boldsymbol{\ell}_1 = [1, 0, -x_1]^\top$, $\boldsymbol{\ell}_2 = [0, 1, -x_2]^\top$. Each correspondence then gives us the following constraint

$$\boldsymbol{\ell}_1^\top (R\boldsymbol{X} + \boldsymbol{t}) = 0 \quad \vee \quad \boldsymbol{\ell}_2^\top (R\boldsymbol{X} + \boldsymbol{t}) = 0 \ , \tag{5}$$

where we only know that one of the two constraints should be satisfied. For n point correspondences, we thus have 2^n possible combinations of constraints to choose from.

While considering all n points leads to a combinatorial explosion and would be computationally intractable, we only need to consider small subsets of matches to hypothesize candidate poses inside RANSAC. For a minimal sample of 6 correspondences, there exist $2^6 = 64$ different combinations of lines in Equation (5). One of these combinations will correspond to having the true (uncorrupted) coordinate in all 6 lines, and should give a correct camera pose for an all-inlier minimal sample. So, to find a camera pose, we can simply iterate exhaustively over all $2^6 = 64$ combinations. Each of these combinations corresponds to a $p6LP$ minimal estimation problem, which can be efficiently solved. If gravity is known, we can efficiently solve the minimal $up4LP$ problem [33] from $2^4 = 16$ combinations.

The key observation is now that, if we have a candidate pose, then for any inlier correspondence, we can easily resolve which of the two axis-aligned lines (*i.e*., coordinates) is the correct one by simply projecting the 3D points to the image and taking the one with the smaller 1D residual d_l. By determining the inlier coordinates $\mathcal{I}$ within the threshold (*i.e*., $d_l < \varepsilon$), we can score candidate poses and thereby embed our method into a standard RANSAC loop to robustly estimate a camera pose. This approach still uses the weaker 1D constraints. The next paragraph explains how we can efficiently recover the original 2D points in this framework to benefit from stronger 2D constraints.

Original Point Recovery After finding a candidate pose using RANSAC, we devise an efficient scheme for recovering the original points $\boldsymbol{x}$ from the set of inliers $\mathcal{I}$. Note that our scheme can only recover the positions for points that are visible in both the query and the map. As such, recovering the original points is not a concern for privacy, as they are known to both client and server.

Let's recall that we randomly picked pairs of points and swapped one of their coordinates to preserve their privacy. To recover the original points, we need to rely on the projected points and the property that the swapped back points should be close to the projected points (within ε distance) if it is an inlier. As for outliers points, there is no proper mechanism to distinguish a correct pair from an incorrect one, thus they remain unrecoverable.

One way of recovering points is to iterate through all pairwise combinations of inlier correspondences and choose a configuration with the most swapped back points close to the projected points. However, this method has a squared complexity in the number of correspondences and is thus computationally expensive to execute inside a RANSAC loop. To achieve a better complexity, it is helpful to notice that each pair of points, that are selected for

coordinate permutation, their original points and the permuted points form a 2D rectangle in the image (*c.f.* Figures 2 and 3). Using this observation, we can devise a more efficient recovery algorithm. For bookkeeping of recovered point coordinates, we first define the two sets

$$S_j = \{(x_{kj}, k) \mid \hat{\boldsymbol{x}}_k = [x_{k1} \quad x_{k2}]^\top\}, \quad j = \{1, 2\} \quad (6)$$

Each element in the sets consists of a single coordinate (x_1 or x_2) and its index k in the ordered list of points. The index uniquely identifies coordinates to enable swapping of point pairs. For example, if the query image has 3 points $\{\hat{\boldsymbol{x}}_1 = [0, 10]^\top,\ \hat{\boldsymbol{x}}_2 = [20, 30]^\top,\ \hat{\boldsymbol{x}}_3 = [40, 50]^\top\}$, then $S_1 = \{(0, 1), (20, 2), (40, 3)\}$, $S_2 = \{(10, 1), (30, 2), (50, 3)\}$. For each projected image point $\tilde{\boldsymbol{x}}$, we first decide which coordinate could be the original one by comparing $\tilde{\boldsymbol{x}}$ and the permuted image point $\hat{\boldsymbol{x}}$. We label coordinate j as correct if the j-th coordinate is within ε from the projected point, that is $|\hat{x}_j - \tilde{x}_j| < \varepsilon$. Denoting the other coordinate as j', we collect the possible swapped point candidates C by lookuping in the set $S_{j'}$, and finding points with the respective coordinate close to the projected point

$$C = \{k \mid |x_{kj'} - \tilde{x}_{j'}| < \varepsilon, (x_{kj'}, k) \in S_{j'}\} \quad (7)$$

Since this set likely contains multiple candidates for each point (having a similar x_1 or x_2 coordinate), and the candidate points after swapping back can be distant from their projected points, we use a symmetric swapping error to filter out the coincident candidates

$$\varepsilon_{sym} = \|\tilde{\boldsymbol{x}} - [x_j \quad x_{kj'}]^\top\|_2 + \|\tilde{\boldsymbol{x}}_k - [x_{kj} \quad x_{j'}]^\top\|_2 \quad (8)$$

We only consider a candidate to be valid, if both points are close to the respective reprojected image points after swapping them back. If multiple candidates are valid, we keep the one with the smallest ε_{sym}. Notice that, with such a selection mechanism, the recovered original points may not be symmetric. However, we have not found any drawbacks from this asymmetry in our experiments.

By implementing the sets S_j using hashing, we achieve amortized linear runtime complexity on average w.r.t. the number of correspondences for our recovery scheme. More details can be found in the suppl. material.

Pose Refinement To further improve the accuracy of our pose estimates, we conduct a non-linear refinement over all inliers using the stronger 2D constraints from Equation (2) for recovered points $\mathcal{I}_p$ and the weaker 1D constraints from Equation (4) for coordinates $\mathcal{I}_l$ that could not be swapped back. The overall cost function is defined as

$$\sum_{k \in \mathcal{I}_p} \rho\left(d_p(\boldsymbol{x}_k, P\bar{\boldsymbol{X}}_k)\right) + \sum_{k \in \mathcal{I}_\ell} \rho\left(\underset{i \in \{1,2\}}{\text{softmin}}\left(d_l(\boldsymbol{\ell}_{ki}, P\bar{\boldsymbol{X}}_k)\right)\right) \quad (9)$$

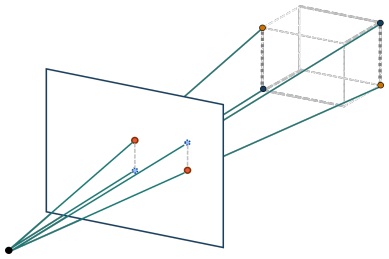


Figure 4: Illustration of point recovery mechanism in the 3D scenario with private maps, where the permuted red points form a cuboid in 3D. The reprojection of the 3D points is used for thresholding of errors in image space.

where ρ is a robustifier. To smoothly select between the two possible axis-aligned lines, we use the (Boltzmann) soft-min operator [5], defined as $\underset{i}{\text{softmin}}\{d_i\} = (\sum_i d_i e^{-d_i})/(\sum_i e^{-d_i})$. We incorporate this non-linear refinement into LO-RANSAC [11] using the Arctan loss as ρ inside the RANSAC loop and Cauchy as the robust loss for the final pose refinement.

Degenerate Configurations Note that when all $\boldsymbol{\ell}_i$ lines are parallel (*i.e.*, originating from the same coordinate), the configuration becomes degenerate as the full 6-DoF of the pose cannot be recovered. This happens for 2 out of 64 configurations that we can trivially discard. For the two degenerate cases, it would be possible to solve for the rotation and two translation parameters using 5 correspondences. We initially explored this direction as well yet found the solutions to be less stable, as they provide less accurate pose estimates, so we do not include them in our final method.

3.3. Localization with Private Maps

In the previous section, we described our coordinate permutation scheme for 2D image points to preserve the privacy of localization queries. Similar to the 2D case, we can also permute coordinates of 3D points to hide the contents of private maps. This is a scenario that was originally addressed by Speciale *et al.* [46] using lifting to random 3D lines. We denote a random 3D line $\boldsymbol{L}$ as

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{d} \end{bmatrix} \in \mathbb{R}^3 \times \mathbb{P}^2 \ , \quad (10)$$

where $\boldsymbol{p}$ and $\boldsymbol{q}$ are 3D points on $\boldsymbol{L}$ subject to $\boldsymbol{d} = \dfrac{\boldsymbol{p} - \boldsymbol{q}}{\|\boldsymbol{p} - \boldsymbol{q}\|_2}$ and such that the line passes through the original 3D point $\boldsymbol{X}$. The reprojection of the 3D line to the image defines a 1D geometric constraint

$$0 = \hat{\boldsymbol{L}}_i^\top \bar{\boldsymbol{x}} = ([\boldsymbol{R}\boldsymbol{d}_i]_\times \boldsymbol{P}\bar{\boldsymbol{p}}_i)^\top \bar{\boldsymbol{x}} \quad (11)$$

that can be used for privacy-preserving pose estimation. To solve for the full 6-DoF pose, at least 6 correspondences are

needed to solve the minimal $p6L$ problem [48] in RANSAC. If gravity is known, only 4 correspondences are needed to solve the $up4$L problem [49].

Instead of lifting to random 3D lines, we construct axis-aligned 3D lines by coordinate permutation of randomly selected pairs of 3D points. Analogous to the 2D case, we can hypothesize pose candidates in RANSAC by generating the different $3^6 = 729$ combinations of line constraints, and we can also recover the original 3D point positions by back-swapping with sets $S_{j=1,2,3}$. Notice that the original 3D point may lie on any of the three axis-aligned lines and the randomly selected point pairs form a cuboid in 3D (see Figure 4). To robustify the back-swapping algorithm against non-isotropic noise on the (triangulated) 3D points, we define the error ε_{sym} in image space through reprojection. Moreover, if runtime is a concern for the final application, we can also choose to expose one of the 3 coordinates, so the number of line combinations would be $2^6 = 64$.

Pose Refinement For pose refinement in the 3D case, we again add constraints from two sources: the ones from inlier lines $\mathcal{I}_L$ and the ones from recovered inlier points $\mathcal{I}_p$. The overall cost function is defined as

$$\sum_{k\in\mathcal{I}_p} \rho\left(d_p(\boldsymbol{x}_k, P\bar{\boldsymbol{X}}_k)\right) + \sum_{k\in\mathcal{I}_L} \rho\left(\underset{i\in\{1,2\}}{\text{softmin}}\left(d_L(\boldsymbol{x}_k, \hat{\boldsymbol{L}}_{ki})\right)\right)$$
$$\text{where} \quad d_L = \frac{|\hat{\boldsymbol{L}}_{ki}^\top \bar{\boldsymbol{x}}_k|}{\sqrt{\hat{L}_{ki1}^2 + \hat{L}_{ki2}^2}} \tag{12}$$

and we use the same setup of LO-RANSAC and choice of robustifiers as in the 2D scenario.

4. Experiments

In our experiments, we first focus on demonstrating the effectiveness of our proposed point recovery mechanism using an ablation study. We then compare our method against the traditional non-privacy preserving approach (PnP) and the original privacy-preserving approach based on random lines. Results are reported on existing benchmark datasets [23, 46, 52] for image-based localization, evaluated for the two scenarios of private localization queries (PnLP) and maps (PnL). All results assume unknown gravity and we show further results with known gravity in the supplementary material. Finally, we show image inversion results using the density-based attack by Chelani *et al.* [9].

Implementation Details We embed our method into LO-RANSAC [11] with Levenberg–Marquardt [2, 26] as non-linear optimizer for pose refinement. Inlier reprojection thresholds on 2D points are set to $12px$ for the Cambridge dataset [23] and to $4px$ otherwise. Reprojection thresholds on lines are chosen as $\frac{12px}{\sqrt{2}}$ and $\frac{4px}{\sqrt{2}}$, respectively.

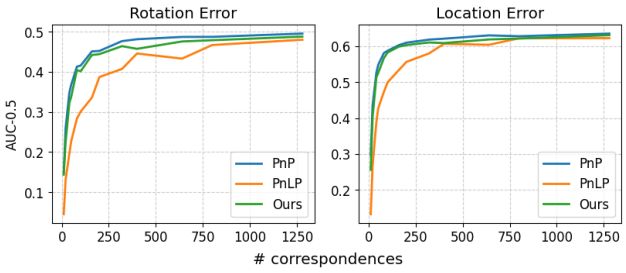


Figure 5: Ablation study on King's college sequence [23] w.r.t. the number of correspondences. Our method closes the performance gap, that is especially pronounced in difficult cases with few correspondences.

We re-implement all methods in the same framework for a fair comparison of accuracy, recall, and runtime. All point clouds are triangulated using COLMAP [41, 43].

4.1. Ablation Study

4.1.1 Impact of Number of Correspondences

As argued in previous sections, the reduced accuracy and recall in privacy-preserving localization methods based on random lines is largely attributed to the weaker geometric constraints. To showcase this performance gap, we conduct an experiment on the King's College sequence in the Cambridge dataset [23]. Localization performance is tested on varying number of correspondences per query image by randomly dropping input matches. Cases with few correspondences typically depict the challenging localization scenarios, where feature matching only yields few matches and only few constraints can be used for pose estimation. The results of this experiment are summarized in Figure 5. First, we observe that the performance gap between the privacy-preserving and traditional approach is especially pronounced for low number of matches, which in turn suggests that the impact of the number of constraints on the localization accuracy saturates eventually. We also see that our method closes the accuracy gap to the traditional approach across the board.

4.1.2 Impact of Point Recovery Mechanism

To analyze the effectiveness of the proposed point recovery mechanism, we conduct experiments on HoloLens data [47] with our method in different settings: with only line constraints, with recovered point constraints, and with the true recovered points using an oracle that has access to groundtruth information. This dataset contains images with a large variance on the inlier ratio, which makes it suitable for this ablation study, as the point recovery mechanism is strongly influenced by it. Results are summarized in Figure 6, which shows a significant improvement in accuracy for our method when using the stronger 2D constraints from

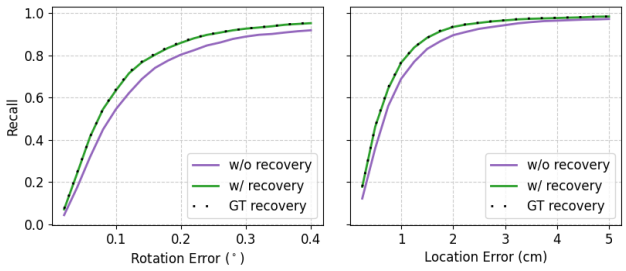


Figure 6: Ablation study on HoloLens data from [47] to measure the effectiveness of our point recovery mechanism.

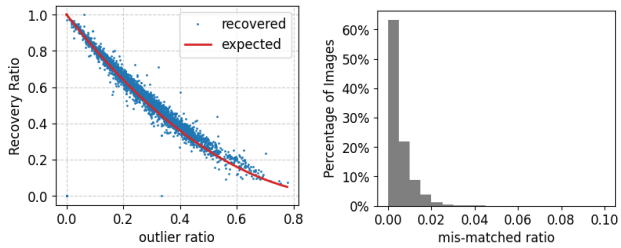


Figure 7: Ablation study on HoloLens data from [47] that shows effective point recovery across all outlier ratios and overall low error rates on the recovered points.

	MED (°)	AUC@1°	MED (cm)	AUC@10cm	(1°, 10 cm)	Time (s)
PnLP [47]	0.08	84.79	0.63	88.88	97.80	0.07
Proposed	0.07	86.69	0.56	90.12	98.41	0.36
PnP	0.07	87.92	0.51	90.85	98.80	0.05

Table 1: Localization performance on HoloLens dataset [47] with indoor and outdoor scenes.

recovered points. In comparison to the oracle that defines a theoretical upper bound for our method, there is no significant difference in performance.

Furthermore, we also report the correlation between the outlier ratio and the ratio of recovered points in Figure 7 The expected number is calculated as p^2 where p is the inlier ratio. From the figure we can see that the ratios of correctly recovered points matches the expected number across all outlier ratios. The ratio of mismatched points stays low in this dataset even when the inlier ratio is as low as 20%. Those observations combined provide strong evidence that the recovering mechanism is effective.

4.2. Localization with Private Queries

HoloLens [47] This dataset features mostly small-scale scenes and unconstrained motion trajectories from a head-worn augmented reality device with VGA-resolution cameras. Equivalent to the original dataset, we use COLMAP for reconstruction of the full image sequence while limiting the number of SIFT features per image to at most 500. There are 3325 query images in total and each query has on average 320 correspondences. After reconstructing the full sequence, we hold out every 5-th image as a query, remove any point correspondences associated to these images from the reconstruction, and retriangulate the remaining database images. The pose of the query images in the full reconstruction serve as the groundtruth for evaluation. The results are summarized in Table 1, where we can see that our method performs almost on par with the traditional non-privacy preserving case in terms of accuracy and recall. The additional combinatorial overhead of evaluating more pose candidates leads to a roughly 5-fold runtime increase for our method.

	MED (°)	AUC@1°	MED (cm)	AUC@10cm	(1°, 10 cm)	Time (s)
PnLP [47]	0.18	71.59	12.13	18.38	41.29	0.36
Proposed	0.18	72.61	11.73	20.03	44.26	2.31
PnP	0.17	73.09	11.66	20.22	43.95	0.20

Table 2: Localization performance on Cambridge dataset [23] with handheld videos in an outdoor scene.

	MED (°)	AUC@1°	MED (cm)	AUC@10cm	(1°, 10 cm)	Time (s)
PnLP [47]	0.02	88.32	2.24	59.57	78.23	0.09
Proposed	0.02	89.67	1.92	63.62	81.33	0.68
PnP	0.02	90.74	1.85	64.58	82.62	0.08

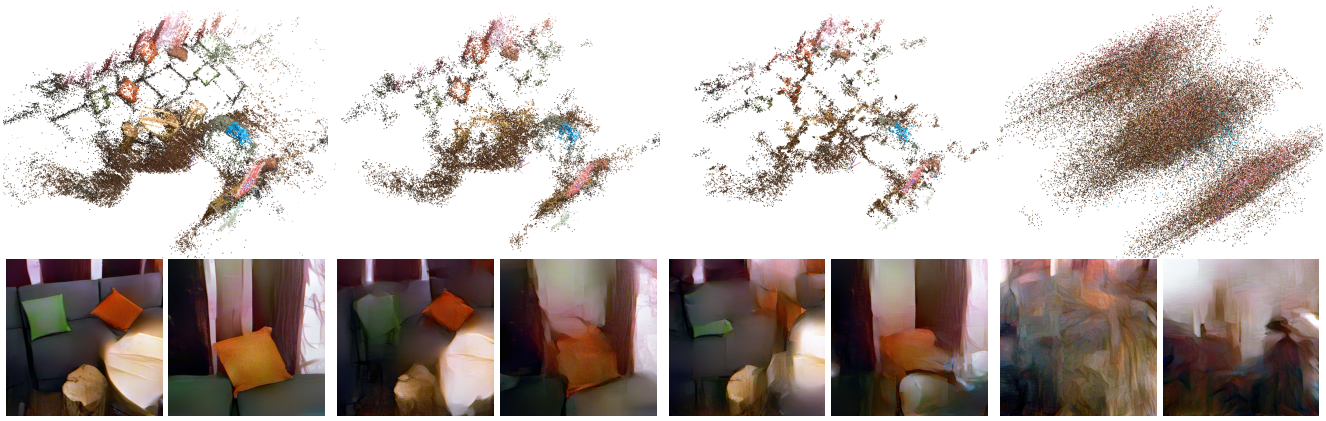
Table 3: Localization performance on crowd-sourced Internet photo collections [29, 53].

	MED (°)	AUC@1°	MED (cm)	AUC@10cm	(1°, 10 cm)	Time (s)
PnL [46]	0.08	84.71	0.64	88.66	97.71	1.16
Proposed	0.07	86.31	0.56	89.71	97.92	6.61
PnP	0.07	87.85	0.51	90.84	98.83	0.07

Table 4: Localization performance on HoloLens dataset [47] in the private map scenario.

Cambridge [23] This localization benchmark dataset contains handheld HD-resolution videos taken around Cambridge University in an outdoor setting. We use the same reference reconstruction as in [6] for the database map and extract SuperPoint+SuperGlue feature correspondences using the hloc toolbox [36, 37]. The dataset has a total of 1918 query images and an average of around 2850 2D-3D correspondences per image. Results are summarized in Table 2 and the performance trend for accuracy, recall, and runtime are all consistent with the results on the lower-resolution HoloLens dataset. This underlines the robustness of our approach against different sensors, scene types, as well as feature detection and matching algorithm characteristics.

Internet Photo Collections [29, 53] This collection of datasets is crowd-sourced from the internet and comprises a large variety in image resolution and quality, lighting conditions, *etc*. We conduct experiments on the Dubrovnik6k, Gendarmenmarkt, Roman Forum, and Trafalgar locations with equivalent settings as used by Speciale *et al*. [47] for generating pseudo-groundtruth and establishing feature correspondences using COLMAP. On average, there are around 1000 SIFT correspondences per image with a total



(a) Original (b) Random line (c) Oracle (d) Proposed

Figure 8: Privacy attack on (private) maps using a combination of 3D image inversion [32] and the density based attack method Line2Point [9]. The top row shows the 3D map that is recovered using Line2Point for all but the leftmost result. The 3D map is used as an input for rendering synthesized images shown at two different viewpoints in the bottom row. From left to right: (a) Results on original, non-privacy preserving point cloud. (b) Results on random line cloud. (c) Results on our method with an attacker that gained access to swapped coordinates. (d) Results on our method with permuted coordinates.

of 2416 query images. Results are summarized in Table 3. The accuracy of all methods on this dataset is surprisingly high, yet we can still observe significant improvements.

4.3. Localization with a Private Map

To evaluate localization scenario with a private map, we conduct experiments on HoloLens data [47] using the same setup as in the private query scenario described in the previous section. We adopt the setting of exposing one correct coordinate in Table 4 and provide results for the computationally more expensive case with 729 combinations in the supplementary material. Like in the private query scenario, we are able to significantly close the accuracy gap but increase run-time by a factor of around six.

4.4. Inversion Attack on Line Cloud

In this section, we analyze and compare the level of privacy protection for the map representation. We perform this analysis by attacking the 3D point maps using image inversion techniques, as proposed by Pittaluga *et al*. [32]. To render images from the privacy preserving 3D line maps, we first execute a density based attack on the map representations using the Line2Point [9] method and then run image inversion on the recovered point cloud. Figure 8 shows results for this experiment, where inverted images from the non-privacy preserving map are nearly indistinguishable from original images of the scene. This again highlights the importance of research on privacy preserving methods for image-based localization. Images rendered from naively constructed random line clouds without sparsification closely resemble the original images as well and thus are not sufficient to protect private information in the map. The results of the attack on our method in column (d) shows mostly unrecognizable images and demonstrates the increased level of privacy that our method provides. Even if an attacker had access to an oracle that knows about which of the coordinates is correct, the rendered images (c) are less recognizable than the ones from a random line cloud. It is important to note that this information is not available to an attacker in practice, but it allows us to understand the relative impact on the increased level of privacy. On the one hand, our method constructs axis-aligned lines, which, by construction, cannot be intersected in a density based attack. On the other hand, we increase the combinatorial complexity of executing a density based attack, as significantly more coordinate pairs must be tried to find a valid intersection. In the supplementary material we provide additional inversion results and more discussion regarding potential attacks.

5. Conclusion

In this paper, we present a significant improvement over existing privacy-preserving image-based localization approaches. Our method improves upon the accuracy limitation of random line based approaches and closes the gap to traditional non-privacy preserving methods. At the same time, our method proves as robust against recently presented density attacks on random 3D line clouds. An interesting avenue for future work could be in addressing the increased runtime of our method. One promising idea involves voting for correct coordinate swaps during RANSAC and progressively sampling from more likely lines.

Acknowledgments We would like to thank Marcel Geppert for reviewing this paper, for the insightful discussions, and his help in the supp. material, as well as Torsten Sattler and Kunal Chelani for the code and data of line cloud inversion experiments. Viktor Larsson was supported by ELLIIT.

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] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver, 3 2022. 6

[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] 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

[5] Kavosh Asadi and Michael L Littman. An alternative softmax operator for reinforcement learning. In *International Conference on Machine Learning*, pages 243–252. PMLR, 2017. 5

[6] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In *International Conference on Computer Vision (ICCV)*, 2021. 7

[7] 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

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

[9] 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. 2, 6, 8

[10] Kunal Chelani, Torsten Sattler, Fredrik Kahl, and Zuzana Kukelova. Privacy-preserving representations are not enough: Recovering scene content from camera poses. In *CVPR*, 2023. 2

[11] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized ransac. In *Joint Pattern Recognition Symposium*, 2003. 5, 6

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

[13] Mihai Dusmanu, Ondrej Miksik, Johannes Lutz Schönberger, and Marc Pollefeys. Cross-Descriptor Visual Localization and Mapping. In *International Conference on Computer Vision (ICCV)*, 2021. 2

[14] 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

[15] 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

[16] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM (CACM)*, 1981. 3

[17] Marcel Geppert, Viktor Larsson, Johannes Lutz Schönberger, and Marc Pollefeys. Privacy Preserving Partial Localization. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2

[18] 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

[19] 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. 1, 2

[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] 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

[22] 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

[23] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 6, 7

[24] 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

[25] Chunghwan Lee, Jaihoon Kim, Chanhyuk Yun, and Je Hyeong Hong. Paired-point lifting for enhanced privacy-preserving visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17266–17275, 2023. 1

[26] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944. 6

[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

[29] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. Location recognition using prioritized feature matching. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II 11*, pages 791–804. Springer, 2010. 7

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

[31] 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

[32] 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. 2, 8

[33] Srikumar Ramalingam and Yuichi Taguchi. A theory of minimal 3d point to 3d plane registration and its generalization. *International Conference on Computer Vision (ICCV)*, 2013. 4

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

[35] 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

[36] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 7

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

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

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

[43] 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

[44] 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

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

[46] 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, 5, 6, 7

[47] 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, 4, 6, 7, 8

[48] Henrik Stewénius, David Nistér, Magnus Oskarsson, and Kalle Aström. Solutions to minimal generalized relative pose problems. In *Workshop on omnidirectional vision*, 2005. 6

[49] Chris Sweeney, John Flynn, and Matthew Turk. Solving for relative pose with a partially known rotation is a quadratic eigenvalue problem. In *International Conference on 3D Vision (3DV)*, 2014. 6

[50] 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

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

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

[53] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *European Conference on Computer Vision (ECCV)*, 2014. 7

[54] 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