Obfuscation Based Privacy Preserving Representations are Recoverable Using Neighborhood Information

Kunal Chelani[1]* Assia Benbihi[2]* Fredrik Kahl[1] Torsten Sattler[2] Zuzana Kukelova[3]

[1]Chalmers University of Technology
[2]Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague
[3]Visual Recognition Group, Faculty of Electrical Engineering, Czech Technical University in Prague
chelani@chalmers.se

Abstract

The rapid growth of AR/VR/MR applications and cloud-based visual localization has heightened concerns over user privacy. This privacy concern has been further escalated by the ability of deep neural networks to recover detailed images of a scene from a sparse set of 3D or 2D points and their descriptors - the so-called inversion attacks. Research on privacy-preserving localization has therefore focused on preventing such attacks through geometry obfuscation techniques like lifting points to higher dimensions or swapping coordinates. In this paper, we reveal a common vulnerability in these methods that allows approximate point recovery using known neighborhoods. We further show that these neighborhoods can be computed by learning to identify descriptors that co-occur in neighborhoods. Extensive experiments demonstrate that all existing geometric obfuscation schemes remain susceptible to such recovery, challenging their claims of being privacy-preserving. Code will be available at https://github.com/kunalchelani/RecoverPointsNeighborhood.

1. Introduction

Visual localization estimates the position and orientation of a camera in a given scene and is central for autonomous navigation [64, 65], Simultaneous Localization and Mapping (SLAM) [14, 18], Augmented and Virtual Reality (AR/VR) [27, 48, 51], and Structure-from-Motion (SfM) [55, 56]. The best performing methods represent the scene with a 3D map, *e.g.*, a Structure-from-Motion (SfM) point cloud [49, 52]. To localize a given query image, they match the descriptors of 2D local features [17, 38] extracted from the query image against the descriptors of the 3D points in the map. The resulting 2D-3D point correspondences are used for camera pose estimation [23, 28, 32–34]. Such feature-based approaches are known to handle challenging conditions and to provide accurate pose estimates [51, 66]. However, they also pose a potential privacy risk because of *inversion attacks* [47]: it is possible to recover the query image in high detail from the 2D image features with inversion networks [16, 47]. One can also recover the map's content from the 3D points and their descriptors [47, 60]. Thus, feature-based methods cannot be directly applied in settings where privacy is of concern [61, 62], such as when a user sends data to a localization service in the cloud or when 3D maps are stored on an external server.

Privacy-preserving localization methods aim to prevent content recovery and mainly fall into two categories: *descriptor obfuscation* approaches, which modify descriptors to prevent inversion while enabling accurate 2D-3D matching [22, 42, 45, 46], and *geometry obfuscation* approaches, which replace each 2D or 3D point with a potentially infinite set of points [24–26, 35, 41, 43, 58, 61, 62]. An example of geometry obfuscation is lifting points to lines, which replaces each point with an infinite set of points lying on a line through the corresponding point [35, 61] (see Fig. 1). By substituting points with potentially infinite sets, these methods prevent the direct application of inversion attacks [22, 47]. Geometric obfuscation approaches carefully design the function mapping a point to a set of points so that the resulting sets still enable pose estimation; for instance, in [61], 2D-3D point matches are replaced by 2D point-to-3D line matches.

Geometry obfuscation methods are considered privacy-preserving by the community since it is unclear how to recover the original point positions from the obfuscations. However, none of the previous work proves that approximating the original point positions is impossible. On the

*Equal Contribution.

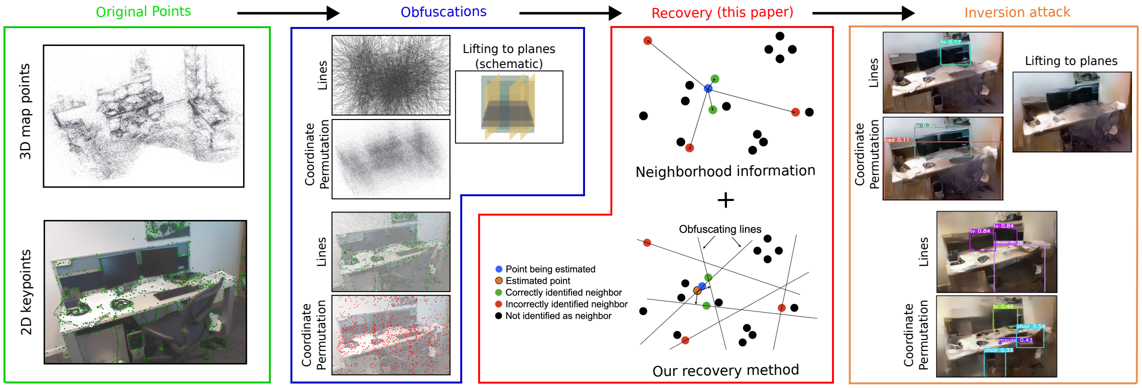

Figure 1. **Geometry obfuscations allow the recovery of image details.** The original point representations are privacy revealing as full images can be recovered from them [47]. Different obfuscation schemes are used to modify them. In this paper, we show that given neighborhood information, it is possible to approximately recover the original point positions, again enabling image recovery.

contrary, [11] reveals the need for more scrutiny before claiming that a method is privacy-preserving: they show that when each 3D point is obfuscated by a line passing through the point with a random direction [61], it is possible to approximate the original 3D point position, thus enabling an inversion attack [47]. Their approach is based on two key insights: (1) the closest points on two such 3D lines are likely to be relatively close to the original 3D points. (2) to recover the position of a point $\mathbf{X}_i$ obfuscated by a line $\mathbf{l}_i$, it is important to know the neighbors of $\mathbf{l}_i$ (defined as the set of lines $\{\mathbf{l}_j\}$ corresponding to original 3D points that are neighbors of $\mathbf{X}_i$). [11] cannot be applied to 2D point obfuscations [43, 62] and does not generalize since property (1) does not hold for all obfuscation schemes [26, 43, 62].

Inspired by observation (2) from [11], we derive a novel and conceptually simple method for approximating the positions of the original points that is applicable for *all* of the currently proposed geometry obfuscation schemes [26, 35, 41, 43, 61, 62]. Our approach uses information about neighborhoods, *i.e.*, about which obfuscated points correspond to nearest neighboring original points, and is computationally efficient[1]. We show that the approximate point positions obtained with our method enable inversion attacks [22, 47]. Through extensive qualitative and quantitative experiments, we demonstrate that the proposed method is robust to errors in neighborhoods, *i.e.*, it does not require access to neighborhoods containing only the nearest neighbors of the original points. Additionally, we present a simple approach for learning neighborhoods from the descriptors associated with geometric obfuscations, which must be provided to enable visual localization. Our method for computing neighborhoods serves as a proof of concept, demonstrating that our scheme for recovering points from geometric obfuscations is practically applicable.

In summary, this paper makes the following contributions: **(1)** we present a novel framework for recovering approximate point positions from obfuscated scene representations. Our framework relies on neighborhood information and is applicable to all geometry obfuscation schemes from the literature. **(2)** we propose a learning-based approach for computing the required neighborhoods from the descriptors used for visual localization. **(3)** extensive experiments with neighborhoods provided by an oracle and our approach show the effectiveness of our recovery framework. Our results show that methods that are currently considered privacy-preserving do not in fact guarantee privacy, and highlight the need to derive clear conditions under which privacy can be guaranteed when proposing privacy-preserving localization approaches.

2. Related Work

Visual Localization. State-of-the-art localization methods rely on features and 2D-3D correspondences between query images and the map. These matches are fed into a robust estimation framework [5, 6, 13, 23] to estimate the pose of a camera [29, 36, 44, 49, 52]. The 3D scene map is generally represented as 3D points generated using Structure-from-Motion [55, 56] or SLAM pipelines [58]. A drawback of feature-based methods is that sparse sets of points and descriptors are vulnerable to inversion attacks in which a neural network recovers detailed images of the scene from the points and their associated descriptors [1, 16, 19–22, 30, 40, 45, 47, 60, 69–71]. While sparsifying the set of points improves privacy by reducing the inversion performance, it comes at the cost of reduced localization accuracy.

Alternative localization methods include Scene Coordinate Regression (SCR) [8, 10, 59] where the 3D map is represented by a neural network that predicts the 3D coordinates of every image pixel, resulting in 2D-3D correspon-

[1] In some cases, *e.g.*, [35], our point recovery is faster to compute than obfuscating the points in the first place.

dences. SCR is said to be inherently privacy-preserving [73] because there is no set of 2D or 3D points to run the inversion attack [22, 47] on. However, current SCR methods [7, 9, 39] do not scale and do not handle challenging conditions as well as feature-based methods although these limitations are investigated [10]. Absolute Pose Regression (APR) [54, 57] and Relative Pose Regression (RPR) [2, 4] methods are end-to-end localization alternatives that share similar characteristics: they are inherently privacy-preserving but their performance falls behind feature-based methods, as pointed in [54]. In this paper, we analyze the privacy properties of features-based methods that remain the gold standard for accurate, robust, and efficient localization.

Privacy Aspects of Visual Localization. Cloud-based localization services require the exchange of information about the scene between the client and the server, leaving several cracks for possible privacy leaks. Naturally, such services should preserve the privacy of the 3D maps stored online from a curious/malicious server [11, 16, 26, 43, 61]. They should also preserve the client's private information that is potentially sent to the server for localization, as part of the query image [16, 42, 62]. As noted in [26], even the server knowing the client's accurate pose can potentially be a privacy risk. It is also shown that even the minimal requirements for running a robust localization service - returning the camera pose to the client - enable the approximate recovery of the scene layout by a malicious third party[12]. In this paper, we analyze the extent to which privacy-preserving geometric obfuscations can reveal private content from a 3D map stored on a server and from a client's query image.

Privacy-Preserving Representations. Inversion attacks [22, 47] take as input sparse feature maps to produce detailed images of the scene. The feature maps are made of descriptors located at keypoint positions and the keypoints are either 2D points or the projection of 3D points onto the image. Therefore, there are two obvious ways to counter such an attack by preventing the construction of the feature map: i) descriptor obfuscations that preserve the point information but modify the descriptors so that localization remains possible but not the inversion [22, 42, 46], ii) and geometric obfuscations that modify the geometry of the points.

The first geometric methods obfuscate points with random lines [24, 26, 61, 62] but [11] later shows that the 3D lines [61] are not as privacy-preserving as originally claimed: the original points can be approximated using the geometry preserved in the random 3D lines. [11] exploit the spatial distribution of the lines to estimate the points' nearest neighbors in the original space and then estimate the point positions that best agree with the neighborhood. Subsequent works account for this important limitation to design geometric obfuscations that are less susceptible to recovery with [11]. One solution is to modify the distribution of line directions by lifting points to paired-point lines [35] so that one line contains two points instead of one or constraining the lines to intersect at specific points [41]. To further reduce the spatial correlation between the original points and their obfuscated representations, [26] lifts points to parallel planes and [43] permutes coordinates of pairs of points, which prevents the estimation of nearest neighbors based on the geometric distances between obfuscations. Overall, these methods obfuscate the position of the points while still allowing localization. Here, we question their claim to be privacy-preserving: we reveal a weakness common to all the obfuscation schemes and propose a generic method that approximately recovers the original points from *all* obfuscations when information on their neighborhood is available.

3. Geometric Obfuscation of Points

This section provides a general definition of obfuscations applied to points. Based on this definition, Sec. 4 then proposes an attack that recovers approximate positions of the original points from an obfuscated representation given the knowledge about the neighborhoods of the original points.

Definition: Geometry obfuscation. A *geometry obfuscation* applied to a point $x \in \mathbb{R}^m$ is a mapping

$$\mathcal{O} : \mathbb{R}^m \to \mathcal{P}(\mathbb{R}^m) \ , \tag{1}$$

where $\mathcal{P}(\mathbb{R}^m)$ is the power set of $\mathbb{R}^m$, *i.e.*, the set of all subsets of $\mathbb{R}^m$. $\mathcal{O}$ maps a point in $\mathbb{R}^m$ to a (potentially infinite) set of points in $\mathbb{R}^m$. Given a set of n original points $P = \{x_j \in \mathbb{R}^m,\ j = 1,\ldots,n\}$, an *obfuscated representation* of P is a set $\mathcal{O}(P) = \{\mathcal{O}(x_j) \in \mathcal{P}(\mathbb{R}^m),\ j = 1,\ldots,n\}$ obtained by obfuscating all of the n points.

This definition can be used to model all obfuscation schemes for 2D and 3D points in the literature. In the case of mapping a point x_j to a line [35, 41, 61, 62] or a plane [26], $\mathcal{O}(x_j)$ contains all points on a line, respectively plane, that passes through x_j.[2] In the case of obfuscation by coordinate permutation [43], the set $\mathcal{O}(x_j)$ contains a single point x'_j obtained by replacing one coordinate of x_j with the corresponding coordinate of another point $x_i \in P \setminus x_j$.

4. Recovering Obfuscated Points using Neighborhood Information

In this paper, we propose an attack designed to recover image content from obfuscated representations. It enables inversion attacks by approximating the original points from the obfuscated representations. Ideally, we would like to

[2] Note that while the set $\mathcal{O}(x_j)$ might be infinite, it can be represented compactly by the parameters of a line or plane.

find the inverse of the obfuscation mapping $\mathcal{O}$ from (1), *i.e.*,

$$\mathcal{O}^{-1} : \mathcal{P}(\mathbb{R}^m) \rightarrow \mathbb{R}^m, \ \ s.t \ \ \mathcal{O}^{-1}(\mathcal{O}(x)) = x \ \ . \quad (2)$$

However, recovering such an inverse mapping is generally impossible [3]. Thus, we aim to find a mapping $\mathcal{R}$

$$\mathcal{R} : \mathcal{P}(\mathbb{R}^m) \rightarrow \mathbb{R}^m \ \ , \quad (3)$$

such that the set of points $\mathcal{R}(\mathcal{O}(P)) = \{\mathcal{R}(\mathcal{O}(x_j)) \in \mathbb{R}^m, \ j = 1, \ldots, n\}$ and their corresponding descriptors can reveal private information through inversion attacks [47]. *I.e.*, the mapping $\mathcal{R}$ should facilitate recognizing objects, text, or persons in images recovered from the point positions in $\mathcal{R}(\mathcal{O}(P))$ and their descriptors.

Naturally, if the points $\mathcal{R}(\mathcal{O}(x_j))$ are close to the original points x_j in the space $\mathbb{R}^m$, it can be expected that an inversion attack recovers a detailed image, potentially containing private information. Thus, if $d(\mathcal{R}(\mathcal{O}(x_j)), x_j) \leq \epsilon$, for some small ϵ [4], the obfuscation $\mathcal{O}$ cannot be considered as privacy preserving. In this paper, we show that having information about the neighborhoods of the original points, we can compute a mapping $\mathcal{R}$ for which ϵ is sufficiently small for most points. Thus, private details can be identified in images recovered from the $\mathcal{R}(\mathcal{O}(P))$.

Recovering points using neighborhood information. Let us assume that for each obfuscated input point $\mathcal{O}(x_j)$ we are given a set of neighbors $\mathcal{N}(\mathcal{O}(x_j)) = \{\mathcal{O}(x_{j_i}) : j_i \in [j_1 \ldots j_K]\}$. Here, $[j_1 \ldots j_K]$ are the indices of the K nearest neighbors (in $\mathbb{R}^m$) of the point x_j among all points in P. In other words, the set $\mathcal{N}(\mathcal{O}(x_j))$ contains the obfuscated representations corresponding to the K nearest neighbors in P of the original point x_j. The assumption that each original point x_j is contained in its set $\mathcal{O}(x_j)$, *i.e.*, $\forall j \ x_j \in \mathcal{O}(x_j)$ holds for approaches that map x to lines [35, 41, 61, 62] or planes [26] passing through x. It does not hold for coordinate permutation-based obfuscation. However, as detailed below, for this case we can extend $\mathcal{O}(x_j)$ to include all points on m lines passing through $\mathcal{O}(x_j)$ as one of them contains x_j.

We propose a strategy for computing a recovery mapping $\mathcal{R}$ (3) based on the following fact: for a recovered point $\mathcal{R}(\mathcal{O}(x_j))$ for which $d(\mathcal{R}(\mathcal{O}(x_j)), x_j) \leq \epsilon$, for some small ϵ, it holds that $d(\mathcal{R}(\mathcal{O}(x_j)), x_{j_i}) \leq \epsilon_2$, for all K nearest neighbors $x_{j_i}, i = 1, \ldots, K$ of x_j and a small ϵ_2 [5]. Since by assumption $x_{j_i} \in \mathcal{O}(x_{j_i})$ for all j_i, it also holds that $d(\mathcal{R}(\mathcal{O}(x_j)), \mathcal{O}(x_{j_i})) \leq \epsilon_3$ for all $\mathcal{O}(x_{j_i}) \in \mathcal{N}(\mathcal{O}(x_j))$ and $\epsilon_3 \leq \epsilon_2$, *i.e.*, the recovered point $\mathcal{R}(\mathcal{O}(x_j))$ has a small distance from all obfuscated representations of the K nearest neighbors of the point x_j. Since $x_j \in \mathcal{O}(x_j)$, we know that $\mathcal{O}(x_j)$ contains a point that is close to x_j and also close to all $\mathcal{O}(x_{j_i}) \in \mathcal{N}(\mathcal{O}(x_j))$. We thus propose to compute a recovery mapping $\mathcal{R}$ by minimizing the cost function:

$$\mathcal{R}(\mathcal{O}(x_j)) = \underset{x \in \mathcal{O}(x_j)}{\arg\min} \sum_{\mathcal{O}(x_{j_i}) \in \mathcal{N}(\mathcal{O}(x_j))} d(\mathcal{O}(x_{j_i}), x) \ \ . \quad (4)$$

Here, $d(\mathcal{O}(x_{j_i}), x)$ is the Euclidean distance of a point $x \in \mathbb{R}^m$ from its closest point in $\mathcal{O}(x_{j_i})$. Note that the point $\mathcal{R}(\mathcal{O}(x_j))$ that minimizes (4) can be far away from the true point position x_j. However, in our experience, using sufficiently many neighbors ”pulls” $\mathcal{R}(\mathcal{O}(x_j))$ towards x_j. Also note that we solve (4) per point x_j, rather than taking point estimates for the neighbors into account. This makes our recovery approach parallelizable.

In the following, we concretely discuss how we compute the recovery mapping for individual obfuscation schemes.

Points lifted to lines. For the obfuscation that maps a point x_j to a line passing through x_j, $\mathcal{O}(x_j)$ can be represented by the parameters of a line in $\mathbb{R}^m$. Thus, each point has a single degree of freedom, *i.e.*, a shift along the line. We solve (4) via least-squares minimization in this variable, which can be easily implemented using existing optimization libraries, *e.g.*, Ceres [3]. In our experience, the choice of initialization for $\mathcal{R}(\mathcal{O}(x_j))$ is not critical (see the supp. mat. for details).

In the case of paired-point lifting [35], each line passes through two original points. Each line contains the descriptors of both points, creating additional confusion as to which descriptor belongs to which point. Although this is not necessary for computing $\mathcal{R}$, it is important for applying inversion attacks. We provide implementation details in Sec. 6.

In the case of 3D ray clouds [41], each line passes through one original point and one of two additional center points. The center points are derived by clustering the point cloud into two clusters which centers are the center points. When solving (4), we ignore all neighbors corresponding to lines passing through the same center as $\mathcal{O}(x_j)$.

Points lifted to planes. The method suggested in [26] first splits the set of points into three disjoint sets P_x, P_y, and P_z. Each set is stored on a separate server. For the server storing P_y, each point $x \in P_y$ is represented by a plane parallel to the xz-plane passing through the y-coordinate of x. Similar obfuscations are used for the points in P_x and P_z [26]. We consider the case where an attacker has access to all three servers - hence having three sets of parallel planes, each orthogonal to the other two. This setting is realistic as access to all three servers is needed for 6D camera pose estimation.

Each obfuscated point $\mathcal{O}(x)$ can be represented by two parameters corresponding to shifts along two basis vectors of a plane. Thus, each point has two degrees of freedom. We solve (4) via a two-variable optimization problem to find

[3] For the coordinate permutation method [43], the recovery of the inverse mapping is theoretically possible; however, it results in a combinatorial problem that can be computationally infeasible to solve [43].

[4] Here, d is the Euclidean distance in $\mathbb{R}^m$.

[5] Here $\epsilon_2 = \epsilon + d(x_j, x_{j_l})$ for the farthest neighbor x_{j_l} from the K nearest neighbors of x_j.

the position on a plane that minimizes the sum of distances to neighboring planes. As for line lifting, the initialization of the point positions is not critical (*cf.* supp. mat.).

Coordinate permutation. The obfuscation scheme based on permuting coordinates [43] randomly subdivides P into pairs of points. For a pair of points x_j and x_i, [43] randomly chooses a coordinate, *e.g.*, the y-coordinate, and exchanges that coordinate between x_j and x_i. The obfuscation thus maps a point x_j to a single point $\mathcal{O}(x_j)$. Clearly, in general, it holds that $x_j \neq \mathcal{O}(x_j)$. However, note that $\mathcal{O}(x_j)$ shares $m-1$ coordinates with x_j. Thus x_j is contained in one of m lines, each parallel to one of the m axes, passing through $\mathcal{O}(x_j)$ [43]. These lines are used for camera pose estimation in [43]. We thus extend the obfuscation $\mathcal{O}(x_j)$ to contain all points on these lines, allowing us to use (4) to compute the mapping $\mathcal{R}$. In essence, this approach corresponds to lifting x_j to m lines, each one parallel to one of the m coordinate axes. In order to recover $\mathcal{R}(\mathcal{O}(x_j))$, we propose a method to determine along which of the m lines x_j has been moved (*i.e.*, which coordinate of x_j was exchanged). For details on this method, please see the supp. mat. Given the line, we then use the same approach as for point-to-line lifting to compute $\mathcal{R}(\mathcal{O}(x_j))$.

Robustness to imperfect neighborhoods. So far, we have assumed that we are given an estimate of the neighborhood $\mathcal{N}(\mathcal{O}(x_j))$ for $\mathcal{O}(x_j)$. Sec. 5 presents a practical approach for computing such estimates. However, the computed neighborhood estimates will contain outliers, *i.e.*, obfuscated representations of points that do not correspond to one of the K nearest neighbors of x_j. As detailed above, we compute the mapping $\mathcal{R}$ via least-squares minimization, which is affected by outliers. To add robustness to outliers in the given neighborhood estimates, we include the minimization problem in a RANSAC-like loop [23]. In each iteration, we select a small number of neighbors and use them to compute an estimate for $\mathcal{R}(\mathcal{O}(x_j))$. We compute the distances of this estimate to all $\mathcal{O}(x_{j_i}) \in \mathcal{N}(\mathcal{O}(x_j))$ and classify $\mathcal{O}(x_{j_i}) \in \mathcal{N}(\mathcal{O}(x_j))$ into inliers and outliers using a pre-decided threshold δ. We obtain the final estimate by solving (4) over the largest inlier set found by this approach. In practice, this approach is more robust than using a robust cost function in (4).

5. Estimating Neighborhoods From Descriptors

Computing the recovered points $\mathcal{R}(\mathcal{O}(x_j))$ from the obfuscations $\mathcal{O}(x_j)$ using (4) assumes that we have information about the neighbors of each original point x_j. In [11], such a neighborhood is geometrically estimated by using the distance between pairs of 3D lines as a proxy for the distance between the original points. However, their approach requires that the line directions are random and that lines are thus unlikely to intersect in 3D. This assumption does not hold for 2D lines, orthogonal 3D planes and ray clouds [41].

In the context of visual localization, each obfuscated point is associated with a descriptor that is used for matching the 2D image query with the 3D map points. We use these descriptors to estimate the required neighborhoods.

Intuitively, local structures (captured by the neighbors of a point) are not unique for each scene, but similar-looking structures can be found in other scenes. This motivates our learning-based approach for estimating the neighborhoods. Given enough scenes as training data, we let a neural network learn about such patterns, which in turn can be used to determine neighborhoods. We pose the task of neighborhood estimation as a feature matching task [37, 50, 63]: given a set of descriptors, we learn a similarity score between all pairs of descriptors that is inversely proportional to the distance between the original points. More specifically, the network takes as input the descriptors and outputs a row-normalized similarity matrix with high entries between the points that are likely to correspond to neighboring points. The network is made of several self-attention blocks [68] that draw contextual cues between the descriptors. It is trained in a supervised manner with the binary cross-entropy loss. The entry (i, j) of the similarity matrix is positive if the j^{th} point is within the K closest points to the i^{th} point.

The network training is only tied to the descriptor, and thus, a network can be trained on any data where point positions and associated descriptors are available, As shown in the experiments, such a simple network predicts neighborhoods that are sufficiently reliable to allow the proposed recovery method to reveal private content robustly. Note that our approach is intended as a proof-of-concept to show that our attack from Sec. 4 is practically feasible. We believe that better results can be obtained by tuning the network architecture and using larger training sets. However, such optimizations are outside the scope of this work.

6. Experimental Evaluation

We evaluate the recovery method based on how well it recovers the points obfuscated by 6 different obfuscation schemes in 3D - random lines [61] (OLC), two variants of paired points lines [35] (PPL and PPL+), the default ray clouds [41], planes [26], and Coordinate Permutation (CP) [43]. In 2D, we evaluate the recovery from points obfuscated using random lines [62] and with CP [43]. We experiment with two widely used descriptors: the handcrafted SIFT [38] and the learning-based SuperPoint [17]. To visually assess the information revealed by the point recovery, we further invert the recovered points and their descriptors into images of the scene with an inversion network [22, 47]. For most of our analysis, we use oracle-provided neighborhoods, *i.e.*, neighborhoods directly obtained from the nearest neighbors of the original points,

	Lines [62]						**Coordinate Permutation** [43]					
	7-scenes [59]			**Cambridge** [31]			**7-scenes** [59]			**Cambridge** [31]		
In.	5px	10px	25px	5px	10px	25px	5px	10px	25px	5px	10px	25px
1.0	47.5	75.1	95.0	60.3	88.1	99.0	45.6	71.7	92.3	61.0	87.4	98.5
0.75	49.3	77.6	96.1	61.4	89.4	99.3	46.4	72.7	91.8	61.7	88.0	98.1
0.5	49.7	78.8	96.9	61.1	89.9	99.4	40.2	63.9	80.2	55.6	80.9	90.5
0.3	44.6	73.1	92.0	56.3	85.6	96.5	19.8	32.3	43.6	25.2	38.2	46.1
0.2	34.3	56.9	74.8	44.5	69.0	80.6	9.7	15.9	24.6	9.7	14.8	20.5
0.1	16.0	26.2	40.0	18.4	28.1	38.2	4.1	6.9	13.2	3.1	4.9	8.4

Table 1. **Geometric accuracy of recovery from obfuscation of 2D SIFT [38] points.** Percentage of points recovered within error thresholds using oracle neighborhoods with different inlier ratios. Sizes: *7-scenes* [59] - 640 × 480 and *Cambridge* [31] - 1024 × 576.

	PPL [35]				**Plane** [26]				**CP** [43]				**Ray** [41]			
	7-scenes [59]		**Cambridge** [31]		**7-scenes** [59]		**Cambridge** [31]		**7-scenes** [59]		**Cambridge** [31]		**7-scenes** [59]		**Cambridge** [31]	
In.	10cm	25cm	25cm	50cm	10cm	25cm	25cm	50cm	10cm	25cm	25cm	50cm	10cm	25cm	25cm	50cm
1.0	94.6	97.3	69.2	83.2	93.4	97.5	65.2	81.1	88.2	94.5	65.3	81.0	94.6	97.9	72.1	83.6
0.75	94.7	97.1	66.9	80.4	93.0	97.0	56.2	67.7	89.1	95.8	66.3	82.0	93.3	96.8	72.9	83.1
0.5	95.0	97.2	67.2	79.3	82.8	88.7	33.2	38.5	67.7	75.0	61.4	72.5	91.9	95.7	74.4	84.1
0.3	94.8	97.1	68.2	78.8	42.1	60.4	15.0	17.1	40.9	46.2	35.4	40.6	86.2	90.5	75.5	84.8
0.2	94.0	96.8	69.0	78.4	20.9	39.6	8.1	9.4	31.1	35.1	24.1	27.2	78.7	83.6	75.0	84.2
0.1	78.2	84.5	69.1	76.2	7.5	20.7	2.9	3.8	22.8	26.2	16.5	18.2	49.9	57.1	63.8	72.7

Table 2. **Geometric accuracy of recovery from obfuscation of 3D points (suing SIFT [38]).** Ratio of points recovered within error thresholds from oracle neighborhoods with different inlier ratios (In.) on 7-scenes [59] and Cambridge [31] datasets. The line obfuscations PPL [35] and [41] are more susceptible to point recovery as compared to plane [26] and point-permutation [43] obfuscations.

rather than neighborhoods computed using our approach from Sec. 5. This provides us full control over the quality of assumed neighborhoods, *i.e.*, the inlier ratios, which is well suited for our analysis. This also allows us to showcase the robustness of our approach. Finally, to show the practical feasibility of the attack, we present results with neighborhoods estimated by our learning-based approach. The point recovery on these estimated neighborhoods reveals private content even though the proposed neighborhood network is only a proof of concept. As such, it is simple and has limited scalability to a few thousand descriptors whereas point clouds usually involve several hundred thousand descriptors. Therefore we run this end-to-end evaluation only in 2D. This network however shows the potential that a similar architecture could be trained for more descriptors given hardware with enough memory.

Implementation details. The minimization problem (4) is formulated as a least-square problem solved using the Ceres solver [3] in a RANSAC [23]-like loop. Given a set of obfuscated representations, each point is recovered individually using only its neighborhood. Although this approach does not model the dependencies between the recovery of each point, it allows for a simple parallelization and efficient runtimes, even on a single CPU (see the supp. mat.).

The oracle based neighborhoods are generated using the original points: a neighborhood of size K with inlier ratio $In.$ is made by first selecting the K nearest neighbor of the original point and replacing $(1 - In.) \cdot K$ of them with points randomly chosen from the set of non-neighbors. The learned neighborhoods are derived as the top-K elements of each row in the similarity matrix output by the network. The network is trained on the top K=20 neighbors of 309K images from 184 Scannet [15] scenes (see supp. mat.).

The recovered positions of the obfuscated points, together with the descriptors are fed to an inversion network [22, 47] to generate images of the scene. As a valid assumption in the context of visual localization, the descriptor is assumed to be not modified. This is true for all the obfuscations discussed in this paper except for PPL/PPL+ [35]. These obfuscations map a pair of points and the corresponding descriptors to the same line, without preserving the mapping between the points and their descriptors. We again use the neighborhood information to recover this point-descriptor mapping (see supp. mat. for details).

Datasets and metrics. We evaluate on the two indoor datasets 7-scenes [59] and 12-scenes [67], and the outdoor dataset Cambridge [31]. Results on 12-scenes are included in the supp. mat as they follow the same trend as results on 7-scenes. Similarly, results for SuperPoint [17] are left for supp. mat as they follow a similar trend as the results using SIFT [38]. We report the geometric accuracy as the fraction of points recovered within chosen error thresholds. The threshold is in pixels for 2D obfuscations and in cm for 3D. Larger thresholds are used for larger (outdoor) scenes in 3D. We compare the quality of the images generated from the recovered points against the ones generated from the original points by comparing their respective similarities to the real image. The similarity is computed with standard perceptual metrics: SSIM, PSNR and LPIPS [72]. We report the last two metrics in the supp. material.

Geometric evaluation. The geometric accuracies for 2D and 3D are reported in Tables 1 and 2, respectively. The proposed generic recovery method can consistently recover the points within a few pixels in the 2D case and within a

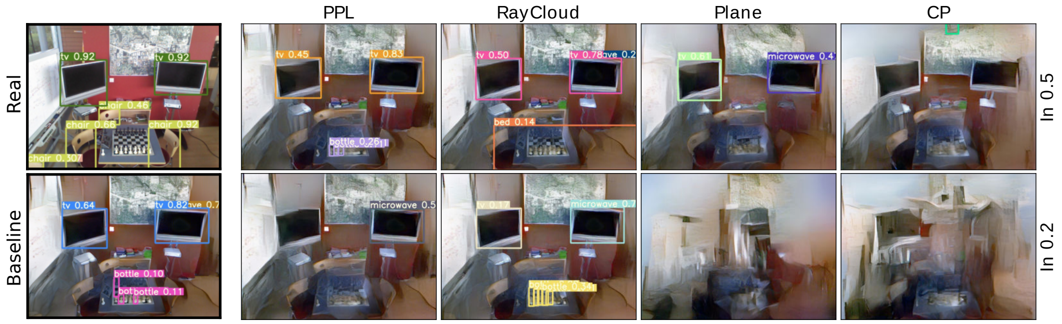

Figure 2. **Visual content revealed** by the inversion [47] from the original points ('Baseline') and the points recovered from the 3D obfuscations with neighborhood information at various levels of inlier ratios (In.). The original points are triangulated from SIFT [38] features. Line obfuscations (OLC) [61, 62], Point-Pair-Lines PPL [35] and RayClouds [41] are more vulnerable to neighborhood-based attacks than Planes [26] and Coordinate Permutation [43].

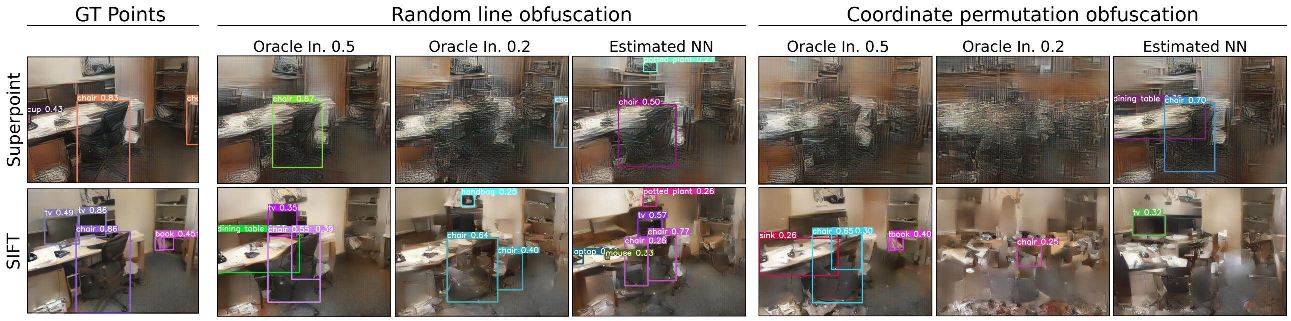

Figure 3. (Best viewed when zoomed in.) **Visual content revealed** by the inversion applied on points recovered from the obfuscated representations when using two different kinds of keypoints extractors and descriptors - SuperPoint [17] and SIFT [38]. The columns titled *Estimated NN* show the content revealed with an end-to-end attack, *i.e.*, starting from only descriptors, we carry out neighborhood estimation, point recovery, and inversion to the image space. The presence of identifiable scene content in the inverted images emphasizes the vulnerability of current geometry obfuscation techniques.

few cms in 3D maps. Note that for 2D and 3D line-based obfuscations the performance can peak when the neighborhood is not perfect, *i.e.*, when the inlier ratio is lower than 1.0, although the variation is small. This is because our robust method identifies outliers in the neighborhood and filters them out and using fewer close points can result in better accuracy using our method.

Perceptual evaluation. All perceptual metrics show consistent results so we report only the SSIM in Sec. 6. 'GT' is the baseline SSIM between the real image and the one generated from the *original* points, which can be interpreted as an upper bound for the SSIM between the real image and the one generated from the *recovered* points. The difference from this bound is higher for 2D than for 3D. This is because even a high geometric error in 3D can reduce to very few pixels upon projection while the recovery from obfuscations in 2D leads to several pixels of error. The larger error in 2D keypoint position estimation leads to worse image reconstructions in case of 2D obfuscations.

Learned neighborhoods. We evaluate the 2D point recovery from lines [62] and CP [43] using learned neighborhoods. We use the top-K=20 neighbors derived from the similarity output by our network from Sec.5. Tab. 4 shows the geometric and perceptual performance of an end-to-end attack, while Fig. 3 shows the inverted images.

An interesting observation is that the network learns the neighborhood more easily for SuperPoint [17] than for SIFT [38] as indicated by the accuracy gap between the two: for SuperPoint [17], the network leads to neighborhoods with acc. between 70% and 80% for $K \in [10, 100]$ while for SIFT [38], the accuracy remains around 35%. Thus one could argue that SIFT is more privacy-preserving than SuperPoint, although there is no guarantee that better neighborhood estimators for SIFT will not become available in the future. Moreover, SIFT [38] typically achieves lower localization performance than SuperPoint [53] and sacrificing performance for privacy might not be a satisfying solution in all scenarios.

Even though the images inverted from the recovered points are not perfect, the outline and the objects in the scene are recognizable. These results highlight an important limitation of pure geometric obfuscations and support the

	7Scenes. GT: 0.74		**Cambridge.** GT: 0.53	
In.	Lines	CP	Lines	CP
1.0	0.62	0.62	0.40	0.41
0.5	0.62	0.58	0.40	0.37
0.2	0.57	0.53	0.31	0.23

	7Scenes. GT: 0.58				**Cambridge.** GT: 0.39			
In.	PPL	Plane	CP	Ray	PPL	Plane	CP	Ray
1.0	0.57	0.55	0.56	0.57	0.36	0.36	0.36	0.37
0.5	0.56	0.49	0.51	0.56	0.36	0.32	0.34	0.37
0.2	0.54	0.43	0.43	0.54	0.36	0.31	0.27	0.37

Table 3. **Perceptual Evaluation** of point recoveries from geometric obfuscations in 2D (left) and 3D (right) with oracle neighborhoods. The original points are derived from SIFT [38] features. The SSIM↑ compares the original image to the images inverted [47] from recovered points. GT refers to the SSIM of the image inverted from the original points and sets the baseline. The SSIM for recovered points is in general close to the baseline, demonstrating that the image content is recovered.

	Superpoint [17]						**SIFT** [38]					
	Geometric				**Perceptual**		**Geometric**				**Perceptual**	
	Lines		**CP**		**Lines**	**CP**	**Lines**		**CP**		**Lines**	**CP**
Neighborhood	10px	25px	10px	25px	SSIM - GT:0.57		10px	25px	10px	25px	SSIM - GT:0.74	
Oracle In. 0.75	61.4	91.2	54.4	83.4	0.46	0.45	77.6	96.1	72.7	91.8	0.62	0.58
Oracle In. 0.5	63.0	92.9	45.6	70.3	0.46	0.42	78.8	96.9	63.9	80.2	0.62	0.58
Oracle In. 0.3	57.8	87.8	23.7	38.4	0.45	0.40	73.1	92.0	32.3	43.6	0.61	0.53
Oracle In. 0.2	45.1	70.8	13.3	23.1	0.40	0.40	56.9	74.8	15.9	24.6	0.57	0.53
Oracle In. 0.1	22.9	39.0	6.7	13.3	0.33	0.40	26.2	40.0	6.9	13.2	0.51	0.52
Estimated (Ours)	53.2	86.2	48.3	80.3	0.46	0.45	47.3	68.1	30.1	45.8	0.57	0.55

Table 4. **End-to-end attack evaluation**. Geometric and perceptual evaluation of the recovery when using neighbors estimated by our network described in Sec. 5 (last row) for two different types of keypoint detectors and extractors—Superpoint [17] and SIFT [38]. The performance when using oracle-provided neighborhoods of different qualities is provided for comparison. The recovery of neighborhoods is observed to be much more effective using Superpoint [17] descriptors compared to SIFT [38].

two claims made in the paper: i) the neighborhood information can be learned from the descriptors, and reiterates that geometric obfuscations alone are not as privacy-preserving as they claim. One needs to also prevent neighborhood information from being inferred from the obfuscations. ii) it shows that the proposed proof of concept to compute the neighborhood information is already sufficient for the proposed point recovery to be applicable. We expect that more complex neighborhood learning will lead to better results. This calls for potential future work on fusing geometric and descriptor obfuscation to prevent neighborhood recovery.

Discussion. The results reveal that the proposed recovery method performs well even if the neighborhoods contain significant fractions of outliers. Fig. 2 and Fig. 3 further show that images generated from recovered points can reveal potentially private user content, which is particularly true for line-based obfuscations [35, 61, 62]. The geometric constraints of parallel planes [26] make recovery difficult, but neighborhoods with reasonable inlier ratios make the plane obfuscation also susceptible to the proposed recovery. The same holds for Coordinate Permutation [43]: the additional step of estimating which coordinate was permuted brings in more noise into our method for recovering points. However, neighborhoods with inlier ratios of 0.5 or more are enough to enable recovery accurate enough to reveal identifiable scene content. The network described in Sec. 5 can produce such informative neighborhoods even with its simple design. We expect methods in future works to improve the estimation of neighborhoods from descriptors, further highlighting the discussed vulnerability of obfuscation schemes. Future methods in de-noising the neighborhood graphs estimated from descriptor and/or geometry can help reduce the error in point position recovery. Similarly, more sophisticated inversion attacks that are robust to small noise in point positions can increase the privacy risk.

7. Conclusion

In this work, we highlight a common vulnerability of all geometry-based obfuscation techniques that have so far been presented as privacy-preserving representations. We present a simple optimization-based method that uses knowledge of point neighborhoods to recover point positions from the discussed obfuscation schemes. We show the robustness of our method and analyze the recovery accuracy by using oracle-provided neighborhoods with varying inlier ratios. Finally, using a neural network that learns to identify local feature descriptors co-occurring across scenes, we show that it is possible to estimate these neighborhoods from the descriptors associated with points. The inverted images from the recovered point positions reveal private scene content, highlighting the drawback of current methods and the need for guarantees on under which circumstances a data representation is indeed privacy-preserving.

Acknowledgements. This work was supported by the Czech Science Foundation (GACR) JUNIOR STAR Grant No. 22-23183M, the EU Horizon 2020 project RICAIP (No. 857306), the Chalmers AI Research Center (CHAIR), WASP and SSF. The compute and storage were partially supported by NAISS projects numbered NAISS-2024/22-637 and NAISS-2024/22-237 respectively.

References

[1] Hoggles: Visualizing object detection features. In *CVPR*, pages 1–8, 2013. 2

[2] Yehya Abouelnaga, Mai Bui, and Slobodan Ilic. Distillpose: Lightweight camera localization using auxiliary learning, 2021. 3

[3] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver, 2023. 4, 6

[4] Vassileios Balntas, Shuda Li, and Victor Adrian Prisacariu. Relocnet: Continuous metric learning relocalisation using neural nets. In *European Conference on Computer Vision*, 2018. 3

[5] Daniel Barath and Gábor Valasek. Space-partitioning ransac. In *European Conference on Computer Vision*, pages 721–737. Springer, 2022. 2

[6] Daniel Barath, Luca Cavalli, and Marc Pollefeys. Learning to find good models in ransac. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15744–15753, 2022. 2

[7] Eric Brachmann and Carsten Rother. Neural-guided ransac: Learning where to sample model hypotheses. In *CVPR*, pages 4322–4331, 2019. 3

[8] Eric Brachmann and Carsten Rother. Visual camera relocalization from rgb and rgb-d images using dsac. *IEEE TPAMI*, 44(9):5847–5865, 2021. 2

[9] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *CVPR*, pages 6684–6692, 2017. 3

[10] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5044–5053, 2023. 2, 3

[11] Kunal Chelani, Fredrik Kahl, and Torsten Sattler. How privacy-preserving are line clouds? recovering scene details from 3d lines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15668–15678, 2021. 2, 3, 5

[12] Kunal Chelani, Torsten Sattler, Fredrik Kahl, and Zuzana Kukelova. Privacy-preserving representations are not enough: Recovering scene content from camera poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13132–13141, 2023. 3

[13] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized ransac. In *Pattern Recognition: 25th DAGM Symposium, Magdeburg, Germany, September 10-12, 2003. Proceedings 25*, pages 236–243. Springer, 2003. 2

[14] Mark J Cummins and Paul M Newman. Fab-map: Appearance-based place recognition and mapping using a learned visual vocabulary model. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 3–10, 2010. 1

[15] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes, 2017. 6

[16] Deeksha Dangwal, Vincent T. Lee, Hyo Jin Kim, Tianwei Shen, Meghan Cowan, Rajvi Shah, Caroline Trippel, Brandon Reagen, Timothy Sherwood, Vasileios Balntas, Armin Alaghi, and Eddy Ilg. Analysis and mitigations of reverse engineering attacks on local feature descriptors. In *British Machine Vision Conference (BMVC)*, 2021. 1, 2, 3

[17] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR workshops*, pages 224–236, 2018. 1, 5, 6, 7, 8

[18] MWM Gamini Dissanayake, Paul Newman, Steve Clark, Hugh F Durrant-Whyte, and Michael Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on robotics and automation*, 17(3): 229–241, 2001. 1

[19] Tien Do, Ondrej Miksik, Joseph DeGol, Hyun Soo Park, and Sudipta N Sinha. Learning to detect scene landmarks for camera localization. In *CVPR*, pages 11132–11142, 2022. 2

[20] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. *Advances in neural information processing systems*, 29, 2016.

[21] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *CVPR*, pages 4829–4837, 2016.

[22] Mihai Dusmanu, Johannes L Schonberger, Sudipta N Sinha, and Marc Pollefeys. Privacy-preserving image features via adversarial affine subspace embeddings. In *CVPR*, pages 14267–14277, 2021. 1, 2, 3, 5, 6

[23] 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*, 24(6):381–395, 1981. 1, 2, 5, 6

[24] Marcel Geppert, Viktor Larsson, Pablo Speciale, Johannes L Schönberger, and Marc Pollefeys. Privacy preserving structure-from-motion. In *ECCV*, pages 333–350. Springer, 2020. 1, 3

[25] Marcel Geppert, Viktor Larsson, Pablo Speciale, Johannes L Schonberger, and Marc Pollefeys. Privacy preserving localization and mapping from uncalibrated cameras. In *CVPR*, pages 1809–1819, 2021.

[26] Marcel Geppert, Viktor Larsson, Johannes L Schönberger, and Marc Pollefeys. Privacy preserving partial localization. In *CVPR*, pages 17337–17347, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[27] Vladimir Guzov, Aymen Mir, Torsten Sattler, and Gerard Pons-Moll. Human poseitioning system (hps): 3d human pose estimation and self-localization in large scenes from body-mounted sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4318–4329, 2021. 1

[28] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1

[29] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar

de Souza, Vincent Leroy, and Gabriela Csurka. Robust Image Retrieval-based Visual Localization using Kapture. arXiv:2007.13867, 2020. 2

[30] Hiroharu Kato and Tatsuya Harada. Image reconstruction from bag-of-visual-words. In *CVPR*, pages 955–962, 2014. 2

[31] 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

[32] Z. Kukelova, M. Bujnak, and T. Pajdla. Real-Time Solution to the Absolute Pose Problem with Unknown Radial Distortion and Focal Length. In *ICCV*, 2013. 1

[33] Zuzana Kukelova, Jan Heller, and Andrew Fitzgibbon. Efficient Intersection of Three Quadrics and Applications in Computer Vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[34] Viktor Larsson, Torsten Sattler, Zuzana Kukelova, and Marc Pollefeys. Revisiting Radial Distortion Absolute Pose. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 1

[35] Chunghwan Lee, Jaihoon Kim, Chanhyuk Yun, and Je Hyeong Hong. Paired-point lifting for enhanced privacy-preserving visual localization. In *CVPR*, pages 17266–17275, 2023. 1, 2, 3, 4, 5, 6, 7, 8

[36] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide Pose Estimation Using 3D Point Clouds. In *ECCV*, 2012. 2

[37] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023. 5

[38] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2), 2004. 1, 5, 6, 7, 8

[39] Zixin Luo, Lei Zhou, Xuyang Bai, Hongkai Chen, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Aslfeat: Learning local features of accurate shape and localization. *Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[40] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, pages 5188–5196, 2015. 2

[41] Heejoon Moon, Chunghwan Lee, and Je Hyeong Hong. Efficient privacy-preserving visual localization using 3d ray clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9773–9783, 2024. 1, 2, 3, 4, 5, 6, 7

[42] Tony Ng, Hyo Jin Kim, Vincent T Lee, Daniel DeTone, Tsun-Yi Yang, Tianwei Shen, Eddy Ilg, Vassileios Balntas, Krystian Mikolajczyk, and Chris Sweeney. Ninjadesc: content-concealing visual descriptors via adversarial learning. In *CVPR*, pages 12797–12807, 2022. 1, 3

[43] Linfei Pan, Johannes L Schönberger, Viktor Larsson, and Marc Pollefeys. Privacy preserving localization via coordinate permutations. In *ICCV*, pages 18174–18183, 2023. 1, 2, 3, 4, 5, 6, 7, 8

[44] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. MeshLoc: Mesh-Based Visual Localization. In *ECCV*, 2022. 2

[45] Maxime Pietrantoni, Martin Humenberger, Torsten Sattler, and Gabriela Csurka. Segloc: Learning segmentation-based representations for privacy-preserving visual localization. In *CVPR*, pages 15380–15391, 2023. 1, 2

[46] Francesco Pittaluga and Bingbing Zhuang. Ldp-feat: Image features with local differential privacy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17580–17590, 2023. 1, 3

[47] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In *CVPR*, pages 145–154, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[48] Gerard Pons-Moll, Vladimir Guzov, Julian Chibane, Riccardo Marin, Yannan He, and Torsten Sattler. Interaction replica: Tracking human-object interaction and scene changes from human motion. 2023. 1

[49] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 1, 2

[50] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, pages 4938–4947, 2020. 5

[51] Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, and Marc Pollefeys. Lamar: Benchmarking localization and mapping for augmented reality. In *European Conference on Computer Vision*, pages 686–704. Springer, 2022. 1

[52] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. *PAMI*, 39(9):1744–1756, 2017. 1, 2

[53] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8601–8610, 2018. 7

[54] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression, 2019. 3

[55] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1, 2

[56] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 501–518. Springer, 2016. 1, 2

[57] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers, 2021. 3

[58] Mikiya Shibuya, Shinya Sumikura, and Ken Sakurada. Privacy preserving visual slam. In *ECCV*, pages 102–118. Springer, 2020. 1, 2

[59] 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 *CVPR*, pages 2930–2937, 2013. 2, 6

[60] Zhenbo Song, Wayne Chen, Dylan Campbell, and Hongdong Li. Deep view synthesis from colored 3d point clouds, 2020. 1, 2

[61] Pablo Speciale, Johannes L Schonberger, Sing Bing Kang, Sudipta N Sinha, and Marc Pollefeys. Privacy preserving image-based localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5493–5503, 2019. 1, 2, 3, 4, 5, 7, 8

[62] Pablo Speciale, Johannes L Schonberger, Sudipta N Sinha, and Marc Pollefeys. Privacy preserving image queries for camera localization. In *ICCV*, pages 1486–1496, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[63] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021. 5

[64] Lauri Suomela, Jussi Kalliola, Atakan Dag, Harry Edelman, and Joni-Kristian Kämäräinen. Benchmarking visual localization for autonomous navigation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2945–2955, 2023. 1

[65] Janine Thoma, Danda Pani Paudel, Ajad Chhatkuli, Thomas Probst, and Luc Van Gool. Mapping, localization and path planning for image-based navigation using visual features and map. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7383–7391, 2019. 1

[66] Carl Toft, Will Maddern, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, Fredrik Kahl, and Torsten Sattler. Long-term visual localization revisited. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):2074–2088, 2022. 1

[67] Julien Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *3DV*, pages 323–332. IEEE, 2016. 6

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 5

[69] Philippe Weinzaepfel, Hervé Jégou, and Patrick Pérez. Reconstructing an image from its local descriptors. In *CVPR*, pages 337–344. IEEE, 2011. 2

[70] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[71] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. Springer, 2014. 2

[72] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[73] Qunjie Zhou, Sérgio Agostinho, Aljoša Ošep, and Laura Leal-Taixé. Is geometry enough for matching in visual localization? In *ECCV*, 2022. 3