Privacy-Preserving Representations are not Enough: Recovering Scene Content from Camera Poses

Kunal Chelani[1] Torsten Sattler[2] Fredrik Kahl[1] 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,fredrik.kahl}@chalmers.se torsten.sattler@cvut.cz kukelzuz@fel.cvut.cz

Abstract

Visual localization is the task of estimating the camera pose from which a given image was taken and is central to several 3D computer vision applications. With the rapid growth in the popularity of AR/VR/MR devices and cloud-based applications, privacy issues are becoming a very important aspect of the localization process. Existing work on privacy-preserving localization aims to defend against an attacker who has access to a cloud-based service. In this paper, we show that an attacker can learn about details of a scene without any access by simply querying a localization service. The attack is based on the observation that modern visual localization algorithms are robust to variations in appearance and geometry. While this is in general a desired property, it also leads to algorithms localizing objects that are similar enough to those present in a scene. An attacker can thus query a server with a large enough set of images of objects, e.g.*, obtained from the Internet, and some of them will be localized. The attacker can thus learn about object placements from the camera poses returned by the service (which is the minimal information returned by such a service). In this paper, we develop a proof-of-concept version of this attack and demonstrate its practical feasibility. The attack does not place any requirements on the localization algorithm used, and thus also applies to privacy-preserving representations. Current work on privacy-preserving representations alone is thus insufficient.*

1. Introduction

Visual localisation refers to the problem of estimating the camera pose of a given image in a known scene. It is a core problem in several 3D computer vision applications, including self-driving cars [17, 18] and other autonomous robots [50], and Augmented Reality [5, 23, 25].

A popular approach for Augmented/Mixed/Virtual Reality (XR) applications is to use a client-server mechanism for localization: the user device (client) sends image data to a cloud-based system (server) that computes and returns the camera pose [23, 25, 46]. Examples of such services include Google's Visual Positioning System [29], Microsoft's Azure Spatial Anchors [24], and Niantic's Lightship [39]. Cloud-based localization services are popular for multiple reasons - *first*, performing localization on the server reduces storage requirements and the computational load, and thus energy consumption, which is important for client devices such as mobile phones and headsets; *second*, it enables using robust mapping and localization algorithms that are too expensive for mobile devices; *third*, in the context of collaborative mapping, *e.g.*, for the AR cloud or autonomous driving, maintaining a single scene representation in a centralized place is far easier than keeping multiple copies on various mobile devices up-to-date.

Naturally, sending user data to a server, *e.g.*, in the form of images to be localized or 3D maps recorded by users that will be used for localization, raises privacy concerns [9, 41, 42]. Work on privacy-preserving localization aims to resolve these concerns by ensuring that private details cannot be recovered from the data sent [14, 26, 42] to or stored on the server [11, 11, 15, 28, 36, 41, 52].

Existing work focuses on scenarios where an attacker gains access to the localization service or can eavesdrop on the communication between client and server. In this work, we demonstrate that it is possible for an attacker to learn about the content of a scene stored on a localization server without direct access to the server. We show that a localization service will reveal scene-related information through estimated camera poses, *i.e.*, through its normal operation process. The attack is based on two recent developments: (1) modern visual localization algorithms are designed to be robust against changes such as illumination and seasonal variations [44]. This is an essential property for cloud-based localization services in order to operate robustly and reli-

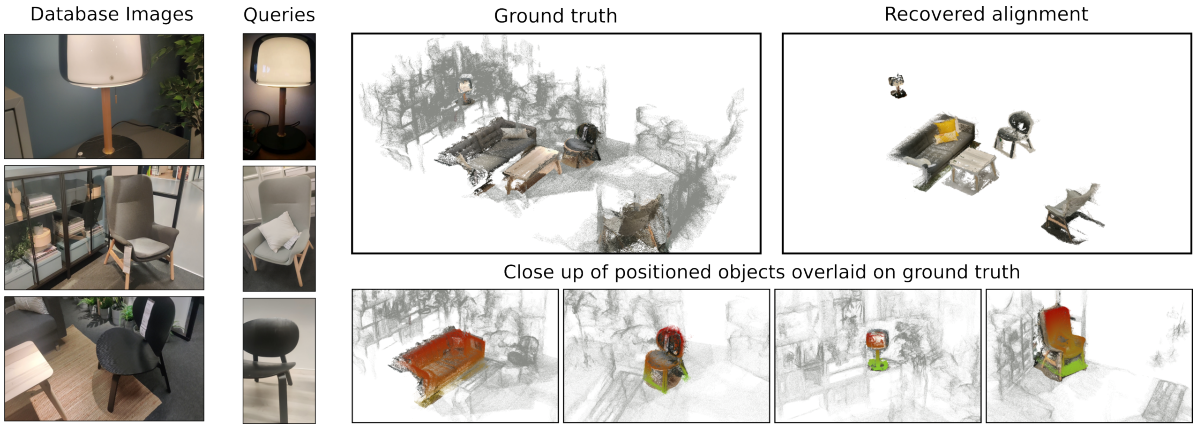

Figure 1. In the context of privacy-preserving localization, we show that it is possible to learn about the content of a scene using camera poses returned by a localization service, without any direct access to the scene representation. (**1st column**) Examples of images from the scene, used to build the scene representation. The images are shown for illustrative purposes and are not available to an attacker trying to learn about the scene. (**2nd column**) The attacker queries the service with images of objects, *e.g.*, downloaded from the Internet. (**3rd & 4th column**) Using the camera poses for the query image returned by the localization service, the attacker is able to identify the types of objects present in the scene and to accurately place them in the scene. We show the estimated object poses overlaid over the ground truth structure of the scene (which is not accessible to the attacker). The attacker is able to faithfully recover the placement of objects. Overall, our results demonstrate that simple feedback such as camera poses is already sufficient to potentially reveal private details.

ably. However, since these algorithms are robust to (slight) variations in appearance and geometry, they will also localize images showing objects that are similar (but not necessarily identical) to those objects present in the scene. (2) massive amounts of images depicting objects in different variations are readily available on the Internet. Taken together, both developments allow an attacker to repeatedly query the server with images and to recover the positions of the objects in the scene based on the camera poses returned by the server (*cf*. Fig. 1). In this paper, we demonstrate the feasibility of this attack by developing a proof-of-concept implementation of the attack.

In summary, we make the following contributions: **(1)** we identify a new line of attack in the context of privacy-preserving visual localization based on the camera poses returned by a cloud-based server. **(2)** we show the feasibility of the attack through a proof-of-concept implementation of the attack. Through experiments, we explore the performance of our implementation as well as the trade-off between localization robustness and potential defenses against the attack. **(3)** the attack is agnostic to the underlying localization algorithm and thus applicable even if the localization system is otherwise perfectly privacy-preserving. This paper thus proposes a new research direction for privacy-preserving localization, where the aim for the localization service is to correctly identify whether a query image was taken in the concerned scene or not, in order to prevent leaking information through camera poses.

2. Related Work

Visual localization. Most state-of-the-art visual localization algorithms are based on establishing 2D-3D matches between a query image and a 3D model of the scene. These correspondences are then used for camera pose estimation. The 3D model can either be stored explicitly [19–21, 27, 31–33, 43], *e.g.*, in the form of a Structure-from-Motion (SfM) point cloud, or implicitly in the form of the weights of a machine learning model [1–3, 6, 38, 45]. In the former case, local feature descriptors are associated with 3D points of the model. It has been shown that this information is sufficient to recover detailed images from the 3D map [28, 40], although sparsifying these models [4, 51] might effectively make them privacy-preserving [7]. Approaches based on implicit representations map image pixels or patches to 3D points by training scene coordinate regression models [3, 38]. Recently, it was claimed that such approaches are inherently privacy-preserving [11]. However, feature-based methods currently scale better to large scenes and are able to better handle condition changes [44], such as illumination or seasonal changes, between the query image and the database images used to build the the scene representation. The resulting robustness is highly important in many applications of visual localization, including AR and robotics. The robustness is a direct consequence of recent advances in local features [10, 13, 30] and effective feature matchers [32, 43, 48, 53]. In this paper, we show that robustness to changing conditions enables an attacker to learn about the content of the scene: robustness to changing conditions not only bridges the gap between (small) varia-

tions in scene appearance and geometry observed in images depicting the same place, but also leads to correspondences between images depicting similar but not identical objects, *e.g.*, different chairs. In turn, these correspondences can be used to localize the object in the scene, which is the basis for the attack described in this work. Note that the properties we exploit are inherent to robust localization algorithms and are not restricted to feature-based methods. Ultimately, any robust localization system needs to handle variations in shape and appearance.

Privacy-preserving visual localization. Existing work on privacy-preserving localization focuses on two points of attack: (1) ensuring that data sent to a localization service does not reveal private information. (2) ensuring that data stored on a localization service does not reveal private information. For the former case, it has been shown that images can be recovered from local features [9, 12, 49]. Work on privacy-preserving queries to a localization server thus mostly aims at developing features that prevent image recovery [14, 26] or on obfuscating the feature geometry [16, 42]. Similarly, work on privacy-preserving scene representation aims to obfuscate the geometry [37,41] (although scene geometry can be recovered under certain conditions [7]), splitting the maps over multiple server for increased data security [15], using implicit representations [11], or storing raw geometry without any feature descriptors [52].

This paper presents a new line of attack that complements existing work. Previous work considers a scenario where the attacker gains access to the service. In contrast, we show that it is possible to recover scene content from the very basic information provided by any localization service, namely the camera poses estimated for query images. As such, the attack is still feasible even if the data send to and stored on the server is completely privacy-preserving. Our work thus shows that existing privacy-preserving localization approaches are not sufficient to ensure user privacy.

3. Recovering Scenes from Camera Poses

Any localization system returns the camera poses of localized query images. At the same time, modern localization algorithms aim to be robust to shape and appearance variations in order to be robust to changes in viewing conditions. This feature, however, opens up the possibility that not only genuine queries, but also images of objects that are similar to the ones present in the scene can be localized. The camera poses of the localized images can then in turn be used to infer the positions of certain objects in the scene, potentially revealing more information about the scene than the cloud-based service / a user would like to disclose.

Naturally, an attacker does not know which objects are present in the scene and thus which images to use for their queries. The Internet is a source of a theoretically unlimited number of images, videos, and 3D models of objects of different types and appearances. This naturally leads to an idea of a potential attack, where an attacker just downloads such images and videos, bombards the server with localization requests, and uses poses of localized images to reveal detailed scene structure.

In the following sections, we investigate this new type of attack, and we try to answer several questions: Can an attacker with access to images and videos of objects similar to those present in the scene easily learn about the presence/absence of different objects and their placement in the scene just from the poses returned by a localization service? What are the challenges of such an attack, and are these challenges easily solvable? Can cloud-based services easily prevent such attacks? To this end, we present a proof-of-concept implementation of the attack.[1] Later, Sec. 6 discusses an approach to potentially mitigate the attack and why its effectiveness is limited.

3.1. Formalization

We assume a localization server $\mathcal{L}$ that is responsible for localizing images in a scene $\mathcal{S}$. $\mathcal{L}$ tries to align each query image it receives with the scene representation as best as possible. If an image can be localized, the server returns a 6-dof camera pose $[\mathbf{R}|\mathbf{t}]$. We assume that the scale of the translation component $\mathbf{t}$ is in known.

An adversary $\mathcal{A}$ is querying $\mathcal{L}$ with many images of different objects, where each image contains only one dominant object to avoid confusion about which object from the image was localized in the scene. $\mathcal{A}$, using the poses returned by $\mathcal{L}$, wants to learn about the presence/absence of objects in the scene $\mathcal{S}$, and wants to infer their (approximate) positions. As such, $\mathcal{A}$ tries to construct an (approximate) "copy" of the scene $\mathcal{S}$ or at least its layout.

In this setting $\mathcal{A}$ needs to deal with two challenges:

1. $\mathcal{A}$ queries $\mathcal{L}$ with images of objects that, in general, differ geometrically from the actual objects in the scene. In the best case, the pose returned by the server provides the best-possible approximate alignment between the query and actual object. In general, the returned poses will be noisy and can be quite inaccurate if only a part of the object, *e.g.*, a chair's leg, is aligned. Creating an accurate "copy" of the scene from such poses is a challenging problem.

2. $\mathcal{A}$ has, in general, no a-priori information about the type of the scene and which objects are visible in it. Since $\mathcal{L}$ can also return poses for objects that are not in the scene, $\mathcal{A}$ needs to have a mechanism for deciding the presence/absence of an object based on the re-

[1] We only aim to show feasibility. We believe that better attack algorithms are certainly possible.

Algorithm 1 Best single camera based alignment between sets of poses

Input $\mathbf{P}_o = \{[\mathbf{R}_i|\mathbf{t}_i]\}$,$\hat{\mathbf{P}}_o = \{[\hat{\mathbf{R}}_i|\hat{\mathbf{t}}_i]\}$, δ_r,δ_t
Output $\mathbf{R}_{best}, \mathbf{t}_{best}, \epsilon$

1: **procedure** GET-BEST-ALIGNMENT
2: $\mathtt{N} \leftarrow |\mathbf{P}_o|$
3: $\mathtt{Inliers\_best} \leftarrow \phi$
4: **for** `i = 1 to N` **do**
5: $\mathbf{R}_{est} \leftarrow \hat{\mathbf{R}}_i^\top \mathbf{R}_i$
6: $\mathbf{t}_{est} \leftarrow \hat{\mathbf{R}}_i^\top (\mathbf{t}_i - \hat{\mathbf{t}}_i)$
7: $\mathtt{Inliers} \leftarrow \phi$
8: **for** `j = 1 to N` **do**
9: $\Delta_r \leftarrow \angle(\mathbf{R}_j \mathbf{R}_{est}^\top \hat{\mathbf{R}}_j^\top)$
10: $\Delta_t \leftarrow ||\hat{\mathbf{R}}_j^\top \hat{\mathbf{t}}_j - \mathbf{R}_{est}\mathbf{R}_j^\top \mathbf{t}_j + \mathbf{t}_{est})||$
11: **if** $\Delta_r < \delta_r$ and $\Delta_t < \delta_t$ **then**
12: $\mathtt{Inliers} \leftarrow \mathtt{Inliers} \cup \{\mathtt{j}\}$
13: **if** $|\mathtt{Inliers}| > |\mathtt{Inliers\_best}|$ **then**
14: $\mathtt{Inliers\_best} \leftarrow \mathtt{Inliers}$
15: $\epsilon \leftarrow |\mathtt{Inliers\_best}|/\mathtt{N}$
16: $\mathbf{R}_{best}, \mathbf{t}_{best} \leftarrow$ **Average(`Inliers_best`)**

turned poses. Naturally, having to deal with noisy and inaccurate poses makes the decision process harder.

In general, it is not possible to overcome these challenges by using a single image of each object. A single camera pose returned by $\mathcal{L}$, without additional information, does not provide enough data for deciding about the presence/absence of the object in the scene and the quality of the pose.

However, given the large amount of images available on the Internet, and in particular the availability of videos, $\mathcal{A}$ can use several images of the same object taken from different viewpoints. Jointly reasoning about all of the corresponding poses obtained for these images can then be used to decide the presence and position of the object.

3.2. 3D Object Placement

Assuming that the attacker knows that an object is present, they still need to predict its position and orientation in the scene based on the pose estimates provided by the server. To this end, the attacker can use that multiple images of the same object taken from different viewpoints are available. These images can be used by $\mathcal{A}$ to build a local 3D model $\mathcal{M}$, *e.g.*, using SfM [34] and MVS [35], and to compute the poses $\mathbf{P}_o$ of these images w.r.t. this model. In turn, $\mathcal{L}$ provides a set $\hat{\mathbf{P}}_o$ of poses for (a subset of) these images in the coordinate system of the scene model $\mathcal{S}$. The problem of placing the object in the copy of the scene $\mathcal{S}$ thus reduces to the problem of aligning both sets of poses (*cf*. Fig. 2). The camera poses $\hat{\mathbf{P}}_o$ provided by $\mathcal{L}$ can be very noisy and can contain outliers. Thus, the alignment process needs to be robust.

As mentioned above, for simplicity we assume that the scale of the 3D model stored by $\mathcal{L}$ is known.[2] Similarly, the scale of the local model $\mathcal{M}$ can be (approximately) recovered using the known size of the object. In this case, the two poses, in the coordinate systems of $\mathcal{M}$ and $\mathcal{S}$, for a single image already provide an alignment hypothesis, *i.e.*, the relative pose between them. As outlined in Alg. 1, we evaluate all hypotheses. The input to Alg. 1 are the two sets of poses, $\mathbf{P}_o$ and $\hat{\mathbf{P}}_o$, and two error thresholds - δ_r for rotation and δ_t for translation. For each pair of corresponding camera poses - local and server-provided, a relative transformation is computed (line 5-6). One set of poses is transformed using this estimated transformation, and errors for rotation and translation between corresponding pairs are computed (Lines 9-10). Using the two thresholds, we determine which other pose pairs are inliers to the pose hypothesis (Lines 11-12). The transformation with the largest number of inliers is selected (Lines 13-14) and refined by averaging the relative poses of all inliers.

Obviously, not knowing the scale of the scene model $\mathcal{S}$ is insufficient to prevent the attacker from placing the object in the scene as the scale and relative transformation can be recovered from two pairs of poses. Additionally, there are ways to further robustify the alignment process. *E.g.*, if images of multiple very similar instance of an object and the corresponding 3D models are available, it seems reasonable to assume that images of different instances taken from similar viewpoints will also result in similar pose estimates by $\mathcal{L}$. These estimates can then be used to average out noise in the poses. Similarly, the relation between different objects, *e.g.*, a monitor standing on a desk, can be used to stabilize the process of placing objects in the scene. However, we do not investigate such advanced strategies in this paper.

3.3. Deciding the Presence/Absence of an Object

We assume that $\mathcal{L}$ is running a localization algorithm that is robust to shape and appearance variations and that is aligning query images to the scene as best as it can. At the same time, $\mathcal{L}$ can also return poses for objects that are not in the scene, as well poses for objects that are not even from the same categories or similar to objects present in the scene. Deciding if an object is present or not in a scene based on the poses returned for its images by the localization server is therefore a challenging problem.

For an attacker $\mathcal{A}$ trying to recover scene information via camera poses, it is impossible to determine which type of objects are present using just a single camera pose returned for one query image of each of the objects.

To overcome this challenge, $\mathcal{A}$ can employ several possible techniques; *e.g.*, they can use statistics about object co-occurrence to select the set of queries and associated camera

[2] In the context of user-generated maps, captured by devices with IMUs such as mobile phones or dedicated XR headsets, it seems realistic to assume that the scale of the maps is provided in meters.

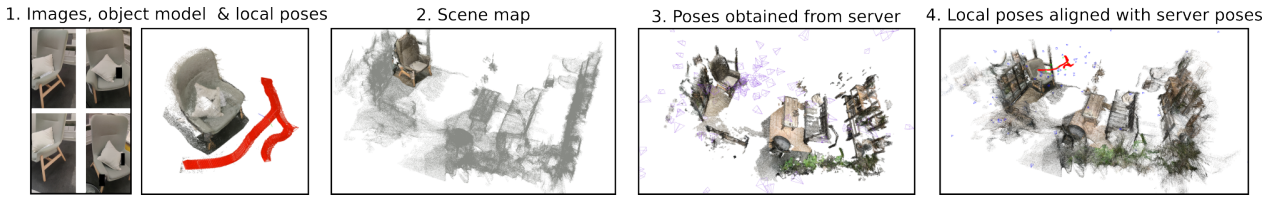

Figure 2. **Object alignment example**: **1.** A 3D model $\mathcal{M}$ of an object and corresponding camera poses $\mathbf{P}_o$ in the attacker's local coordinate system, built from a sequence of object images. **2.** The server scene with a similar object. **3.** The noisy poses returned by the server for the queried object images. **4.** Sequences of local and server-provided poses aligned to approximately place the object in the scene.

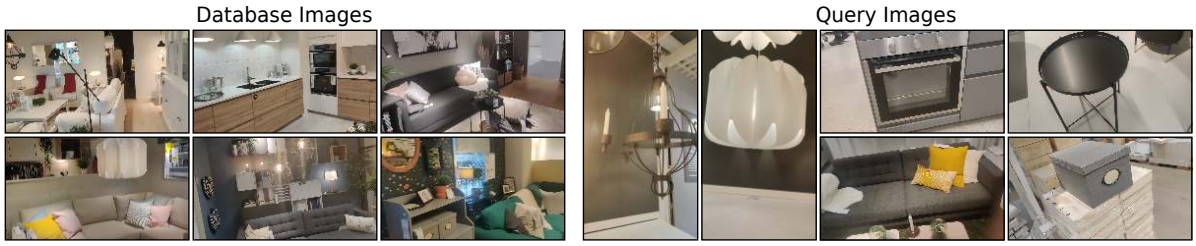

Figure 3. Example images from *IKEA-Scenes* (left) and one of the objects of corresponding scenes in *IKEA-Objects* (right).

poses having a high probability of their spatial distribution. Another simple solution is to use multiple images of the same object taken from different viewpoints or to cluster query images into groups depicting similar objects that are assumed to be matched with the same object in the scene $\mathcal{S}$. $\mathcal{A}$ can then use different images from these groups to query $\mathcal{L}$ and decide on the presence/absence of the object based on the consistency of returned poses. Even though the returned poses can be noisy and can contain outlier poses, in general, it is expected that a reasonably large subset of images depicting the same object from different viewpoints or depicting objects from the same group will show consistency of returned poses if a similar object is present in $\mathcal{S}$. On the other hand, poses obtained for images of an object that is absent can be expected to exhibit a much higher variance.

In this paper, we discuss and evaluate another strategy for presence/absence decision that allows us to show the completeness of the attack and present its proof-of-concept implementation. We assume that the attacker $\mathcal{A}$ learns certain statistics for each object or category from a curated training data that comprises of scenes with known presence/absence of these objects or object categories. This can be done for different types of localization schemes over huge amounts of 3D data. The attacker can then use these learned statistics to infer the presence/absence of objects when attacking an unknown scene $\mathcal{S}$.

For experimental results in the later sections, we use the inlier-ratio ϵ obtained from the object positioning step (Line 15 in Alg. 1) as this statistic. We can assume that for each object (or a class of objects) o, $\mathcal{A}$ has inlier-ratios ϵ_o^+ and ϵ_o^- that are trained on scenes with known presence or absence of o. *E.g.*, ϵ_o^+ and ϵ_o^- can be computed as the medians of ϵ_o over all "present(+)/absent(-)" scenes. Based on these statistics, the presence or absence of o in the unknown scene $\mathcal{S}$ can be decided by comparing the distances of $\epsilon_o^{\mathcal{S}}$ to ϵ_o^+ and ϵ_o^-. We provide concreteness to this idea when assessing its effectiveness over a real world dataset in Section 5.2.

4. Datasets

We use multiple datasets for our experiments:

IKEA-Scenes and **IKEA-Objects** - We captured image sequences of seven different inspiration-rooms at an IKEA furniture store (*cf*. Fig.3). 1,000-2,500 images were captured for each room, depending on its size. 4-10 objects from each room were selected, and a separate sequence of images was captured for each of them in the inventory section of the store, where the surrounding environment was different from that of the inspiration rooms. Note that the two instances of each object have the same model, but in many cases differ in color and size. Presence/absence of additional objects such as cushions on a sofa, or a computer on a desk can additionally change the overall appearance of the two instances. In total, the dataset comprises 38 object instances covered by 100-200 images each. While capturing the dataset, we tried to only have a single object occupying a large part of each image. However, this was not always possible and no post processing has been applied to mask out objects. We call the inspiration-room data *IKEA-Scenes* and the data from the inventory section *IKEA-Objects*.

ScanNet-Office-Scene - To show that the objects do not need to be of the exact same model for the proposed attack to work, we consider a generic office scene - *scene0040* from the ScanNet [8] dataset.

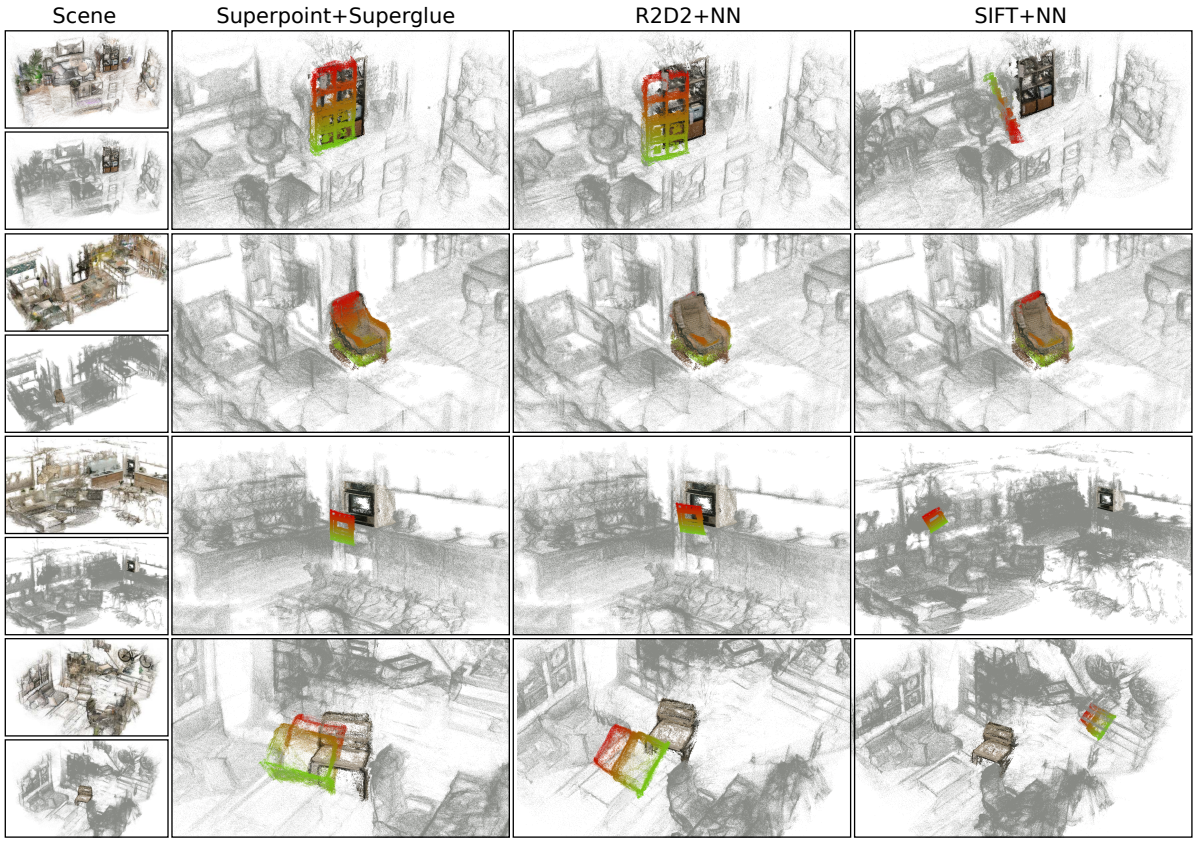

Figure 4. Qualitative results for aligning objects in different scenes of the *IKEA-Scenes* dataset. We evaluate three combinations of local features and matchers. Aligned objects are color-coded green to red along the gravity direction to make their orientation better visible.

Office-Objects - We collected image sequences of 5 common office room objects - a *door*, a *whiteboard*, an *office chair*, a *desk with computer*, and a *bookshelf*. These images are used as queries by the attacker.

RIO10 - RIO10 [47] is a localization benchmark dataset which we use to evaluate the effectiveness of a potential defence strategy that a localization server might employ.

We manually scale all local 3D models constructed by the attacker to roughly metric scale.

5. Experimental Evaluation

This section presents a series of experiments that show the practical feasibility of the attack introduced in Sec. 3. First, we show via qualitative results that the method proposed in Sec. 3.2 allows the attacker to place the 3D models of relevant objects close to the actual corresponding objects in the scene. We then explain and evaluate a simple implementation of the method described in Sec. 3.3 that the attacker can use to decide the presence/absence of objects.

For querying the localization server, we use images from the datasets described above. To implement the server, we use HLoc [31, 32] (with default thresholds and parameters), a state-of-the-art visual localization approach. HLoc uses feature descriptors to establish 2D-3D matches between features extracted from the query image and 3D scene points. The resulting correspondences are then used for pose estimation. We demonstrate the reliance of the attack on the robustness of the localization process by evaluating three different local image features and matchers: Superpoint [10] features with the SuperGlue [32] (most robust), R2D2 [30] with Nearest Neighbor (NN) matching, and SIFT [22] with NN matching (least robust).

5.1. 3D Object Placement

We qualitatively evaluate the accuracy of the 3D object placements obtained using the approach from Sec. 3.2 for the *IKEA-Scenes* and *ScanNet-Office-Scene* datasets. We use qualitative results rather than quantitative metrics since it is hard to quantify when a placement is realistic enough. *E.g.*, consider the predicted positions of the oven in the 3rd row of Fig. 4. The first two predictions are far enough from the ground truth position that a metric such as the IoU of the 3D bounding boxes of the objects will discard them as wrong. Yet, the estimated positions are close enough to the ground truth to provide the attacker with a good layout of the scene.

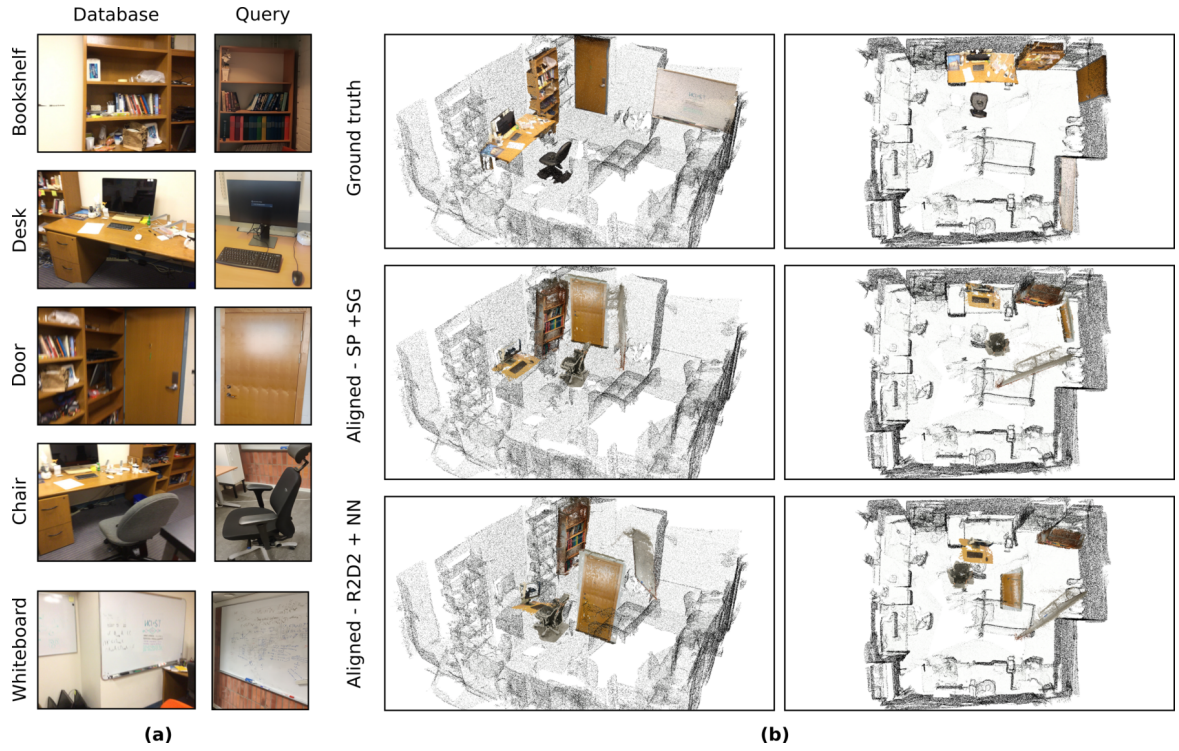

Figure 5. **(a)** Example images from *ScanNet-Office-Scene* and corresponding objects in *Office-Objects*. **(b)** Qualitative results for aligning generic office objects in ScanNet [8] *scene0040*, using Superpoint+Superglue and R2D2+NN.

Fig. 4 shows results for placing selected items from the *IKEA-Objects* dataset in 4 different scenes from the *IKEA-Scenes* dataset. Fig. 5 shows results for placing objects from the *Office-Objects* dataset in the *ScanNet-Office-Scene* dataset. As can be seen, using a robust localization process based on Superpoint features and the Superglue matcher or R2D2 features allows the attacker to place the objects close to their ground truth positions. In particular, the results from Fig. 5 show that the alignment also works well when the queried object is not the same model of different color/size but also a very different one in terms of shape and overall appearance. The results clearly demonstrate the practical feasibility of the placement strategy.

We used slightly different values for the error thresholds required by the positioning algorithm based on the object size and obtained poses. Such an approach is feasible if a human supervises the attack. Code and data is available at https://github.com/kunalchelani/ObjectPositioningFromPoses.

5.2. Deciding the Presence/Absence of an Object

In Sec. 3.3, we suggested strategies which an attacker can use to decide whether an object is present or not in a scene $\mathcal{S}$. for each object.

Concretely, using a set of training scenes, the attacker has learned representative values ϵ^+ and ϵ^- for the inlier-ratio returned by Alg. 1 for cases where the object is present(+) respectively absent(-). When deciding the presence of an object $\mathbf{o}$ in a scene $\mathcal{S}$, the attacker uses the inlier ratio (ϵ) from Alg. 1 to make their decision. The object $\mathbf{o}$ is considered to be present in the scene if $|\epsilon - \epsilon^+| < |\epsilon - \epsilon^-|$ and otherwise considered as absent.

We use the *IKEA-Scenes* and *IKEA-Objects* dataset for this experiment. When deciding the presence/absence of an object in a scene, the other 6 scenes are used as training scenes. Many of the objects from *IKEA-Objects* are only present in one of the scenes from *IKEA-Scenes*. In these cases, no reference value for ϵ^+ is available for these scenes. In such cases, the object is considered as present if $\epsilon > \epsilon^-$. This strategy is motivated by the assumption that correctly placing an object that is present results in a higher inlier-ratio than placing objects that are not present.

Tab. 1 shows precision and recall of this strategy. Since the computation of the inlier-ratio ϵ depends upon the error thresholds, we present the results for three different sets of thresholds. The results show that for most scenes, it is possible to obtain a precision/recall of approx. 0.4/0.6, which, *e.g.*, translates to 3 out of 5 present, and around 29 out of 33 absent objects from *IKEA-objects* being correctly classified. The average precision using random guessing in these scenes is 0.19. This, together with the quality of the placement, clearly validates the feasibility of the proposed attack.

Scene	Superpoint+Superglue						R2D2+NN						SIFT + NN					
	$10°, 0.25m$		$30°, 0.5m$		$60°, 2m$		$10°, 0.25m$		$30°, 0.5m$		$60°, 2m$		$10°, 0.25m$		$30°, 0.5m$		$60°, 2m$	
	Precision	Recall	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R
Scene1	0.6	0.85	0.75	0.85	0.67	0.85	0.57	0.57	0.36	0.57	0.28	0.57	0.33	0.57	0.45	0.71	0.33	0.43
Scene2	0.36	0.4	0.36	0.5	0.37	0.6	0.34	0.4	0.3	0.3	0.35	0.6	0.33	0.4	0.26	0.5	0.28	0.6
Scene3	0.55	0.71	0.36	0.57	0.25	0.43	0.31	0.71	0.47	1	0.41	1.0	0.3	0.42	0.5	0.42	0.44	1.0
Scene4	0.17	0.4	0.23	0.6	0.14	0.4	0.34	0.6	0.28	0.4	0.2	0.4	0.15	0.4	0.15	0.4	0.17	0.4
Scene5	0.33	0.6	0.4	0.8	0.44	0.8	0.5	0.6	0.34	0.4	0.5	0.6	0.22	0.4	0.25	0.4	0.33	0.4
Scene6	0.25	0.6	0.28	0.6	0.22	0.4	0.22	0.4	0.3	0.6	0.33	0.8	0.14	0.2	0.2	0.2	0.25	0.6
Scene7	0.5	0.5	0.5	0.33	0.33	0.5	0.6	0.5	0.5	0.5	0.38	0.5	0	0	0.14	0.17	0	0

Table 1. Precision (P) and recall (R) of our method to determine the presence of objects for the *IKEA-scenes* and *IKEA-Objects* datasets.

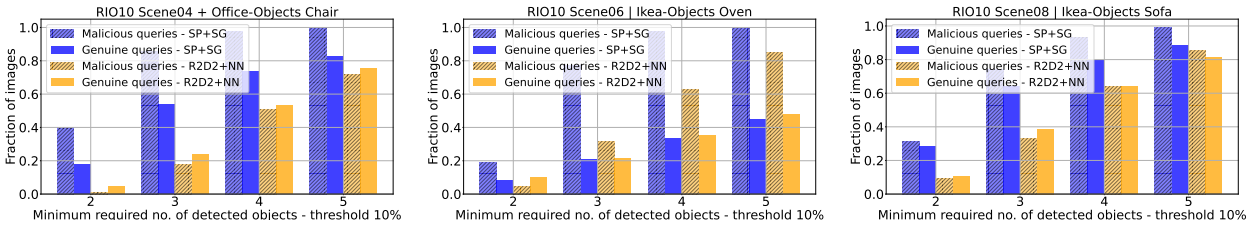

Figure 6. Effectiveness of a potential approach to prevent the proposed attack based on not providing poses for queries containing only a few objects. Only objects contributing at least 10% of the inliers found on the object with the most inliers are considered. As can be seen, finding a suitable threshold for the minimum number of visible objects can be difficult.

6. Preventing the Attack?

A natural way to prevent the presented attack is to try to distinguish between genuine and malicious queries. By not sending poses for query images deemed as (potentially) malicious, the localization service effectively prevents the attacker from using pose estimates to learn about the scene.

One potential classification strategy is based on the fact that the attacker sends images focusing on a single object. In this case, we expect that most of the 3D points from the inlier 2D-3D matches found by HLoc lie on a single 3D object. We thus count the number of 3D objects that contribute at least a certain fraction of inliers (X% of the inliers of the object contributing the largest number of inliers). If the number is too small, the query image is considered to be malicious and is rejected.

Fig. 6 shows results for three different objects used to attack three different scenes of the RIO10 dataset [47]. Here, we use the instance-level labels provided by the dataset, which include background classes such as floor and walls, to define objects. As can be seen, rejecting the majority of malicious queries while retaining genuine queries can be challenging. The reason is that even while focusing on a single object, other objects might be partially visible in the queries, *e.g.*, part of a desk for monitors, different pillows on a couch, books on a shelf, *etc*. In addition, genuine queries might focus on small parts of the scene or even individual objects. Thus, finding a suitable threshold on the minimum number of visible objects can be hard. Furthermore, note that this defense strategy requires the service to have knowledge about the objects in the scene, either extracted from the queries or the scene representation. This requirement creates a potential privacy risk if an attacker is able to gain access to the service.

7. Conclusions and Future work

In this paper, we have considered the problem of privacy-preserving localization. Prior work aims to defend attacks for the case where the attacker gains access to a cloud-based localization service. In contrast, we show that it is possible for an attacker to recover information about the scene by using the service as intended: by querying the server with images of different objects, an attacker is able to determine which objects are present and to estimate their position in the scene. The attack is based on the minimum amount of information that a localization service needs to provide to its users, *i.e.*, camera poses for query images, and exploits that modern localization systems are robust to changing conditions. Experiments with our proof-of-concept implementation show the practical feasibility of the attack. The attack is applicable even if the localization algorithm used by the server is otherwise perfectly privacy-preserving.

Our results show that existing privacy-preserving approaches are not sufficient to ensure user privacy, creating the need for further research. In particular, first experiments show that preventing the attack proposed in this paper without reducing localization performance and creating other angles of attack is a non-trivial task and interesting direction for future work.

Acknowledgements. This work was supported by the EU Horizon 2020 project RICAIP (grant agreement No. 857306), the European Regional Development Fund under project IMPACT (No. CZ.02.1.01/0.0/0.0/15_003/0000468), the Czech Science Foundation (GAČR) JUNIOR STAR Grant No. 22-23183M, Chalmers AI Research Center (CHAIR), WASP and SSF.

References

[1] Eric Brachmann and Carsten Rother. Learning Less is More - 6D Camera Localization via 3D Surface Regression. In *CVPR*, 2018. 2

[2] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *ICCV*, 2019. 2

[3] Eric Brachmann and Carsten Rother. Visual camera re-localization from RGB and RGB-D images using DSAC. *arXiv:2002.12324*, 2020. 2

[4] Federico Camposeco, Andrea Cohen, Marc Pollefeys, and Torsten Sattler. Hybrid Scene Compression for Visual Localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[5] R. O. Castle, G. Klein, and D. W. Murray. Video-rate localization in multiple maps for wearable augmented reality. In *ISWC*, 2008. 1

[6] T. Cavallari, L. Bertinetto, J. Mukhoti, P. Torr, and S. Golodetz. Let's take this online: Adapting scene coordinate regression network predictions for online rgb-d camera relocalisation. In *3DV*, 2019. 2

[7] Kunal Chelani, Fredrik Kahl, and Torsten Sattler. How Privacy-Preserving Are Line Clouds? Recovering Scene Details From 3D Lines. In *(CVPR)*, pages 15668–15678, June 2021. 2, 3

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 5, 7

[9] 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. 2021. 1, 3

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description, 2018. 2, 6

[11] Tien Do, Ondrej Miksik, Joseph DeGol, Hyun Soo Park, and Sudipta N. Sinha. Learning to detect scene landmarks for camera localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 1, 2, 3

[12] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *CVPR 2016*, pages 4829–4837, 06 2016. 3

[13] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A trainable CNN for joint detection and description of local features. In *CVPR*, 2019. 2

[14] Mihai Dusmanu, Johannes L. Schönberger, Sudipta N. Sinha, and Marc Pollefeys. Privacy-preserving visual feature descriptors through adversarial affine subspace embedding. 2020. 1, 3

[15] Marcel Geppert, Viktor Larsson, Johannes L. Schönberger, and Marc Pollefeys. Privacy preserving partial localization. In *CVPR*, 2022. 1, 3

[16] Marcel Geppert, Viktor Larsson, Pablo Speciale, Johannes L. Schönberger, and Marc Pollefeys. Privacy Preserving Structure-from-Motion. In *ECCV*, 2020. 3

[17] Marcel Geppert, Peidong Liu, Zhaopeng Cui, Marc Pollefeys, and Torsten Sattler. Efficient 2D-3D Matching for Multi-Camera Visual Localization. In *ICRA*, 2019. 1

[18] L. Heng, B. Choi, Z. Cui, M. Geppert, S. Hu, B. Kuan, P. Liu, R. Nguyen, Y. C. Yeo, A. Geiger, G. H. Lee, M. Pollefeys, and T. Sattler. Project AutoVision: Localization and 3D Scene Perception for an Autonomous Vehicle with a Multi-Camera System. In *ICRA*, 2019. 1

[19] 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

[20] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From Structure-from-Motion Point Clouds to Fast Location Recognition. In *CVPR*, 2009. 2

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

[22] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2), 2004. 6

[23] Simon Lynen, Bernhard Zeisl, Dror Aiger, Michael Bosse, Joel Hesch, Marc Pollefeys, Roland Siegwart, and Torsten Sattler. Large-scale, real-time visual–inertial localization revisited. *IJRR*, 39(9):1061–1084, 2020. 1

[24] Microsoft. Spatial Anchors, 2020. 1

[25] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt. Scalable 6-DOF Localization on Mobile Devices. In *ECCV*, 2014. 1

[26] 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. 2021. 1, 3

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

[28] 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

[29] Tilman Reinhardt. Using Global Localization to Improve Navigation, 2019. 1

[30] Jerome Revaud, Philippe Weinzaepfel, César Roberto de Souza, and Martin Humenberger. R2D2: repeatable and reliable detector and descriptor. In *NeurIPS*, 2019. 2, 6

[31] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 2, 6

[32] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2, 6

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

[34] Johannes L. Schönberger and Jan-Michael Frahm. Structure-From-Motion Revisited. In *CVPR*, June 2016. 4

[35] 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. 4

[36] Mikiya Shibuya, Shinya Sumikura, and Ken Sakurada. Privacy Preserving Visual SLAM. In *ECCV*, 2020. 1

[37] Mikiya Shibuya, Shinya Sumikura, and Ken Sakurada. Privacy preserving visual SLAM. In *(ECCV)*, 2020. 3

[38] 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*, 2013. 2

[39] Tory Smith. Niantic lightship. 2022. 1

[40] Zhenbo Song, Wayne Chen, Dylan Campbell, and Hongdong Li. Deep Novel View Synthesis from Colored 3D Point Clouds. In *ECCV*, 2020. 2

[41] Pablo Speciale, Sing Bing Kang, Marc Pollefeys, Johannes Schönberger, and Sudipta Sinha. Privacy preserving image-based localization. In *CVPR*. IEEE, June 2019. 1, 3

[42] Pablo Speciale, Johannes L. Schonberger, Sudipta N. Sinha, and Marc Pollefeys. Privacy Preserving Image Queries for Camera Localization. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 3

[43] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021. 2

[44] 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. *PAMI*, 2020. 1, 2

[45] J. Valentin, A. Dai, M. Niessner, P. Kohli, P. Torr, S. Izadi, and C. Keskin. Learning to Navigate the Energy Landscape. In *International Conference on 3D Vision (3DV)*, 2016. 2

[46] Jonathan Ventura, Clemens Arth, Gerhard Reitmayr, and Dieter Schmalstieg. Global Localization from Monocular SLAM on a Mobile Phone. *IEEE Transactions on Visualization and Computer Graphics*, 20(4):531–539, 2014. 1

[47] Johanna Wald, Torsten Sattler, Stuart Golodetz, Tommaso Cavallari, and Federico Tombari. Beyond controlled environments: 3d camera re-localization in changing indoor scenes. In *European Conference on Computer Vision (ECCV)*, 2020. 6, 8

[48] Qing Wang, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching, 2022. 2

[49] P. Weinzaepfel, H. Jégou, and P. Pérez. Reconstructing an image from its local descriptors. In *CVPR*, 2011. 3

[50] A. Wendel, A. Irschara, and H. Bischof. Natural landmark-based monocular localization for mavs. In *ICRA*, 2011. 1

[51] Luwei Yang, Rakesh Shrestha, Wenbo Li, Shuaicheng Liu, Guofeng Zhang, Zhaopeng Cui, and Ping Tan. Scenesqueezer: Learning to compress scene for camera relocalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8259–8268, June 2022. 2

[52] Qunjie Zhou, Sérgio Agostinho, Aljoša Ošep, and Laura Leal-Taixé. Is geometry enough for matching in visual localization? In *ECCV*, 2022. 1, 3

[53] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. Patch2pix: Epipolar-guided pixel-level correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2